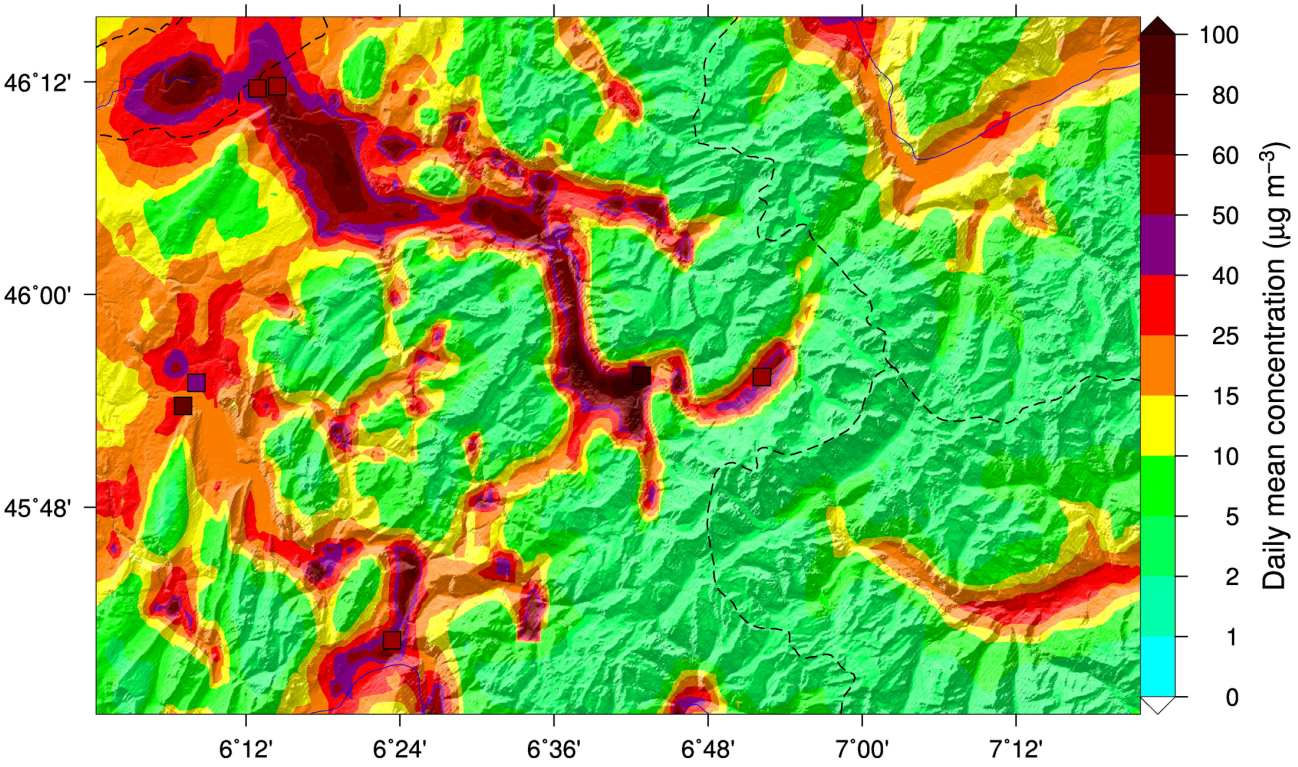
Chemistry-transport model *v2020r1*

Documentation



PM_{10} surface concentrations in the Alps with 1km resolution.

This documentation and the model are freely available at:

`http://www.lmd.polytechnique.fr/chimere/`

CHIMERE is distributed under the GNU General Public License
Copyright (C) 1997-2020 LMD (CNRS), INERIS, LISA (CNRS)
For questions, send an e-mail to chimere@lmd.polytechnique.fr
Last update of this documentation: December 22, 2020

Contents

1 Model overview and last changes **5**
- 1.1 Short description of the model . . . 5
- 1.2 Short history . . . 6
- 1.3 The CHIMERE software . . . 7
- 1.4 Last changes in version *v2020r1* . . . 7
- 1.5 The development team . . . 8

2 Getting started : first download, installation and simulation **11**
- 2.1 List of required external libraries . . . 11
- 2.2 Download the model . . . 12
- 2.3 Compile the CHIMERE system . . . 13
- 2.4 Run the model . . . 16
- 2.5 Hands-on : a first test case . . . 18

3 The CHIMERE parameter file **23**
- 3.1 General structure . . . 23
- 3.2 Analysis or forecast . . . 23
- 3.3 Nesting, parallelism and vertical mesh . . . 23
- 3.4 Simulation output parameters . . . 24
- 3.5 Databases . . . 25
- 3.6 Emissions . . . 25
- 3.7 Meteorology Driver . . . 26
- 3.8 Chemical boundary conditions . . . 27
- 3.9 Chemistry and deposition / resuspension options . . . 28
- 3.10 Numerical solution parameters . . . 29
- 3.11 CHIMERE Subgrid Processes . . . 29
- 3.12 Flags whether to run individual dynamic interfaces and the Core . . . 30

4 CHIMERE general code structure **33**
- 4.1 Directories structure . . . 33
- 4.2 The chimere.sh script . . . 33
- 4.3 Parallelism . . . 34
- 4.4 Routine calls sequence . . . 35
- 4.5 How to add a new diagnostic variable to the code? . . . 37
- 4.6 The several time-steps . . . 38
- 4.7 Numerical integration . . . 40

5 How to define a new CHIMERE simulation? **43**
- 5.1 Define the simulation domain . . . 43
- 5.2 Surface and soil input data . . . 46
- 5.3 Nested simulations . . . 49

6 Management of offline or online simulations **55**
- 6.1 How to perform an offline run? . . . 55
- 6.2 How to perform an online run? . . . 58
- 6.3 Main changes in OASIS-MCT . . . 60
- 6.4 Main changes in WRF . . . 61
- 6.5 Main changes in CHIMERE . . . 63
- 6.6 Rules and advice for online coupling with CHIMERE . . . 63

7 Meteorology, turbulence, direct and indirect effects **65**
- 7.1 Diagnostics of turbulent parameters: . . . 65
- 7.2 Transport and mixing . . . 71
- 7.3 The direct aerosol effects . . . 75
- 7.4 The indirect aerosol effects . . . 76

8 Initial and boundary conditions **79**
- 8.1 Options activation . . . 79
- 8.2 The boundary and initial conditions interface . . . 81
- 8.3 Available global models climatologies . . . 82

9 Anthropogenic surface emissions **89**
- 9.1 The emitted species for the CHIMERE simulation . . . 89
- 9.2 The emiSURF program . . . 91
- 9.3 A subgrid scale model to account for heterogeneous surface emissions . . . 99

10 Natural emissions **105**
- 10.1 Biogenic emissions . . . 105
- 10.2 Sea salt emissions . . . 106
- 10.3 DMS emissions . . . 106
- 10.4 Mineral dust emissions . . . 107
- 10.5 Fire emissions . . . 117
- 10.6 The urban resuspension . . . 120

11 [chemprep] the chemical preprocessor **121**
- 11.1 Preparation of new chemical data . . . 121
- 11.2 The preparation of gaseous and aerosol chemistry . . . 122
- 11.3 The aerosols size distribution calculations . . . 130
- 11.4 The preparation of tracers . . . 133

12 Gas-phase chemistry **135**
- 12.1 Option activation . . . 135
- 12.2 The MELCHIOR mechanims . . . 135
- 12.3 The SAPRC mechanims . . . 137
- 12.4 The DMS chemical scheme . . . 138

13 Aerosol chemistry **139**
- 13.1 The aerosols species . . . 139
- 13.2 Main processes . . . 139
- 13.3 Thermodynamic equilibrium of inorganic compounds . . . 141
- 13.4 Thermodynamic equilibrium of organic compounds . . . 142
- 13.5 Multiphase chemistry . . . 143
- 13.6 Secondary organic aerosol (SOA) . . . 143

14 [FastJX] Radiation, AOD and photosynthesis **149**
- 14.1 Radiative processes in the atmosphere . . . 149
- 14.2 Optical preprocessing using prep_mie . . . 149
- 14.3 Fast-JX initialisation . . . 150
- 14.4 Online calculation of photochemical reaction rates . . . 151
- 14.5 Aerosol and Cloud Optical Depth . . . 151
- 14.6 Surface albedo . . . 151
- 14.7 The lidar vertical profiles . . . 151

15 Depositions **155**
- 15.1 Dry deposition . . . 155
- 15.2 Wet scavenging . . . 165

Bibliography **167**

A References using CHIMERE **181**
- A.1 Reference papers . . . 181
- A.2 List of papers using CHIMERE . . . 182

B Changes in WRF for the coupling **199**
- B.1 Introduction . . . 199
- B.2 Registry files . . . 199
- B.3 Solver part . . . 203
- B.4 Coupling modules . . . 205
- B.5 Driver part . . . 226
- B.6 Physics packages . . . 227

C Example of an OASIS namcouple file for the WRF/CHIMERE coupling **241**

D How to install MPI and NetCDF **247**
D.1 Libraries to install 247
D.2 Installation of MPI 248
D.3 Installation of NetCDF-4 with HDF5 248

E Structure of the CHIMERE netCDF files **251**
E.1 The anthropogenic and fires emissions in the RAW format 251
E.2 The anthropogenic, biogenic, dust and fire emissions in the CHIMERE output format 252
E.3 The main results file "out.[label].nc" 256
E.4 The results file for deposition fluxes "dep.[label].nc" . . . 262
E.5 The results file for restart "end.[label].nc" 269
E.6 The results file "exdomout.[label].nc" 280

F VOC speciation **283**

G Chemical reactions **291**
G.1 Reaction list of the reduced MELCHIOR1 chemical mechanism 291
G.2 Reaction list of the reduced MELCHIOR2 chemical mechanism 298
G.3 Reaction list of the SAPRC-07-A chemical mechanism . 301
G.4 Reaction list of the chlorine SAPRC-07-A chemical mechanism 308
G.5 Reaction list of the VBS scheme with functionalization process for POA 311
G.6 Reaction list of the VBS scheme with functionalization and fragmentation process for POA 312
G.7 The SOA simple scheme with single-step oxidation . . . 313
G.8 The SOA medium scheme with single-step oxidation . . . 314
G.9 The SOA complex scheme with single-step oxidation . . 315
G.10 Reaction list of the VBS scheme with functionalization process 316
G.11 Reaction list of the VBS scheme with functionalization and fragmentation process 316

1 Model overview and last changes

1.1 Short description of the model

All informations, documentation and source code are available on the CHIMERE web site:

```
http://www.lmd.polytechnique.fr/chimere
```

CHIMERE is a Eulerian chemistry-transport model (CTM). External forcings are required to run a simulation: meteorological fields, primary pollutant emissions and chemical boundary conditions. Using these input data, CHIMERE calculates and provides the atmospheric concentrations of hundreds of gas-phase and aerosol species over local to continental domains (from 1 km to few degrees resolution). The key processes affecting the chemical concentrations and represented in CHIMERE are emissions, transport (advection and mixing), chemistry and deposition, as presented in Figure 1.1 . The model can be used in offline or online mode. In offline mode, meteorological fields for the period and geographical region of the simulation are pre-calculated by a meteorological model. They are interpolated in space and time on the CHIMERE temporal and spatial grids. Any meteorological model can be used provided that all mandatory variables for the CHIMERE simulation are given in the appropriate format. Interfaces for WRF and ECMWF models are provided with the CHIMERE code. For online simulations, the meteorological model runs at the same time as CHIMERE and fields are exchanged between models during the simulation allowing for interactions to take place. The model is currently set up to use the WRF 3.7.1 regional model which is provided with the CHIMERE code.

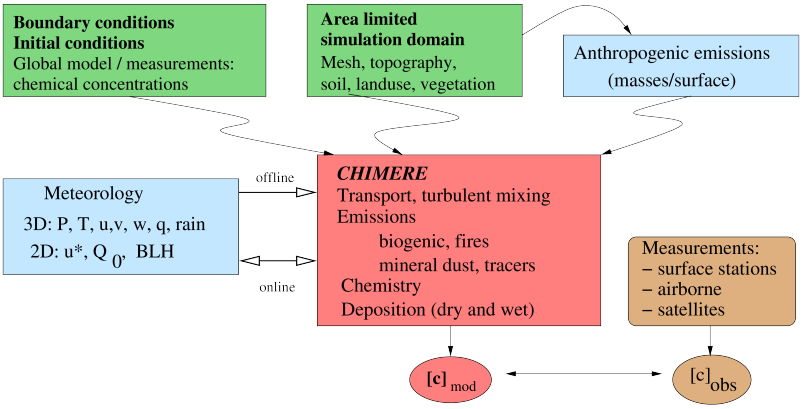


Figure 1.1: *General principle of a chemistry-transport model such as CHIMERE. In the box 'Meteorology', u* stands for the friction velocity, Q_0 the surface sensible heat flux, L the Monin-Obukhov length and BLH the boundary layer height. c_{mod} and c_{obs} are for the chemical concentrations fields for the model and the observations, respectively.*

CHIMERE is designed to be used for several applications: analysis of pollution events, ensemble modelling, long-term simulations or forecast. The model is able to run with nested domains. For each process, several schemes are implemented and can be easily changed using the parameter file. The user can easily change some parameterizations, input databases of horizontal or vertical resolution. Several chemical mechanism are provided, simplified or more complete, with or without aerosols.
Currently, the model is used for research on transport and mixing, turbulence, gas and aerosol chemistry, anthropogenic, biogenic, natural (mineral dust, fires, volcanos and marine) emissions and dry and wet deposition. It is also used for oerational forecast: for example by regional air quality networks (France, Italy, Spain, Netherlands, Portugal etc.), national (PREV'AIR) or continental scale (CAMS).
The main parameterizations implemented in CHIMERE are described in Table 1.1 .

Process	Methodology
Meteorological forcing	ECMWF, WRF
Landuse	GlobCover or USGS global datasets
Horizontal resolution	from 1km to few degrees
Horizontal transport	Upwind, Van Leer or PPM
Vertical transport	Upwind, Van Leer or Després-Lagoutière
Deep convection	[Tiedke] scheme
Boundary-layer turbulence	[Troen and Mahrt] scheme
Anthropogenic emissions	EMEP, HTAP, CAMS or local inventories
Biogenic emissions	online MEGAN model
Sea salt emissions	[Monahan], [Martensson] or [Grythe] scheme
DMS emissions	[Blend], [Nightingale], [Wanninkhof] or [Liss and Merlivat] scheme
Mineral dust emissions	[Marticorena and Bergametti], [Alfaro and Gomes] or [Kok] scheme
Chemical mechanism	SAPRC-07-A, MELCHIOR1 or MELCHIOR2
Aerosols representations	bins ($\approx$ 10)
Secondary Organic Aerosols	[Pun] scheme, VBS and H^2O shemes
Photolytic rates	online FastJX radiative transfer model
Numerical solver	TwoStep solver or operator splitting
Aerosol thermodynamic equilibrium	online ISORROPIA or SOAP
Dry deposition	[Wesely] or [Zhang] parameterizations

Table 1.1: *Main properties of the CHIMERE model*

1.2 Short history

The main evolutions of the CHIMERE code are:

1997 The first model version was released in 1997 and was a box model covering the Paris area including only gas-phase chemistry ([Honoré and Vautard, 2000], [Vautard et al., 2001]).

1998 The model is implemented for its first forecast version (Pollux) during the ESQUIF experiment ([Menut et al., 2000b]) still over the Paris area ([Vautard et al., 2000]).

1999 The geographical domain was extended over Europe with a cartesian mesh ([Schmidt et al., 2001]) and the new experimental forecast platform (PIONEER) is set up

2000 The adjoint model was developed to estimate the sensitivity of concentrations to all parameters ([Menut et al., 2000a]).

2003 The experimental forecast became operational with the PREVAIR system operated at INERIS ([Honoré et al., 2008], [Rouïl et al., 2009]).

2004 The aerosol module is implemented ([Bessagnet et al., 2004]) with further improvements concerning the dust natural emissions and resuspension over Europe ([Vautard et al., 2005b], [Hodzic et al., 2006a] [Bessagnet et al., 2008]) and evaluated against long-term and field measurements ([Hodzic et al., 2005], [Hodzic et al., 2006b]).

2005 Development and implementation of a simple dust emissions scheme for Europe, in order to close the aerosol mass budget for air quality studies ([Vautard et al., 2005b]). This development was done over Europe only due to the availability of anthropogenic emissions (EMEP only for the CHIMERE version previous to 2008). At the same time, development of the transport model CHIMERE-dust dedicated to long range transport of dust over the northern hemisphere ([Menut et al., 2005, Menut et al., 2007, Menut, 2008, Menut et al., 2013a]). This model was developed for studies of mineral aerosols processes (emissions and transport) over the Northern Atlantic and did not contain chemistry.

2006 Development of the parallel version of the model and its first implementation on a massively parallel computer (the ECMWF computer in the framework of the FP6/GEMS project

2007 The deep convection is added to ensure correct mixing for large scale domains. The dry deposition scheme of Zhang is added.

2007 *CHIMERE becomes a CNRS french national tool.* It means that the model organization now contains a development team, several e-mail adresses (chimere@lmd, chimere-users@lmd), an annual training course, a web site and a documentation.

2008	The air quality forecast at the European scale is created within the GEMS European project. CHIMERE is one of the CTM used daily for gas and aerosol forecast. A kriging approach is developed to optimize the initial conditions by assimilating surface ozone concentrations of the days before, [Honoré et al., 2008, Rouïl et al., 2009]
2009	Improvement and validation of Secondary Organic Aerosols, [Bessagnet et al., 2009].
2011	The development of CHIMERE-dust is frozen. Its specific parameterizations are implemented into the distributed version of CHIMERE.
2012	MACC Operational forecast
2013	Release of the 2013 model version, with a reference paper for the scientific documentation, [Menut et al., 2013a]
2014	The online FastJX Radiative transfer and photolysis model is implemented in CHIMERE and replace old look-up table. The SAPRC chemical mechanism is added, [Mailler et al., 2016].
2016	The global database for mineral dust are updated, including aeolian roughness length.
2017	Release of the 2017 version: CHIMERE is able to run over a hemispheric domain as well as for smaller regions anywhere in the world ([Mailler et al., 2017]). Development of the meteo/chemistry coupling with the aerosol direct effect, [Briant et al., 2017].
2018	Development of a new aerosol schemes, [Couvidat et al., 2018, Cholakian et al., 2018].
2019	Development of the meteo/chemistry coupling with the aerosol indirect effect, [Tuccella et al., 2019].
2020	Development of the DUST mineralogical composition and solubility, [Menut et al., 2020b].

Finally, the development of CHIMERE follows three main rules.

1. concentrations of main pollutants are calculated with the best possible accuracy using well evaluated and state-of-the-art parameterisations.
2. a modular framework is maintained to allow updates to the code by developers but also all interested users.
3. the code is kept computationally efficient to allow long-term simulations, climatological studies and operational forecast.

1.3 The CHIMERE software

In order to facilitate software distribution, CHIMERE is protected under the General Public License. The source code and the associated documentation is available on a web site http://www.lmd.polytechnique.fr/chimere. The documentation is both technical and scientific. It includes a chapter dedicated to the set-up of a test case simulation that allows new users to easily carry out a CHIMERE simulation: model configuration and data (meteorology and emissions files) are provided to simulate 2009 winter-time particulate pollution over Europe.
The code is mostly written in Fortran 90, and running scripts are written in shell (using gnu-awk for input datafiles processing). The required software is a Fortran compiler and C++ compiler for SOAP thermodynamic model. CHIMERE model is distributed with makefiles designed for gfortran and ifort. The required libraries are NetCDF (4.x.x), blitz, MPI, python, and GRIB API. The model includes tools that can help the user to configure the model's Makefiles for the libraries already installed.
The model parallelism results from a Cartesian division of the main geographical domain into several sub-domains. Each worker performs the model integration in its geographical sub-domain as well as boundary condition exhanges with its neighbours and file output. To configure the parallel sub-domains, the user has just to specify the number of available processors in the CHIMERE parameter file.

1.4 Last changes in version *v2020r1*

An important novelty is that the CHIMEE code is distributed with two others codes:

1. **WRF 3.7.1**: This model version is modified to host the OASIS-MCT routines used for the online coupling between WRF and CHIMERE. The WRF model is distributed with the WRF "public domain notice"[1]:

[1] https://www2.mmm.ucar.edu/wrf/users/public.html

WRF was developed at the National Center for Atmospheric Research (NCAR) which is operated by the University Corporation for Atmospheric Research (UCAR). NCAR and UCAR make no proprietary claims, either statutory or otherwise, to this version and release of WRF and consider WRF to be in the public domain for use by any person or entity for any purpose without any fee or charge. UCAR requests that any WRF user include this notice on any partial or full copies of WRF. WRF is provided on an "AS IS" basis and any warranties, either express or implied, including but not limited to implied warranties of non-infringement, originality, merchantability and fitness for a particular purpose, are disclaimed. In no event shall UCAR be liable for any damages, whatsoever, whether direct, indirect, consequential or special, that arise out of or in connection with the access, use or performance of WRF, including infringement actions. WRF is a registered trademark of the University Corporation for Atmospheric Research (UCAR).

The reference paper linked to this version is:
Skamarock, W. C., J. B. Klemp, J. Dudhia, D. O. Gill, D. M. Barker, M. G Duda, X.-Y. Huang, W. Wang, and J. G. Powers, 2008: A Description of the Advanced Research WRF Version 3. NCAR Tech. Note NCAR/TN-475+STR, 113 pp. doi:10.5065/D68S4MVH

2. **OASIS-MCT 3:** For the coupling, the OASIS-MCT 3.0 code is used. OASIS is a widely used external coupler developed by the CERFACS (Centre Européen de Recherche et de Formation Avancée en Calcul Scientifique, Toulouse, France). The reference paper of the model used and distributed with CHIMERE is:
Craig, A.; Valcke, S.; Coquart, L. Development and performance of a new version of the OASIS coupler, OASIS3-MCT_3.0. Geosci. Model Dev. 2017, 10, 3297-3308.

The other novelties are:

- **Numerics:**
  - Read NetCDF data in parallel with NetCDF4
  - Create your model domain with WPS software
  - Offline with any meteorological model or online with WRF
- **Processes:**
  - Aerosol direct and indirect effects with the online mode, [Briant et al., 2017, Tuccella et al., 2019]
  - Volatil Basis Set aerosol schemes, [Cholakian et al., 2018]
  - DMS emissions and chemistry
  - The H^2O aerosol scheme developed by [Couvidat et al., 2018]
  - Mineral dust mineralogy (with required input global databases from LMDz-INCA), [Menut et al., 2020b]
  - Addition of several sea salt emissions schemes
  - SOAP thermodynamic model developed by [Couvidat and Sartelet, 2015]
  - NO_x emissions by lightning, [Menut et al., 2020a]
  - Sub-grid scale variability of concentrations due to anthropogenic emissions by activity sectors, [Valari and Menut, 2010]
  - Updated strategy for vertical advection of pollutants and implementation of the [Després and Lagoutière, 1999] antidiffusive advection scheme

1.5 The development team

1.5.1 How to contact the developers?

CHIMERE is a National Tool of the French Institut des Sciences de l'Univers, meaning that support is provided to the model's users. The model is developed at IPSL/LMD (Palaiseau), INERIS (Verneuil en Halatte) and IPSL/LISA (Creteil) in France.

- **For questions about the model development**:
  Regarding parameterizations or variables choices, you can send an e-mail to the developers: *chimere@lmd.polytechnique.fr*
  This will send an e-mail to: Arineh Cholakian, Sylvain Mailler, Laurent Menut, Romain Pennel, Solène Turquety, Myrto Valari (LMD), Bertrand Bessagnet (CITEPA), Augustin Colette, Florian Couvidat, Gaël Descombes, Frédérik Meleux (INERIS) and Guillaume Siour (LISA).

- **For questions about the model use**:
  To share pre- or post-processing tools with the other users, or just present your result, you can register to the chimere users mailing list here: *http://www.lmd.polytechnique.fr/chimere/subscribe.php*
  Note that by registering to download the code, you will be automatically registered on the chimere-users mailing list. You can initiate discussions, exchange programs or data between users by using the e-mail adress: *chimere-users@lmd.polytechnique.fr*

1.5.2 Acknowledgments

The CHIMERE developers acknowledge:

- Laure Cocquart and Sophie Valcke for the training and use of the OASIS-MCT code
- The WRF developers team for the WRF model
- Richard Engelen for the global concentrations dataset in the framework of the MACC project. If you are using this dataset and publish your results, please add this acknowledgment: *"This data set was provided by the MACC-II project, which is funded through the European Union Framework 7 programme. It is based on the MACC-II reanalysis for atmospheric composition; full access to and more information about this data can be obtained through the MACC-II web site (`http://www.copernicus-atmosphere.eu`)."*
- Eric Chaxel for the WRF model meteorological interface
- Christian Seigneur and Betty Pun (AER) for the reduced SOA scheme,
- Mian Chin and Paul Ginoux (NASA) for the GOCART aerosols concentrations fields used as boundary conditions,
- Sophie Szopa and Didier Hauglustaine (IPSL/LSCE) for the LMDz-INCA gas concentrations fields used as boundary conditions,
- Athanasios Nenes and the ISORROPIA team for the free use of ISORROPIA model,
- Catherine Prigent for the satellite aeolian roughness length data. If you are using this dataset and publish your results, please add this acknowledgment: *"Comparison of satellite microwave backscattering (ASCAT) and visible/near-infrared reflectances (PARASOL) for the estimation of aeolian aerodynamic roughness length in arid and semi-arid regions by [Prigent et al., 2012, Atmos. Meas. Tech.]."*
- Emilie Journet and Yves Balkanski for the surface databases of the sol mineral composition.
- Claire Granier and the ECCAD plateform for the CAMS global emissions data.
- The CHIMERE evaluations were granted access to the HPC resources of [TGCC/CINES/IDRIS] under the allocations gen10274 attributed by GENCI (Grand Equipement National de Calcul Intensif)

1.5.3 This documentation

This documentation was written by Laurent Menut, Régis Briant, Paolo Tuccella, Sylvain Mailler, Romain Pennel, Guillaume Siour, Bertrand Bessagnet, Florian Couvidat, Arineh Cholakian and Myrto Valari.

2 Getting started : first download, installation and simulation

This chapter presents :

- §2.1: the necessary external libraries
- §2.2: how to download the model
- §2.3: how to compile the model
- §2.4: how to run the model
- §2.5: a hands-on trainning on a first test case which exhibits some of the outputs

2.1 List of required external libraries

The CHIMERE system has to be installed with GNU **bash** Bourne shell, **awk** and **make**. If no **bash** is installed, the model may work with a baseline Bourne-shell, but the user may have to edit the scripts to take into account some syntactic features specific to **bash** . The same remark apply to the **awk** and **make** utilities.

CHIMERE has been tested on GNU/Linux systems with the Open MPI implementation of the Message Passing Interface library. The model requires several numerical tools, compilers and libraries presented in table 2.1. The version used in the development team is indicated, other versions or implementations could work for CHIMERE but without any warranty. Standard python modules *sys*, *re*, *os*, *subprocess* and *datetime* are also required.

Important notice:
1. ALL libraries must be compiled with the same compiler as the CHIMERE model.
2. This computer installation is under the user's responsability and the CHIMERE developers are not able to install users computers or clusters.
3. Hints and installation examples are provided in Annex D

Software	URL	Tested versions
Fortran/C/C++ compilers		
GNU	`https://gcc.gnu.org/`	GCC 4.7.2
INTEL	`https://software.intel.com/parallel-studio-xe`	composer_xe_2013 parallel_studio_xe_2017
MPI		
Open MPI	`https://www.open-mpi.org/software/ompi/`	openmpi-2.0.1 openmpi-2.0.4
Librairies		
HDF5	`https://support.hdfgroup.org/ftp/HDF5/releases/`	HDF5 1.8.20
Unidata NetCDF-C	`https://github.com/Unidata/netcdf-c/releases`	netCDF-C 4.6.0
Unidata NetCDF-Fortran	`https://github.com/Unidata/netcdf-fortran/releases`	netCDF-Fortran 4.4.4
Blitz	`https://sourceforge.net/projects/blitz`	blitz-0.10
Grib API	`https://confluence.ecmwf.int/display/GRIB/Home`	grib-api 1.12.0 grib-api 1.26.1
Jasper	`https://www.ece.uvic.ca/~frodo/jasper/#download`	JasPer 1.900.1

Table 2.1: *List of required external libraries.*

netCDF-4 parallel:
The use of PnetCDF for I/O is deprecated in this CHIMERE *v2020r1* version. Parallel I/O are available through netCDF-4 and HDF5 parallel libraries.
Libraries have to be compiled from source in this order :

1. HDF5 with enable-parallel (C library)
2. netCDF-C (using previously compiled HDF5)
3. netCDF-Fortran (using previously compiled HDF5 and netCDF-C)

   Details and examples are provided in Annex D

2.2 Download the model

All codes and data have to be downloaded directly from the website:

```
https://www.lmd.polytechnique.fr/chimere/2020_getcode.php
```

If you are a new user, you have to register first here:

```
https://www.lmd.polytechnique.fr/chimere/CW-dwl-newuser.php
```

Data will be examined and if approved login/password will be sent to access the download page. Note that this step is made manually by the developers, thus it is not instantaneous.
If you are an already registered user, you can directly use your login/password. It is true even if you changed of e-mail adress: this address is only used as a login and its validity is not checked.
The download page is as in Figure 2.1 .

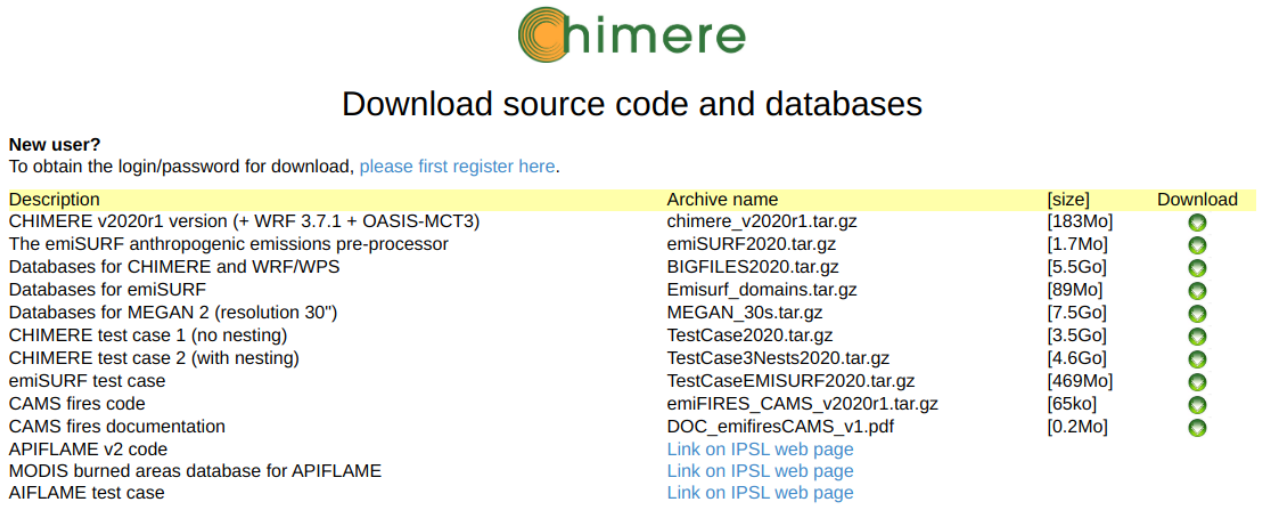


Figure 2.1: *The download page on the model web site*

At the end, you have model code and data as listed in Table 2.2 .
The user should find in the CHIMERE directory:

- the top-level script chimere.sh and compile scripts build-chimere.sh and build-wrf.sh
- chemprep/: Contains data files for gas, aerosols and chemical mechanisms.
- domains/: Files describing the simulation domain (mesh, landuse, roughness, LAI etc.)

Models	
WRF and WPS models (free)	https://www.mmm.ucar.edu/weather-research-and-forecasting-model
ISORROPIA	http://isorropia.eas.gatech.edu
MEGAN	http://www.lar.wsu.edu/megan
Databases	
EMEP	http://www.emep.int
EDGAR-HTAP	http://edgar.jrc.ec.europa.eu/
GlobCover Portal	http://due.esrin.esa.int/page_globcover.php
USGS	http://www.usgs.gov
Eolian roughness length	[Prigent et al., 2012]
Green Fraction	http://ncar.ucar.edu
LAI from NASA/EOSDIS	https://modis.gsfc.nasa.gov/data/dataprod/mod15.php
Biogenic emissions factor	[Guenther et al., 2006]
DMS sea surface concentrations	[Lana et al., 2011]

Table 2.2: *URL adresses for the development and the use of the CHIMERE model and its modules*

- mychimere/: Contains configuration files for compilation of CHIMERE, WRF and WPS with various architectures and compilers
- scripts/: Contains the scripts called by the top-level script.
- src/: Contains the model source code.
- oasis3-mct/: The OASIS-MCT code for the coupling between WRF and CHIMERE
- WPS361/: The WPS 3.6.1 version necessary to make the domain for WRF
- WRFV3_cpl371/: The WRF 3.7.1 version with the modified routines for the coupling

2.3 Compile the CHIMERE system

The CHIMERE system is made of shell scripts driving the initialization of the simulation and of executables and libraries that need to be compiled. Main executables are :

1. **chimere.e** : the CHIMERE model
2. **wrf_n0.exe** : the WRF model (used in online mode)
3. the WRF Preprocessing System : **geogrid.exe**, **metgrid.exe**, **ungrib.exe**. **geogrid** is needed both in offline and online mode to build the domain grid.
4. the **oasis3-mct** library, needed for online simulation coupled with WRF.

Everything has to be compiled the first time :
1. oasis3-mct is strongly embedded into CHIMERE and needs to be compiled even if the user uses only the offline mode.
2. WPS is used for the preparation of the CHIMERE domains even in offline mode. Therefore, WPS needs to be compiled even for offline simulations. For WPS to compile properly you need to have WRF compiled. So WRF too must be compiled even if the user uses only the offline mode.

2.3.1 Setup the architecture configure file

Compilers and external libraries path as well as compiler options are controlled using an architecture configure file in ./mychimere/mychimere-'my_arch', where **'my_arch'** is a name the user decides to give to its computing architecture. This name format for the achitecture configure is mandatory, 'my_arch' suffix is left free.

It is recommended to do the following :

```
cd mychimere
cp mychimere-void mychimere-my_arch
```

Then, the user needs to fill in its **mychimere-my_arch** file with correct paths towards compilers and external libraries (see section 2.1) present on its computer.

Some mychimere-xxxx files are provided as examples, but are related to specific clusters (local development cluster, IPSL mesocenter and TGCC HPC center), they most probably won't work on the specific user achitecture.

Explanations about each fields needed for correct compilation are provided in **mychimere-void**. The first section :

```
#-------------------------------------------------------------------------------
#       Compilers
#-------------------------------------------------------------------------------
export my_compilerF90=                          # Path to Fortran 90 compiler
export my_compilerC=                            # Path to C compiler
export my_compilerCpp=                          # Path to C++ compiler
```

corresponds to the path of the compilers needed to compile the system.

The following sections :

```
#-------------------------------------------------------------------------------
#       MPI - parallel execution of chimere
#-------------------------------------------------------------------------------

#-------------------------------------------------------------------------------
#       HDF5  - parallel version
#-------------------------------------------------------------------------------

#-------------------------------------------------------------------------------
#       NETCDF-C  - link with HDF5 parallel
#-------------------------------------------------------------------------------

#-------------------------------------------------------------------------------
#       NETCDF-Fortran  - link with HDF5 parallel and NETCDF-C
#-------------------------------------------------------------------------------

#-------------------------------------------------------------------------------
#       GRIB  - link with jasper
#-------------------------------------------------------------------------------

#-------------------------------------------------------------------------------
#       BLITZ
#-------------------------------------------------------------------------------
```

are related to the required external libraries.
Most often the manual compilation and installation of external libraries listed in section 2.1 will produce 3 kind of directories under the /.../NameOfTheLibrary location :

- bin/ : with binary executables such as for instance : *mpirun*, *nc-config*, *nf-config*, ...
- include/ : with include files such as (extension can be *.h* ,*.mod*, ...) : *mpif.h*, *netcdf.mod*, *grib_api.mod*, *h5f.mod*, ...
- lib/ : with binary library files (extension can be *.so* or *.la*) such as : *libmpi.so*, *libnetcdf.so*, *libnetcdff.so*, *libgrib_api.so*, ..

In the mychimere-my_arch file, **my_*lib*bin**, **my_*lib*inc**, **my_*lib*lib** fields correspond to those **bin**, **include** and **lib** directories.

If the installation of the external libraries is done through a package manager (for instance synaptic, apt-get on Ubuntu), it is most likely that libraries will be installed together under system directories such as : */usr/bin*, */usr/lib*, */usr/lib/x86_64-linux-gnu/*, */usr/include*. It is under the responsibility of the user to find the correct library paths.

The last section :

```
#-------------------------------------------------------------------------------
#       Makefile header needed to compile CHIMERE and WRF
#            - with this architecture configuration -
#-------------------------------------------------------------------------------
```

corresponds to the files containing compiler rules for CHIMERE, WRF and WPS. Those rules are written in Makefile header files located in mychimere/makefiles.hdr, mychimere/config_wrf and mychimere/config_wps.
Depending on your compiler, you can most likely fill in the following :

	GCC	Intel
my_hdr	Makefile.hdr.gfortran-64-ompi	Makefile.hdr.ifort-64-ompi
configure_wrf_file_name	configure.wrf.gfortran	configure.wrf.ifort
configure_wps_file_name	configure_gfortran.wps	configure_ifort.wps

2.3.2 Compilation of the models

In any case and any use, the user needs to first compile the 4 components of the system :

1. OASIS3-MCT
2. CHIMERE
3. WRF
4. WPS

2 compilation scripts take care of those tasks : build-chimere.sh and build-wrf.sh. General help messages are displayed this way :

```
./build-chimere.sh --help
./build-wrf.sh --help
```

2.3.2.1 Compilation of OASIS and CHIMERE

After filling out the mychimere/mychimere-**my_arch** file, to compile OASIS and CHIMERE, the user has to execute the following :

```
./build-chimere.sh --arch my_arch
```

Normally, last lines of the output should be :

```
Compiling OASIS...
Compiling CHIMERE...
Compilation OK. Saving compiled code to /home/myaccount/CHIMERE/exe_PROD...
```

Log files of the compilations will be found in : compilogs/make.oasis.*???*.log and compilogs/make.*???*.log.
By default compilation is done in PROD mode, for advanced modelling switch to DEVEL mode is available by using the **- - devel** option of build-chimere.sh.
Afterwards, if the user makes changes in the chimere source code, recompilation of oasis is not mandatory and the following can be executed :

```
./build-chimere.sh --no-oasis
```

2.3.2.2 Compilation of WRF and WPS

To compile WRF and WPS, the user has to execute the following :

```
./build-wrf.sh --arch my_arch
```

Normally, last lines of the output should be :

```
Compiling WRF
1. Compiling /home/myaccount/CHIMERE/WRFV3_cpl371/main/real_nO.exe without OASIS
2. Compiling  /home/myaccount/CHIMERE/WRFV3_cpl371/main/wrf_O.exe with OASIS

   Compiling WPS...
      Compilation Success
```

Log files of the compilations will be found in : compilogs/make.em_real_without_oasis.????.log, compilogs/-make.wrf_with_oasis.????.log and make.wps.????.log.
The user can also decide to compile only WRF or WPS by doing the following :

```
./build-wrf.sh --arch my_arch --wrf
./build-wrf.sh --arch my_arch --wps
```

In any case a first compilation of WRF is mandatory to compile WPS.

2.3.2.3 Whole compilation

Instead of running the previous steps manually, you can add these lines in a script to make the whole compilation (keeping in mind that the first time everything needs to be compiled) :

```
# Clean and recompile OASIS
icompoasis=1

# Clean and recompile CHIMERE
icompchim=1

# Clean and recompile  WPS
icompwps=1

# Clean and recompile WRF and WPS
icompwrf=1

#-----------------------------------------------------------------------
# arch configuration file of the form : machine.compilo (ex : irene.ifort)
machine=my_machine
compilo=my_compiler

# prod or devel
typmod=prod

#-----------------------------------------------------------------------
# CHIMERE compilation

if [ $icompchim == "1" ] ; then
         touch  exe_PROD exe_DEVEL
        \rm -r exe_PROD exe_DEVEL

        if [ $icompoasis == "1" ] ; then
                ./build-chimere.sh  --arch ${machine}.${compilo} --${typmod}
        else
                ./build-chimere.sh  --arch ${machine}.${compilo} --${typmod} --no-oasis
fi

#-----------------------------------------------------------------------
# WRF compilation

if [ $icompwrf == "1" ] ; then
   ./build-wrf.sh  --arch ${machine}.${compilo} --wrf
fi

#-----------------------------------------------------------------------
# WPS compilation

if [ $icompwps == "1" ] ; then
   ./build-wrf.sh  --arch ${machine}.${compilo} --wps
fi
```

2.4 Run the model

2.4.1 Start from command line

The script chimere.sh drives the run of the CHIMERE system. General help messages are displayed this way :

```
./chimere.sh --help
```

4 informations in the following order are compulsory to start CHIMERE.

- the chimere.par parameter file (see Chapter 3)

- the task to execute (options s/p/f):

  s : for **s**equential (i.e. not parallelized) code. Only the pre-processing of CHIMERE will be executed (e.g. preparation of boundary conditions).

  p : for **p**arallellized code. Only the parallel computing stage of CHIMERE will be executed (here we assume that you have already run the pre-processing).

  f : for **f**ull (both of the above). The complete CHIMERE system will be executed.

- the starting date of the simulation in format YYYYMMDDHH (e.g. 2020030100)
- the number of hours of the simulation

2 optional informations may be provided :

- the excution mode –devel for debugging purpose or –prod for production mode (default is –prod)
- the restart mode –restart yes/no/optionnal (default is optionnal)

  yes : a restart end.* file is mandatory and the run will abort if the file does not exist

  no : no need for a restart file, the model will initiate with a cold start

  optionnal : if a restart end.* file exists the mode will use it as initial condition, otherwise a cold start will occure

The following command line is then used to start a CHIMERE run :

```
./chimere.sh parameter_file task starting_date number_of_hours [--devel] [--restart yes/no/optionnal]
```

This command will start the simulation and generates prints on the screen.

2.4.2 Start with a script

Instead of running the previous step manually, you can add these lines in a script to drive the full simulation.
Following this example you can redirect the prints on the screen to both the screen and a log file, to keep track of your run.
This script gives also an example of how to run a long simulation divided into several smaller steps (here a 2 month long run by step of 5 days).

```
source ./mychimere/statcodes_paths.sh

firstdate=2013061500
lastdate=2013083100

incday=5
parfile=my_chimere.par

#-------------------------------------------------------------------

curdate=${firstdate}
nh=`expr ${incday} \* 24`

#-------------------------------------------------------------------

# prod or devel
typmod=prod

#-------------------------------------------------------------------
# simulation


# First block, no restart
chimrestart=no

while [ $curdate -le $lastdate ] ; do

   datestr=`date +"%Y%m%d%H%M%S"`
   curlog=${logdir}/${logfile}.${datestr}.log

   ./chimere.sh ${parfile} f ${curdate} ${nh} --${typmod} --restart ${chimrestart} 2>&1 | tee ${curlog} || exit 1

    echo "Log file for this simulation: "${curlog}
    chimrestart=yes
```

```
    curdate=$(date -u -d "${curdate:0:8} ${curdate:8:2} ${nh} hour" +%Y%m%d%H)
done
```

2.5 Hands-on : a first test case

The easiest way for a beginner to get started with CHIMERE is to simulate the two test-cases proposed on the CHIMERE download page. All case dependent input and desired output data are available at the TestCase.tar.gz archive.
Once untarred, the TestCase.tar.gz archive contains two configuration files, one for an an online and one for an offline simulation (chimere.par.test.offline and chimere.par.test.online) and two directories.

Directories

- INPUTS/
- DESIRED_OUTPUTS/

INPUTS contains three subdirectories:

- WRFtest/ contains two netCDF files. The geo_em.d01.nc file is the WPS geog outpout and will be used to define the coordinates of all points in the CHIMERE simulation grid. The wrfout file contains WRF calculated meteorological fields that will force the CHIMERE simulation at the offline run.
- EMI/ contains emission fluxes over the CHIMERE simulation grid, calculated with the emiSURF model for the chemical mechanism of the CHIMERE simulation. datasets (produced by emiSURF model).
- AVN/ contains the GRIB files from the GFS global forecast data, necessary to force WRF with large-scale conditions in the online run.

DESIRED_OUTPUTS/ contains the outputs expected from the online and offline test runs respectively, for comparison with the results obtained by the user on his/her machine.
To run the test cases the user also needs some other case-independent input data. They are climatologies for boundary and initial conditions, landuse databases, aerosol chemistry related parameters and biogenic emission related data. They are packed in the fichier BIGFILES.tar.gz archive and may be download at the CHIMERE webpage.

Configuration Files

1. chimere.par.test.offline is the configuration file for the offline run. In this case meteorological conditions are read from a fully-prepared WRF simulation covering the CHIMERE simulation period. For this case the WRF simulation domain is identical to the CHIMERE simulation domain.
2. chimere.par.test.online is the configuration file for an online run. In this case CHIMERE and WRF models run in online-coupled mode exchanging data at *[dtphys]* frequency (look at Section 3 for detailed description of the chimere.par file). In the case simulated here both direct interactions between aerosol particles and radiation (incoming and outgoing) and indirect effects implemented by a microphysics parameterization including a scheme for cloud droplet nucleation (see Sections 7.3 and 7.4 for more details).

The section of the chimere.par configuration files that needs to be modified in both offline and online cases cases is the following.

```
[simuldir] Simulation output directory                : __FILL__
[bigfilesdir]   Big files (AEROMIN.bin, etc)          : __FILL__
[emissdir]       Anthropogenic emission Dir (stage 1)  : __FILL__
```

Note that when running a new simulation, if you don't wish to lose the previous simulation you need to change the *[simuldir]* flag

- *[simuldir]* : where outputs will be written. A directory with sufficient space and where the user has write rights.

- *[bigfilesdir]* : where the BIGFILES directory is on the disc. The subdirectory structure needs to be preserved as in the BIGFILES.tar.gz archive
- *[emissdir]* : where anthropogenic emission files are stored (monthly fluxes)

2.5.1 Offline test case

The section of the chimere.par.test.offline file that needs to be modified is the "Meteorlogy Driver". All required input datafiles are prepared. You just have to setup the configuration file chimere.par.test.offline to provide full paths to these files and directories. The places where chimere.par.test.offline should be modified are indicated by the a __FILL__ sign:

```
[$meteo_DIR]  Where are your meteo files              : __FILL__      # Path to file wrfout
[$wrf_geog]   If WRF, where is geogfile               : __FILL__      # Path to file geo_em.d01.nc
```

- *[meteo_DIR]* : full path to the directory where the wrfout file is
- *[wrf_geog]* : full path to the geog file. This file is obtained by the WRF model preprocessor (WPS) and contains the geographical coordinates of the CHIMERE simulation domain

Once these paths are set, the program can be run with the following command:

```
./chimere.sh chimere.par.test.offline f 2017010100 24
```

When the run is over, the simuldir should have a content similar to DESIRED_OUTPUTS/test.offline/.
The three netcdf files :

- **out**.2017010100_24_test.offline.nc : this file contains the simulated concentrations for each point of the 3D simulation grid and each hour of the simulation period
- **dep**.2017010100_24_test.offline.nc: this file contains 3D deposition fluxes at every *[nsdepos]* hours (cf chimere.par flag)
- **end**.2017010100_24_test.offline.nc : this file contains concentrations at the final simulation test and will be read to initialize fields at the next simulation in case of a continuous run

Make sure that the out.2017010100_24_test.offline.nc is present in the simuldir directory, and that it contains 25 time steps. Also check that there are no substantial differences between the values of your out file the DESIRED_OUTPUTS/test.offline/out.2017010100_24_test.offline.nc.
You will also find a directory named data_IDF5_test.offline. There are stocked in netcdf format all emission fluxes used as input for your simulation. They are interpolated on the simulation grid at an hourly frequency.

- **AEMISSIONS**.2017010100_24_test.offline.nc : Anthropogenic emissions
- **BEMISSIONS**.2017010100_24_test.offline.nc : Biogenic emissions
- **DEMISSIONS**.2017010100_24_test.offline.nc : Dust emissions
- **SEMISSIONS**.2017010100_24_test.offline.nc : Sea-salt emissions

2.5.2 Online test case

The user has to modify chimere.par.test.online to run this test case. The lines that need to be modified are indicated with the __FILL__ sign. Apart from the modifications than are common in both offline and online cases (see previous section), for the online run you also need to specify the path to the necessary input meteorological data for the WRF simulation (boundary and initial conditions):

```
[dgrb]   Dir for input meteo grib files : __FILL__
```

The *[dgrb]* provides the directory of the GFS global forecast data. These data are provided for this test-case simulation in the AVN subdirectory of the in TestCase/ directory.
Once these paths are set, the program can be run with :

```
./chimere.sh chimere.par.test.online f 2017010100 24
```

When the run is over, the simuldir should have a content similar to DESIRED_OUTPUTS/test.online/.
The same outputs as in the online case should be found in your simuldir directory, with extension test.online.nc. Additionally, the online run has produced a wrfout file : wrfout_d01_2017-01-01_00:00:00. Same as for the out file, this file should be compared to DESIRED_OUTPUTS/test.online/wrfout_d01_2017-01-01_00:00:00.

2.5.3 Output on screen at run time

The outputs on screen give the correct reading of input file or not, shows the meteorological pre-processing phase and the CHIMERE simulation itself by printing some variables to give informations about the run. The output on screen may be very simple (with the flag chimverb=0) to very very verbose (with the flag chimverb=5).
After the initialization phase, the outputs on screen have an hourly frequency and are as:

```
 DATE            O3       NO2      T_2m(C)    PBLH(m)   DTphys(mn) DTchem(mn)  CPU(mn)   end(h)
2016062600      18.5       0.5      27.5       378.      20.00      5.00       0.80       1h36
2016062601      18.3       0.5      27.2       384.      20.00      5.00       3.44       4h10
2016062602      18.2       0.4      26.8       439.      20.00      5.00       1.20       3h32
2016062603      18.7       0.4      26.3       540.      20.00      5.00       1.17       3h11
2016062604      19.2       0.4      25.4       376.      20.00      5.00       1.23       3h00
2016062605      19.3       0.4      24.9       301.      20.00      5.00       1.20       2h51
2016062606      17.6       0.3      24.9       312.      20.00      5.00       1.16       2h44
2016062607      17.7       0.2      25.8       388.      20.00      5.00       1.18       2h39
...
```

These outputs are:

- The current date in format `yyyymmddhh`
- The first two species defined in the OUTPUT_SPECIES file. In this case, the output is for O_3 and NO_2
- The 2m temperature (in degree)
- The boundary layer height (in meters)
- The physical time-step (in mn)
- The chemical time-step (in mn)
- The CPU used for each hour of simulation (in mn)
- An estimation of the remaining time before the end of the simulation (in mn)

All these values are given for a cell located at the middle of the horizontal domain.

2.5.4 The output files

Four different output files are delivered:

1. out.[label].nc: contains concentrations fields and meteorology regridded on the CHIMERE mesh
2. dep.[label].nc: contains dry and wet deposition fluxes
3. end.[label].nc: for initialization and restart
4. wrfout_${dom}_YYYY_MM_DD_HH:mm:ss: in case of online runs only, this is the WRF model output with all the simulated meteorological fields

These files are encoded in the NetCDF format, which presents the following advantages:

- portability among different architectures of computers
- self-documentation since a lot of meta-data are included in the file itself
- direct access
- compatibility with many free or commercial post-processing tools

Output of chemical and meteorological fields in file "out.[label].nc" Three-dimensional chemical fields are written in an output out.[label].nc file. The selected species correspond to those listed in the OUTPUT_SPECIES parameter file. Default concentrations values are expressed in ppb. These values may be transformed into μg.m^{-3} if the molar mass of each species is added as a second column in the parameter files OUTPUT_SPECIES located in the chemprep directory (the default value is zero). In case of use of FastJX for the photolysis rates, an output of the optical depth for clouds and aerosols is automatically added as output in the out.[label].nc file.
example: In the file chemprep/data/output_species/OUTPUT_SPECIES...: If the line is "`O3 0.`", ozone results will be in ppb. If the line is "`O3 48.`", the results will be in μg.m^{-3}.
Hourly 2-D and 3-D meteorological and turbulence related fields calculated on the CHIMERE grid are also provided in the output out file. For an "off-line" run, meteorological fields (e.g. from WRF or CAMS) are just interpolated on the CHIMERE grid. Turbulence related fields are calculated with a meteorological diagnostic program embedded in the CHIMERE core. The variables are listed in Table E.1 , p.257.

Output deposition fluxes in file "dep.[label].nc" This file contains fluxes integrated over the chemical time-step duration. The calculations are done in the model core and are written in independent files. For the dry deposition flux, only the first vertical cell is taken into account. On the other hand, for the wet deposition, fluxes are sumed up the whole column since, in the model, each vertical level may contribute to a net sink. Units of these two integrated fluxes are g/cm^2.

All required fields for the initialization of a CHIMERE simulation in file "end.[label].nc" The structure of this file is the same as the "out.[label].nc" file. The main goal of this file is to save all concentrations fields every $nsconcs hours. For example, for a second run following in time the first one, this "end..." file will be link to a "ini..." file and thus used to give realistic initial conditions during the restart.

The WRF model output in the file "wrfout_${dom}_YYYY_MM_DD_HH:mm:ss" This is the WRF model's output. All fields have one record calculated at each CHIMERE simulation time-step.

3 The CHIMERE parameter file

3.1 General structure

All parameters driving a simulation, except from the starting date/hour and the duration of the simulation (in hours) which are provided with the command line when launching the chimere.sh script, have to be filled in the chimere.par file. This file is an ascii file and is organized in columns. The first column provides the parameter name and should not be modified by the user. Each of the following columns corresponds to a different simulation over a different geographical domain in case of nesting. The user needs to provide the appropriate values in the corresponding column: $1^s t$ column for a single run in case of not nested simulation (or the parent domain in case of nested runs) and the following columns for the inner domains in case of nested runs.

```
   [$parameter_name] This parameter description  : value1, value2,..  valueN
```

All parameters initialized this way will be visible by the CHIMERE scripts during the simulation. Simulation parameters, such as lab, dom, etc., are given as comma-separated values after the colon character (":") in chimere.par. For each simulation, you need to specify in the top of the chimere.par file the run value which corresponds to the column from which CHIMERE should read the parameters in the chimere.par file. For example, to run CHIMERE over three nested domains (cf. Section 5.3) you would set

```
#-------------------------------------------------------------------------------------------
#    CHIMERE parameter file
#       Separate parameters for multiple runs with commas, e.g., Nested... : no, yes
#       Only use absolute paths for files and directories
#-------------------------------------------------------------------------------------------
   runs :  1 2 3
```

Therefore, the user can run multiple simulations with a single call to chimere.sh. The script will subsequently run simulations with parameters defined by the columns whose numbers are listed in the run" parameter in the top of chimere.par file. If for some parameter the number of comma-separated values is less than the number of the simulation requested, the last comma-separated value will be used for that simulation. For instance, you would typically save concentrations to the restart file *end....nc* at the same frequency for all your simulations, thus having only one value of the *$nsconcs* parameter. The complete chimere.par file is presented and commented below.

3.2 Analysis or forecast

```
# Analysis or forecast
   [$forecast] Forecast(1) or analysis(0) run                : 0
```

This flag remains in this parameter file but is used only for biomass burning data management.

3.3 Nesting, parallelism and vertical mesh

```
# CHIMERE Simulation Domain
   [$nested]               Nested run? (yes/no)               : no
   [$dom]                  CHIMERE domain                     : MED60
   [$ntotprocs]            Number of available processors     : 60
   [$ratioproc]            Ratio procs #chimere/#wrf (real)   : 1
   [$nlevels]              Number of vertical layers          : 20
   [$top_chimere_pressure] Top layer pressure                 : 300
   [$first_layer_pressure] First layer pressure               : 998
```

1. *[nested]*: If you want to use the nesting, you have to specify 'yes' for each column corresponding to a nested domain. For example, for three nested domains, with 'dom2' nested inside 'dom1' and 'dom3' nested inside 'dom2', the lines become:

```
   [$nested]              Nested run? (yes/no)              : no, yes, yes
   [$dom]                 CHIMERE domain                    : dom1, dom2, dom3
```

2. *[dom]* defines the name of the simulation domain. This parameter has to be defined in the domains/domainlist.nml file.

3. *[ntotprocs]* option is for the parallel mode. If you are 'offline' the number of subdomain are calculated for CHIMERE only. If you are in 'online' mode, the number of available processor is split between WRF and CHIMERE. The *[ratioproc]* number is a real and is to manage the number of processors affected to WRF and CHIMERE respectively.

4. For the vertical mesh, there is only three parameters:

   - *[nlevels]*: the number of vertical levels
   - *[first_layer_pressure]*: the pressure of the top of the first cell
   - *[top_chimere_pressure]*: the top of the domain , in hPa. The surface is, by default, considered as p_0=1000 hPa. The thickness of the first cell is thus p_0-'first_layer_pressure'.

3.4 Simulation output parameters

```
# Simulation output parameters
 [$online]    on-line [1] or off-line [0] modeling          : 1
 [$cpl_case]  coupling case                                 : 3
 [$lab]       Simulation label                              : test5
 [$clab]      Coarse domain label                           : none
 [$sim]       Current run label (output files' ending)      : ${idatestart}_${nhours}_${lab}
 [$simuldir] Simulation output directory                    : /${myresults}/CHIMOUT-${lab}

# Output parameters
 [$accur]     Output species detailed (low/full)            : full
 [$nsconcs]   Conc. save freq. (hr), file end...            : 24
 [$nsdepos]   Deposition save freq. (hr), file dep...       : 6
 [$accurmet] Meteo detailed (low/medium/full)               : full
```

These flags manage the online coupling and the outputs. This part is new and more details are provided in $6.2, p.58.

1. *[online]* and *[cpl_case]*: The online management depends on these two flags only.

Mode	Online choice	Coupling type	Exchange frequency	Interactions
Offline	*[online]*=0		1 hour	No
Online	*[online]*=1	*[cpl_case]*=1	*[dtphys]*	No
	[online]=1	*[cpl_case]*=2	*[dtphys]*	Direct effect
	[online]=1	*[cpl_case]*=3	*[dtphys]*	Indirect effect
	[online]=1	*[cpl_case]*=4	*[dtphys]*	Direct + indirect effects

2. *[lab]* and *[clab]*: You can define here a character chain to identify your output files. This is the label of files out, dep, end.

3. *[accur]* may be full or low.

   - accur=low → all gaseous species and aerosols as the total mass of each species
   - accur=full → all gaseous species and aerosol with their size distribution.

4. *[nsconcs]* is the frequency of the outputs. *[nsconcs]* is only for the restart, generally 24 is enough (i.e one possible restart per day).

5. *[nsdepos]* is the frequency of the outputs in the dep file for th edeposition fluxes. For example:

   - nsdepos=1: deposition fluxes are written every hour

- nsdepos=6: deposition fluxes are written every six hours and are cumulated during the last six hours
- nsdepos=0: The output file dep is not created. This saves disk place and the simulation is faster.

6. *[accurmet]* The same as *[accur]* but for the meteorological variables written in the out file.
7. *[sim]* is the other part of the output files label. It is strongly recommended to not change this path.
8. *[simuldir]* is the directory path where you want to put the results.

3.5 Databases

```
# CHIMERE mandatory databases
 [$bigfilesdir]   Big files (AEROMIN.bin, etc)             : /${myaccount}/BIGFILES
 [$lunom]         Land Cover GLCF/GLOBCOVER/USGS           : USGS
 [$landcover_dir] Land Cover directory                     : ${bigfilesdir}/LANDCOVER
 [$megan_data]    MEGAN data root                          : ${bigfilesdir}/MEGAN
 [$imegan]        MEGAN database choice (1=2.5', 2=30")    : 2
 [$emissdir]      Anthropogenic emission Dir (stage 1)     : /${myaccount}/anthro_emis
 [$fire_emissdir] Fire emission Dir (stage 1)              : /${myaccount}/fires_emis
```

1. *[bigfilesdir]*: This is the directory where the user can store all useful databases. This is created when downloading the CHIMERE data.
2. *[lunom]*: The name of the database to use to deifne the landuse.
3. *[landcover_dir]*: The directory where are stored these landuse databases.
4. *[megan_data]*: The directory where are stored tthe MEGAN model databases.
5. *[imegan]*: Choice for the resolution of the MEGAN model database. The use of the fine resolution of 30s is possible with this CHIMERE model version.
6. *[emissdir]*: The directory where are stored the anthropogenic emissions (possibly generated using the emiSURF program).
7. *[fire_emissdir]*: The directory where are stored the biomass burning emissions (possibly generated using the API-FLAME program).

3.6 Emissions

```
# Emissions
 [$emistyp]          Anthropogenic emission format type    : emisurf
 [$asurfemis]        anthro surface emissions? (1/0)       : 1
 [$apointemis]       anthro point emissions?   (1/0)       : 0
 [$iuse_firemis]     Fire emissions?   (1/0)               : 1
 [$ifirevprof]       Fire vertical profile shape (1/2/3)   : 1
 [$iusebemis]        Biogenic emissions? (1/0)             : 1
 [$iusedms]          DMS marine emissions? ([1-4]/0)       : 1
 [$iusedust]         Mineral dust emissions? (0-3)         : 1
 [$icuth]            u* threshold scheme                   : 1
 [$ifluxv]           vertical flux scheme                  : 2
 [$ifecan]           Use of soil moisture [1] or not [0]   : 1
 [$iusesalt]          Sea salts emissions? (0/1/2/3/4)     : 1
```

All emissions are managed in this section.

1. *[emistyp]* describes the kind of input files you have. 'emisurf' is a generic name for all emissions if they are generated with the emiSURF program.
2. For *[asurfemis]*, *[point_emissions]*, *[iuse_firemis]*, *[iusebemis]* and *[iusedust]* the possible cases are:
   - 0 = emissions not used
   - 1 = emissions used

3. *[ifirevprof]* is the way to define the vertical shape of the biomass burning emissions.
4. *[iusedms]* is designed to manage the DMS emission parameterizations. The possible options are:
   - iusedms=0 → no DMS emissions
   - iusedms=1 → scheme of [Liss and Merlivat, 1986]
   - iusedms=2 → scheme of [Wanninkhof, 1992]
   - iusedms=3 → scheme of [Nightingale et al., 2000]
   - iusedms=4 → scheme of [Elliott, 2009]
5. Mineral dust emissions:
   - *[iusedust]*: when this flag is set to 1, mineral dust emissions are calculated. The main options are described below (more details are proposed in §10.4, p.107).
     - iusedust=0 → No mineral dust emissions
     - iusedust=1 → Mineral dust emissions with only one species called DUST
     - iusedust=2 → Mineral dust emissions with mineralogy. Emissions are calculated for 8 chemical elements: magnesium (mg), phosphorus (P), calcium (Ca), manganese (Mn), iron (Fe), aluminium (Al), silicon (Si) and potassium (K).
     - iusedust=3 → Mineral dust emissions with mineralogy. Emissions are calculated for 12 mineral species such as Smectite, Illite, Hematite etc.
   - *[icuth]* is for the u* threshold estimation with:
     - icuth=0 → [Iversen and White, 1982] scheme
     - icuth=1 → [Shao and Lu, 2000] scheme
   - *[ifluxv]* is for dust production model for saltation and sandblasting, with:
     - ifluxv=0 → [Marticorena and Bergametti, 1995] scheme
     - ifluxv=1 → [Alfaro and Gomes, 2001]
     - ifluxv=2 → [Alfaro and Gomes, 2001] + optimization following [Menut et al., 2005]
     - ifluxv=3 → [Kok et al., 2014] scheme
   - *[ifecan]* is for the use of soil moisture to inhibit mineral dust emissions, following the [Fecan et al., 1999] scheme.
     - ifecan=0 → no use
     - ifecan=1 → use
6. *[iusesalt]* is the way to parameterize the sea-salt emissions:
   - isaltp=0 → no emissions
   - isaltp=1 → scheme of [Monahan, 1986]
   - isaltp=2 → scheme of [Monahan, 1986] with size distribution of [Martensson et al., 2003]
   - isaltp=3 → scheme of [Martensson et al., 2003]
   - isaltp=4 → scheme of [Grythe et al., 2014]

3.7 Meteorology Driver

```
# Meteorology Driver

# in case of offline
 [$meteo]    What is your meteo driver (WRF,ecm)            : WRF
 [$metdom]   Meteo driver domain (offline and online)       : d01
 [$meteo_DIR]  Where are your meteo files                   : __myWRFoutputs__
 [$meteo_file] Prefix of meteo files                        : WRFOUTan_${dom}*
 [$wrf_geog]   If WRF, where is geogfile                    : ${meteo_DIR}/WPS/geo_em.${metdom}.nc

# in case of online
 [$geodata]   Static input data for WPS                     : __mygeog_data__/geog_data_v41
 [$dgrb]      Dir for input meteo grib files                : __myglobal_meteo__/AVN
 [$wrfrestart] WRF restart param (true / false)             : optional
 [$wrf_restart_dir] WRF restart dir                         : ${simuldir}
 [$wrf_tstep_grid] (WRF default is 6 in 6*dx)               : 4
```

1. *[meteo]*: Depending on the flag, different programs will be used. The two possibilities are *WRF* and *ecm*. If the user wants to use his own data, he has to write, for example, *mesonh* and write the corresponding program interf_mesonh to read the data and build the exdoumout.nc datafile.

2. *[metdom]*: The name of the domain is, in general, in the filename. To provide this name enables to automatically search in the directory even if there is outputs from different simulations.

3. *[meteo_DIR]*: If you want to use the model offline, you have to specify the directory where are the WRF output files.

4. *[meteo_file]*: In order to select the right files to make a list, you also have to give the prefix of these files.

5. *[wrf_geog]*: It is also possible to specify the path of the geog file used for the WRF domain. In this case, CHIMERE will run on the same horizontal grid than WRF, avoiding horizontal interpolation of meteorological variables.

6. *[geodata]* and *[dgrb]*: If you want to use the online version, the use of WRF as meteorological model is mandatory (the coupling being implemented in this model only). You have to specify some paths corresponding to the *[geog]* data and where are the global fields to force WRF.

7. *[wrfrestart]*: For the WRF simulation management in case of chained run, we added the *[optional]* value able to check if this is the first simulated period (cold start) or a period needing a restart. To keep this flag as *[optional]* is strongly recommended.

8. *[wrf_tstep_grid]]*: In the WRF documentation a value between 3 and 6 is recommended. For the time-step calculation, a maximum is Δt=6 $\times$ dx, where dx is the horizontal grid spacing in km. But the choice of this value is not extremely important since we distribute the coupled version with a WRF namelist designed to use the adaptive time-step.

3.8 Chemical boundary conditions

```
# Chemical boundary conditions
#    0 - do not use
#    1 - use and build
#    2 - check if files exist and build only
 [$iusebound]    Use and build Boundary conditions? (0/1/2) : 2
 [$iuseini]      Use and build Initial conditions? (0/1/2)  : 2
 [$nest_topconc] Use global top conc? (when same vert grid) : 1
 [$bcdir]        Boundary conditions                        : ${bigfilesdir}
 [$bcgas]        BC Gaseous Species                         : LMDz4_INCA3_96x95x19_v2013c
 [$bcgasdt]      Use of gas bc (0/1/2)                      : 1
 [$bcaer]        BC non-dust  Aerosols                      : LMDz4_INCA3_96x95x19_v2013c
 [$bcaerdt]      use of aer bc (0/1/2)                      : 1
 [$bcdust]       BC Dust                                    : GOCART
 [$bcdustdt]     Use of dust bc (0/1/2)                     : 1
```

1. For *[iusebound]* and *[iuseini]*, the possible values are:

   - 0 = do not use
   - 1 = forced build then use
   - 2 = check if files exist, build only if not existing then use

2. *[nest_topconc]* is as:

   - nest_topconc=0 $\rightarrow$ use coarse simulation's
   - nest_topconc=1 $\rightarrow$ use global model's

3. The use of boundary conditions is possible with three sub-choices: for gaseous species (bcgas), non-dust aerosols (bcaer) and mineral dust aerosols (bcdust). For the way to use it, three choices:

   - bcgasdt=0 $\rightarrow$ no use
   - bcgasdt=1 $\rightarrow$ use of climatological values
   - bcgasdt=2 $\rightarrow$ use of time varying values using linear temporal interpolation

3.9 Chemistry and deposition / resuspension options

```
# Chemistry and deposition / resuspension options
 [$mecachim]  Chemistry mechanism (0..2)                    : 0
 [$aero]      Chemically-active aerosols (1/0)              : 1
 [$nbins]     Number of aerosol size sections               : 10
 [$carb]      Include carboneceous species? (0..4)          : 1
 [$soatyp]    Secondary organic aerosol scheme (0..2)       : 2
 [$trc]       Include tracers?                              : 1
 [$nequil]    ISORROPIA coupling (1) / tabulation (0)       : 0
 [$npeq]      - equilibrium calc freq (phys steps, < 5)     : 1
 [$ithermo]   Choice of SOA thermodynamics  (0..3)          : 1
 [$iadrydep]  Aero dry deposition (1=Wesely, 2=Zhang)       : 2
 [$resusp]    Resuspension [1] or not [0]                   : 1
```

1. For *[mecachim]*, the options are (see also §12,p.135):
   - mecachim=0 → No gas phase chemistry
   - mecachim=1 → melchior complete
   - mecachim=2 → melchior reduced
   - mecachim=3 → SAPRC-07-A
2. *[aero]* designs the fact to model chemically active aerosols (1) or not (0).
3. *[nbins]* is the number of bins for the aerosols and a value between 6 and 10 is recommended to have realistic results.
4. The *[carb]* option manages the SOA formation from Primary Organic Aerosols (POA). It mainly depends on your anthropogenic emissions and the availability about the speciation of OCAR and BCAR.
   - carb=0 → no carboneceous species
   - carb=1 → carboneceous species (OCAR and BCAR) are used.
   - carb=2 → carboneceous species (POA and BCAR) are used. POA (POA+IVOC+SVOC) are splitted in 9 volatility bins. Only the functionalization process is taken into account. See [Zhang et al., 2013].
   - carb=3 → carboneceous species (POA and BCAR) are used. POA (POA+IVOC+SVOC) are splitted in 9 volatility bins. Functionalization, fragmentation and non volatil bin are taken into account. See [Cholakian et al., 2018].
5. The *[soatyp]* manages the SOA formation from VOC. It is active only if aero=1. The possible values are:
   - soatyp=0 → no SOA chemistry
   - soatyp=1 → single-step oxydation (simple)
   - soatyp=2 → single-step oxydation (medium)
   - soatyp=3 → single-step oxydation (complex)
   - soatyp=4 → H^2O (Hydrophilic/Hydrophobic Organics) reduced mechanism [Couvidat et al., 2018]
   - soatyp=5 → H^2O reduced mechanism [Couvidat et al., 2018]
   - soatyp=6 → VBS scheme [Zhang et al., 2013]
   - soatyp=7 → VBS scheme with fragmentation process [Cholakian et al., 2018]
6. *[trc]*: select if you want to add tracers or not.
7. *[nequil]*: the choice for the inorganic equilibrium calculation by using a look-up table (0) or the ISORROPIA model (1).
8. *[npeq]*: Frequency of call of ISORROPIA and SOAP. These modules are called every *npeq* chemical time-step.
9. *[ithermo]* is the choice of the thermodynamics method for the SOA.
   - ithermo=0 → no use of SOAP
   - ithermo=1 → SOAP with ideality for Raoult and Henry laws
   - ithermo=2 → non-ideality is computed with UNIFAC
   - ithermo=3 → long and medium range interactions are taken into account
10. *[iadrydep]*: There is two parameterizations for the dry deposition fluxes calculations:

- iadrydep=1 → scheme of [Wesely, 1989]
- iadrydep=2 → scheme of [Zhang et al., 2001]

The fact to calculate and apply dry deposition fluxes is mandatory.

11. *[resusp]* is to select if the resuspension scheme from [Loosmore, 2003] is used or not. This resuspension is applied only for the percentage of horizontal cell being 'urban' and for the PPM species. See details §10.6, p.120. The fact to apply resuspension is optional.

3.10 Numerical solution parameters

```
# Numerical solution parameters
 [$dtphys]      Physical time steps (in mn)                 : 20
 [$dtchem]      Chemical time step (in mn)                  : 20
 [$ngs]       Number of Gauss-Seidel iterations             : 2
 [$nsu]        - during spinup                              : 3
 [$irs]        - number of spinup hours                     : 1
 [$iadv]      Advection scheme (0..2)                       : 2
 [$iadvv]     Vertical advection scheme (0..1)              : 1
 [$imethod]   Numerical method (twostep:0, splitting:1)     : 0
 [$splitcfl]  (if splitting) use CFL for trsp only (1/0)    : 1
 [$is_diagwinw] Diagnose vertical wind from horiz components : .true.
```

1. *[dtphys]* and *[dtchem]* are the physical and chemical time-step, in minutes. The physical time-step is fixed but the chemical time-step may change during the simulation, depending on the CFL. This is automatically managed in the model (see §4.6, p.38).

2. With *[ngs]*, the number of Gauss-Seidel iteration ensures the stability of the chemical concentrations evolution. A minimum value of ngs=2 is recommended.

3. *[nsu]* and *[irs]*: It is also important to well start a simulation and the default values of nsu=3 and irs=1 are also recommended.

4. *[iadv]*: The advection schemes are for the horizontal part as:

   - iadv=0 → upwind
   - iadv=1 → PPM
   - iadv=2 → VanLeer

   [iadvv] for the vertical part:

   - iadvv=0 → upwind
   - iadvv=1 → VanLeer

5. *[imethod]*: A new choice is available for the time integration: using the splitting operators or Two-Step. This is described in §4.6, p.38.

6. *[splitcfl]*: This flag can be used only if *[imethod=1]*, representing the use of the operator splitting. In this case, you can authorize the code to use different time-step for transport anc chemistry. If *[imethod=0]*, the value of *[splitcfl]* is not considered.

7. *[is_diagwinw]*: a new choice where you can use a diagnostic vertical wind component or directly the one calculated by the meteorological model. Possible values are *.true.* or *.false.*.

3.11 CHIMERE Subgrid Processes

```
# CHIMERE Subgrid Processes
 [$ideepconv]   Deep convection activation (1/0)             : 1
 [$urbancorr]   Urban correction [1] or not [0]              : 0
 [$ilidar]      Lidar profiles [1] or not [0]                : 0
 [$idiagblh]    Use BLH of meteo driver (0) or diagnose (1)  : 0
 [$idiagustar]  Use u* of meteo driver (0) or diagnose (1)   : 1
 [$idiagshf]    Use Surf.Heat Flux of meteo (0) or diag. (1) : 1
 [$sgmodel]      Subgrid scale model for anthropogenic emi   : 0
```

These processes are activated (=1) or not (=0).

- *[ideepconv]*: For the calculation of deep convection fluxes with the [Tiedtke, 1989] scheme (more details p.69).
- *[urbancorr]*: By activating this flag, the wind speed is slightly reduced in the urban model cells and the surface sensible heat flux slightly increase (more details p.67).
- *[ilidar]*: With this flag, model-to-observations calculations are activated. In this case, pseudo-lidar profile are estimated (zenith and nadir), (more details p.151).
- *[idiagblh]*: This flag enables to read the BLH provided by the forcing meteorological model (WRF, IFS, Arpege etc.) or to diagnose it using the schemes implemented in CHIMERE, [Troen and Mahrt, 1986] (more details p.66).
- *[idiagustar]* and *[idiagshf]*: same as for BLH. Read from meteo file or diagnose in CHIMERE the friction velocity u_* and the surface heat fluxes Q_0 and L_0.
- *[sgmodel]* : This flag activates the subgrid scale calculation of chemical species concentration by taking into account the heterogeneity of surface anthropogenic emissions [Valari and Menut, 2010]. More details are given in Section 9.3.

3.12 Flags whether to run individual dynamic interfaces and the Core

```
# Flags whether to run individual dynamic interfaces and the Core
 [$imakechemprep] Make chemical input data                   : 2
 [$istopchemprep] Make chemical input data and exit          : 0
 [$istopdom]      Make domains and exit                      : 0
 [$imakemeteo]  Run meteo interface? (0-2)                   : 1
 [$imakeaemis]  Build anthropogenic emissions? (0-2)         : 1
 [$imakefemis]  Build fire emissions? (0-2)                  : 1
 [$imakerun]    Run Chimere Core? (0-1)                      : 1
 [$runwrfonly]  Run WRF without CHIMERE                      : 0
 [$chimverb]    Print on screen verbose or not [0:5]         : 2
 [$timeverb]         Print exec time on screen [0:5]         : 0
# Miscellaneous
 [$clean]            Clean mode (full, light, or none)     : none
```

1. *[imakechemprep]*: The chemical data depends on all options. For each set of options, an inputdata directory is created. If the directory is already created, the code continues. Except if the user selects 1 to force the directory creation.

   imakechemprep = 0 : No preparation. We consider the directory already exists and is OK
   = 1 : Force building of data
   = 2 : Check is directory exists. If yes, continue.

2. *[istopchemprep]*: This flag enables to create the inputdata and stops just after, to manually modify the chemical mechanism before a simulation.

   istopchemprep = 0 : Prepare the data if necessary and continue.
   = 1 : Prepare and forced stop.

3. *[istopdom]*: This flag enables to create an horizontal domain using the WPS and WRF tools and to stop. It is useful if the user wants to check the domain before launching a simulation or use the geog created file for the emiSURF program.

4. *[imakeaemis]*:

- imakeaemis=0: The emission file in the CHIMERE format, AEMISSIONS.${sim}.nc, already exists and will be directly read
- imakeaemis=1: Emissions are built from annual regional/global databases (if the file already exists, it is overwritten). The emiSURF program is used.
- imakeaemis=2: The program checked the existenz of the AEMISSIONS.${sim}.nc file. If the file already exists, emissions are directly read. If it doesn't exist, the case *[imakeaemis=1]* is applied.

5. *[imakefemis]*: Same as *[imakeaemis]* but for the biomass burning emissions.

6. *[imakerun]*: This option enables to run the whole pre-processing (grid, emissions, meteo, chemical data etc.) but to stop just before the simulation. It is a way to check all required data before starting a costly simulation.

7. *[runwrfonly]* : By activating this option only the WRF related executables will be launched (preprocessing: geogrid.exe, ungrib.exe, metgrid.exe and real.exe, and WRF simulation with wrf.exe). CHIMERE will not run. The frequency of time-slots in the WRF simulation output file is equal to the *[dtphys]* parameter. Setting *[dtphys]* to 60 you will have an hourly frequency (one value every 60 minutes) in your wrfout file.

8. *[chimverb]* flag manages the verbosity during a simulation: from 0 (quiet) to 5 (very verbose). chimverb=0 is recommended.

9. *[clean]* is the way to clean the temporary files at the end of a simulation. This impacts the size of your output directory.

   - clean=full: all is cleaned except the out files
   - clean=light: only temporary files are removed
   - clean=none: all is keeped

4 CHIMERE general code structure

4.1 Directories structure

The $chimere_root top directory contains individual files, the top-level script chimere.sh, and subdirectories. These subdirectories are:

- chemprep/: Contains data files for gas, aerosols and chemical mechanisms.
- domains/: Files describing the simulation domain (mesh, landuse, roughness, LAI etc.)
- mychimere/: Contains configuration files for compilation of CHIMERE, WRF and WPS with various architectures and compilers
- scripts/: Contains the scripts called by the top-level script.
- src/: Contains the model source code.
- oasis3-mct/: The OASIS-MCT code for the coupling between WRF and CHIMERE
- WPS361/: The WPS 3.6.1 version necessary to make the domain for WRF
- WRFV3_cpl371/: The WRF 3.7.1 version with the modified routines for the coupling

Other subdirectories are created when running the model:

- compilogs/, containing the compilation log files
- exe-${compil_mode}/, where ${compil_mode}/ is DEVEL for the execulables compiled with the –devel option, PROD otherwise.
- $simuldir/: An *output* directory where simulation output files are created during the run. The PATH to this directory is given in chimere.par (variable $simuldir). It does not need to be created before the run.
- $chimere_tmp/: CHIMERE *temporary directory* defined in chimere.par. This allows several simulations to be subsequently launched from a unique $chimere_root directory, given the name of the temporary directory. $chimere_tmp/ is typically a subdirectory of $simuldir/

4.2 The chimere.sh script

The chimere.sh script is the main script for the simulation. A priori, the user does not need to edit this file: all user-defined parameters are set in the chimere.par file. When this script is launched, it is considered that WRF, OASIS and CHIMERE are already correctly compiled (see Section 2.3 for the compilation procedure). A short description of this script is provided here for information about the structure of a simulation.
The main steps in this script are:

1. Read the chimere.par file
2. If online, check WRF executables files, input data and read namelist
3. chemprep: prepare the chemistry for CHIMERE if not already present (this produces a chemprep/inputdata.* directory generating the files describing the chemical mechanism for the simulation
4. domains: prepare the geo_em file for the simulation domain, if not already present. Here a stop is possible to check whether the geog file corresponds to what the user wants.
5. domains: prepare all other files related to the surface, LAI, landuse, roughness etc.
6. Initialization of all parameters and input files
7. Step 1: sequential: during this step, all processes are done on one processor. In fact, several prep-processing are launch together to save time.

- ◇ Creation of the namcouple for OASIS if online
- ◇ Creation of the boundary conditions, preparation of namelist for the meteo, check which anthropogenic emission files exist, preparation of the FastJX input files with optical properties for all the aerosol species needed for the simulation.

8. Step 2: parallel: this is the simulation with the model. Depending on the online configuration CHIMERE is launched alone or with WRF.

4.3 Parallelism

Only the model core (chimere.e) has been parallelized so far, while CHIMERE preprocessing tasks (i.e., prior to chimere.F90 call), such as the boundary condition interface, remain sequential.

To allow for a parallel execution of CHIMERE by several processors, the main domain is horizontally divided in rectangular subdomains. This cartesian division is performed in the main execution script, which takes as inputs the user-defined parameters **ntotprocs** and **ratioproc**. **ntotprocs** defines the total number of processors the user wishes to affect to his jobs, **ratioproc** defines the number of processors that will be affectd to CHIMERE for each processors affected to WRF (for online runs only : for offline simulations all the available processors are affected to the CHIMERE executable). Each CHIMERE process will be called a **worker**. There are as many workers as CHIMERE subdomains. In a first order analysis, we could consider that :

1. all workers read the chimere.nml namelist
2. worker 0 performs all initializations
3. worker 0 broadcasts initialization data to other workers (see Section 4.4.1)
4. all workers work on their own subdomains
5. each worker writes its portion of the results to the corresponding sections in the output files using netDF4 parallelized I/O capability.

Since CHIMERE has been parallelized, the computation of transport terms needed additional interactions. A worker process requires not only its subdomain's data but also a stripe of boundary data belonging to its neighbours. The reason is that to calculate a target value at point **(i,j)**, you need values for all surrounding points from **(i-3,j-3)** to **(i+3,j+3)**. So, steps 3 and 4 in the the algorithm above become:

3. worker 0 broadcasts intialization data *plus a surrounding "halo"*, to other workers
4. all workers work on their own subdomain *but communicate with other workers to update their surrounding "halo"*

In fact, to calculate a target value at point **i** and time **tn+1**, you need source values at point **i-3, i-2, i-1**, at time **tn+1**, and source values **i, i+1, i+2, i+3** at time **tn**. You cannot simply calculate target values at time **tn+1** from a set of source values at time **tn**. That is, to start the calculation of the 5th line of a subdomain, you need to know the new values of the 3 last pixels of the 5th line of its left neighbour. This is done by setting a dependency relationship between lines, and by copying the 3 last pixels of a line to the halo of the corresponding line in the right-hand subdomain. This copy occurs for each line of each subdomain, thus at a frequency much higher than the fine time step.
What we have just explained in the previous paragraph for the **i** direction also applies to the **j** direction. The dependancy relationship in both directions leads to a straightforward conclusion : *it is impossible to parallelize CHIMERE.*
We have practically solved the problem by starting the computation of each row of subdomains as if it were an independant domain, with no upper and lower neighbour, exactly as we do for the main domain in the sequential scheme. But at the end of each fine time step, we update their upper and lower halos. This provides a reasonnable smoothing between subdomains, albeit not being numerically identical to the sequential scheme.
This leads to an imperceptible numerical difference between the results of the sequential model and those of the parallel model, showing patterns parallel to the X direction when we display the difference. The amplitude of the discrepancy is negligible, it damps very quickly, and we have never observed the development of numerical instabilities from these patterns.
Each worker process directly communicates with external files. This is implemented using the NetCDF4 parallel input/output routines.

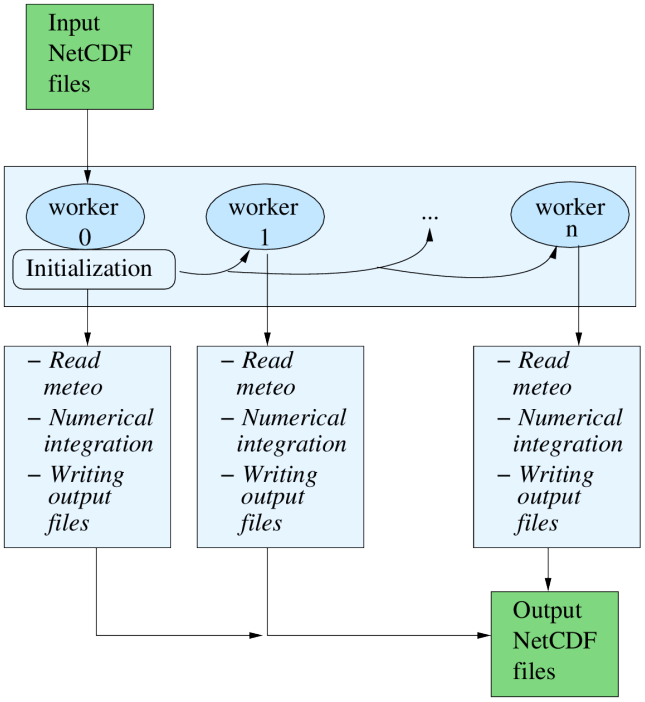


Figure 4.1: *Input/output management of NetCDF files and role of each worker.*

4.4 Routine calls sequence

The general schematic diagram of calls of the CHIMERE model src/chimere.F90 is given in Figure 4.2 . Only major routines are cited, not the utility ones.

1. The main program chimere.F90 first calls the chimere.nml reading routine read_chimere_nml on all processors. *In online runs, just after this stage each worker declares itself to the OASIS coupler.* Then the worker 0 calls the initialization routines (in inichimere.F90 and further calls) where all "static data" are read and many variables defined. At the end of the initialization step, data are broadcasted from worker 0 to other processors (see Section 4.4.1).

2. Then all processors call successively the iniworker.F90 which initialize arrays and output files in each worker and then worker.F90, the main routine of the CHIMERE model which will:

   (a) reads hourly data every hour (routine readhour.F90)

   (b) loops on the coarse time step,

   (c) outputs results on files (routine outprint.F90) and

   (d) shifts hourly data (routine renewhour).

3. At each *hourly* model time step, one needs hourly data for the start and the end of the hour in order to linearly interpolate, for meteo data, boundary conditions, biogenic and anthropogenic emissions. The code keeps only these two instants in memory. Within the coarse time step, the locvalues.F90 routine is called in order to assign all variables that will be kept constant during the *physical* time step (linearly interpolated meteo data, reaction rates etc...). *in online runs, freshly computed meteorological fields from WRF are received at each physical time step, and the requested variables are sent from CHIMERE to WRF as well.* Then a loop on the *fine* time step is carried out, with a call to the numerical solver twostep_mod.F90.

4. The twostep_mod.F90 routine calls the core routine prodloss.F90 which calculates, for all processes, the loss and production terms and sums them up. Thus prodloss.F90 calls emissions.F90 (emission terms), transmix.F90 (trans-

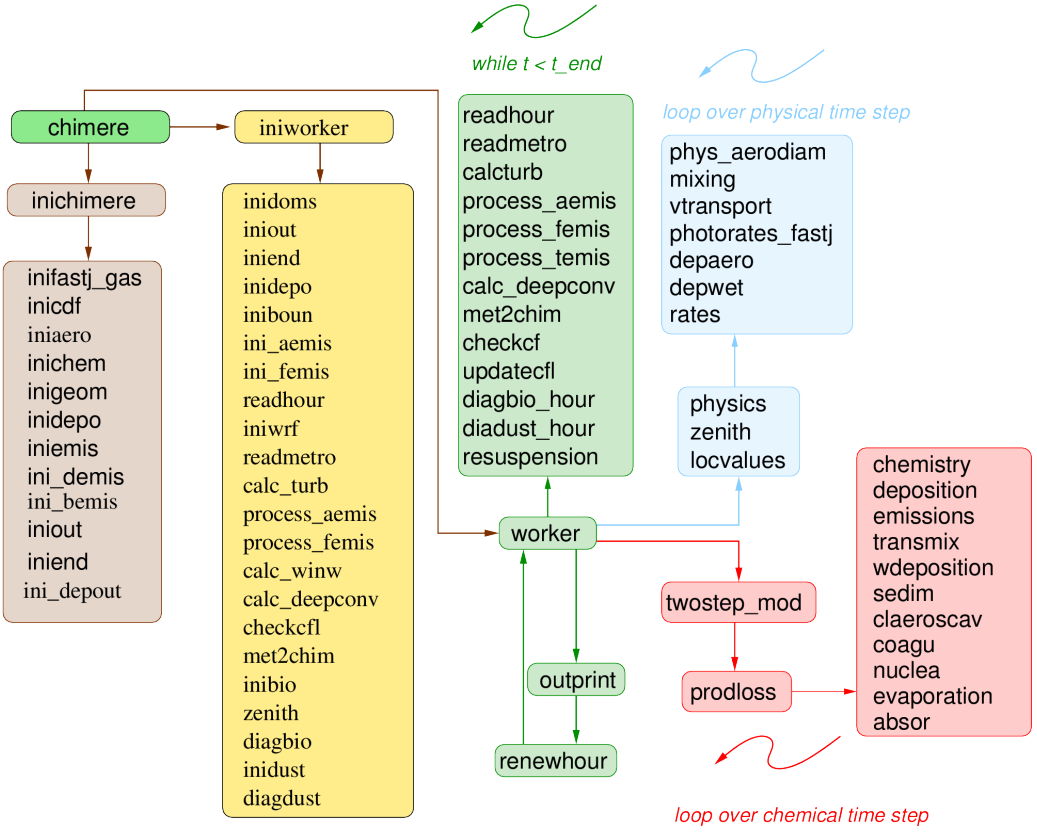


Figure 4.2: *Detailed list of CHIMERE routines and time loops*

port and mixing terms), chemistry.F90 (chemical terms), deposition.F90 and wdeposition.F90 (dry and wet deposition terms). For aerosol species prodloss.F90 calls sedim.F90 (sedimentation terms), evaporation.F90 (evaporation terms) end nucelation.F90 (nucleation term).

4.4.1 Parameter broadcast

Once the worker 0 has finished the initialization step, parameters are broadcasted to other workers. This is done by the call to the broadcast_all_once subroutine (worker_message_subs module), from the inidoms subroutine.
The broadcast_all_once subroutine will perform the following operations:

- Initialization of worker 0 domain parameters with the domains_init subroutine (pncvar module) (e.g., number of zonal cells, number of meridian cells, longitude, latitude, sub-domain indexes within the entire domain).
- These domain related parameters are then scattered from worker 0 to the other workers (subroutine scatter_params).
- All parameters are broadcasted from worker 0 to other workers (e.g., array size, time parameters).
- Now that all workers know all parameters, they are able to allocate array with a call to the worker_allocall subroutine (chimere_common module).
- Once arrays are allocated on all workers, worker 0 may broadcast the array data (subroutine bcast_arrays) and types structure for species, output and emissions (subroutine bcast_types). Note that spatialized arrays do not require to be broadcast entirely (e.g., xlong, xlati, dland) as each worker only need to know the part of the array related to its subdomain. These arrays are then scatter from worker 0 to other workers using mpi_send / mpi_recv.
- Finally, the subroutine domains_init is called by non-zero workers.

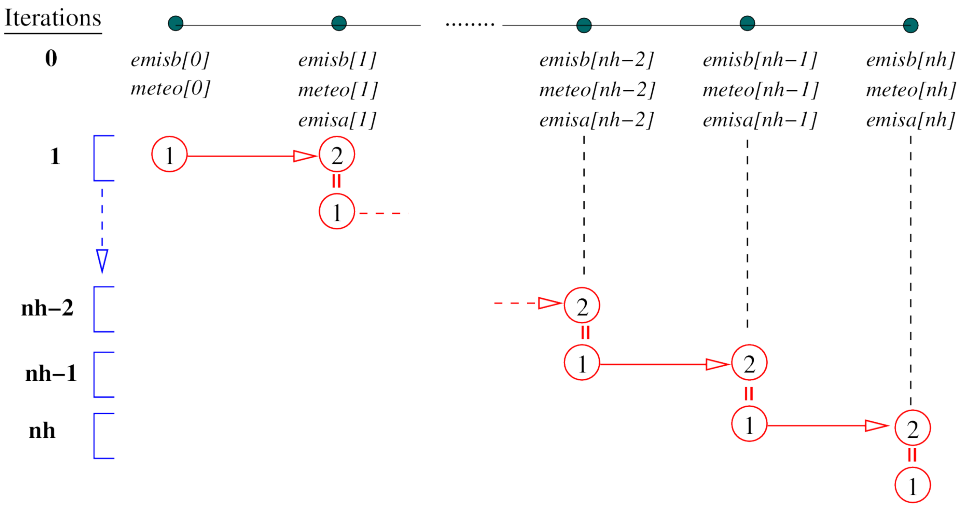


Figure 4.3: *Time processing in CHIMERE*

After the call to the broadcast_all_once subroutine, the worker 0 has transmitted every information that he has acquire during the intialization step, to the other workers. From this point, communications among workers will only consist in halos updates at each time steps.
Note that each workers, allocate arrays on its own subdomain only. However, workers sometimes require information over the whole domain, thus some variables are allocated over the whole domain. Such variables, related to the whole domain, always have the "entire_domain" suffix in their names.

4.5 How to add a new diagnostic variable to the code?

These are brief guidelines to adding a new diagnostic variable to the code. As an example let us examine how the `atb_mol_nad` variable is implemented.

1. Compute the variable: This is done in a subroutine called from the worker subroutine, e.g.,

```
if (ilidar == 1) call lidar
```

   In our example, the molecular attenuated backscatter signal `atb_mol_nad` is computed in the lidar subroutine called from worker.

2. Declare the variable in chimere_common.F90:

   We need both the variable itself and its structure in the output NetCDF file:

```
real(kind=iprec),allocatable,dimension(:,:,:,:)  :: atb_mol_nad
....
type(ncvar_type)                                  :: var_atb_mol_nad
```

3. Allocate the variable in chimere_allocs.F90:

```
allocate( atb_mol_nad(nzonalmax,nmeridmax,nverti,nwl) )
```

4. Define the corresponding NetCDF variable:

   First in iniout.F90: create and define attributes. Output NetCDF variables are currently created using the traditional NetCDF library:

```
integer :: out_atb_mol_nad_varid
....
ncstat=nf90_def_var(out_ncid,'atb_mol_nad',NF90_DOUBLE,
```

```
    & d_dimids,out_atb_mol_nad_varid)
....
ncstat=nf90_put_att(out_ncid,out_atb_mol_nad_varid,'units', &
    '[m-1 sr-1]')
ncstat=nf90_put_att(out_ncid,out_atb_mol_nad_varid,'long_name', &
    'ATB molec nadir')
```

In worker_iniout: read the NetCDF structure for parallel I/O. Once the NetCDF file is created, and the variable is defined using the standard NetCDF library, its structure can be read with the pncvar module routine for future input/output:

```
call pncvar_get_var_info(out_nc, 'atb_mol_nad', var_atb_mol_nad)
```

5. Save the variable to the output file. In the outprint subroutine, the variable's portion of each subdomain is written to the output file, using the pncvar_put_var subroutine of the pncvar module:

```
toprint4d(:,:,:,:)=real(atb_mol_nad(1:nzonal,1:nmerid,1:nivout,1:nwl))
call pncvar_put_var(domains, var_atb_mol_nad, toprint4d, &
       rank, (/nzonal,nmerid,nivout,nwl,1/), (/iprint+1/))
```

4.6 The several time-steps

CHIMERE simulation is split into two main parts: initialization and the time integration (Figure 4.4).

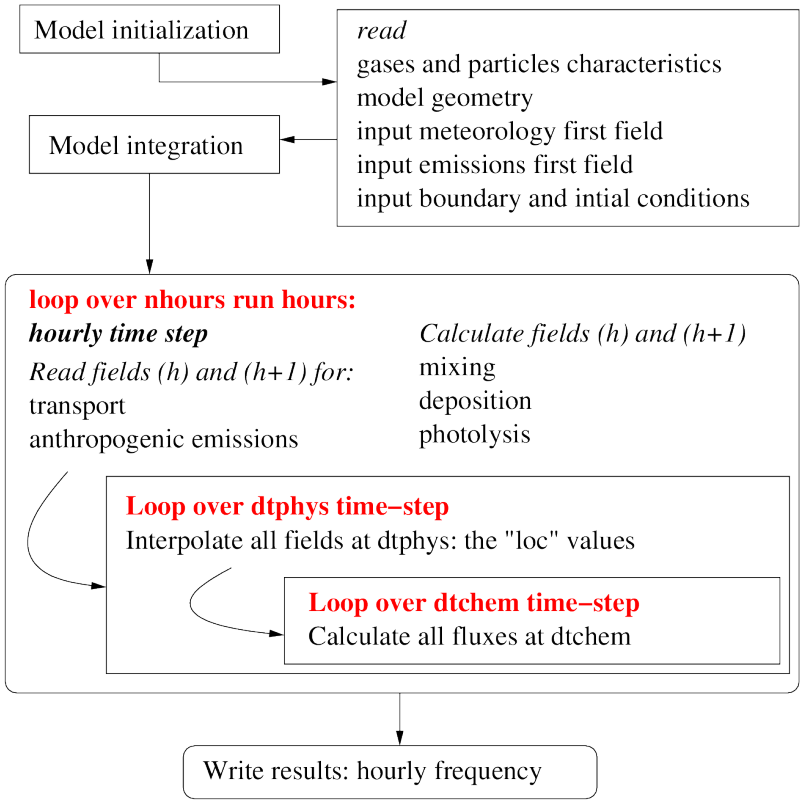


Figure 4.4: *General CHIMERE structure for the time integration of all processes*

At the **initialization** stage the model reads all input parameters, as well as two first meteorological and emissions fields (in order to start the run with realistic meteorological and concentrations fields). The **time integration** stage is itself split into two parts:

- A user-defined *physical* time step called dtphys
- A user-defined *chemical* time step called dtchem

These parameters can be changed in the chimere.par file as:

```
[dtphys]      Physical time steps (in mn)      : 15
[dtchem]      Chemical time step (in mn)       : 5
```

4.6.1 The physical time-step

The main loop in the model is a **do while** loop acting on a counter nseconds that is incremented at the physical time step and counts the number of seconds that have been simulated since the beginning of the run. Apart from the operations that are performed at the physical or chemical time steps, some operations, essentially reading the anthropogenic emission files and, in the case of offline runs, reading the forcing meteorological files, are performed at each hour.
The physical time-step is a user-defined value, independant of resolution of transport or chemistry : the forcing variables (wind, temperature, humidity, emissions) being hourly provided or diagnosed, this step is used (in the case of offline runs) to refine these values in order to have more realistic sub-hourly values. *For online runs, the Ãźmeteorological values are updated by receiving real-time meteorological values instead of interpolated values*. For offline runs, the refined values are calculated in src/locvalues.F90 with a simple linear interpolation between the previous and the next exact hour.
The weak point of linear time interpolation in offline runs is that in reality, processes vary nonlinearly even inside one hour. Therefore, using online simulation without coupling (cpl_case=1 in the chimere.par file) is interesting because it permits to have realistic variations of the meteorological fields even inside one hour. If an offline simulation is performed, then the physical time step should be sufficiently short to ensure a smooth variation of these parameters. Typical recommanded values of the time step parameter are dtphys=15 mn for a large domain with $\Delta x \approx 0.5^o$ and dtphys=5 mn for $\Delta x \approx 5$ km.

4.6.2 The chemical time-step and the CFL criteria

The chemical time-step called $dtchem$ has to respect the Courant-Freidrich Levy (CFL) conditions, since the fluxes (momentum, turbulent, deep convection) are used to update the chemical concentrations at this chemical time-step. It the user can proposed a time-step, this is also calculated to have an adaptative time-step. There is two ways to manage this step:
If the user proposed:

```
[dtchem]      Chemical time step (in mn)       : 5
```

The adaptative time-step is calculated in src/checkcfl.F90 but if the value if found to be larger that the user request (here dtchem=5mn), the user request is finally saved for the model integration.
The other possible choice for the user is:

```
[dtchem]      Chemical time step (in mn)       : 0
```

The adaptative time-step is also calculated and is directly used. This ensures to have the lowest possible duration for the whole simulation. Note that since the adaptative time-step is diagnosed using the meteorological data with (h) and $(h+1)$, this time-step remains constant during one hour.
The CFL criteria is defined as:

$$u \times \frac{\Delta t}{\Delta x} < CFL_{max} \tag{4.1}$$

where CFL_{max}=1, u is the wind speed in a grid cell, Δt the time-step and Δx the horizontal resolution. The time-step has to be selected in order to have this criterion being respected. Originally used for horizontal transport only, we extended in CHIMERE this criterion to the vertical wind speed and the deep convection fluxes. The main goal of this estimation is to ensure that a parcel of air (including chemically active concentrations) is not allowed to travel accross a grid cell during a single time-step.
In order to ensure a maximum of robustness in the model, this criterion is increased with the following (tunable) value in the src/chimere_params.F90:

```
real(kind=iprec),parameter :: cflmax=0.8
```

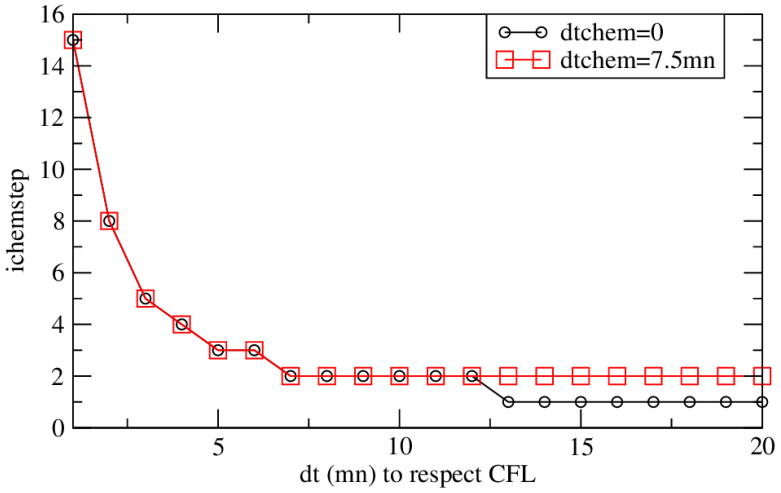


Figure 4.5: *Calculation of the number of integration chemical steps per physical steps to satisfy the CFL criterion.*

An example of adaptative time-step is displayed in Figure 4.5 . For a specific meteorology, a $dtchem$ time-step is evaluated (here from 1 to 20mn). In this case, we consider that the user wants, at least, a time-step of 7.5mn. The physical time-step is fixed to 15mn. If $dtmin$=1, we have to have 15 $ichemstep$ values. The more the $dtmin$ increases the more the $ichemstep$ value decreases. If the user wants a full adaptative time-step (i.e $dtchem$=0), $ichemstep$ may decrease until $ichemstep$=1 (for $dtmin$ >13). If the users wants at least $dtchem$=7.5, then $ichemstep$ is constrained and will decrease until $ichemstep$=2 only (even if it was found to be not necessary, following the CFL criterion).

4.7 Numerical integration

Two methods are proposed for the numerical integration of all sources and sinks for the chemical species: the TwoStep solver for all processes, following the P-L appraoch and the splitting operator approach. The choice has to be made in the chimere.par parameter file as:

```
[$imethod]   Numerical method (twostep:0, splitting:1)      : 1
[$splitcfl]  (if splitting) use CFL for trsp only (1/0)     : 1
```

4.7.1 The TWO-STEP time numerical solver

The numerical method for the temporal solution of the stiff system of partial differential equations is adapted from the second-order TWOSTEP algorithm originally proposed by [Verwer, 1994] for gas phase chemistry only. The calculation is performed in src/twostep_mod.F90. It is based on the application of a Gauss-Seidel iteration scheme to the two-step implicit backward differentiation (BDF2) formula:

$$c^{n+1} = \frac{4}{3}c^n - \frac{1}{3}c^{n-1} + \frac{2}{3}\Delta t R(c^{n+1}), \tag{4.2}$$

with c^n being the vector of chemical concentrations at time t_n, Δt the time step leading from time t_n to $t_n + 1$ and $R(c) = \dot{c} = P(c) - L(c)c$ the temporal evolution of the concentrations due to chemical production and emissions (P) and chemical loss and deposition (L). Note that L is a diagonal matrix here. After rearranging and introducing the production and loss terms this equation reads

$$c^{n+1} = \left(I + \frac{2}{3}\Delta t L(c^{n+1})\right)^{-1}\left(\frac{4}{3}c^n - \frac{1}{3}c^{n-1} + \frac{2}{3}\Delta t P(c^{n+1})\right). \tag{4.3}$$

The implicit nonlinear system obtained in this scheme can be solved pertinently with a Gauss-Seidel method.

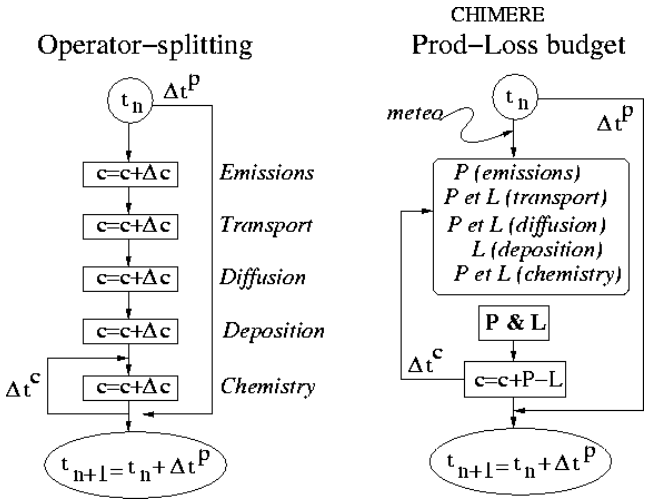


Figure 4.6: *Principle of 'operator-splitting' versus Chimere integration*

In CHIMERE the production and loss terms P and L in Equation 4.3 are replaced by the modified terms $\tilde{P} = P+P_h+P_v$ and $\tilde{L} = L + L_h + L_v$, respectively.

- P_h and P_v denote the temporal evolution of the concentrations due to horizontal (only advection) and vertical (advection and diffusion) inflow into the concerned grid box,
- L_h and L_v the temporal evolution due to the respective outflow divided by the concentration itself.

With this integration scheme, at each time step all physical and chemical processes are updated simultaneously. This numerical avoid the "splitting error" created by the "operator splitting" technique. Further advantages of the scheme are its stability even for quite long time steps due to the implicitness of the formulation and the simplicity of the code which facilitates the development of secondary models (adjoint, tangent linear) enormously. However, this method can create some numerical instabilities for fast reversible processes and flux limiter may be necessary to stabilize the model. Moreover, the results depend on the order into which concentrations of species are solved (slightly different results may be obtained for another order). It also necessitate the concentrations at the previous time step (not known at the first time step) and fluxes need to be continuous and differentiable whereas some fluxes (anthropogenic emissions) do not respect this condition.

4.7.2 Operator-splitting

The operator-splitting of CHIMERE can be used with the option imethod=1 in chimere.par. The operator will solve separatly transport (including deposition and emissions), chemistry and aerosol formation. Whereas this method can create "splitting error", this method has the advantage of solving individual processes with the appropriate time step. The chemical time step must remain low to prevent significant "splitting error". The principle of the opertor-splitting in Chimere is shown by Figure 4.7. The operator-splitting can be used with the option splitcfl in chimere.par equal to 0 or 1. For the option 0, each processes are constrained to respect the time step required by CFL (ΔT_{CFL}). Transport, chemistry and aerosol formation are solved one after another on ΔT_{CFL} until the chemical time step (ΔT_{chem}) is reached. For the option 1, only tranport and chemistry are constrained to respect the CFL. In that case, transport and chemistry are solved one after another on ΔT_{CFL} until the chemical time step. After that, aerosol formation on ΔT_{chem} is solved. This option, is an efficient way to reduced the computation time for high resolution simulations.
For the **transport integration**, emissions and deposition are solved together with transport. This is done to ensure that the conditions on the ground level (which is the consequence of deposition, emissions and vertical diffusion all together) are respected. These processes are solved with a semi-implicit iterative method :

$$c^{n+1} = \frac{c_0 + 0.5\Delta T\left(P[c^n] + P[c^0] - L[c^0]\right)}{1 + 0.5\Delta T L[c^n]} \tag{4.4}$$

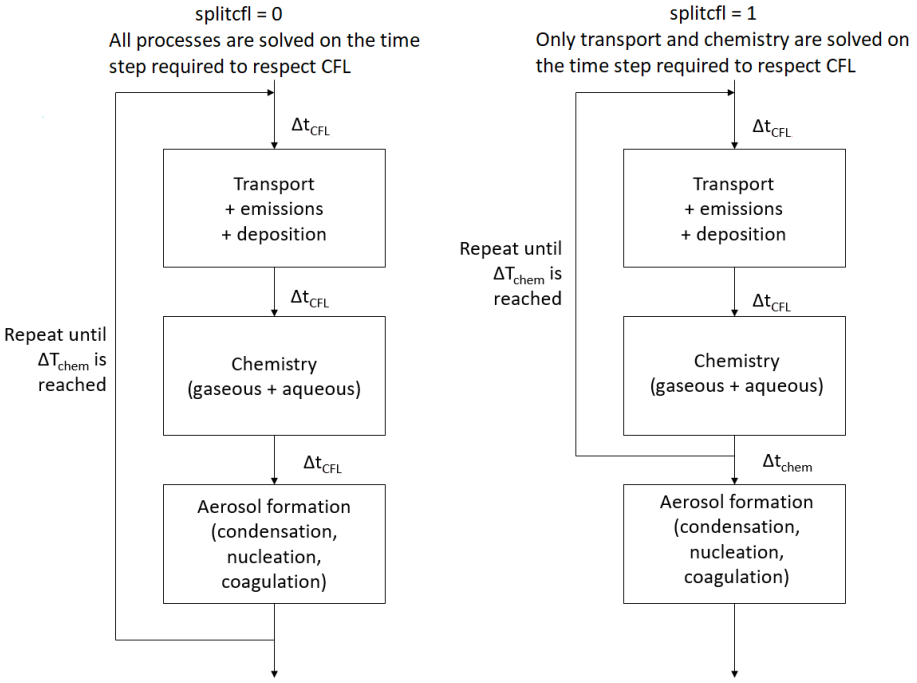


Figure 4.7: *P-L approach or 'Operator-splitting' used in CHIMERE.*

with c^0 the initial concentration, c^n the concentration at iteration n, P[c^n] and P[c^0] are the production fluxes at iteration n and 0, L[c^n] and L[c^0] are the production fluxes at iteration n and 0.
For the **chemical integration**, chemistry is solved with a two-step method by using an adaptative time. The model selects the most appropriate time step (down to 10 s) to solve the chemical equations.
For the **aerosol formation**, aerosols formation is solved with Eq. 4.4. Condensation/evaporation is an almost instantaneous process. However, a kinetic equal to 3/(4ΔT) is used in Chimere (for both splitting and Two-step) to limit the flux and solve the equations. To ensure with splitting that equilibrium is almost reached, 3 subsequent integration on $\Delta T=\Delta T_{integration}/3$ ($\Delta T_{integration}$ being the time step of integration). To improve precision with operator-splitting, the option npeq in chimere.par may be set to 0 (not valid for twostep). In that case, operator-splitting will call the equilibrium thermodynamic ISORROPIA and SOAP just before the aerosol integration.

4.7.3 Gauss-Seidel iterations

For both numerical methods, it is strongly recommended, in order to have accurate results, to use 2 Gauss-Seidel iterations, which increases by 2 the computer time. This parameter has to be selected in the chimere.par file as:

```
[ngs]       Number of Gauss-Seidel iterations              : 2
[nsu]        - during spinup                               : 3
[irs]        - number of spinup hours                      : 1
```

This change affects mostly concentrations during morning. However, for values of ozone concentration, the proposed version does not appear to degrade simulations, especially during afternoon hours.
To ensure a correct initialization of the model (mostly when climatological boundary conditions are used for the first hour integration), it is also recommended to force a higher number of Gauss-Seidel iterations: here, for example, 3 Gauss-Seidel iterations for a spin-up time of 1 hour.

5 How to define a new CHIMERE simulation?

The simulation of the predefined test case showed that the model is well installed and ready to run(Section 2.5). In this chapter, we show how to prepare a new simulation. The pre-processing consists of four steps :

1. Define the simulation domain over the geographical region of the study
2. Calculate anthropogenic emissions fluxes on the simulation grid and for the chosen period. This step depends on user's choices and is fully described in chapter 9, p.89.
3. Define the chemical mechanism and the aerosol options: this step is fully described in chapter 11, p.121.
4. Prepare all other domain-dependent input data (this step is automatically managed by CHIMERE)

5.1 Define the simulation domain

The first step is to define the domain over the region you want to study: CHIMERE being an eulerian model, it is necessary to define a grid, horizontally and vertically. CHIMERE will create these grids based on information given in the domainlist.nml file (see Section 5.1.2 below). To create these grid files, it is useful to activate the *[istopdom]* option in the chimere.par file as explained here:

```
[istopdom]       Make domains and exit : 1
```

This will execute the model but only for the preparation of the meshes. It is useful if the user wants to check the simulation domain before running the simulation. You may also want use the genarated geog file to run the emiSURF program and prepare anthropogenic emissions on the simulation domain. The functionality of this flag is illustrated in the Figure 5.1.

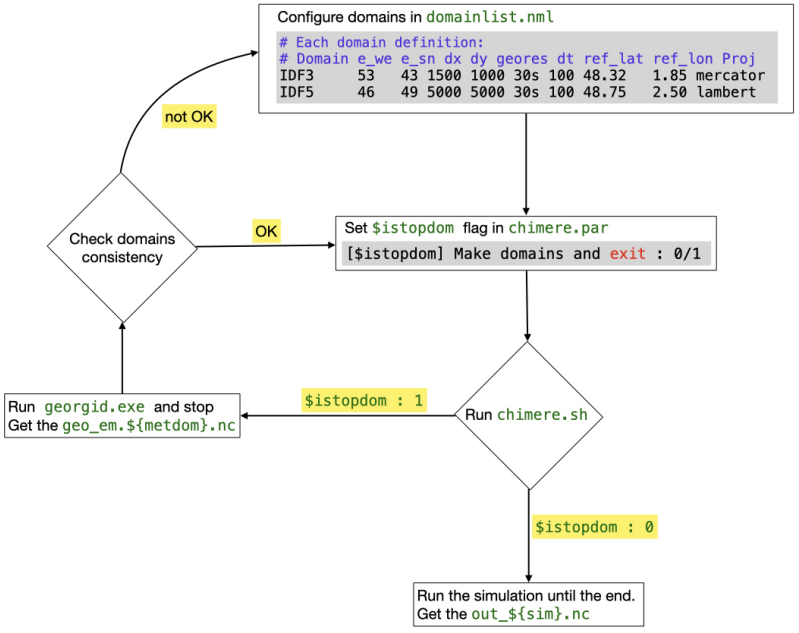


Figure 5.1: *Flowchart illustrating the functionality of the [istopdom] flag.*

5.1.1 The vertical grid

The vertical mesh of CHIMERE is completely independent of the horizontal domain. The model uses any number of vertical layers, described in hybrid $\sigma - p$ coordinates. The pressure in hPa at the top of each layer k is given by the following formula:

$$P_k = a_k 10^5 + b_k P_{surf} \tag{5.1}$$

P_{surf} is the surface pressure and the coefficients a_k and b_k are calculated by the geom.awk script, called from chimere-domain.sh via define_geom.sh. The a_k and b_k coefficients ($1 \leq k \leq nlayer$) are written to the VCOORD file which is then stored in the domains/VCOORD/ directory of the chimere root directory, and in the domains/ subdirectory of the temporary chimere directory.
The generation of the VCOORD file depends on three parameters of the chimere.par file :

```
[nlayer_chimere]       Number of vertical layers           : 20
[top_chimere_pressure] Top layer pressure (mbar)           : 200
[first_layer_pressure] First layer pressure (sigma * 1e3)  : 997
```

with:

1. *nlayer_chimere* is the number of vertical layers needed,

2. *top_chimere_pressure* is the pressure of the top of the domain (mbar), and

3. *first_layer_pressure* is the pressure of the top of the lowest chimere level, expressed in $\sigma * 10^3$ units. For example, if `$first_layer_pressure` is set to 997, the thickness of the first layer (in pressure) will be equal to $(1 - 0.997)\, P_{surf} \simeq 3\,\mathrm{hPa}$.

The algorithm coded in scripts/domain-geom.awk produces a VCOORD file in which the thickness of the layers increase exponentially with height in the lowest 200 hPa of the atmosphere, and have constant thickness (in pressure coordinates) from 200 hPa to the top of the model. The number n_{200} of layers in the lowest 200 hPa of the model is determined so that:

1. No layer is thicker than its upper neighbour. Since the layer thickness increase exponentially from $k = 1$ to $k = n_{200}$ and is constant from $k = n_{200} + 1$ to $k =$`$nlayer_chimere`, this condition is equivalent to imposing that layer n_{200} is not thicker than layer $n_{200} + 1$

2. n_{200} is the highest possible value such that condition 1 is verified

These two conditions are coded in scripts/domain-geom.awk and produce a VCOORD file that ensures a fine vertical resolution for the lowest atmospheric layers while not having too thick layers close to model top. Another script scripts/domain-geom-verbose.awk is also provided in order to make a (relatively) user-friendly output on the screen so that the user can check whether the vertical resolution obtained with their parameters is reasonable. A sample output of scripts/domain-geom-verbose.awk is as follows:

```
Vertical layers will be as follows :
lowest model layer will be  5 hPa thick
will have  6  layers below 800 hPa
           2  layers from 800 hPa to  500 hPa
k    ca        cb        p(hPa)       thk(hPa)
  1  0.00000   0.99500   995.00000     5.00000
  2  0.00902   0.97705   986.07113     8.92887
  3  0.02513   0.94500   970.12617    15.94496
  4  0.05389   0.88777   941.65206    28.47411
  5  0.10525   0.78555   890.80371    50.84835
  6  0.19697   0.60303   800.00000    90.80371
  7  0.34848   0.30152   650.00000   150.00000
  8  0.50000   0.00000   500.00000   150.00000
```

Users may want to generate their own VCOORD files depending on their modelling priorities (long-range transport, urban boundary layer . . .). To do that, the easiest possibility is to modify the geom.awk script according to one's needs.

5.1.2 The horizontal grid

The horizontal domain of the simulation is defined with geographical longitude and latitude coordinates. For CHIMERE, the mesh has not to be regular but must be cartesian (cells with four sides and the most squared as possible). In the previous model versions, the mesh was defined by longitude/latitude pairs of the grid-cell centers in an ascii file (the COORD_{dom} file. In this model version, we adopt the netCDF format and the WRF's preprocessor (WPS) geog_em.{dom}.nc file. CHIMERE is often used with WRF as meteorological forcing, in offline mode but also now in online mode. Using identical definitions of the simulation domain has several advantages: (i) avoid costly horizontal interpolation of the meteorological fields during the simulation, (ii) benefit from updates in landuse databases of the WRF system, (iii) use consistent surface and soil information between the two models, iv) have access to additional arrays such as the orography.
When configuring a new CHIMERE simulation you may be found in one of the following situations concerning your horizontal domain:

1. You want to generate a new simulation domain from scratch
2. You already have a geog_em.{dom}.nc file and want to run CHIMERE on it
3. You want to force your CHIMERE simulation with a meteorological fields on a different mesh grid.

5.1.2.1 Need to generate a 'geog' file

If the user wants to make a simulation on a new domain, or wants to migrate from 'offline' to 'online' simulation, it is necessary to have a common 'geog' file for WRF and CHIMERE. In order to simplify the creation of this file, the parameter file domains/domainlist.nml is defined.
The 'geog' will be automatically generated if the file specified in the chimere.par is not found. The test is performed by the scripts/chimere-domain.sh script. After its creation, the path will be used to rename the file and to store it.
The main part of domainslist.nml file lists the parameters necessary to generate a grid as:

```
# Domain  e_we  e_sn    dx      dy    geores  dt   ref_lat  ref_lon     Proj
FRA15       88   85   15000   15000    30s   200    49.115   9.25      lambert
EUR45      100  100   45000   45000    30s   200    49.115   9.25      lambert
```

with the following parameters (using the WRF names and definitions for consistency):

1. Domain: Name of the domain
2. e_we, e_sn: The number of grid cells on the east-west and south-north directions respectively
3. dx, dy: Resolution (meters) of the grid in the east-west and south-north directions respectively
4. geores: Resolution of the WPS database to use. This is the place where you can define the use of MODIS or USGS databases for the generation of the geog file. This data corresponds to the geog_data_res variable in WPS namelist. If you want to automatically distinguish the sea and the lakes (important for sea salt emissions) you can also use the USGS data with lakes such as:

   ```
   MED60 160  110   60000 60000 usgs_lakes+30s 200 29 8. lambert
   ```

5. dt: A time-step (seconds) for WRF. May be overriden by the model depending on the CFL. But the user may specify here a very small time-step even if this is not required.
6. ref_lat, ref_lon (degrees): The latitude/longitude coordinates (degrees) of a reference point in the domain (a point you absolutely want to include in your mesh).
7. Proj: Projection specifications. Lambert is typically used for central latitudes. The possible choices are:
   - 'lambert': Lambert conformal
   - 'mercator': Mercator
   - 'polar': polar stereographic
   - 'lat-lon': regular latitude-longitude grid (including global)

We note here, that additional parameters are mandatory for the creation of geographical domains with the WPS program. These parameters are hard-coded in the namelists_wrf/namelist.wps.sed as follows:

```
stand_lon =  11.5,
truelat1  =  30.0,
truelat2  =  60.0,
```

These parameters are appropriate for mid-latitude European domains but may need to be changed for other regions of the world. For more control over the domain creation with the WPS program, additional parameters may be used in the namelists_wrf/namelist.wps.sed such as:

1. REF_X : A real value specifying the i part of an (i, j) location whose (latitude, longitude) location in the simulation domain is known. Default value is (((E_WE-1.)+1.)/2.) = (E_WE/2.).

2. REF_Y : A real value specifying the j part of an (i, j) location whose (latitude, longitude) location in the simulation domain is known. Default value is (((E_SN-1.)+1.)/2.) = (E_SN/2.)

For more details on the use of the WPS program and its namelist the user is invited to read the WPS User's guide.
Some other specific arguments need to be specified in case of nesting. These specific arguments are specified below in Section 5.3.1.

5.1.2.2 Already existing 'geog' file

In the case where the user has already a WRF 'geog' file, the full path to the file must be provided in the chimere.par file as:

```
   [$wrf_geog]         Where is geog from WRF  : /home/myaccount/chimere/domains/geog_USGS_${dom}.nc
```

In addition, the domain definition still needs to be added to the CHIMERE domainslist.nml (cf. Section 5.1.2.1) and all parameters must correspond exactly to the existing 'geog' file.
The code will search automatically for the existenz of this file. If the file is found, CHIMERE will use it as horizontal domain.

5.1.3 Nudging

The CHIMERE-WRF coupled suite can use the WRF nudging option set to avoid strong divergences inside the models. A coarse model simulation (typically above 25km resolution) is driven by global scale simulations commonly issued from GFS or ECMWF datasets reanalysed with synoptic observations. At higher resolution over nested domains a model nudging should be not necessary to let the model develop the local physics and dynamics. For the chemistry, model nudging is never used but assimilation of observation data (from satellite and ground stations) is a research areas but this is not fully operational yet in CHIMERE and not proposed in this version.

5.2 Surface and soil input data

Once the model grid is defined and the geog file is created, several other input files will be automatically created by CHIMERE. These files are necessary for different physical and chemical processes during the simulation such as mineral dust, biogenic and DMS emissions.
At the end of this data preprocessing, the directory named with the simulation domain <dom> should contain:

dms_<dom>.nc	DMS data
EFMAP_LAI_<dom>_2.nc	MEGAN input parameters
erodibility_<dom>.nc	Erodibility for dust emissions
geog_USGS_<dom>.nc	Geometry file
LANDUSE_USGS_<dom>.nc	Land-use categories
z0_CHIMERE_<dom>_ASCAT.nc	Aeolian roughness length

5.2.1 The landuse

After creating the horizontal mesh, several scripts are launched by the scripts/chimere-domain.sh to create additional datasets required for the CHIMERE simulation. Among them, landuse fractions for each model grid-cells have to be defined. This information is necessary for the deposition, the biogenic emissions, the resuspension and the meteorological changes in urban areas.
The landuse dataset is generated with the scripts/domains-prep_landuse.sh script. High resolution landuse datasets are processed, and area fractions for different landuse categories are calculated on the simulation domain. The only action the user has to take is to define which input database will be processed among those implemented in the CHIMERE preprocessor. The related flags in the chimere.par namelist are:

```
[$lunom]         Land Cover GLOBCOVER/USGS          : USGS
[$landcover_dir] Land Cover directory               : ${bigfilesdir}/LANDCOVER
```

There are two options: 'GLOBCOVER' or 'USGS'. We recommend to use USGS as in WRF. If the output file for the landuse is not found, the script is launched automatically. The output file is created in the directory domains/<dom> and with the name LANDUSE_<lunom>_<dom>.nc.
CHIMERE uses a simplified landuse classification with 9 categories. This choice is made to speed up the model, when usual landuse has more than 20 categories. Correspondance matrixes are defined for both implemented landuse databases (USGS and LANDCOVER) to convert input landuse data to the CHIMERE native classification as shown in Table 5.1 .

Category Number	Description	Category Number	Description
1	Agricultural land / crops	6	Shrubs
2	Grassland	7	Needleaf forest
3	Barren land/bare ground	8	Broadleaf forest
4	Inland Water	9	Ocean
5	Urban		

Table 5.1: *Landuse categories used in CHIMERE*

GlobCover Land Cover: This is a global land cover map at 10 arc second (300 meter) resolution [Bicheron et al., 2011]. It contains 22 global land cover classes defined within the UN Land Cover Classification System (LCCS). GlobCover database is based on the ENVISAT satellite mission's MERIS sensor (Medium Resolution Image Spectrometer) Level 1B data acquired in Full Resolution (FR) mode with a spatial resolution of 300 meters. GlobCover LC was derived from an automatic and regionally-tuned classification of a time series of MERIS FR composites covering the period December 2004-June 2006. Raw global data to read are in the file globcover_freshwater.nc and this file is preformatted and provided on the CHIMERE web site.
The matrix to convert GLOBCOVER landuse to CHIMERE landuse is as follows:

```
CROP GRAS BARR IWAT URBA SHRB CONI DECI OCEA
1.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00    1 Post-flooding or irrigated cropland
1.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00    2 Rainfed cropland
0.64 0.12 0.00 0.00 0.00 0.12 0.06 0.06 0.00    3 Mosaic cropland (50-70%) / vegetation (20-50%)
0.40 0.20 0.00 0.00 0.00 0.20 0.10 0.10 0.00    4 Mosaic vegetation (50-70%) / cropland (20-50%)
0.00 0.00 0.00 0.00 0.00 0.00 0.00 1.00 0.00    5 Broadleaved evergreen or semi-deciduous forest (>5m)
0.00 0.00 0.00 0.00 0.00 0.00 0.00 1.00 0.00    6 Closed broadleaved deciduous forest (>5m)
0.00 0.00 0.00 0.00 0.00 0.00 0.00 1.00 0.00    7 Open broadleaved deciduous forest/woodland (>5m)
0.00 0.00 0.00 0.00 0.00 0.00 1.00 0.00 0.00    8 Closed needleleaved evergreen forest (>5m)
0.00 0.00 0.00 0.00 0.00 0.00 1.00 0.00 0.00    9 Open needleleaved deciduous or evergreen forest (>5m)
0.00 0.00 0.00 0.00 0.00 0.00 0.50 0.50 0.00    10 Closed to open mixed broadleaved and needleleaved forest
0.00 0.40 0.00 0.00 0.00 0.30 0.15 0.15 0.00    11 Mosaic forest or shrubland (50-70%) / grassland (20-50%)
0.00 0.60 0.00 0.00 0.00 0.20 0.10 0.10 0.00    12 Mosaic grassland (50-70%) / forest or shrubland (20-50%)
0.00 0.00 0.00 0.00 0.00 1.00 0.00 0.00 0.00    13 Shrubland
0.00 1.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00    14 Herbaceous vegetation
0.00 0.25 0.00 0.00 0.00 0.25 0.25 0.25 0.00    15 Sparse (<15%) vegetation
0.00 0.00 0.00 0.00 0.00 0.00 0.00 1.00 0.00    16 Closed to open broadleaved forest regularly flooded
0.00 0.00 0.00 0.00 0.00 0.40 0.00 0.60 0.00    17 Closed broadleaved forest / shrubland permanently flooded
0.00 0.40 0.00 0.00 0.00 0.00 0.30 0.30 0.00    18 Grassland / woody vegetation regularly flooded
0.00 0.00 0.00 0.00 1.00 0.00 0.00 0.00 0.00    19 Urban areas >50%
0.00 0.00 1.00 0.00 0.00 0.00 0.00 0.00 0.00    20 Bare areas
0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 1.00    21 Water bodies
0.00 0.00 1.00 0.00 0.00 0.00 0.00 0.00 0.00    22 Permanent snow and ice
0.00 0.00 0.00 1.00 0.00 0.00 0.00 0.00 0.00    23 Inland water
```

This matrix is reported in the file domains/LAND_AGGREGATION_GLOBCOVER.

The U.S. Geological Survey Landuse: This is a global land cover map at 30 arc second (1 km) resolution. It contains 24 global land cover classes defined within the UN Land Cover Classification System (LCCS). USGS database is based on the data set derived from 1 km Advanced Very High Resolution Radiometer (AVHRR) data spanning a 12 month period (April 1992-March 1993). This data was generated by the U.S. Geological Survey's (USGS) National Center for Earth Resources Observation and Science (EROS), the University of Nebraska-Lincoln (UNL) and the Joint Research Centre of the European Commission, [Loveland et al., 2000]. Raw global data are in the file usgs_soil_landuse.nc and this file is preformatted and provided on the CHIMERE web site.
The matrix to convert USGS landuse to CHIMERE landuse is as follows:

```
CROP GRAS BARR IWAT URBA SHRB CONI DECI OCEA
0.00 0.00 0.00 0.00 1.00 0.00 0.00 0.00 0.00    1 Urban and Built-up Land
0.80 0.20 0.00 0.00 0.00 0.00 0.00 0.00 0.00    2 Dryland Cropland and Pasture
0.50 0.50 0.00 0.00 0.00 0.00 0.00 0.00 0.00    3 Irrigated Cropland and Pasture
0.50 0.50 0.00 0.00 0.00 0.00 0.00 0.00 0.00    4 Mixed Dryland/Irrigated Cropland and Pasture
0.00 1.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00    5 Cropland/Grassland Mosaic
0.50 0.00 0.00 0.00 0.00 0.00 0.25 0.25 0.00    6 Cropland/Woodland Mosaic
0.00 1.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00    7 Grassland
0.00 0.00 0.00 0.00 0.00 1.00 0.00 0.00 0.00    8 Shrubland
0.00 0.50 0.00 0.00 0.00 0.50 0.00 0.00 0.00    9 Mixed Shrubland/Grassland
0.00 0.00 0.00 0.00 0.00 1.00 0.00 0.00 0.00    10 Savanna
0.00 0.00 0.00 0.00 0.00 0.00 0.00 1.00 0.00    11 Deciduous Broadleaf Forest
0.00 0.00 0.00 0.00 0.00 0.00 1.00 0.00 0.00    12 Deciduous Needleleaf Forest
0.00 0.00 0.00 0.00 0.00 0.00 0.00 1.00 0.00    13 Evergreen Broadleaf
0.00 0.00 0.00 0.00 0.00 0.00 1.00 0.00 0.00    14 Evergreen Needleleaf
0.00 0.00 0.00 0.00 0.00 0.00 0.50 0.50 0.00    15 Mixed Forest
0.00 0.00 0.00 1.00 0.00 0.00 0.00 0.00 0.00    16 Water Bodies
0.00 0.00 0.00 0.00 0.00 1.00 0.00 0.00 0.00    17 Herbaceous Wetland
0.00 0.50 0.00 0.00 0.00 0.00 0.25 0.25 0.00    18 Wooden Wetland
0.00 0.00 1.00 0.00 0.00 0.00 0.00 0.00 0.00    19 Barren or Sparsely Vegetated
0.00 0.00 0.00 0.00 0.00 1.00 0.00 0.00 0.00    20 Herbaceous Tundra
0.00 0.00 0.00 0.00 0.00 0.00 0.50 0.50 0.00    21 Wooded Tundra
0.00 0.25 0.00 0.00 0.00 0.25 0.25 0.25 0.00    22 Mixed Tundra
0.00 0.00 1.00 0.00 0.00 0.00 0.00 0.00 0.00    23 Bare Ground Tundra
0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00    24 Snow or Ice
0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00    25 Playa
0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00    26 Lava
0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00    27 White sand
0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 1.00    28 Ocean
```

This matrix is reported in the file domains/LAND_AGGREGATION_GLOBCOVER.
Note that in the files globcover_freshwater.nc and usgs_soil_landuse.nc, an additional class "inland water" has been added to the classifications to distinguish between the sea water and fresh water. This was needed to avoid model emissions of sea salt over the fresh water surfaces. The separation was done using a land-sea mask. So instead of the original 22 GlobCover classes or 24 USGS classes, CHIMERE land use pre-processor takes 23 or 25 classes, respectively.

5.2.2 Surface data for the biogenic emissions

The scripts/domains-prep_megan.sh creates MEGAN biogenic emission factors and leaf-area indices (LAI) on the domain grid. The output file is domains/<dom>/EFMAP_LAI_MED60_2.nc. The index '1' or '2' depends on the database selected by the user for the MEGAN raw data. This is a user parameter in the chimere.par file as:

```
[$megan_data]    MEGAN data root                          : ${bigfilesdir}/MEGAN
[$imegan]        MEGAN database choice (1=2.5', 2=30")    : 2
```

5.2.3 Surface data for mineral dust emissions

A z_0 file is required to compute mineral dust emission along with USGS landuse. The z_0 file will be created by the program domains-extract_z0.F90, called by the script domains-prep_z0.sh, itself called by chimere-domain.sh script. If the file already exist or if $iusedust is "0", the z_0 file will not be created. An example of output is presented in Figure 5.2 (left).
The domains-extract_z0.F90 program will use the 6km spatial resolution GARLAP (Global Aeolian Roughness Lengths from ASCAT and PARASOL) dataset from [Prigent et al., 2012] (file z0_CHIMERE_usgs_ASCAT_m.nc) to create the z_0 file. The z_0 output file will contain aeolian roughness length values along with landuse index and landuse percentage, for each landuse category and for each soil types over the given CHIMERE domain.
The script chimere-domain.sh also calls the script domains-prep-erod.sh to project MODIS erodibility factor on the CHIMERE grid. An example of output is presented in Figure 5.2 (right). The data used are fully described [Beegum et al., 2016].

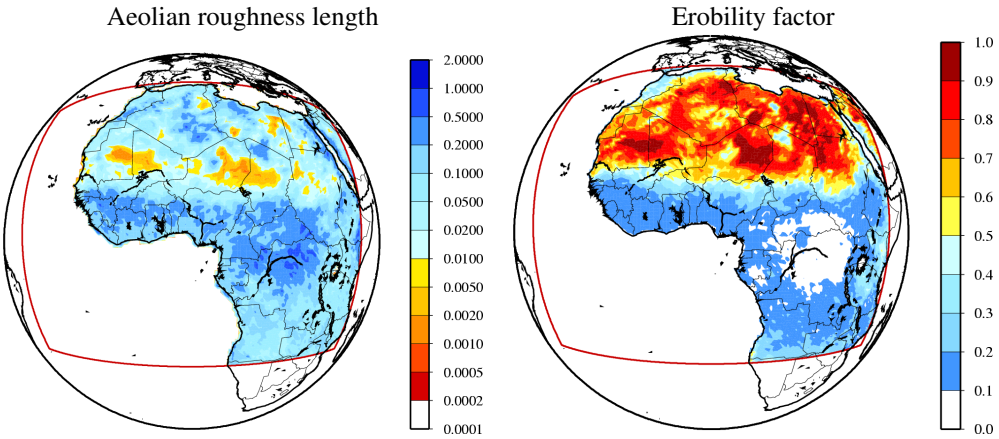


Figure 5.2: *Examples of (i) aeolian roughness length z_0 derived from GARLAP dataset and projected on a CHIMERE grid, (ii) erodbility map.*

5.2.4 Surface data for dimethyl sulfide (DMS) emissions

The script chimere-domain.sh also calls the script domains-prep-dms.sh to project DMS sea surface concentrations (nM) from [Lana et al., 2011] on the CHIMERE grid. An example of output is presented in Figure 5.3 .

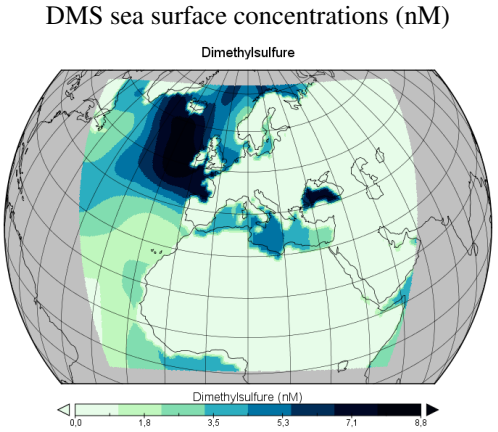


Figure 5.3: *DMS sea surface concentrations in july derived from [Lana et al., 2011]*

5.3 Nested simulations

CHIMERE offers a nesting possibility, with a one-way nesting strategy, meaning that a small, fine resolution domain can be embedded within a large, coarse resolution domain. This approach has advantages and drawbacks. The main advantage is its flexibility: the simulation on the fine domain can be performed after the simulation over the coarse domain is finished and quality-checked. On the other hand, this nesting does not permit retroaction of the fine domain on the coarse domain. Therefore, offline nesting is adapted if one wants to simulate accurately the atmospheric composition over a small domain while taking into account long-range advection of polluted plumes from distant areas into the target area.

5.3.1 domain definition for nesting

For now, the user may run up to three-level nested simulations. For more than three nests, additional modifications have to be applied to the CHIMERE scripts and namelists.
Similar to not nested runs, the first thing to do in order to prepare a simulation over a new domain is to provide the necessary information in the domains/domainlist.nml file. In the specific case of nested domains, in addition to the coarser domain definition in the upper section of the domains/domainlist.nml file, you should add one line for each nest at the dedicated section at the bottom of the domains/domainlist.nml file.

```
# Domain  e_we  e_sn    dx      dy    geores  dt    ref_lat  ref_lon     Proj
MED60      160   110   60000   60000        modis_landuse_20class_30s_with_lakes  200     29      8.        lambert
EUR20      163   100   20000   20000        modis_landuse_20class_30s_with_lakes   60     29      8.        lambert
FRA07      151   151   6666.667  6666.667   modis_landuse_20class_30s_with_lakes   20     29      8.        lambert
# Nested domains (parent i,j are given by DomainWizard or manually chosen)
#   Dom1  Dom2  grid_ratio  iparst  jparst
NEST EUR20    MED60     3  50  68
NEST FRA07    EUR20     3  50  37
```

1. Dom1: Name of the parent domain (coarse)

2. Dom2: name of the nested domain

3. grid_ratio: Refinement ratio from the parent grid. Warning: with WRF, only values of 3 or 5 are supported.

4. iparst, jparst: The east-west and south-north indexes on the parent grid of the lower left corner of the nested grid (indexes start from 1).

5.3.2 Nesting of offline runs

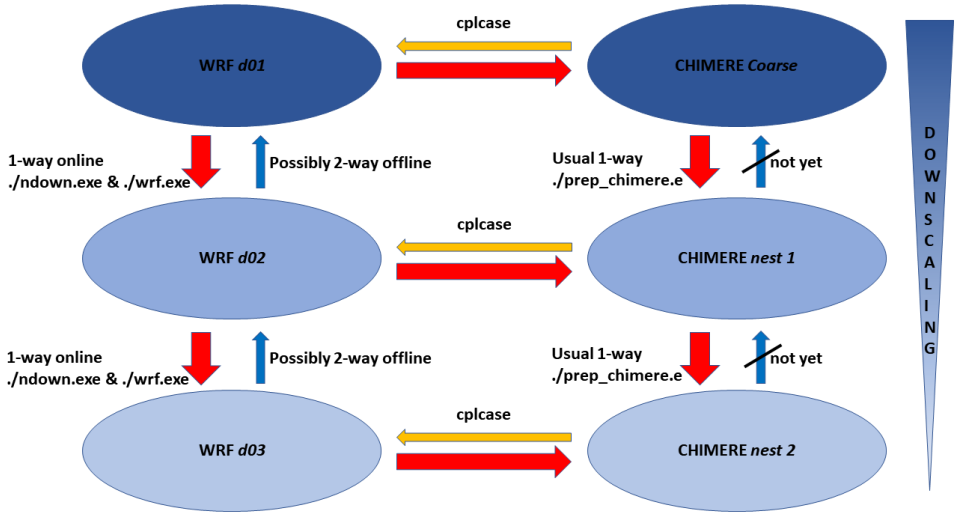


Figure 5.4: *Principle of the nesting in case of offline or online coupling.*

As in the previous version, CHIMERE supports nesting up to three levels but the namelist files can be easily changed to increase the depth of nesting. For this version, CHIMERE can be used as usual in offline mode with a meteorological forcing using WRF, ECMWF or other sources of meteorological data. In this case WRF can be priory run in 1-way or 2-way nesting, then CHIMERE is run after as usual for each domain with a 1-way nesting.

5.3.3 Nesting of online runs

If the online flag is activated (online=1), the model is fully coupled with WRF with different complexity of coupling (cplcase flag). In this case, the 2-way nesting option for WRF cannot be used. Meteorology will be downscaled with WRF's ndown.exe program. At each domain, WRF and CHIMERE both run, exchanging information frequently but without feedback from the inner domain to the parent domain either for chemical species concentrations nor meteorological fields. A nested domain may have different number of vertical levels from its parent domain as long as the top height of the nested domain is inferior to the top height of the parent domain.

5.3.4 A test-case with three nests

In this section we present a step-by-step guide on how to configure and run a simulation with nested domains. The configuration of the test-case consists of three domains, as shown in Figure 5.5. The online=1 flag is activated and the cpl_case option is set to 1. This corresponds to a simulation where both WRF and CHIMERE models run simultaneously but with no interactions between them. However, the same steps are to be followed for other values of the cpl_case flag.

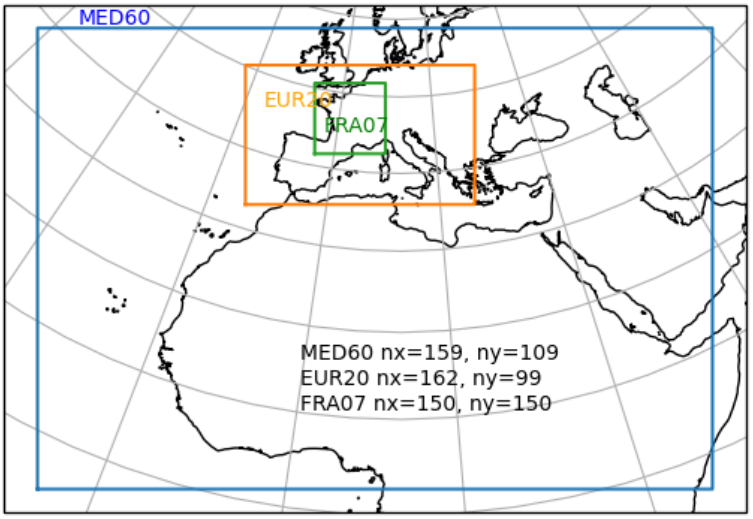


Figure 5.5: *Domains configuration for the 3-levels nest test case.*

All necessary scripts, configuration files and input for the test case can be found in the dedicated Test-Case3Nests2020.tar.gz archive. Once decompressed this archive the sub-directory "INPUTS" with two subdirectories:

1. AVN : The NCEP meteorology for the simulation period required for the initial and boundary conditions of the WRF simulation.
2. EMI : Anthropogenic emissions for January 2017 have already been prepared for all three domains. They have been processed with the emiSURF program from the CAMS inventory.

We note here, that no geog files are needed for the domains definition in this test-case since they will be created by geogrid.exe program as part of the WRF/CHIMERE preprocessing procedure.

The mylaunch-nest-test.sh bash script

This script is not part of the CHIMERE model. It is just a utility that may help you understand the new handling of CHIMERE nested domains. There are three flags in the chimere.par.test.nest generic configuration file that are manipulated with sed commands from the mylaunch-nest-test.sh bash script to provide the appropriate values for each configuration:

1. the *[$run]* flag: This flag spans from 1 , 2 , 3 indicating which of the three domains (here MED60, EUR20, FRA07) will be processed each time. Depending of the value pf this flag, the corresponding column will be read in the namelist.

2. the *[$istopdom]* flag: This flag is first activated to prepare the geog files of the three domains and stop (see Section 5.1).

3. the *[$wrfrestart]* flag: For the first run in the simulation period, this flag is desactivated and initial conditions for WRF meteorology are taken from climatology.

To use the mylaunch-nest-test.sh bash script, you need to understand the sed commands that are used by the script to set and activate the *[run]* and *[stopdom]* and *[wrfrestart]* flags of the chimere.par.test.nest file.
There are three sections in the mylaunch-nest-test.sh script. Each section will create the appropriate chimere.par configuration file for each one of the three domains and run the three simulations one after the other.

1. SECTION 1 Create the three domains : Here, only the preprocessor of the WRF/CHIMERE system will run (the *[$istopdom]* flag is activated). The first three lines activate the I*[$istopdom]* flag to the chimere.par configuration file and the next three lines run the CHIMERE preprocessor for each domain. At the end of this process you will find the three geog files of each domain in the $CHIMERE_ROOT/domains/$dom directory. We remind here, that for the creation of the geog files (the 1^{st} section of the script) you need to have the definition of the three domains in your domains/domainlist.nml file as presented in Section 5.3.1.

2. SECTION 2 First run with no restart : By uncommenting these lines (make sure to comment the first section), a 24 hours simulation will run over each domain. In the first three lines of this section the *[$istopdom]* flag is set to 0 in the chimere.par configuration file of all three domains and the *[$run]* takes values 1,2,3 for the simulation at the outer, intermediate and inner nests respectively. The next three lines of the section will run the three simulations. You may uncomment these lines one-by-one to be sure that simulation over each domain has ended up smoothly. Boundary conditions for the first simulation over the coarser domain stem from global models for both WRF and CHIMERE runs. For the simulations over the second and third nests, the coarser domain's meteorology is downscaled with the ndown.exe program and pollutant concentrations are downscaled with the src/prep_chimere program. Initial meteorological conditions for all three runs step from climatology (the *[wrfrestart]* flag is set to 0 in all three chimere.par configuration files).

3. SECTION 3 Second run with restart : Here the *[wrfrestart]* flag is activated and initial conditions for meteorological conditions stem from the previous run. You will use this section (i.e. you will comment the previous sections only if Section 1 has run successfully. This means that you will have obtained the first 24-hour CHIMERE and WRF output to use as initial conditions for your next run.

WE REPEAT HERE : Depending on what you want to do you will run the script with only the relevant section command lines (i.e. you must comment the other sections). For this test-case, only the first and second sections are used.

The chimere.par.test.nest configuration file

There is a specific chimere.par configuration file that we propose you to use in this nested case study. It goes together with the mylaunch-nest-test.sh bash script. It is the chimere.par.test.nest configuration file and you will find it in the ficTestCase3Nests2020 directory.
As shown in the general test-case of a CHIMERE simulation presented in Section 2.5, you need to provide the appropriate paths to your input and output directories in the CHIMERE model namelist chimere.par. To do so you need to search the chimere.par.test.nest file for the $__FILL__$ signs and replace them with full paths of the location of your corresponding files and directories in your file system. These flags are:

```
[simuldir] Simulation output directory                    : __FILL__
[bigfilesdir]   Big files (AEROMIN.bin, etc)              : __FILL__
[emissdir]       Anthropogenic emission Dir (stage 1)     : __FILL__
[dgrb]    Dir for input meteo grib files                  : __FILL__
```

BE CAREFUL : the *[$simuldir]* flag here must be composed from a principal directory (where ever you wish in your computer) and a subdirectory that **must** be named after the *[$lab]* flag. This is the only way the inner domain will know where to look for its boundary conditions, namely in the $simuldir/../$clab/ directory. Bellow is shown the directory structure that works fine, with

```
[$lab]      Simulation label                              : test-nest-MED60, test-nest-EUR20, test-nest-FRA07
[$clab]     Coarse domain label                           : none, test-nest-MED60, test-nest-EUR20
[$simuldir] Simulation output directory                   : principal_directory_path/${lab}
```

You will not modify in the chimere.par.test.nest configuration file the following flags. The mylaunch-nest-test.sh bash script will do so with the corresponding sed commands at each of the three sections.

```
   runs :  _RUN_
   [wrfrestart] WRF restart param (true / false)          : _WRFRESTART_
   [istopdom]      Make domains and exit                  : _ISTOPDOM_
```

Creating your nested domains

This is the first Section of the mylaunch-nest-test.sh bash script. By running the following six commands you will create the geog_USGS_${dom}.nc files in the $CHIMERE_ROOT/domains/$dom directory (where dom=MED60, EUR20, FRA07). After execution make sure that these files are there.

```
cat chimere.par.test.nest | sed 's/_RUN_/1/' | sed 's/_ISTOPDOM_/1/' > chimere.par.test.nest.makedom1
cat chimere.par.test.nest | sed 's/_RUN_/2/' | sed 's/_ISTOPDOM_/1/' > chimere.par.test.nest.makedom2
cat chimere.par.test.nest | sed 's/_RUN_/3/' | sed 's/_ISTOPDOM_/1/' > chimere.par.test.nest.makedom3

# CREATE GEOG FILES

./chimere.sh chimere.par.test.nest.makedom1 f ${datecur} 24
./chimere.sh chimere.par.test.nest.makedom2 f ${datecur} 24
./chimere.sh chimere.par.test.nest.makedom3 f ${datecur} 24
```

These commands will not run the CHIMERE and WRF models. Only the WPS preprocessor will be run to prepare domains and initial and boundary conditions for the WRF simulations.

Running CHIMERE and WRF nested simulations

Once you have verified that the three domains are created you may run the CHIMERE and WRF nested simulations. Anthropogenic emission input fluxes are already generated for the three domains and you can find them in the Test-Case3Nests2020/INPUTS/EMI directory.
You will now comment the first section of the mylaunch-nest-test.sh bash script and un comment the econd one:

```
#### SECTION 2 : RUN THE WRF/CHIMERE SYSTEM FOR ONE DAY
# FIRST RUN WITH COLD START

cat chimere.par.test.nest.mval | sed 's/_RUN_/1/' | sed 's/_ISTOPDOM_/0/' | sed -s 's/_WRFRESTART_/false/'
> chimere.par.test.nest.1
cat chimere.par.test.nest.mval | sed 's/_RUN_/2/' | sed 's/_ISTOPDOM_/0/' | sed -s 's/_WRFRESTART_/false/'
> chimere.par.test.nest.2
cat chimere.par.test.nest.mval | sed 's/_RUN_/3/' | sed 's/_ISTOPDOM_/0/' | sed -s 's/_WRFRESTART_/false/'
> chimere.par.test.nest.3


./chimere.sh chimere.par.test.nest.1 f ${datecur} 24
./chimere.sh chimere.par.test.nest.2 f ${datecur} 24
./chimere.sh chimere.par.test.nest.3 f ${datecur} 24
```

The three last commands will run the CHIMERE and WRF simulations, the one after the other starting from the outer domain and ending at the inner nest. You may want to uncomment them one-by-one to be sure that the coarser simulation has ended up properly before starting the next one.

6 Management of offline or online simulations

WRF and CHIMERE are coupled using the OASIS-MCT external coupler. To facilitate the use of this coupled model, all developments done are distributed: (i) the modified CHIMERE model, (ii) the modified WRF model and (iii) the OASIS-MCT model. All new specific parameters files are provided with this new version and the user has a minimum of parameter to change to switch between offline and online modes.
Of course, the coupling between the two models is not mandatory and the users can continue to use CHIMERE in off-line mode and with other meteorological drivers, such as IFS/ECMWF or other.
Whatever you want to perform an offline or online simulation, there is only one launching script to use: chimere.sh. The management of the offline/online is done only with the chimere.par file as:

```
# Online run parameters
   [$online]    on-line [1] or off-line [0] modeling  : 1
```

- *[online]*=0: offline mode: In this case, The CHIMERE model will read hourly meteorological files, from WRF, IFS/ECMWF or any other meteorological driver.
- *[online]*=1: online mode: In this case, the script will launch at the same time OASIS, WRF and CHIMERE.

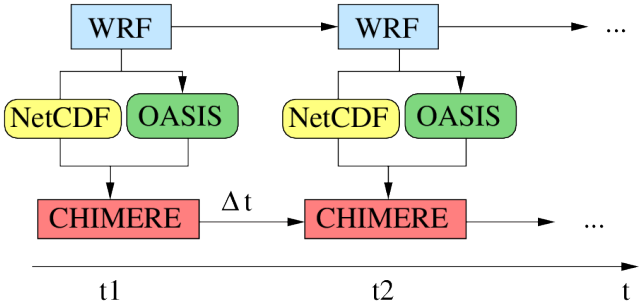


Figure 6.1: *Case of offline (*[online=0]*) or online coupling (*[online=1] *and* [cpl_case=1]*). In this case, the online coupling only uses OASIS to send WRF meteorological fields to CHIMERE in place of reading WRF NetCDF files.*

6.1 How to perform an offline run?

We present here the way to use the meteorological variables when the offline mode is selected.
CHIMERE is driven by meteorological fields from regional models, such as WRF or global models such as IFS/ECMWF. Depending on these data, there is two ways for the data management:

- WRF: If the meteorological model is WRF, the output file is directly and completely read by CHIMERE, during the simulation. There is no pre-processing in that case. In addition, CHIMERE may run over the same horizontal grid than WRF, provided in the chimere.par file.
- IFS/ECMWF or another: Several pre-processor are proposed to homogeneize these various data and deliver a standard meteorological file to be read by the CHIMERE core: the NetCDF file exdomout.nc.

The choice of the meteorological driver is done in the chimere.par file. Pre-processors are already developed and proposed with the CHIMERE code for the meteorological models WRF and IFS/ECMWF.

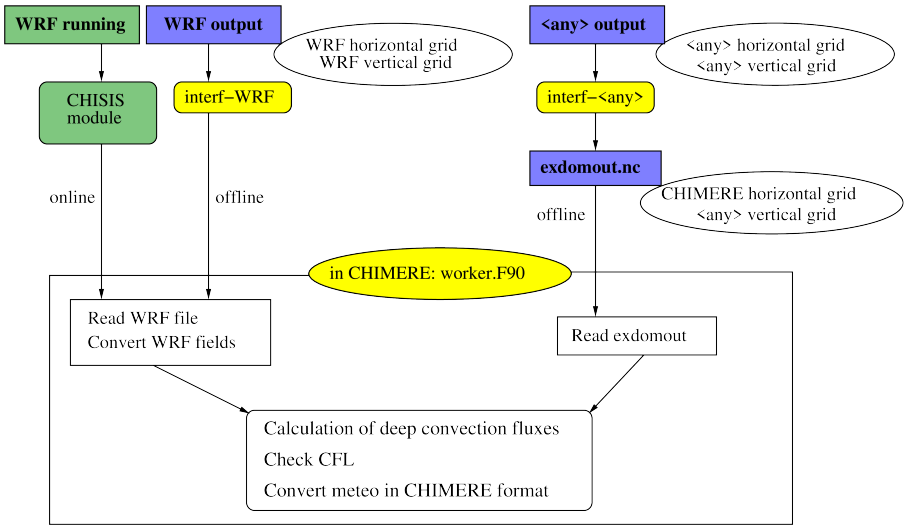


Figure 6.2: *Flow chart of the meteorological CHIMERE pre-processor. The online case corresponds to the 'green' boxes. The offline cases correspond to the 'blue' and 'yellow' boxes.*

```
# in case of offline
 [$meteo]    What is your meteo driver (WRF,ecm)             : WRF
 [$metdom]   Meteo driver domain (offline and online)        : d01
 [$meteo_DIR]  Where are your meteo files                    : /${myWRFoutputs}
 [$meteo_file] Prefix of meteo files                         : WRFOUTan_${dom}*
 [$wrf_geog]   If WRF, where is geogfile                     : ${meteo_DIR}/WPS/geo_em.${metdom}.nc
```

1. *[meteo]*: Depending on the flag, different programs will be used. The two possibilities are *WRF* and *ecm*. If the user wants to use his own data, he has to write, for example, *mesonh* and write the corresponding program interf_mesonh to read the data and build the exdoumout.nc datafile.
2. *[metdom]*: The name of the domain is, in general, in the filename. To provide this name enables to automatically search in the directory even if there is outputs from different simulations.
3. *[meteo_DIR]*: If you want to use the model offline, you have to specify the directory where are the WRF output files.
4. *[meteo_file]*: In order to select the right files to make a list, you also have to give the prefix of these files.
5. *[wrf_geog]*: It is also possible to specify the path of the geog file used for the WRF domain. In this case, CHIMERE will run on the same horizontal grid than WRF, avoiding horizontal interpolation of meteorological variables.

6.1.1 Case of WRF model output

Most of users are using the WRF model as meteorological driver for CHIMERE simulation. It is justified since WRF is a state of the art model, with many improvements every year. All the subroutines dedicated to the WRF outputs treatments are in the src/meteo-prepwrf.F90 program. This program will directly read all necessary meteorological variables used in CHIMERE. These variables are described in Table 6.1 .
with:

- "soil_layers_stag"=4. The depths and thicknesses of the soil layers are described in the tables ZS and DZS (but not read and not used in this CHIMERE version).
- Following the WRF-ARW documentation (but not the WRF output value), we use always the constant T00=+300 to get the total potential temperature.
- When sea ice is present, an artificial coverage of 10cm of snow is added in order to retrieve the correct albedo. There is no impact on the result (the meteorology being done) but this is a correction for the albedo used later for FastJX.

WRF name	Variable	Dimension	Units
ZNU	Eta values on half (mass) levels	(t,z)	ad.
ZNW	Eta values on full (w) levels	(t,zstag)	ad.
PSFC	Surface pressure	(t,y,x)	pa
SWDOWN	Downward short wave flux at ground surface	(t,y,x)	w m^{-2}
GLW	Downward long wave flux at ground surface	(t,y,x)	w m^{-2}
Q2	Qv at 2 m	(t,y,x)	kg kg^{-1}
RAINC	Accumulated total cumulus precipitation	(t,y,x)	mm
RAINNC	Accumulated total grid scale precipitation	(t,y,x)	mm
LH	Latent heat flux at the surface	(t,y,x)	j m^{-2}
T2	Temp at 2 m	(t,y,x)	k
SNOW	Snow water equivalent	(t,y,x)	kg m^{-2}
U10	U at 10 m	(t,y,x)	m s^{-1}
V10	V at 10 m	(t,y,x)	m s^{-1}
SMOIS	Soil moisture	(T,Sstag,y,x)	m^3 m^{-3}
PB	Base state pressure	(T,z,y,x)	pa
P	perturbation pressure	(T,z,y,x)	pa
U	x-wind component	(T,z,y,xstag)	m s^{-1}
V	y-wind component	(T,z,ystag,x)	m s^{-1}
W	z-wind component	(T,zstag,y,x)	m s^{-1}
T	perturbation potential temperature (theta-t0)	(T,z,y,x)	k
QCLOUD	Cloud water mixing ratio	(T,z,y,x)	kg kg^{-1}
QVAPOR	Water vapor mixing ratio	(T,z,y,x)	kg kg^{-1}
PH	perturbation geopotential	(T,zstag,y,x)	m^2 s^{-2}
PHB	base-state geopotential	(T,zstag,y,x)	m^2 s^{-2}
optional tables (diagnosed in chimere if not available			
QICE	Ice mixing ratio	(T,z,y,x)	kg kg^{-1}
P_HYD	hydrostatic pressure	(T,z,y,x)	pa
QRAIN	Rain water mixing ratio	(T,z,y,x)	kg kg^{-1}
QSNOW	Snow mixing ratio	(T,z,y,x)	kg kg^{-1}
CLDFRA	Cloud fraction	(T,z,y,x)	ad.
SNOWH	Physical snow depth	(t,y,x)	m
SEAICE	Sea ice flag	(t,y,x)	ad.
UST	U* in similarity theory	(t,y,x)	m s^{-1}
HFX	Upward heat flux at the surface	(t,y,x)	w m^{-2}
PBLH	Pbl height	(t,y,x)	m

Table 6.1: *Mandatory and optional variables in wrf.nc meteo file. The dimensions are "S" for soil layers, "T" for Time, "x" for west_east, "y" for south_north and "z" for bottom_top. Some tables are on the "staggered grid" and has the "stag" postfix.*

- A minimum of specific humidity is forced to avoid division by zero: min=10^{-6} kg/kg

Up to now, CHIMERE can only process WRF processed meteorological fields written at an hourly frequency in the wrfout file.
If you intend to use as input for your offline CHIMERE simulation a wrfout file created with the CHIMERE/WRF modeling suite you have two options:

1. If the wrfout file is produced from an online CHIMERE/WRF run then the time frequency of the variables in the file is equal to the value of the flag *[dtphys]* in minutes. In other words, probably not hourly. In this case you need to post process the netcdf file to obtain one value every hour.

2. If you have used the flag *[runwrfonly]* to create the wrfout file (i.e. without running CHIMERE) then you can set the *[dtphys]* flag to 60 minutes. The produced wrfout file can be used as it is to force your offline CHIMERE simuation

6.1.2 Case of any other model output

In case of not using WRF, the user can use already prepared programs (in case of IFS/ECMWF) or prepare by himself new codes.
If the *[meteo]* flag in chimere.par is ecm, the programs already exist and will be automatically used. If the user wants to use another model, he has to prepare by himself an input file having the format of the exdomout.nc file described in Appendix E.6, p.280

For the the preparation of this exdomout.nc, the script scripts/chimere-meteo.sh calls another script scripts/meteo-interf-[meteo].sh. This script calls the prep[meteo].F90 fortran program to prepare a generic exdomout.nc file, as presented in Figure 6.2 .
The programs such as src/meteo-prepecm.F90 read the raw meteorological model output and does several operations:

- It transforms original meteorological variables given on the meteo model grid, at a frequency that is not necessarily one hour, to standard variables given on the CHIMERE horizontal grid as hourly values.
- The operations performed are horizontal and temporal interpolation, wind vector rotation, transformation from perturbation and mean values to the total ones, etc., depending on the original data format.
- Note there is no vertical interpolation (from the meteo file to CHIMERE) to conserve the full vertical resolution, for a better accuracy on turbulent diagnostics made later in CHIMERE (see 7.1)

Table 6.2 lists the variables which must or can be output by the interpolation stage.

CHIMERE name	Variable	Dimension	Units
Mandatory Variables			
lon	longitude of gridpoints	2D	degrees_east
lat	latitude of gridpoints	2D	degrees_north
tem2	2m Temperature	2D	K
soim	Soil moisture	2D	m^3/m^3
rh2m	2m Relative humidity	2D	0-1
lspc	Large-scale Precipitation	2D	$kg/m^2/hour$
copc	Convective Precipitation	2D	$kg/m^2/hour$
temp	Temperature	3D	K
cliq	Cloud liquid water content (excluding rain water)	3D	kg/kg
sphu	Specific humidity	3D	kg/kg
pres	Pressure	3D	Pa
alti	Altitude of half layer	3D	m
winz	Zonal component of the wind	3D	m/s
winm	Meridional component of the wind	3D	m/s
swrd	Short Wave Radiation	2D	W/m^2
Optional Variables			
lwrd	Long Wave Radiation	2D	W/m^2
weas	Water equiv. accum. snow depth	2D	mm
sshf	Surface sensible heat flux	2D	W/m^2
slhf	Surface latent heat flux	2D	W/m^2
usta	Friction velocity	2D	m/s
hght	Boundary layer height	2D	m
snow	Snow depth	2D	cm
psfc	Surface pressure	2D	Pa
rain	rain water profile	3D	kg/kg
cice	ice profile	3D	kg/kg

Table 6.2: *Mandatory and optional variables in exdomout.nc meteo file. Note that all meteorological 3D variables must be provided at half-layers levels. If the optional variable are not provided by the raw meteorological model, there are intialized to zero and diagnosed if possible.*

6.2 How to perform an online run?

Once the option *[online=1]* is selected, the user has to select what kind of coupling with WRF has to be performed. This choice is done with the *[cpl_case]* flag in the chimere.par file. The several options are summarized in the Table 6.3 :
In online mode, these four cases may be summarized as:

1. *[cpl_case]*=1: The exchanges only go in the direction from WRF to CHIMERE. This is the same type than in offline mode, except that in this case the flow is through OASIS and not by reading previously prepared WRF NetCDF files inputs.

2. *[cpl_case]*=2: The Aerosol Direct Effect. This case is equivalent to the case 1 with, in addition, the exchange of aerosol optical properties from CHIMERE to WRF (Section 7.3).

Mode	Online choice	Coupling type	Exchange frequency	Interactions
Offline	*[online]*=0		1 hour	No
Online	*[online]*=1	*[cpl_case]*=1	*[dtphys]*	No
	[online]=1	*[cpl_case]*=2	*[dtphys]*	Direct effect
	[online]=1	*[cpl_case]*=3	*[dtphys]*	Indirect effect
	[online]=1	*[cpl_case]*=4	*[dtphys]*	Direct + indirect effects

Table 6.3: *Summary of options to switch between offline and online modes.*

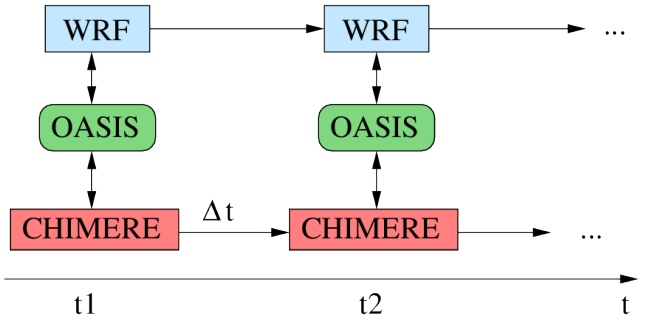


Figure 6.3: *Case of online coupling (*[online=1] *and* [cpl_case=2,3,4]*).*

3. *[cpl_case]*=3: The Aerosol Indirect Effect. This case is equivalent to the case 1 with, in addition, the exchange of the cloud fraction and the droplet number from CHIMERE to WRF (Section 7.4).

4. *[cpl_case]*=4: The Aerosol Direct and Indirect Effects.

The "direct effect" cases have been evaluated in [Briant et al., 2017] and the "indirect effect" cases in [Tuccella et al., 2019].
When selecting *[online=1]*, the main script chimere.sh performs the following operations:

- Compilation of WRF exe files (geogrid.exe, ungrib.exe, metgrid.exe, real.exe and wrf.exe) with and without OASIS.
- CHIMERE preprocessing. The chimere.sh script along with all its sub-scripts is executed to the chimere-step2.sh step. So chimere.e is not executed at this stage, as it is executed later together with wrf.exe.
- Creation of the OASIS namcouple file. The file is automatically generated from chimere.par parameters.
- WRF preprocessing (geogrid.exe, ungrib.exe, metgrid.exe and real.exe)
- Check online parameters consistency (script scripts/check-online-params.sh).
- Launch both wrf.exe and chimere.e executables concurrently via mpirun.

To finalize the configuration of an online simulation, some other parameters have to be set in the chimere.par file:

```
# in case of online
 [$geodata]    Static input data for WPS                    : /${mygeog_data}/geog_data_v4.2
 [$dgrb]       Dir for input meteo grib files               : ${myglobal_meteo}/AVN
 [$wrfrestart] WRF restart param (true / false)             : optional
 [$wrf_restart_dir] WRF restart dir                         : ${simuldir}
 [$wrf_tstep_grid] (WRF default is 6 in 6*dx)               : 4
```

1. *[geodata]* and *[dgrb]*: If you want to use the online version, the use of WRF as meteorological model is mandatory (the coupling being implemented in this model only). You have to specify some paths corresponding to the *[geog]* data and where are the global fields to force WRF.

2. *[wrfrestart]*: For the WRF simulation management in case of chained run, we added the *[optional]* value able to check if this is the first simulated period (cold start) or a period needing a restart. To keep this flag as *[optional]* is strongly recommended.

3. *[wrf_tstep_grid]*: In the WRF documentation a value between 3 and 6 is recommended. For the time-step calculation, a maximum is Δt=6 $\times$ dx, where dx is the horizontal grid spacing in km.

Note that in the beginning of a coupled simulation you might notice a screen message "Initialization of workers", and the model seems to freeze after that. Don't worry. This can take a few minutes and is due to the fact that the Thompson mirophysics scheme takes time to prepare its lookup tables: freezeH2O.dat, qr_acr_qg.dat, qr_acr_qs.dat, which are written in the temporary directory chimere_tmp. To avoid waiting during your next simulations, you can copy those files to the dir_wrf/test/em_real directory after the first coupled run.
As three-dimensional field exchanges are not possible with the OASIS3-MCT release, CHIMERE three-dimensional variables are decomposed into one-dimensional unstructured array prior to the exchange. OASIS is then not able to perform any spatial interpolation between both CHIMERE and WRF grid, as it only considered unstructured arrays. Therefore, both WRF and CHIMERE horizontal grids are required to be identical.
Since the 2016 version, CHIMERE is able to directly read a WRF geog file. In case of the coupling, it is mandatory to use the WRF geog file.

6.3 Main changes in OASIS-MCT

CHIMERE and WRF are interfaced with the OASIS3-MCT coupler, developed by the CERFACS (Centre Européen de Recherche et de Formation Avancée en Calcul Scientifique, Toulouse, France), [Craig et al., 2017].

6.3.1 The CHISIS module

An interface module between CHIMERE and OASIS, called CHISIS (CHImere / oaSIS), has been developed. The CHISIS module contains all necessary subroutines for CHIMERE to communicate with OASIS in order to exchange fields with another model. In addition, CHISIS includes a subroutine able to read the OASIS configuration file (i.e. namcouple file), therefore allowing CHIMERE to be aware of coupling parameters (e.g. exchanged variable names, time steps, partitions, grids, involved models). The CHISIS subroutines are generic and do not contain any CHIMERE specific material, therefore it can be used in other models.

6.3.2 The namcouple file

The namcouple file of OASIS-MCT describes the coupling fields and other coupling parameters. In the WRF/CHIMERE online-coupled version namcouple is automatically generated. An example of complete namcouple file is presented in Chapter C, p.241 for the case with direct and indirect effects. An extract of this namcouple is displayed and commented here:

```
# This is a typical input file for OASIS3-MCT.
# Keywords used in previous versions of OASIS3
# but now obsolete are marked "Not used"
# Don't hesitate to ask precisions or make suggestions (oasishelp@cerfacs.fr).
#
# Any line beginning with # is ignored. Blank lines are not allowed.
#
#############################################################################
 $NFIELDS
# The number of fields described in the second part of the namcouple.
#
51
 $END
#############################################################################
 $NBMODEL
# The number of models in this experiment + their names (6 characters)
#
2 chimer wrfexe
 $END
#############################################################################
 $RUNTIME
# The total simulated time for this run in seconds
#
86400
 $END
#############################################################################
 $NLOGPRT
# Amount of information written to OASIS3-MCT log files (see User Guide)
0  0
 $END
#############################################################################
 $STRINGS
```

```
#
# The above variables are the general parameters for the experiment.
# Everything below has to do with the fields being exchanged.
#
#####################################################
#
# Field 1: model1 to model2
#
#   First line:
# 1) and 2) Symbolic names for the field before and after interpolation
#           (8 characters maximum)
# 3) Index of field in cf_name_table.txt
# 4) Exchange frequency for the field in seconds (here 1 day)
# 5) Number of analysis to be performed
# 6) Restart input NetCDF file names
# 7) Field status: EXPORTED, EXPOUT, INPUT, OUTPUT
#    (For OASIS3-MCT, IGNORED/IGNOUT are converted to EXPORTED/EXPOUT)
#
#   Second line:
# 1)-2) and 3)-4) Source and target grids first and 2nd dimensions (optional)
# 5) and 6) Source and target locator prefix (4 characters)
# 7) LAG index if needed
# 8) SEQ index if needed
#
#   Third line:
# Overlap (P or R) and nbr of overlap grid points for source and target grids.
#
# List of analyses
#
SND_PSFC RCV_PSFC 1 900 1 not_used EXPORTED
wrf1 chm1
R 0 R 0
LOCTRANS
INSTANT
#
SND_SWDOWN RCV_SWDOWN 1 900 1 not_used EXPORTED
wrf1 chm1
R 0 R 0
LOCTRANS
INSTANT
#
```

For each variable, a block is defined. The important parameters in these blocks are, with the values for PSFC for example:

- SND_PSFC: The name of the variable sent by WRF (defined in frame/module_cpl.F)
- RCV_PSFC: The name of the variables received by CHIMERE (defined in src/aerosol_effects.F90)
- 1: Index of field
- 900: the time frequency of the exchanges between the two models
- 1: Number of analysis to be performed
- wrf1: the format of the table send by WRF
- chm1: the format of the table received by CHIMERE

Note that the OASIS3-MCT code has been integrated in the CHIMERE release and is automatically compiled together with CHIMERE. Once compiled, the OASIS compiled code is placed in the $chimere_root/oasis3-mct/bin directory.
30 variables are exchanged from WRF to CHIMERE, which are the meteorological fields that are required for a CHIMERE run. Sub-hourly interpolations of meteorological fields is no longer done, as CHIMERE can receive meteorological fields at a sufficient rate. The frequency at which the exchanges are made is set by the CHIMERE *dtphys* parameter in the chimere.par file.
OASIS send operation is non-blocking. As the WRF model ran faster than CHIMERE, it does not need to wait for the OASIS receive instructions from CHIMERE in order to continue its run (see Figure 6.1). In practice, it has been observed that the WRF model can be a few iterations ahead of CHIMERE only. Indeed, the sending buffer gets full, forcing WRF to wait for the CHIMERE receive instructions. However, this does not induces any additional burden as the overall computational time is defined by the CHIMERE model (i.e. the slowest of the two models). An evaluation of the WRF-CHIMERE computational performances has been made in [Briant et al., 2017].

6.4 Main changes in WRF

This section is a brief overview of the changes on the WRF side, [Skamarock et al., 2007]. Starting from version 3.7.1, WRF already comes with an OASIS interface developed for the coupling with the ocean model NEMO. The two modules

frame/module_cpl_oasis3.F and frame/module_cpl.F implement this interface and contain the routines cpl_oasis_rcv, cpl_oasis_snd, etc. encapsulating OASIS library routines (oasis_get, oasis_put, etc.). These modules have been modified in order to enable the coupling with a new model (CHIMERE). The following changes have been made.

6.4.1 The modified scripts and routines

- Registry files have been modified to define the coupling fields and namelist parameters.
- Conditional compilation using the flag cpl_wrf_chimere and two logical variables *wrf_nemo* and *wrf_chimere*; if **-Dcpl_wrf_chimere** option is used in the Makefile, *wrf_chimere* is set to **.True.** and *wrf_nemo* is set to **.False.**
- New subroutines cpl_oasis_read_namcouple, etc. read the coupling parameters in the namcouple file and store them in the new variable namcouple_info. The parameters read are broadcasted to be available for all processors.
- New subroutine cpl_rcv2 receives the required fields from CHIMERE, on the grid and partition defined for coupling.
- The subroutines cpl_sndfield, cpl_rcv, cpl_oasis_snd, and cpl_oasis_rcv are adapted to the CHIMERE coupling grid (1D).
- Subroutine cpl_oasis_define is adapted to take into account the "Points" partition used for data transfer between WRF and CHIMERE.

The Chapter B gives a detailed guide on how to implement the changes in WRF needed for coupling with CHIMERE.

6.4.2 WRF configuration files

Generic WRF configuration files are placed in the namelists_wrf directory. These files can be adapted by users to their own needs. However, some parameters are fixed by the chimere.par file because they must be consistent with the selected coupling case. A list is presented in Table 6.4 .

WRF namelist.input	
Parameter	Comments
start_date	Fixed by the run_WRF_CHIMERE.sh script as it must be consistent with the selected CHIMERE parameters
end_date	Fixed by the run_WRF_CHIMERE.sh script as it must be consistent with the selected CHIMERE parameters
radt	Frequency at which the RRTMG scheme is called by WRF. It is set automatically to 1 minute per kilometer of the grid distance between each grid cell. In addition, for coupling case 2 and 3, "radt" must be fixed to a multiple of the OASIS exchange frequency in order to ensure that CHIMERE is able to send aerosol optical properties whenever WRF requires them.
direct_feed_chimere	= 0 if *[cpl_case]*=3 = 1 if *[cpl_case]*=2 or 4
indirect_feed_chimere	= 0 if *[cpl_case]*=2 = 1 if *[cpl_case]*=3 or 4
ra_sw_physics	= 4 if *[cpl_case]*=2, 3 or 4
ra_lw_physics	= 4 if *[cpl_case]*=2, 3 or 4
bl_pbl_physics	= 1 if *[cpl_case]*=3 or 4
mp_physics	= 28 in case of *[online]*=1
cu_physics	= 3 or 5 if *[cpl_case]*=2, 3 or 4
cu_diag	= 1 if *[cpl_case]*=2, 3 or 4

Table 6.4: *List of WRF parameters to adapt in case of coupling with CHIMERE. These parameters are automatically changed in the WRF namelist depending on the user's choice in the chimere.par file regarding the [cpl_case] value.*

Some comments and recommendations:

- Setting *[direct_feed_chimere]* to 1 in the WRF namelist enables the WRF radiative scheme to receive the aerosol optical parameters from CHIMERE, rather than using a constant value. This is done automatically when setting the option *[cpl_case]*=2 or 3.

- When *[cpl_case]*=3, the WRF option *[indirect_feed_chimere]* is set automatically to 1. This option allows WRF to receive from CHIMERE the aerosol size distribution (in particles per m^3). At the same time, the *[direct_feed_chimere]* is set to 0 with *[cpl_case]*=3 (since we just want the indirect effects).
- The *[radt]* parameter is the frequency of calling the WRF radiative scheme. It also determines the frequency at which CHIMERE sends the optical properties to WRF. This parameter is also set automatically depending on the horizontal resolution (every 30 min at 30 km resolution). The value is automatically adjusted to be a multiple of the physical time step *[dtphys]* of CHIMERE.
- *[ra_sw_physics]* and *[ra_lw_physics]*: The radiative scheme must be set to RRTMG (option 4) for both shortwave and longwave radiation schemes. (If the user forgets this rule, the namelist parameter is automatically corrected).

```
ra_lw_physics    = 4, 4, 4
ra_sw_physics    = 4, 4, 4
```

- Set the *[n_bin]* variable in the WRF namelist (namelists_wrf/namelist.input.sed) to the number of bins in CHIMERE.
- The physics section of the WRF namelist must contain the CHIMERE size distribution sections (in μm)> This line is automatically updated from CHIMERE to WRF. Example:

```
chm_sections         = 0.01,0.02,0.05,0.1,0.2,0.5,1.1,2.5,5.0,10.0,40.0,
```

- The *[mp_physics]* parameter (microphysics option) is set to 28 (the Thompson aerosol-aware scheme). This option is mandatory since this is the only routine with implementation of OASIS routines to receive CHIMERE fields.

6.5 Main changes in CHIMERE

For the coupling the most important amount of changes was done in the CHIMERE model. First, additionnal parameters are added to the chimere.par file as described in chapter 3, p.23. Second, and depending on these parameters, several changes were done in the CHIMERE code. In addition, and for the aerosol indirect effect, note that a new file is created: optical_lw.txt in the chemprep/ directory.
The changes in CHIMERE are done for the whole code structure: bash script, input ascii parameters files and Fortran. All changes have been foreseen not to change the original use of the model in offline mode if necessary.

1. **Scripts:**
   - create_namcouple.sh & template namcouple_template used to generate the namcouple file.
   - wrf-compile.sh, wrf-params.sh, wrf-run-wps.sh, etc., as well as the WRF configuraiton files found namelists_wrf/ directory.
2. **Routines:**
   - Interface CHIMERE-OASIS: New module chisis_exchanges.F90: It contains routines: chisis_rcv, chisis_snd, chisis_def_var, chisis_def_partition. New subroutines were also added in iniworker and called during meteorological initializations if *online* option is set to 1: chimere_wrf_def_vars, chimere_wrf_def_partitions.
   - Data flow: New module aerosol_effects.F90 contains the routines receive_wrf_meteo, send_direct_effect_params_to_wrf, send_drop_nucl_to_wrf
   - Physical calculations: The module aerosol_effects.F90 also contains routines calc_aod, optica_lw, aero_properties. The prep_mie.F90 module contains modifications related to the aerosol long-wave properties.
   - Interpolations and initializations: The module aerosol_effects.F90 contains chm2wrf_vert_grid, aerocloud_init subroutines.

6.6 Rules and advice for online coupling with CHIMERE

Note that this section is intended for advanced users as some significant modifications of CHIMERE scripts and code could be required.
CHIMERE is interfaced with the OASIS coupler in a way that it can be coupled with other models or other interactions can be implemented between already coupled models (i.e. WRF). This section intend to gives some advice and set some rules for coupling CHIMERE with another code:

- The online coupling in CHIMERE allows several cases, each one representing a new coupling scenario. The creation of a new cases can be done in the scripts/create_namecouple.sh script. This script will create the namcouple file and all exchanged variables must be defined.
- As CHIMERE only send 1-dimensional unstructured array, a new grid must be defined for each different variable size. 5 grids have been required for the case 1, 2 and 3 (see src/iniworker.F90):

Grid name	comments	size
chm1	2-dimensional grid	nzonal × nmerid
chm2	3-dimensional	nzonal × nmerid × nlev_wrf
chm3	3-dimensional vertically staggered	nzonal × nmerid × (nlev_wrf + 1)
chm4	3-dimensional zonally staggered	(nzonal + 1) × nmerid × nlev_wrf
chm5	3-dimensional meridionally staggered	nzonal × (nmerid + 1) × nlev_wrf

  Grid names are defined as such: chm1, chm2, etc and their equivalent in WRF code are wrf1 wrf2 wrf3, etc. It is important to respect this convention as some hard-coded tests are made using these names. This allow the model to know whether a given variable is sent or received.
- A similar convention has been set on exchanged variable names. The sent variable names start with *SND_* and the received variable names start with *RCV_*. These prefix are automatically added by the create_namecouple.sh script.
- If possible, the send / receive operations must be put at the same point of the time iteration in both models (e.g. beginning or end). This will ensure that both models perform their time iteration concurrently, thereby optimizing the computational burden.

7 Meteorology, turbulence, direct and indirect effects

Once meteorological variables are read, through the OASIS coupler or just read in NetCDF files, the CHIMERE model needs additionnal parameters such as turbulent quantities. It also needs to manages the meteorological wind fields to compute transport. These two uses of the meteorological variables are described in this chapter. In addition, and in case of online mode, the aerosols calculated by CHIMERE are send to WRF to calculate the direct and indirect effects of aerosols on radiation and clouds respectively.

7.1 Diagnostics of turbulent parameters:

The second stage of the meteorological fields treatment is directly in the CHIMERE model, and during the hourly integration phase. Note that compared to the previous versions of CHIMERE, all these programs replace the *diagmet* programs. All programs are called by the worker.F90 routine and are:

- readmetro: routine for the hourly reading of the *exdomout.nc* file (in meteorological fields are not from WRF model)
- calc_turb: calculation of turbulence on raw meteo profiles
- calc_winw: calculation of the vertical wind speed on chimere profiles
- calc_deepconv: calculation of deep convection fluxes (if required by the user)
- checkcfl: Estimation of the adaptative time-step
- met2chim: Conversion of units from the meteo format to the specifc CHIMERE format (molecules, centimeters and seconds).

The calculations of additionnal parameters, mainly turbulent parameters, are done in the src/calc_turb.F90 routine. The first calculation is done to estimate several meteorological parameters, mandatory for CHIMERE but not always available with the meteorological input files. These parameters correpond to the optional parameters listed in Table 6.2 . The calculations are done using the meteorological fields provided in the *exdomout.nc* file, i.e already projected on the horizontal CHIMERE grid but still on the native meteo model vertical grid.
The availability of these parameters is checked during the reading of the exdomout.nc file or directly into the WRF file. If the parameters exist, they are not recalculated, contrarily to the previous versions of CHIMERE where it was a user's option. Indeed, we consider that if the variables exist, they correspond to recent parameterization in the meteorological model used and are consistent with other variables.

7.1.1 The friction velocity u_*

The friction velocity u_* is used for the deposition and calculation of diffusivities. It is a particularly sensitive parameter for ozone in summer through the calculation of the aerodynamic resistance r_a. Friction velocity is thus sensitive to the land use type, which is critical to deposition. In large scale meteorological models, roughness lengths are often too coarse for the implementation of high-resolution deposition. Therefore an update of u_* is proposed following the [Louis et al., 1982] formulation which is particularly robust. We recommend to use this alternative formulation, no matter whether u_* is available in the input fields, in order to have a deposition that is consistent with the high-resolution land use.

$$u_* = \sqrt{C_{DN}^2 \, F_m \, |U|^2} \tag{7.1}$$

with $|U|$ the mean wind speed at 10 m above ground level, F_m is the [Louis et al., 1982] stability function and C_{DN} the momentum transfer coefficient as:

$$C_{DN} = \frac{k}{ln\left(\frac{z}{z_{0m}}\right)} \tag{7.2}$$

with k=0.41 the Karman constant, z the altitude where the wind speed $|U|$ is known and z_{0m}, the momentum roughness length. The momentum stability function F_m is estimated depending on the bulk Richardson number, R_{ib}, value. R_{ib} is estimated at each altitude z as:

$$R_{ib}(z) = \frac{g\ z}{\overline{\theta_v(z)}} \frac{\overline{\theta_v(z)} - \overline{\theta_v(z_0)}}{|U(z)|^2} \tag{7.3}$$

with g=9.81 m^2 s^{-2} the gravitational acceleration and θ_v the virtual potential temperature in Kelvin.

- In unstable cases, if $R_{ib} < 0$:

$$F_m = 1 - \frac{2b\ R_{ib}}{1 + 3b\ c\ C_{DN}^2 \cdot \sqrt{\frac{z}{z_{0m}}} \sqrt{|R_{ib}|}} \tag{7.4}$$

- In stable cases, if $R_{ib} > 0$:

$$F_m = \frac{1}{1 + \frac{2b\ R_{ib}}{\sqrt{1 + d\ R_{ib}}}} \tag{7.5}$$

with the constant $b = c = d$=5. In neutral case, i.e R_{ib}=0, F_m=1.

7.1.2 The surface sensible heat flux Q_0

The surface sensible heat flux, Q_0, is used to compute w_* and therefore mixing, and the height of the boundary layer. In fact only the virtual heat flux is required, which can be recomputed from an empirical formula [Priestley, 1949] using temperatures in the first model layers. This formula is not very accurate. It is strongly advised to use heat fluxes from the meteorological model if available. If the surface sensible heat flux Q_0 is provided by the meteorological model, it is directly used for the computation of the convective velocity w_*:

$$w_* = \left(\frac{g\ Q_0\ \overline{h}}{\rho\ C_p\ \overline{\theta_v}} \right)^{1/3} \tag{7.6}$$

where $\overline{h}$ is the convective boundary layer height, C_p the specific heat of air at constant pressure, $\overline{\theta_v}$ the mean virtual potential temperature in the first model vertical level.

7.1.3 The Monin-Obhukov length L

The Monin-Obhukov length is estimated as:

$$L = \frac{-\overline{\theta_v}\ u_*^3}{k\ g\ Q_0} \tag{7.7}$$

7.1.4 The boundary layer height $\overline{h}$

The boundary layer height may be directly read from the meteorological forcing or diagnosed in CHIMERE. The user has to set the following option in the chimere.par parameter file:

```
# CHIMERE Subgrid Processes
 [$idiagblh]    Use BLH of meteo driver (0) or diagnose (1)  : 0
```

The boundary layer height ($\overline{h}$) is derived from different formulation depending on the atmospheric static stability. When stable, i.e when $L > 0$, $\overline{h}$ is estimated as the altitude when the Richardson number reaches a critical number here chosen as R_{ic}=0.5, following [Troen and Mahrt, 1986].
In unstable situations (i.e. convective), $\overline{h}$ is estimated using a convectively-based boundary layer height calculation. This is based on a simplified and diagnostic version of the approach of [Cheinet and Teixeira, 2003] which consists in the resolution of the (dry) thermal plume equation with diffusion. The in-plume vertical velocity and buoyancy equations are solved and the boundary layer top is taken as the height where calculated vertical velocity vanishes. Thermals are initiated with a non-zero vertical velocity and potential temperature departure, depending on the turbulence similarity parameters in the surface layer.

7.1.5 The vertical diffusivity profile K_z

Once the depth of the boundary layer is computed, vertical turbulent mixing can be applied following the K-diffusion framework following the parameterisation of [Troen and Mahrt, 1986], without counter-gradient term. In each model column, the diffusivity coefficient profile K_z (m^2 s^{-1}) is calculated as:

$$K_z = k w_s \frac{z}{\overline{h}} \left(1 - \frac{z}{\overline{h}}\right)^2 \tag{7.8}$$

where w_s is a vertical scale given by similar formulas:

- In the stable case (when the surface sensible heat flux is negative):

$$w_s = \frac{u_*}{(1 + 4.7 z/L)} \tag{7.9}$$

- In the unstable case:

$$w_s = (u_*^3 + 2.8\, e\, w_*^3)^{1/3} \tag{7.10}$$

where $e = max(0.1, z/\overline{h})$. A minimal K_z is assumed, with a value of 0.01 m^2 s^{-1} in the dry boundary layer and of 1 m^2 s^{-1} in the cloudy boundary layer. K_z is capped to a maximal value of K_z=500 m^2 s^{-1} to avoid unrealistic mixing. Above the boundary layer, a fixed value of K_z=0.1 m^2 s^{-1} is prescribed. As for many CTMs considered as numerically diffusive, horizontal turbulent fluxes are not considered.

7.1.6 Urban corrections and other tunable parameters

Finally, all three-dimensional meteorological fields are averaged over the CHIMERE vertical grid. Note that several fixed parameters are present in this src/calc_turb.F90 routine. They were chosen are the more realistic as possible and it is not recommended to modify these values. These constant valeus are listed in Table 7.1 .

Parameter	Value	Description	Unit
pblmin	20.	Minimum PBL height	m
vkminup	0.1	Minimum Kz above PBL	m^2/s
addwind	0.8	Urban wind correction factor	m/s
addflux	0.	Urban Q_0 correction additive term	m^2/s^2
z10m	10.	10 meters above ground level	m
soimdef	1.0	Default Soil Moisture (no resuspension/erosion)	m^3/m^3
woff	0.5	Wind offset to smooth Richardson numbers	m/s
crhx	0.90	Min RH for cloud BLH enhancement	[0:1]
vkmax	500.	Maximum Kz	m^2/s
odclw	180.0	Optical depth rate for liquid water	ad.
odcic	66.66	Optical depth rate for ice	ad.
topcldmax	1000.	Maximum altitude of low clouds top	m
cloh	1	High cloud option for attenuation	ad.
crhh	0.95	High cloud option for attenuation	ad.
cloh2	0.005	High clouds optical depth /m for RH=1	ad.
ztherm	25.0	Height of thermals start	m
ric	0.5	Troen and Mahrt suggestion for critical Ri for BL top	ad.
dvsca	1000.0	Entrain./Detrain. first guess vertical scale	m
vkmindry	0.01	Minimum Kz in the dry boundary layer	m^2/s
vkminwet	1.0	Minimum Kz in cloudy boundary layer	m^2/s
rlam	150.	Upper air mixing length	m

Table 7.1: *Constant values used for the calculation of additionnal parameters in the routine calc_turb.*

Note that a urban correction is available in the model. This is very simple and just implemented to retrieve some observed trends in urban areas: less surface wind speed and more surface sensible heat flux.
This option may be activated with in the chimere.par parameter file as:

```
[urbancorr]    Urban correction [1] or not [0]    : 0
```

This flag induces a slight change for the 10m wind speed and Q_0 the surface sensible heat flux, only over urbanized areas as:

$$U_{10m,urb} = addwind \times U_{10m} \quad (7.11)$$
$$Q_{0,urb} = addflux + Q_0 \quad (7.12)$$

For the diagnostics calculations, some other parameters are fixed and depends on the landuse. They are tabulated in the src/calc_turb.F90 routine and for the 9 landuse classes of CHIMERE.

Category	1	2	3	4	5	6	7	8	9
Landuse	Agricultural	Grassland	Barren land	Inland water	Urban	Shrubs	Needleaf forest	Broadleaf forest	Ocean
z_0 (m)	monthly	0.1	0.01	0.0001	1.0	0.15	1.0	1.0	0.0001
UV albedo (no snow)	0.035	0.04	0.1	0.07	0.035	0.05	0.025	0.025	0.07
UV albedo (snow)	0.376	0.720	0.836	0.07	0.3	0.558	0.278	0.558	0.836

Table 7.2: *Tabulated values needed for the diagnostic of additional meteorological parameters. The values of UV albedo are extracted from (Laepple et al. 2005) for all meteorological conditions, except when snow is detected, where they are extracted from (Tanskannen et al. 2007).*

For the first landuse type, Agricultural, a typical monthly variability is tabulated as presented in Table 7.3 .

Month	1	2	3	4	5	6	7	8	9	10	11	12
z_0 (m)	0.05	0.05	0.05	0.1	0.15	0.15	0.15	0.1	0.1	0.05	0.05	0.05

Table 7.3: *Tabulated monthly variability of the roughness length for the landuse category # 1 "Agricultural".*

7.1.7 Estimation of the vertical velocity w

The vertical velocity may be read from the meteorological forcing or diagnosed. The user has to set the following option in the chimere.par file:

```
[$is_diagwinw] Diagnose vertical wind from horiz components : .true.
```

In case of *[is_diagwinw=.true.]*, the calculation is done in the src/calc_winw.F90 routine. The vertical velocity is diagnosed by using the horizontal fluxes. A simple momentum conservation relation is applied to calculate the vertical flux for the top of each model box (assuming that the first one, near the ground, is zero).

7.1.8 Calculation of the solar zenithal angle

The solar zenithal angle has to be diagnosed for two different processes:

- The calculation of the photolysis rates using the FastJX module
- The calculation of the biogenic emissions

Its calculation is done following the method described in [Jacobson, 1999]. This calculation is done in the src/zenith.F90 routine.
The cosine of the solar zenithal angle is defined as:

$$cos(\theta_s) = sin(\phi)sin(\delta) + cos(\phi)cos(\delta)cos(H_a) \quad (7.13)$$

with:

- θ_s the solar zenithal angle
- ϕ the latitude
- δ the solar declination angle

- H_a the local hour angle of the sun

The solar declination angle is estimated as:

$$\delta = arcsin(sin(\epsilon_{ob})sin(\lambda_{ec})) \tag{7.14}$$

with:

- ϵ_{ob} the obliquity of the ecliptic
- λ_{ec} the ecliptic longitude of the Earth

The obliquity of the ecliptic, ϵ_{ob}, is calculated as:

$$\epsilon_{ob} = 23.439 - 10^7 \times N_{jD} \tag{7.15}$$

with N_{jD} the number of days since the the 1st January 2000. This estimation is automatically and hourly done using the src/calendar.F90 time calculations routines. The ecliptic longitude λ_{ec} is calculated as:

$$\lambda_{ec} = L_M + 1.915sin(g_M) + 0.02sin(2g_M) \tag{7.16}$$

with L_M and g_M the mean longitude of the sun and the mean anomaly, respectively. They also depend on the current Julian day and are estimated as:

$$L_M = 280.46 + 0.9856474N_{jD} \tag{7.17}$$
$$g_M = 357.528 + 0.9856003N_{jD} \tag{7.18}$$

Finally, the local hour angle of the sun H_a is calculated as:

$$H_a = \frac{2\pi t_s}{86400} \tag{7.19}$$

where t_s is the number of seconds past local noon.

7.1.9 Deep convection fluxes

The calculation of deep convection fluxes is a user's choice and can be activated or not in the chimere.par namelist file as:

```
[ideepconv]   Deep convection activation (1/0)   : 1
```

If the option is on, the calculations of deep convection fluxes are done in the calc_deepconv routine.

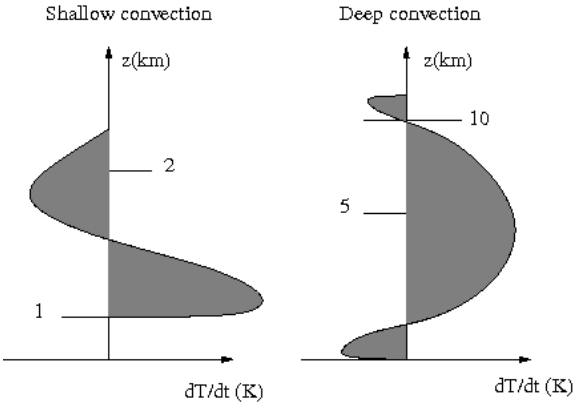


Figure 7.1: *Temperatures changes due to shallow and deep convection*

The deep convection scheme describes effects of subgrid scale clouds on tropospheric convection, after [Tiedtke, 1989] and as presented in Figure 7.1 . Deep convection occurs when cumulus or cumulo-nimbus clouds (referenced as convective clouds) are present. These clouds are formed when air masses are unstable, when warm air is at the surface or cold air is transported in upper layers (cold front). High vertical wind speed are observed leading to vertically extended cloud structures. On the other hand, when clouds are only due to mechanical forcings (mountains, warm fronts), they are named stratiform clouds and generally exhibit low vertical speed values.
Air masses may be quickly mixed in the troposphere when convective unstabilities occured under a cloud. To describe this phenomenon, schemes generally consider a cloud (and the whole column including this cloud) and the environment. In the major part of deep convection parameterizations, the hypothesis of small cloud surface compared to the total studied surface is done.

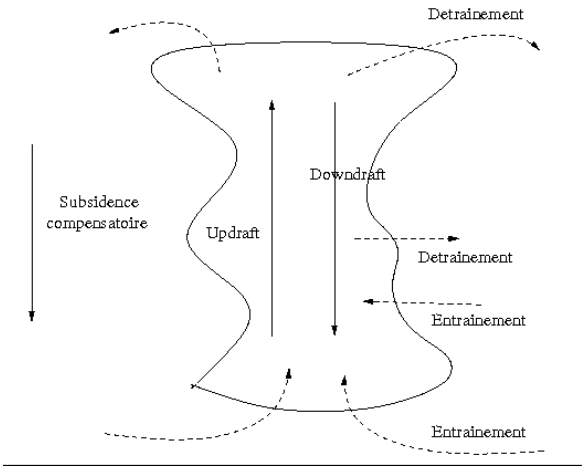


Figure 7.2: *Schematic view of a convective cloud*

Under the cloud, updrafts and downdrafts are observed, Figure 7.2 . The updraft originates from air masses lighter than their environment when downdrafts represent fall of colder air (often with rain). In the updraft and the downdraft, air may be exchange between the cloud and the environment: if air of environment flows to the cloud, this is the **entrainment** and if air of clouds flows towards environment, this is the **detrainment**. In order to ensure mass conservation, a compensatory **subsidence** is observed in the environment.
The scheme of [Tiedtke, 1989] estimates a convection mass flux such as:

$$M(z) = \rho(z)(a_{up}\overline{w_{up}(z)} + a_{dw}\overline{w_{dw}(z)}) = -\rho(z)a_{env}\overline{w_{env}(z)} \tag{7.20}$$

with $\overline{w_{up}(z)} > 0$, $\overline{w_{dw}(z)} < 0$ et $\overline{w_{ent}(z)} < 0$ the vertical wind speed in the updrafts, downdrafts and environment, respectively. The vertical gradient of this mass flux is:

$$\frac{\partial M(z)}{\partial z}(z) = E(z) - D(z) \tag{7.21}$$

with $E(z) > 0$ et $D(z) > 0$, the entrainment and detrainment fluxes. This equation works both for updrafts and downdrafts, with, for example for the updraft:

$$E(z) = E_{up}(z) + E_{dw}(z) \text{ and } D(z) = D_{up}(z) + D_{dw}(z) \tag{7.22}$$

An example of estimated fluxes are displayed in the Figure 7.3 .
These fluxes are used to estimates chemical concentrations fluxes as:

$$\frac{\partial M_{up}(z)c_{up}(z)}{\partial z} = E_{up}(z)c_{env} - D_{up}(z)c_{up} \tag{7.23}$$

$$\frac{\partial M_{dw}(z)c_{dw}(z)}{\partial z} = E_{dw}(z)c_{env} - D_{dw}(z)c_{dw} \tag{7.24}$$

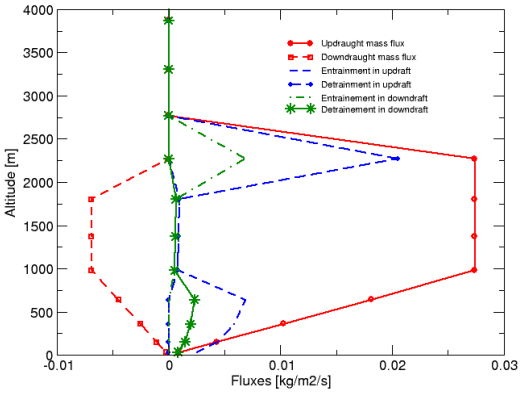


Figure 7.3: *Entrainment and detrainment fluxes in the updrafts and downdrafts.*

with the concentrations c_{up}, c_{dw} and c_{env} in the updraft, downdraft and environment, respectively. The parametrization is designed with the parameter names such as in the Figure 7.4 .

The net flux is first estimated in the src/locvalues.f90 routine. At this stage, the fluxes are in molecules.cm^{-2}.s^{-1} (after conversion in src/iniphys.f90). For each grid cell, the Equation 7.21 is applied to give 'updraught mass flux' (flxuloc) and 'downdraught mass flux' (flxdloc). The calculation is done from surface to top domain in order to ensure mass conservation following the relation:

$$\begin{aligned} FLXU(z) = & \frac{\partial M_{up}(z)}{\partial z}(z) = E_{up}(z) - D_{up}(z) + FLXU(z-1) \\ FLXD(z) = & \frac{\partial M_{dw}(z)}{\partial z}(z) = E_{dw}(z) - D_{dw}(z) + FLXD(z-1) \end{aligned} \tag{7.25}$$

The top level of the cloud and the detrainment (when $E_{dw}(z) > 0$) are estimated separately and names 'nvcloudtop' and 'nvdettop', respectively.

These fluxes are used in the src/transmix.f90 routine. One have to note that the 'concu', 'concd' parameters are not concentrations but ratios of pollutants molecules number on air molecules number. This conversion is done to ensure 'trpr' and 'trlo' production and loss fluxes terms in molecule.cm^{-3}.s^{-1}. 'concu' and 'cond' are estimated using the mass conservation relation, as for example:

$$\begin{aligned} FLXU(nv) \times concu(nv) \quad = & \, FLXU(nv-1) \times concu(nv-1) \\ & + dpeu(nv) \times conc(nv) \\ & - dpdu(nv) \times concu(nv) \end{aligned} \tag{7.26}$$

and we have:

$$concu(nv) \quad = \frac{FLXU(nv-1) \times concu(nv-1) + dpeu(nv) \times conc(nv)}{dpdu(nv) + FLXU(nv)} \tag{7.27}$$

7.2 Transport and mixing

7.2.1 Horizontal transport

Three schemes for horizontal transport of chemical concentrations in the model are now available. The choice of the scheme is activated in the chimere.par file as:

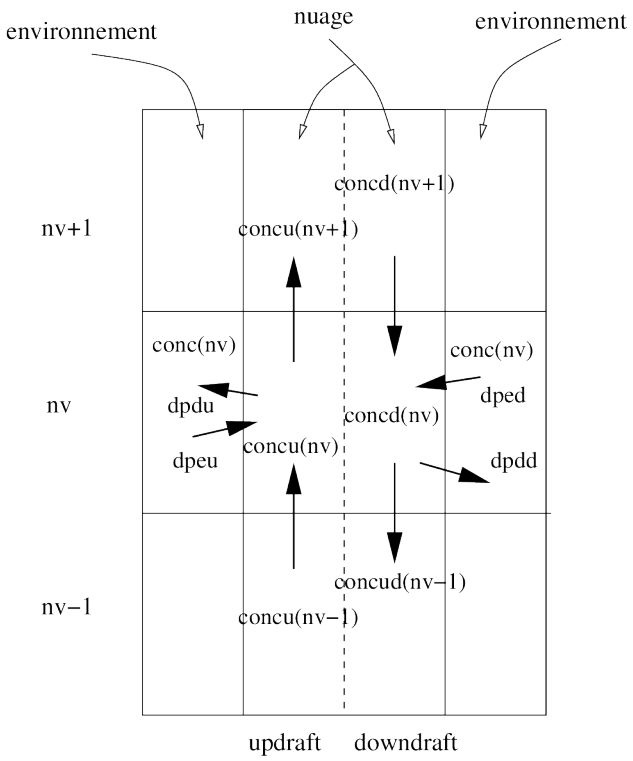


Figure 7.4: *Convective atmospheric column and its environment*

```
[iadv]      Advection scheme (0..2) : 2
```

where the three possible values are:

1. *[iadv=0]*: The **upwind scheme**. The simple first-order upwind Godunov scheme with a low computational cost but not very accurate and very diffusive.
2. *[iadv=1]*: The third-order **PPM** scheme (PPM stands for 'Parabolic Piecewise Method'), proposed by [Colella and Woodward, 1984]): high computational cost but more accurate.
3. *[iadv=2]*: The second-order **Van Leer** ([Van Leer, 1977]) scheme: intermediate computational cost, more accurate than upwind. This is in intermediate choice..

By default, all species are transported with the same scheme. But the user's has the option to manually change that, by modifying the second column of the ACTIVE_SPECIES file in its chemprep inputdata directory.
These schemes are defined in dedicated source files (except upwind, because its formulation involves no calculation): src/ppm.F90 for PPM, src/vanleer.F90 for Van Leer and src/dl99.F90 for Després-Lagoutière (the latter one can only be used in the vertical dimension). The diagnosed fluxes are then used for transport and mixing in the src/transmix.F90 program.
For a given chemical species with concentration c, the following conservation equation is numerically solved:

$$\partial_t (c\rho) + \partial_i \mathbf{F^i} = 0 \tag{7.28}$$

with ρ is the air density and F indicating the mass flux corresponding to velocity $\mathbf{v}$:

$$\mathbf{F} = \rho c \mathbf{v} \tag{7.29}$$

As CHIMERE is designed for structured grids where the grid cells are nearly parallelepipedic, this equation can be discretized and solved separately for each of the three orthogonal directions: zonal, meridional and vertical. This strategy

is known as *directional splitting*, which is very widely used in weather modelling and chemistry-transport modelling, both because it is usually more computationnally efficient and more stable than a straightforward bi- or tri-dimensional approach ([Byun et al., 1999]). Therefore, the tendency c_t in the concentration c is equal to $c_t^{(1)} + c_t^{(2)} + c_t^{(3)}$, where $c_t^{(i)}$ is the time derivative of concentration due to transport in the i^{th} direction. It is also assumed that the time variations of ρ are much slower than the time variations of c ($|c\rho_t| \ll |\rho c_t|$), so that Eq. 7.28 becomes:

$$\rho c_t^{(i)} = -\partial_i \mathbf{F^i} \tag{7.30}$$

After time and space discretisation, if we note $\delta^{(i)}c$ the variation of c due to transport in direction i, the discretized transport equations are the following:

$$\delta^{(i)}c = -\left(\frac{F^i_{n+\frac{1}{2}}(t) - F^i_{n-\frac{1}{2}}(t)}{\Delta x}\right)\Delta t \tag{7.31}$$

The concentration increments are calculated successively for each direction from the initial concentration field (parallel strategy). This equation secures mass conservation because the inward and outward fluxes cancel out in each direction. The key issue in solving this equation is the estimation of the fluxes at the cell interfaces ($F^i_{n\pm\frac{1}{2}}(t)$). The way these fluxes are estimated numerically determines the characteristics of the transport scheme (scheme type and order, diffusivity, numerical stability etc.).
In chemistry-transport models, various types of numerical schemes have been tested and are proposed to users to solve the advection equations. These numerical schemes range from simple order-1 numerical schemes such as the classical upwind method to higher-order methods such as the Piecewise-parabolic method ([Colella and Woodward, 1984]) proposed in CHIMERE.
In the horizontal directions, it is assumed that the grid cell length does not vary substantially from one grid cell to its neighbours. As in the CHIMERE model, species concentrations and meteorological variables are represented on the same grid, the wind speed at the interfaces is interpolated linearly from the wind speeds at the centers of the two gridcells separated by the interface. The wind speed at the interface will be noted $u_{n\pm\frac{1}{2}}(t)$, and is assumed constant during each coarse time step. In the following we will drop the superscript i to indicate the direction. The definition of the wind speed at the cell interface is independent of the scheme used, so that the definition of the fluxes at the interfaces only depends on the evaluation of the concentration at the interface.
Three horizontal transport schemes are available in the model.

7.2.1.1 Upwind scheme

In the upwind scheme ([Courant et al., 1952]), the fluxes at the cell interfaces are defined by the following equations according to the sign of the wind speed at the cell interface:

- If $u_{n+\frac{1}{2}}(t) > 0$

$$F_{n\pm\frac{1}{2}}(t) = c_n \rho_n u_{n+\frac{1}{2}}(t) \tag{7.32}$$

- If $u_{n+\frac{1}{2}}(t) < 0$

$$F_{n\pm\frac{1}{2}}(t) = c_{n+1} \rho_n u_{n+\frac{1}{2}}(t) \tag{7.33}$$

The assumption made in this scheme is that the tracer concentration is uniform in each grid cell, so that the mass flux at the interface is the product of the wind at the interface by the tracer concentration in the upwind cell.

7.2.1.2 Van Leer scheme

The Van Leer scheme used in CHIMERE is scheme I described in the seminal paper of [Van Leer, 1977], including slope-limiting as described in Eq. 74 of the same paper. This particular form of slope-limiting has been chosen because it is the most permissive one, reducing numerical diffusion as much as possible in this framework.
This scheme, which is recognized in meteorology for its good numerical accuracy and smaller diffusion than the first-order upwind scheme, is of course slightly more time-consuming than the first-order upwind scheme. In meteorology, it can be considered as a good compromise solution between numerical accuracy and computational efficiency for long-range transport ([Hourdin and Armengaud, 1999]).

7.2.1.3 The Parabolic Piecewise Method scheme

Another scheme that is proposed in CHIMERE for horizontal transport is the 3^{rd} order Parabolic Piecewise method (PPM) scheme, with slope-limiting and monotonicity-preserving conditions, such as presented in [Colella and Woodward, 1984]. This scheme is applied for each dimension separately, which formally limits the model to 2^d order accuracy in solving the two-dimensional transport problem because the cross derivative $\partial c^2/\partial x \partial y$ is not taken into account. However, since the scheme is symmetric, it can be considered that the errors due the neglect of cross-derivatives approximately compensate ([Ullrich et al., 2010]). Therefore, since the treatment of transport in each horizontal dimension has 3^{rd} order accuracy, the PPM scheme as implemented in CHIMERE is much less diffusive than the simple 2^d order Van Leer scheme (see, e.g. [Vuolo et al., 2009]).

7.2.2 Vertical transport

7.2.2.1 vertical advection schemes

Three schemes for horizontal transport of chemical concentrations in the model are now available. The choice of the scheme is activated in the chimere.par file as:

```
   [iadvv]  Vertical advection scheme (0..2)   : 1
```

where the three possible values are:

1. *[iadvv=0]*: The **upwind scheme**. The simple first-order upwind Godunov scheme with a low computational cost but not very accurate and very diffusive.
2. *[iadvv=1]*: The second-order **Van Leer** ([Van Leer, 1977] scheme.
3. *[iadvv=2]*: The first-order **Després-Lagoutière** ([Després and Lagoutière, 1999]) antidissipative scheme.

The Després-Lagoutière scheme is first-order accuracy but designed to reduce numerical diffusion even with a coarse resolution. This scheme is very useful to represent long-range advection of concentrated plumes ([Lachatre et al., 2020]).

7.2.2.2 Vertical wind calculation

```
[is\_diagwinw] Diagnose vertical wind from horiz components : .true.
```

New in release 2020a: setting `is_diagwinw` to `.true.` is the historic choice made by CHIMERE as most other CTMs. setting `is_diagwinw` to `.false.` is a new possibility offered by CHIMERE 2020a.

7.2.2.2.1 is_diagwinw| set to .true. In this configuration, vertical transport is assumed to balance horizontal mass divergence. To enforce this assumption, the vertical mass flux of dry air is reconstructed columnwise for each cell interface in the vertical direction, successively from model bottom to model top, in the following way :

- In the lowest layer, the lateral mass imbalance is calculated by summing up all the incoming mass fluxes at the lateral boundaries of the cell and substracting the outgoing mass fluxes at lateral boundaries, the wind at the upper interface of the lowest model layer is then calculated in order to compensate this mass imbalance by an opposite flux at the cell's upper boundary. This calculation is performed at the first order, assuming that the density of the air crossing a cell boundary is equal to the average density of the upstream cell, both for horizontal and vertical transport.
- The same calculation is then performed successively for each cell in all the column up to the model top, balancing the total imbalance of the mass flux from lateral and lower boundaries by an opposite mass flux at the upper boundary. These vertical transport mass fluxes (independent of species) are calculated first at each coarse time step in routine src/vtransport.f90

As discussed in [Lachatre et al., 2020], this method permits to ensure mass conservation for all species, to the expense of errors in the vertical wind. These errors are moderate in the lowest model layers but substantial in the free troposphere. The calculation of the concentration tendancies due to vertical transport for each model species is then performed at each

fine time step in routine src/transmix.f90. As the latter transport scheme needs two concentration values upstream of the considered boundary and one concentration value downstream, and this can not be secured next to model bottom and model top, CHIMERE switches back to upwind vertical transport for the cells where only one value of upstream concentration is available within the vertical simulation domain (including the boundary condition at model top).

7.2.2.2.2 is_diagwinw set to .false. Unlike the previous case, the vertical wind is directly interpolated from the forcing model (both in offline and online mode). As discussed in [Lachatre et al., 2020], this method should permit a more realistic representation of vertical transport, but makes it necessary to introduce a term to compensate the mass-wind inconsistencies that are introduced due to interpolation errors for winds and density fields. The detailed methodology for this new option as well as its advantages and drawbacks is discussed in [Lachatre et al., 2020].

7.2.3 Turbulent mixing

At each interface between layers k and k+1, one calculates an equivalent turbulent vertical velocity V_k and vertical mixing is expressed in layer k by the mass flux:

$$\text{Net Flux} = \frac{V_k(C_{k+1}.D_k/D_{k+1} - C_k)}{H_k} \tag{7.34}$$

where C denotes concentrations, D densities and H layer thicknesses. Density ratios are applied since turbulent mixing must conserve mass in each model cell. An equivalent formula is used for the upper layer. The turbulent velocity V_k is deduced from K(z) using:

$$V_k = \frac{K_z}{(\frac{1}{2}(H_k + H_{k+1}))} \tag{7.35}$$

When the top of the boundary layer is above the model top, H_{k+1} is replaced by h - <model-top> in the previous equation. The K(z) profile is calculated in routine src/calc_turb.F90 and the turbulent velocity profile in routine src/mixing.F90. Vertical mixing mass fluxes (independent of species, first order) are calculated first at each coarse time step in routine src/vtransport.F90, then the complete calculation using the species concentration is performed at each fine time step in routine src/transmix.F90.

7.3 The direct aerosol effects

7.3.1 Model configuration

In this case, we are in the configuration:

```
# Online run parameters
   [$online]   on-line [1] or off-line [0] modeling  : 1
   [$cpl_case] coupling case                         : 2 or 4
```

The direct interactions between the aerosol particles and the solar radiation has been implemented for both short-waves and long-waves. CHIMERE send informations about aerosols under the form of optical depth to WRF.
A new Fortran module, called aerosol_effects (in the new aerosol_effects.F90 file) was created in order to gathers all WRF-CHIMERE aerosols effects related subroutines.
In the WRF input file namelist.input, the following configuration must be set-up:

```
ra_sw_physics  =  4
ra_lw_physics  =  4
direct_feed_chimere  =  1
```

7.3.2 The aerosol optical properties feedback

Aerosol optical properties are computed in the CHIMERE model before being sent to WRF, through the OASIS coupler. These variables are:

1. The aerosol optical depth (AOD),

2. The single scattering albedo (SSA),
3. The asymmetry factor (AF)

After receiving the fields, WRF will used them as an input for its RRTMG (Rapid Radiative Transfer Model for General circulation models) scheme [Iacono et al., 2008]. CHIMERE will send the aerosol optical properties right after receiving the meteorological fields. Both ways OASIS exchanges are done at the beginning of both models time iterations, thus allowing the two models to perform their time iteration concurrently. Each model can need to wait for the other in order to receive the required fields that will allow it to continue the run. Note that initial aerosols optical properties that are sent to WRF can be provided through CHIMERE end...nc file in case of a continuation run, or are set to zero otherwise.
Aerosol optical properties that are send to WRF are the SSA and AF at 400nm and 600nm along with AOD at 300nm, 400nm, 999nm and at 16 long-wavelengths ranging from 3400mn to 55600nm.
The mass concentration distribution of each aerosol species is interpolated on a finer resolution than the size distribution used by CHIMERE. However, the initial 12 bins mass is conserved. Mie theory is used to find the scattering and absorption efficiencies. The photo_jx subroutine (fjx_sub_mod.F90 file) has been modified in order to include the SSA and AF calculations. The SSA is calculated for each CHIMERE layer and wave-lengths as follows:

$$\omega(z,\lambda) = \frac{\sum_{i=1}^{N_b}\sum_{j=1}^{N_s}\omega_{i,j}(\lambda)\tau_{i,j}(z,\lambda)}{\tau(z,\lambda)}$$

where $\omega(z,\lambda)$ is the SSA in a given layer z at the length wave λ, N_b and N_s are the number of bins (12) and of aerosol species respectively, i and j the $i-th$ bin and $j-th$ aerosol species respectively, $\omega_{i,j}(\lambda)$ and $\tau_{i,j}(z,\lambda)$ the SSA and AOD of the species j in the bin i at length wave λ, $\tau(z,\lambda)$ the total AOD.

The AF is calculated following [Irvine, 1976]:

$$g(z,\lambda) = \frac{\sum_{i=1}^{N_b}\sum_{j=1}^{N_s}\frac{\alpha_{1_{i,j}}(\lambda)}{3}\omega_{i,j}(\lambda)\tau_{i,j}(z,\lambda)}{\sum_{i=1}^{N_b}\sum_{j=1}^{N_s}\omega_{i,j}(\lambda)\tau_{i,j}(z,\lambda)}$$

where $\alpha_{1_{i,j}}(\lambda)$ is the first coefficient of Legendre polynomials of the species j in the bin i at length wave λ.
Short-wave AOD are already calculated in CHIMERE at 200nm, 300nm, 400nm, 600nm and 999nm. Those AOD are calculated by assigning a complex refractive index to each aerosol species and by assuming an external mixing state. The 16 long-wavelengths AOD are following the same method as the short-waves AOD.
A new file containing the refractive indexes of the aerosol species at the long-wavelengths was created (i.e. optical_lw.txt) in a new sub-directory (i.e. data_optical) of the chemprep directory. Moreover, the prep_mie.F90 program that initializes the aerosol optical properties has been modified in order to include the initialization of aerosol optical properties in the long-wave bands following closely the code already implemented for visible bands.
Aerosol optical properties computed by CHIMERE are interpolated on WRF vertical grid, using the chm2wrf_vert_grid subroutine (module aerosol_effects), before being sent through the OASIS coupler. WRF top levels usually goes up to 50 hPa for most applications. However the CHIMERE maximum allowed top level is 200 hPa. Therefore, WRF optical properties climatology [Péré et al., 2014] is used for the highest vertical levels. However, it should be noted that the climatology is used only for the short-waves and that the infrared AODs are set to zero above CHIMERE top level.
The aerosol optical properties received from CHIMERE are used in WRF as input for radiative schemes. The AODs at 300nm, 400nm and 999nm are interpolated on the bands of RRTMG-SW by using an Ångström power law, while the SSA and AF at 440nm and 600nm are interpolated assuming a linear relation with wave lentgh. The AODs computed in the long-wave range are added in RRTMG-LW to the gases optical depth.

7.4 The indirect aerosol effects

7.4.1 First and second indirect effects

The first step to take into account the aerosol indirect effects in the model is to diagnose the droplet number concentration from activation of aerosol particles predicted by CHIMERE. Indirect effects are implemented through the cloud microphysics scheme of [Thompson and Eidhammer, 2014]. This scheme is an aerosol aware microphysics parameterization including already a scheme for cloud droplet nucleation, where aerosol particle number concentration is bases

on a multi-years global model simulation, and aerosols are activated using a lookup table created by an explicit treatment of the Kohler theory assuming the lognormal distribution of the aerosol particles. Because CHIMERE uses a sectional treatment for aerosols, the activation scheme of [Thompson and Eidhammer, 2014] has been replaced with the [Abdul-Razzak and Ghan, 2002] parameterization. In this scheme, the Kholer theory is used to relate the aerosol size distribution and composition to the number activated for each section as a function of maximum supersaturation.
According to the Kholer theory, the dry radius of the smallest activated particle is given by

$$r_c = \frac{A}{3}\left(\frac{\sqrt{B}}{2}S_{max}\right)^{-\frac{2}{3}}$$

where A is the curvature coefficient (Kelvin term), B the hygroscopicity paramater and S_{max} the maximum supersaturation of an adiabatic rising parcel. Following [Abdul-Razzak and Ghan, 2002], S_{max} is parameterized as

$$S_{max} = \frac{S_e}{\left[0.5\left(\frac{\zeta}{\eta}\right)^{\frac{3}{2}} + \left(\frac{S_e^2}{\eta+3\zeta}\right)^{\frac{3}{4}}\right]^{\frac{1}{2}}}$$

where η and ζ are functions of the updraft velocity and total aerosol particle number concentration, S_e is an effective critical supersaturation given by

$$S_e^{\frac{2}{3}} = \frac{\sum_{i=1}^{N_b} N_i}{\sum_{i=1}^{N_b} N_i / S_i^{\frac{2}{3}}}$$

where N_i is the aerosol number concentration in the model section i, N_b the number of section, and S_i the critical supersaturation of a particle with dry radius equal to the mean radius r_i of a section i:

$$S_i = \frac{2}{\sqrt{B}}\left(\frac{A}{3r_i}\right)^{\frac{3}{2}}.$$

Mean hygroscopicity parameter is based on the volumetric average of hygroscopicity of the single model species:

$$B = \frac{\sum_i^{N_b}\sum_j^{N_a} B_j m_{i,j}/\rho_j}{\sum_i^{N_b}\sum_j^{N_a} m_{i,j}/\rho_j}$$

where i and j are the $i-th$ section and $j-th$ aerosol species respectively, $m_{i,j}$ mass concentration of the species j in the section i, and ρ_j the density of the species j. B_j is the hygroscopicity parameter of the species j and is based on the values reported by [Petters and Kreidenweis, 2007]. The values of B used in the model are reported in the Table below.

Species	Hygroscopicity	Species	Hygroscopicity
H_2S0_4	0.50	PPM	0.10
$HN0_3$	0.50	BCAR	1.0×10^{-6}
NH_4	0.50	DUST	0.03
ASOA	0.14	SEA SALT	1.16
BSOA	0.20	NA	1.16
OCAR	0.10	CL	1.16

Table 7.4: *Hygroscopicity parameters used in CHIMERE.*

Once r_c is known, the fraction f_i of activated particles in each model section is determined comparing r_c with the lower r_{il} and upper r_{iu} bounds of a given section i:

$$f_i = \begin{cases} 0 & \text{if } r_c > r_{iu} \\ 1 & \text{if } r_c \leq r_{il} \\ \dfrac{\log\left(r_{iu}/r_c\right)}{\log\left(r_{iu}/r_{il}\right)} & \text{if } r_{il} < r_c \leq r_{iu} \end{cases}$$

The number of aerosol activated as cloud droplet is calculated over a Gaussian spectrum of updraft velocities. For a single updraft of the spectrum w, the number of activated aerosols is determined integrating the model size distribution as:

$$N(w) = \sum_{i}^{N_b} f_i(w) N_i$$

thus, the mean number of activated aerosol in a grid cell is obtained averaging $N(w)$ over the probability distribution $p(w)$

$$N^* = \frac{\int_0^\infty N(w)p(w)dw}{\int_0^\infty p(w)dw}.$$

$p(w)$ is a Gaussian function, the grid cell mean vertical velocity $\bar{w}$ is given by large scale vertical velocity and the standard deviation σ_w represents the subgrid variablity that we assume that is due to the turbulence, [Ghan et al., 1997]. σ_w is diagnosed from vertical eddy diffusity K_z as:

$$\sigma_w = \sqrt{2\pi}\frac{K_z}{\Delta z}.$$

Droplet nucleation is done at cloud base, in forming cloud and every where N^* is greater than the existing droplet concentration N_d. As a consequence, the droplet nucleation rate over a model time step Δt is given by:

$$R_d = max\left(\frac{N^* - N_d}{\Delta t}, 0.0\right)$$

Aerosol activation as cloud droplet is calculated in WRF. Aerosol number size distribution and mean hygroscopicity are diagnosed in from CHIMERE aerosol mass in the routine aero_properties (module aerosol_effects.F90). These are interpolated on WRF vertical grid through the routine chm2wrf_vert_grid, before being sent to WRF with OASIS. Within WRF, they are used to form the cloud droplet in the module module_mp_thompson.F90, where a new routine droplet_activation has been implemented in order to calculate the aerosol activation following the scheme described above.
For the cloud ice formation, it is calculated in the [Thompson and Eidhammer, 2014] using the IN calculated in CHIMERE. For IN estimation, only mineral dust concentrations with mean mass median diameter $D_p > 0.5\mu m$ are taken into account. The scheme for heterogeneous ice nucleation is the one of [DeMott et al., 2015]. Homogeneous freezing of deliquesced aerosols is parameterized as in [Thompson and Eidhammer, 2014] following the method of [Koop et al., 2001]. The climatology of deliquesced aerosol number concentration is replaced with the CHIMERE prediction and is based on a mixture of hygroscopic particles with the diameter larger than 0.1 μm. Further details about the implementation of the aerosol indirect effects within WRF-CHIMERE are provided by [Tuccella et al., 2019].
When taking into account the effect of aerosols on cloud formation, and when the regional model has a top in the middle of the troposphere or close to the UTLS, the problem of the top boundary condition has to be solved. Having no realistic information about the AOD and CCN above the model top, climatological values are used and then interpolate to have realistic values.

8 Initial and boundary conditions

While some regional chemistry transport model rely on observation-based boundary concentrations, the CHIMERE strategy relies exclusively on simulations from global models. The added value in geographical and temporal variability as well as the larger number of chemical species available argue in favour of this second option, despite the increased sensitivity of the limited area CTM to any bias in the large scale forcing data. In order to cope with possible limitations in the global model forcing, we choose to increase the modularity of the global to regional nesting by allowing the use of several sources of boundary conditions (i.e. simulations from different models), or even combinations of such sources.

8.1 Options activation

It is possible to run the model without any boundary or initial conditions. But the best way is to use available informations, coming from previous simulations or other models outputs.

```
# Chemical boundary conditions
#    0 - do not use
#    1 - use and build
#    2 - check if files exist and build only
 [iusebound]     Use and build Boundary conditions? (0/1/2) : 2
 [iuseini]       Use and build Initial conditions? (0/1/2)  : 2
 [nest_topconc]  Use global top conc? (when same vert grid) : 1
```

These options correspond to:

- Initial conditions:
  - ◇ iuseini=0 : all chemical species are set to zero at the first time-step
  - ◇ iuseini=1 or 2
    - ◇ Use climatological data if the simulation starts (no end.nc file available)
    - ◇ Use the previous simulation if the run is chained (end.nc file available)
- Boundary conditions:
  - ◇ iusebound=0 : all chemical species are set to zero around the domain and during the whole simulation
  - ◇ iusebound=1 or 2
    - ◇ Use climatological data if nested=no
    - ◇ Use another CHIMERE simulation if nested=yes

The *nest_topconc* enables to select the way to manage the concentrations at the top of the domain.

- nest_topconc=0 → use coarse simulation's
- nest_topconc=1 → use global model's

Note: in case of nesting, and if simulations are not with the same vertical levels, the coarse domain must always be higher than the fine one, as displayed in Figure 8.1 .

These several options are summarized in Figure 8.2 . The initial and boundary conditions are different all along the run but their preparation is managed in the same way. The main difference is that the **initial conditions** are necessary for the first time-step only but over the whole simulation domain when the **boundary conditions** are necessary every hour but only for the model cells surrounding the simulation domain.
In case of use of boundary and initial conditions, there is two possibilities.

- First, it is possible to use a CHIMERE simulation with a larger domain to have these conditions.

```
[nested]      Nested run? (yes/no)                          : no, yes
[dom]         CHIMERE domain                                : CONT5, IDF15C
```

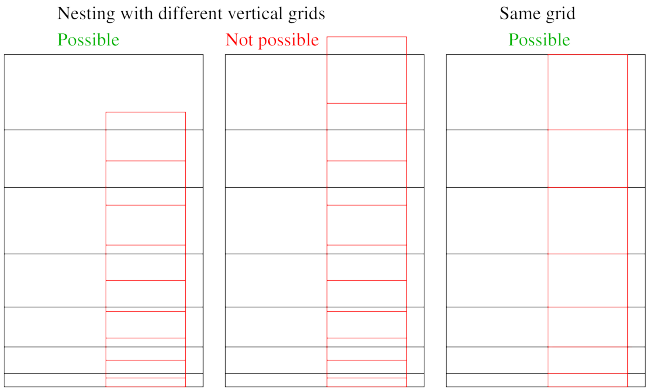


Figure 8.1: *Different possibilities for the nesting.*

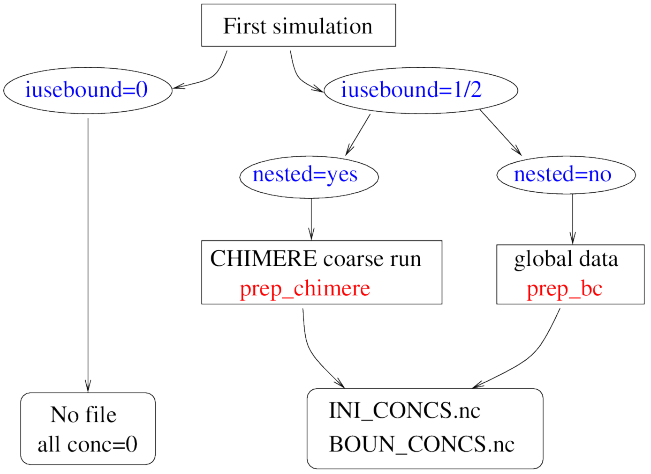


Figure 8.2: *Flowchart of the several options to build the initial and boundary conditions.*

In this case, the user needs to preselect iusebound=1 or 2. For the domain *IDF15C*, the nesting is selected to *yes*, then files will be prepared from the simulation with the *CONT5* domain.

- Second, it is possible to use global datasets only.

```
[nested]      Nested run? (yes/no)                          : no
[dom]         CHIMERE domain                                : CONT5
```

In this case, and if, for example, there is only one domain, boundary conditions are from climatologies. It is necessary to provide the path where the global datafiles are stored: bcdir, together with a label pointing towards the relevant source (INCA, MACC, GOCART) as the name of the subdirectory where global data where extracted. The source of boundary data can be a composite of several global models for gas, dust and non-dust aerosol by using the right flags: bcgas, bcdust, and bcaer. And for each of these flag the user can indicate whether climatological data (monthly means) are desired (with bcgasdt, bcaerdt, or bcdustdt = 0), or if a higher temporal frequency is available (setting these flags to 1) in that case the fortan code extract automatically the highest available temporal resolution.
The use of these conditions is managed by the following options available in the chimere.par file:

```
[bcdir]        Boundary conditions                          : {bigfilesdir}
[bcgas]        BC Gaseous Species                           : LMDz4_INCA3_96x95x19_v2013c
[bcgasdt]      Use of gas bc (0/1/2)                        : 1
[bcaer]        BC non-dust  Aerosols                        : LMDz4_INCA3_96x95x19_v2013c
[bcaerdt]      use of aer bc (0/1/2)                        : 1
[bcdust]       BC Dust                                      : GOCART
[bcdustdt]     Use of dust bc (0/1/2)                       : 1
```

Note there is no two-ways nesting in CHIMERE, but only a nesting from the coarse domain to the fine one. Then even if WRF has the capability to manage two-ways nesting, this option is not possible with the online coupling WRF-CHIMERE.

8.2 The boundary and initial conditions interface

After reading the chimere.par file, and if the boundary conditions are activated, the script scripts/chimere-bound.sh will prepare required data. There is mainly two options:

1. This is not a nested simulation: the script scripts/chemprep-prep-boun-data.sh is launched during the chemprep stage. Then the script scripts/chimere-bound.sh launchs the program src/prep_bc to extract the global climatological data.
2. This is a nested simulation: the script scripts/chimere-bound.sh launchs the program src/prep_chimere to read the previously done coarse CHIMERE simulation.

At the end of preparation of these files, the results are stored in a dedicated directory defined in the scripts/define_params.sh script:

```
iniboundir={simuldir}/IBC
```

It is recommended to not change this pathname. The directory contains several files, depending on the simulated period and named:

1. INI_CONCS.nc-<speciestype> for the initial conditions
2. BOUN_CONCS.nc-<speciestype> for the boundary conditions

with <speciestype> the three flags: gas, aer and dust. A list of these files is also generated in order to have the exact name of the files to open during the CHIMERE simulation.

8.2.1 In case of a nested run

If the flag nested=yes, the concentrations fields from a previous coarse CHIMERE simulation are used both for initial and boundary conditions. The complete extraction is performed with the src/prep_chimere program. This program will prepare the initial and boundary conditions files using an existing out.[coarse-label].nc simulation output.
For the initial conditions: if the simulations starts, the concentrations of the coarse run are interpolated into the nested domain. If this is a restart simulation, the previous simulation created a end.nc file. This file is renamed as ini.nc file and read durectly during the initialization phase of CHIMERE by the program src/iniconc.F90. The format of this end.nc file is fully described in **Appendix E.5, p.269**.
For the boundary conditions, the src/prep_chimere program interpolates the coarse simulation on the nested domain with an hourly frequency. In this case, the calculations are relatively fast, the model species being the same between the coarse and the nested, thus not needing chemical conversions.

8.2.2 In case of a climatology use

If the simulation if not a nested run, it is necessary to read climatological data to ensure realistic initial and boundary conditions. The calculations are done by the script scripts/chimere-bound.sh launching the src/prep_bc.F90 program.
By reading the arguments in the chimere.par file, the script scripts/chimere-bound.sh will link towards the right type of input data for each type of compound (gas, aer, dust). If several sources are required, the boundary condition extraction will be performed sequentially before being merged into a single file. In the case of time varying data, the script will try to locate the boundary conditions of the relevant years. Otherwise, for climatological data, the code looks for monthly data in the "clim" subdirectory. The boundary condition year is then handled as "0000".
The geometrical extraction as well as unit conversion is ultimately performed in the src/prep_bc.F90 program. This code is generic and ignores the origin of the input data as long as the correct input format is provided. It can handle any temporal resolution of the boundary conditions and makes the difference between climatological and time-varying fields by checking whether the boundary condition year is "0000".

8.3 Available global models climatologies

Three sources of boundary conditions were kindly made available by global chemistry-transport modelling groups to the CHIMERE users:

1. The MACC climatology
2. A climatology from the global model LMDZ-INCA
3. A climatology from the global model GOCART

Their main characteristics are summarised below. The users are invited to take good care of acknowledging the work of their global modelling colleagues in their products and publications.
All the global fields provided for download are monthly climatologies (ie. averages over several years for a given month of the year) but the code has the capacity to use time-varying boundary conditions if the user has such fields available (see section 8.2).

8.3.1 The MACC climatology

The global chemical reanalysis produced as part of the MACC II project (Monitoring Atmospheric Composition and Climat II) is available. The global MACC service provides a 5-years reanalysis for the period 2004-2009 of trace gases and aerosol concentrations. A global monthly climatology at a resolution of 1.125^o is made available to CHIMERE users. The MACC modelling system relies on the coupled IFS-Mozart [Horowitz, 2003] modelling and assimilation system for reactive gases and on the MACC prognostic aerosol module for particulate matter.
Detailed validation of the MACC reanalysis can be found at:
http://www.copernicus-atmosphere.eu/services/aqac/global_verification/validation_reports/ .
For users, in case of publication using these MACC data, please add in your acknowledgements:

> *This data set was provided by the MACC-II project, which is funded through the European Union Framework 7 programme. It is based on the MACC-II reanalysis for atmospheric composition; full access to and more information about this data can be obtained through the MACC-II web site http://www.copernicus-atmosphere.eu).*

The list of chemical species available in the MACC reanalysis is given in the list below.

- BC ..... Black carbon
- BIGALK ..... Lumped alkans C>3
- BIGENE ..... Lumped alkenes C>3
- C10H16 ..... lumped monoterpense
- C2H6 ..... Ethane
- CH2O ..... Formaldehyde
- CH3CHO ..... Acetaldehyde
- CH4 ..... Methane
- CO ..... Carbon monoxide
- DUST1 ..... Desert dusts (0.03-0.55 μm)
- DUST2 ..... Desert dusts (0.55-0.9 μm)
- DUST3 ..... Desert dusts (0.9-20 μm)
- GLYOXAL ..... Glyoxal
- H2O2 ..... Hydrogen peroxide
- HNO3 ..... Nitric acid
- ISOP ..... Isoprene
- NH3 ..... Ammonia
- NO2 ..... Nitrogen dioxide
- O3 ..... Ozone
- OM ..... Particulate organic matter (0.1-1 μm)
- PAN ..... Peroxy acetyl nitrate
- SO2 ..... Sulfur dioxide
- SO4 ..... Sulfate
- TOLUENE ..... Toluene

8.3.2 The LMDz-INCA climatology

The historical and still commonly used set of boundary conditions is provided by the LMDz-INCA model (Laboratoire de Météorologie Dynamique General Circulation Model coupled with INCA: Interaction with Chemistry and Aerosols). The original INCA model is described in ([Hauglustaine et al., 2004] and [Folberth et al., 2006]) while developments related to aerosol modelling are presented in ([Bauer et al., 2004], [Schulz et al., 2006] and [Balkanski et al., 2007]). The specific version presented here is LMDZ4-INCA3 with the nitrate chemistry introduced in [Hauglustaine et al., 2014]. It has 96x96 horizontal grid points and 19 vertical levels.
The list of chemical species available in the LMDz4-INCA3 model is given below. An example of surface ozone concentrations modelled with LMDz-INCA and MACC is presented in Figure 8.3 .

- ALKAN ..... Lumped alkans C > 3
- AROM ..... Lumped aromatics
- BC ..... Black carbon
- C2H4 ..... Ethylene
- C2H6 ..... Ethan
- C3H6 ..... Propen
- CH2O ..... Formaldehyde
- CH3CHO ..... Acetaldehyde
- CH4 ..... Methane
- CNO3 ..... Coarse Nitrate (2.5-5 μm)
- CO ..... Carbon monoxide
- DUST ..... Desert dusts (1-10 μm)
- H2O2 ..... Hydrogen peroxide
- HNO3 ..... Nitric acid
- NH3 ..... Ammonia
- NH4 ..... Particulate ammonium (0.1-1 μm)
- NO2 ..... itrogen dioxide
- NO3 ..... Particulate nitrate (0.1-1 μm)
- O3 ..... Ozone
- OC ..... Particulte organic carbon (0.1-1 μm)
- PAN ..... Peroxy acetyl nitrate
- SO2 ..... Sulfur dioxide
- SO4 ..... Sulfate

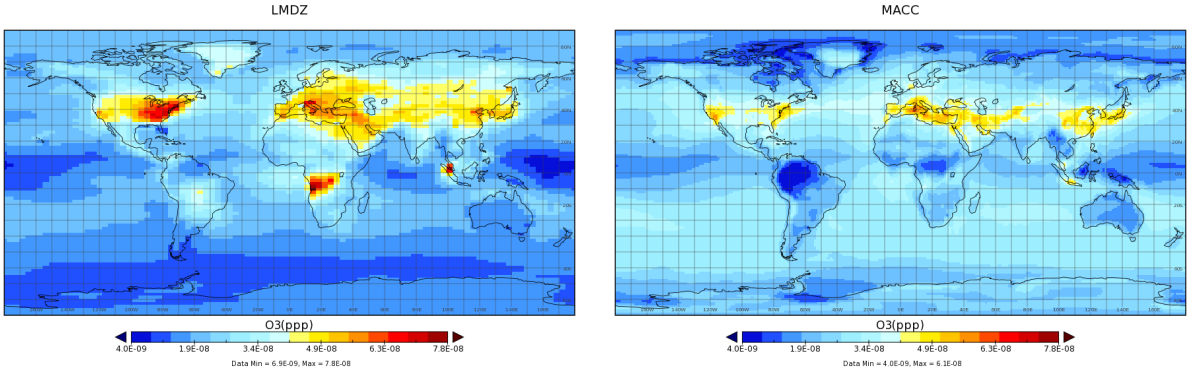


Figure 8.3: *Surface ozone concentrations modelled with LMDz-INCA and MACC for july.*

8.3.3 The GOCART climatology

The Global model GOCART (NASA,[Ginoux et al., 2001]) was historically the only source of boundary conditions for particulate matter, it can still be used in combination with other sources for gaseous boundary conditions.

The list of chemical species available in the GOCART model is given in the list below. For each species, the size distribution is given in μm. An example of surface mineral dust concentrations modelled with GOCART and MACC is presented in Figure 8.4 .

- DUST1 ..... Desert dusts (0.2-0.36)
- DUST2 ..... Desert dusts (0.36-0.6)
- DUST3 ..... Desert dusts (0.6-1.2)
- DUST4 ..... Desert dusts (1.2-2)
- DUST5 ..... Desert dusts (2-3.6)
- DUST6 ..... Desert dusts (3.6-6)
- DUST7 ..... Desert dusts (6-12)
- HBBC1 ..... Hydrophobic black carbon (0.02-0.06)
- HBBC2 ..... Hydrophobic black carbon (0.06-0.18)
- HBBC3 ..... Hydrophobic black carbon (0.18-0.54)
- HBBC4 ..... Hydrophobic black carbon (0.54-1.62)
- HBBC5 ..... Hydrophobic black carbon (1.62-4.86)
- HBBC6 ..... Hydrophobic black carbon (4.86-14.58)
- HBOC1 ..... Hydrophobic organic carbon (0.02-0.06)
- HBOC2 ..... Hydrophobic organic carbon (0.06-0.18)
- HBOC3 ..... Hydrophobic organic carbon (0.18-0.54)
- HBOC4 ..... Hydrophobic organic carbon (0.54-1.62)
- HBOC5 ..... Hydrophobic organic carbon (1.62-4.86)
- HBOC6 ..... Hydrophobic organic carbon (4.86-14.58)
- HLBC1 ..... Hydrophylic black carbon (0.02-0.06)
- HLBC2 ..... Hydrophylic black carbon (0.06-0.18)
- HLBC3 ..... Hydrophylic black carbon (0.18-0.54)
- HLBC4 ..... Hydrophylic black carbon (0.54-1.62)
- HLBC5 ..... Hydrophylic black carbon (1.62-4.86)
- HLBC6 ..... Hydrophylic black carbon (4.86-14.58)
- HLOC1 ..... Hydrophylic organic carbon (0.02-0.06)
- HLOC2 ..... Hydrophylic organic carbon (0.06-0.18)
- HLOC3 ..... Hydrophylic organic carbon (0.18-0.54)
- HLOC4 ..... Hydrophylic organic carbon (0.54-1.62)
- HLOC5 ..... Hydrophylic organic carbon (1.62-4.86)
- HLOC6 ..... Hydrophylic organic carbon (4.86-14.58)
- SULF1 ..... Sulfates (0.02-0.06)
- SULF2 ..... Sulfates (0.06-0.18)
- SULF3 ..... Sulfates (0.18-0.54)
- SULF4 ..... Sulfates (0.54-1.62)
- SULF5 ..... Sulfates (1.62-4.86)
- SULF6 ..... Sulfates (4.86-14.58)

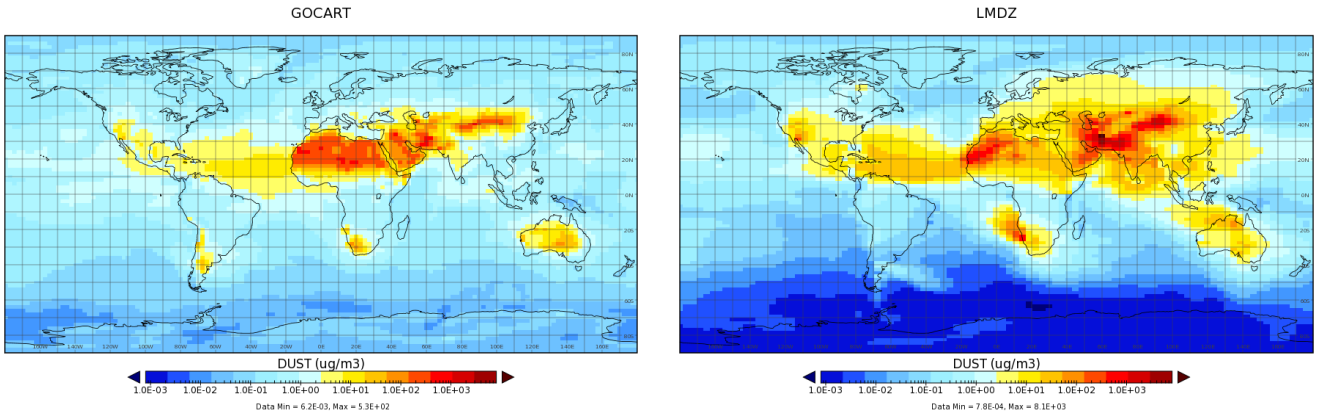


Figure 8.4: *Surface mineral dust concentrations modelled with GOCART and LMDzINCA for july.*

8.3.4 The boundary conditions builder files

In case of use of a climatology, it is necessary to prepare a 'builder' file. The main goal is to be able to convert the global files model species into the CHIMERE model species. It is also necessary to manage the several units available, the type of the species (gaseous or aerosol). This part is managed with the scripts/chemprep-prep-boun-data.sh script.
The file XXX_builder (XXX is the name of a global concentrations dataset, GOCART, LMDZ-INCA or MACC at the moment) is used to build the interface between XXX and CHIMERE species. These files are split in 5 files types.

- For gas species : XXX_builder.meca where meca represents the chemical mechanism name (SAPRC/MELCHIOR1/MELCHIOR2)
- For SOA : XXX_buider.aer.soa where soa represents the SOA mechanism used (simple/medium/complex)
- For dust : XXX_buider.dust if the DUST specie are taken into account
- For primary aerosol species : XXX.car if the carbonaceous species are taken into account and ppmcarb if there are no carbonaceous species.
- For sea salts : xxx_builder.nacl if the sea salts are taken into account

Should any modification be added to the chemical scheme, these builders have to be modified accordingly. For example, the builder for LMDz-INCA and the SAPRC chemical mechanism is chemprep/boundaries_spec/INCAv2012_builder.saprc as:

```
# Coarse species, typde, number of corresponding CHIMERE species,
# and for each species: name, molar mass, conversion factor
ALKAN      gas  1    ALK3   59.    1.
AROM       gas  1    ARO2  119.    1.
C2H4       gas  1    C2H4   28.    1.
C2H6       gas  1    ALK1   30.    1.
C3H6       gas  1    OLE1   72.    1.
CH2O       gas  1    HCHO   30.    1.
CH3CHO     gas  1    CH3CHO 44.    1.
CH4        gas  1    CH4    16.    1.
CO         gas  1    CO     28.    1.
H2O2       gas  1    H2O2   34.    1.
HNO3       gas  1    HNO3   63.    1.
NH3        gas  1    NH3    17.    1.
NO2        gas  1    NO2    46.    1.
O3         gas  1    O3     48.    1.
PAN        gas  1    PAN   100.    1.
SO2        gas  1    SO2    64.    1.
```

All global forcing data are given in netcdf format with one file per month. Additional information on the input data are provided in the netcdf attributes for each variable. The most important attributes are "unit":

- **ppp** for part per part (mixing ratio),
- **ppb** for part per billion of 1e9 ppp,
- **ug** for micro-gram per cubic meter)

"typ" indicates if the variable is a gas ("gas"), desert dust ("dust") or non-dust aerosol ("aer"). The molar mass as used in the global model should also be provided.
In addition to concentration fields, the input files have latitude and longitude coordinates as well as the vertical grid (either 3D pressure on model levels or surface pressure + alpha and beta coefficients for the sigma-p grid). Would some conversion be required to provide concentrations in ppb to Chimere from the native unit in the global model (e.g. μg m^{-3}), the 3D temperature field is also requested in order to derive air density. Alternatively a constant and 3D uniform temperature of 15^{o}C is used.
The correspondence between the chemical species of the global model and Chimere is handled in the chemprep part of the code. This way, it is possible to use any source of input data for any of the chemical mechanism of Chimere.
It is also **possible to add a new global dataset**: Basic netcdf manipulation using NCO or CDO tools should allow advanced users to convert any global model fields into the appropriate format following the structure of the existing forcings and keeping one file per month, even if high frequency data are available. A new builder also needs to be written for each type of forcing.

Emissions

CHIMERE accounts for most possible types of emissions. While the anthropogenic and fire emissions need to be processed before the beginning of the simulation, all the other types of emissions (mineral dust, biogenic emissions, sea-salt emissions and tracers) are estimated by the CHIMERE main program itself during the simulation.
The emissions fluxes calculations are described in:

1. §9, p.89: anthropogenic emissions
2. §10.1, p.105: biogenic emissions
3. §10.2, p.106: sea salt emissions
4. §10.3, p.106: DMS emissions
5. §10.4, p.107: mineral dust emissions
6. §10.5, p.117: vegetation fires emissions
7. §10.6, p.120: the resuspension
8. §11.4, p.133: tracers emissions (in the chemprep chapter)

9 Anthropogenic surface emissions

Anthropogenic emission fluxes for CHIMERE simulations may be provided in two alternative file formats:

- The EMIS.$dom.$MON.$SPECIES.s.nc set of files: This is the output of the emiSURF program developed and distributed along with the CHIMERE code (Section 9.2). The user may produce these files with a different program but needs to include all required information in the appropriate format. These files are further processed within CHIMERE to (i) provide hourly fluxes for the simulation period and ii) to vertically interpolate emission fluxes on the CHIMERE vertical layers (module (prep_aemis.F90)).
- The AEMISSIONS.${sim}.nc file. This file contains the actual hourly fluxes for the CHIMERE simulation on the 3D simulation grid. This file has to be placed in the ${simuldir}/data_${dom}_${lab} directory.

The parameters related to these options in the chimere.par file are the following:

```
[$asurfemis]   anthro surface emissions? (1/0)       : 1
[$imakeaemis]  Build anthropogenic emissions? (0-2)  : 2
[$emistyp]     Anthropogenic emission format type    : emisurf
```

The *[asurfemis]* indicates whether surface, anthropogenic emissions will be used for the simulation (asurfemis=1) or not (asurfemis=0). The *[imakeaemis]* indicates whether the AEMISSIONS.${sim}.nc needs to be created from the monthly fluxes or not, as follows:

- imakeaemis=0: The user has already created the AEMISSIONS.${sim}.nc file in the appropriate format and has placed it in the appropriate directory.
- imakeaemis=1: The CHIMERE preprocessor will read the monthly emission fluxes prepared by the emiSURF or some equivalent program and create the AEMISSIONS.${sim}.nc file.
- imakeaemis=2: This option only makes sense if the *[$emistyp]* is set to 'emisurf'. CHIMERE will check if the AEMISSIONS.${sim}.nc file exists and if not it will create it from the monthly emission fluxes (output of the emisurf program).

The flag *[$emistyp]* indicates the program used to prepare the EMIS.$dom.$MON.$SPECIES.s.nc set of files. Only the 'emisurf' value is actually implemented and in this case the *[$imakeaemis]* needs to be set to either 1 or 2. If an equivalent program has been used, the user still needs to indicate 'emisurf'. If the user has created her own AEMISSIONS.${sim}.nc, the value here needs to be anything other than 'emisurf' and *[$imakeaemis]*=0.

9.1 The emitted species for the CHIMERE simulation

Independently of the file format, to prepare anthropogenic emission input for the CHIMERE simulation, the user needs to know in advance the chemical mechanisms of the simulation for both gas-phase chemistry and secondary organic aerosol formation. Depending on these mechanisms the list of possible emitted species changes.
The parameters related to these options in the chimere.par file are the following:

```
[$mecachim]  Chemistry mechanism (0..2)                  : 2
[$aero]      Chemically-active aerosols (1/0)            : 1
[$carb]      Include carboneceous species?               : 1
[$soatyp]    Secondary organic aerosol scheme (0..2)     : 2
```

See Section 3 for the description of these options.
Table9.1 shows all the possibly emitted species for the MELCHIOR and SAPRC gas-phase chemical-mechanisms (*[$mecachim]* flag)
If *[$aero]*>0 primary particulate matter emissions are accounted for in the CHIMERE simulation. Depending on the reported pollutants in the emission inventories and the *[$carb]* and *[$soatyp]* options of the chimere.par namelist the emitted "model species" are different. Table9.2 shows all the possibly emitted species for primary particulate matter.

MELCHIOR2		SAPRC			
NO	Nitrogen monoxide	NO	Nitrogen monoxide	GLY	Glyoxal
NO2	Nitrogen dioxide	NO2	Nitrogen dioxide	HCHO	Formaldehyde
HONO	Nitrous acid	HONO	Nitrous acid	MEOH	Methanol
SO2	Sulphur dioxide	SO2	Sulphur dioxide	MACR	Methacrolein
NH3	Ammoniac	CO	Carbon monoxide	OLE1	Alkens k_{OH}<7x10^4
CO	Carbon monoxide	CH4 [1]	Methane	OLE2	Alkens k_{OH}>7x10^3
CH4 [1]	Methane	ACET	Acetone	CRES	Phenols and Cresols
C2H6	Ethane	ALK1	Ethane	PRD2	Ketones (k_{OH}>0.7x10^4 ppm^{-1}.mn^{-1})
NC4H10	n-Butane	ALK2	Propane	RCHO	Lumped C3+ Aldehydes
C2H4	Ethene	ALK3	Alkans 2.5x10^3 <k_{OH}< 5x10^3	C5H8	Isoprene
C3H6	Propene	ALK4	Alkans 5x10^3<k_{OH}<1x10^4	APINEN	α-pinene
C5H8	Isoprene	ALK5	Alkans 1x10^4<k_{OH}	ACYE	Acetylen
OXYL	o-Xylene	ARO1	Aromatiques k_{OH}<2x10^4	BENZ	Benzen
HCHO	Formaldehyde	ARO2	Aromatiques k_{OH}>2x10^4	AACD	Acetic Acid
CH3CHO	Acetaldehyde	BALD	Aromatic aldehydes	IPRD	Lumped isoprene product species
CH3COE	Methyl Ethyl Ketone	CCHO	Acetaldehyde		
APINEN	α-pinene	C2H4	Ethen		

Table 9.1: *List of emitted species for the MELCHIOR2 and SAPRC gas-phase chemical mechanisms*

Aerosols			
H2SO4_fin [5]	Primary sulfuric acid	POAlP	low volatility (h2o)
TPPM_big [4]	PPM mixed with OCAR and BCAR	POAmP	medium volatility (h2o)
TPPM_coa [3]	PPM mixed with OCAR and BCAR	POAhP	high volatility (h2o)
TPPM_fin [2]	PPM mixed with OCAR and BCAR	POAlP_fin [2]	low volatility (h2o)
OCAR_coa [3]	Organic carbon	POAmP_fin [2]	medium volatility (h2o)
OCAR_fin [2]	Organic carbon	BOAlP	low volatility from wood-burning residential (h2o, carb=3)
BCAR_coa [3]	Black carbon	BOAmP	medium volatility from wood-burning residential (h2o, carb=3)
BCAR_fin [2]	Black carbon	BOAhP	high volatility from wood-burning residential (h2o, carb=3)
PPM_coa [3]	Primary particulate matter	BOAlP_fin [2]	low volatility from wood-burning residential (h2o, carb=3)
PPM_fin [2]	Primary particulate matter	BOAmP_fin [2]	medium volatility from wood-burning residential (h2o, carb=3)
TOL	benzene, toluene and other mono-substituted aromatics	POA1_fin [2]	1^{st} volatility bin (vbs)
TMB	Trimethylbenzene and other poly-substituted aromatics	POA2_fin [2]	2^{nd} volatility bin (vbs)
OLE1	Alkens k_{OH}<7x10^4 (vbs)	POA3_fin [2]	3^{rd} volatility bin (vbs)
OLE2	Alkens k_{OH}>7x10^3 (vbs)	POA4_fin [2]	4^{th} volatility bin (vbs)
ALK4	Alkans 5x10^3<k_{OH}<1x10^4 (vbs)	POA5_fin [2]	5^{th} volatility bin (vbs)
ALK5	Alkans 1x10^4<k_{OH} (vbs)	POA6_fin [2]	6^{th} volatility bin (vbs)
ARO1	Aromatiques k_{OH}<2x10^4 (vbs)	POA7	7^{th} volatility bin (vbs)
ARO2	Aromatiques k_{OH}>2x10^4 (vbs)	POA8	8^{th} volatility bin (vbs)
		POA9	9^{th} volatility bin (vbs)

Table 9.2: *List of chemical species available for anthropogenic aerosol emissions.*

- [1]: CH4 emissions are always set to zero in this model version
- [2]: Species_fin are for diameters $\phi < 2.5$ μm
- [3]: Species_coa are for diameters $2.5 < \phi < 10$ μm
- [4]: Species_big are for diameters $\phi > 10$ μm
- [5]: These species are assumed to be emitted only in the fine mode.

If the user passes through the emiSURF program to prepare anthropogenic emission fluxes the handling of the emitted species list depending on the chemical mechanisms is automatic (see Section 9.2.4).

9.2 The emiSURF program

The emiSURF model is the pre-processor of the anthropogenic emission fluxes used in atmospheric composition simulations with the CHIMERE chemistry transport model. The emiSURF model can be used as a stand-alone model but in this case, the user needs to be aware of all the modeling stages before adapting the model output to other uses. The model takes as input gridded monthly or annual emission totals from several available inventories. The main operations applied by the emiSURF model on the inventory data are listed bellow:

- horizontal regridding of emissions from the inventory grid to the CHIMERE simulation grid
- distribution of emissions to vertical layers
- chemical speciation profiles to map emissions from inventory pollutants to 'model species'.
- temporal profiles to obtain hourly fluxes

This top-down approach results in a set of output files: one file per emitted "model species" and month. Monthly totals in these files are distributed across the seven days of the week and the 24 hours of the day through the "type_day" and "Time" dimensions respectively.

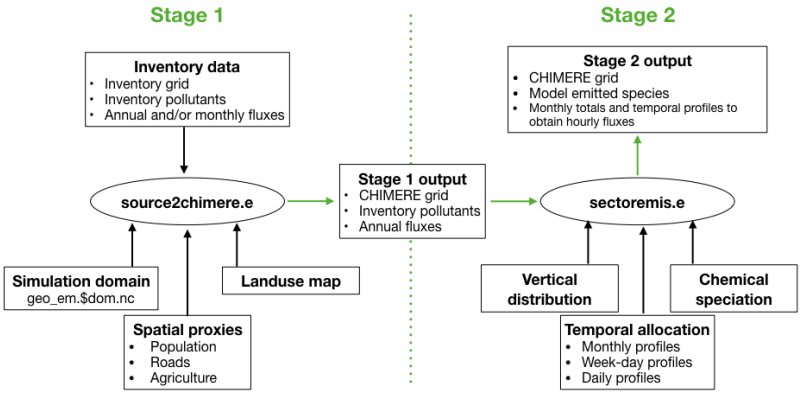


Figure 9.1: *Top-down processing of anthropogenic emissions with the emiSURF program.*

9.2.1 Download and install emiSURF

The emiSURF program is available at the CHIMERE download page. To compile the software you need to type the following command:

```
./build-emisurf.sh --par emisurf.par
```

In the emisurf.par namelist you need to provide the path to your chimere directory (see Section 9.2.2)

9.2.2 The emiSURF model namelist: emisurf.par

All user-defined parameters to run the emiSURF model are written in the emisurf.par file.

```
   [$parameter_name] This parameter description  : value
```

Definition of the simulation domain The domain of the CHIMERE simulation (i.e. longitudes and latitudes of the center of the grid-cells in degrees) may be provided in either netCDF format through the WPS geog file or in as an ascii file (the old CHIMERE format of the fichier COORD_${dom}. This option is managed with the following parameters.

```
[$dom]                                  : Euro
[$domainsdir]                           : ${chimeredir}/domains
# link to geog file used only if use_geog=True
[$geogfile]                             : /data/PLT8/pub/cchambon/geo_em.d01.nc
#[$geogfile]                             :
[$use_geog]                             : True
```

The name of the simulation domain needs to be given with the *[$dom]* flag. If the user has already created the geog file of the CHIMERE simulation domain then the *[$use_geog]* flag needs to be set to "True" and the path to the geog file must be provided. If the user wants to use an ascii file with the coordinates of the grid-cell centers the *[$use_geog]* is "False" and the COORD_${dom} file has to be found in the $domainsdir/HCOORD directory.

Input databases There are several mandatory or not databases that the emiSURF program will look for in your file-system:

- The pollutants inventory
- Land-use fractions
- Other spatial proxies
- Point sources

This information is communicated through the following namelist variables:

```
# possible sources: EMEP01x01, CAMS, HTAP01, RCP85
[$emisdatabase]                          : HTAP01
# HTAP/CAMS raw database informations
[$htap_resol]                            : 0.1
[$cams_resol]                            : 0.1
[$htapdirannual]                         : ${bigfiles}/HTAP_01_${scenlist}/HTAP_annual_Tons_${scenlist}
[$htapdirmonth]                          : ${bigfiles}/HTAP_01_${scenlist}/HTAP_monthly_Tons_${scenlist}
[$camsdir]                               : ${bigfiles}/CAMS_GLOB_ANT
[$version_cams]                          : 4.2
# RCP raw database informations
[$rcpdatabase]                           : ${bigfiles}/EmisRCP85
[$popfile]                               : ${bigfiles}/Population/pop_country_europe.nc
# Road database dir
[$roaddir]                               : ${bigfiles}/Proxies/ProxyRoad/
# Agriculture proxy
[$agridir]                               : ${bigfiles}/Proxies/ProxyAgri/
# LSP file
[$lspfile]                               : ${pdir}/annual-LSP/EPER_2004.txt
```

Emission sectors With this version of the emiSURF program, the user may retrieve the information on the emission fluxes per activity sector in the output files. This option is managed with in the following section of the emisurf.par namelist.

```
# Emissions by SNAP sectors or not
#
# Format for the output: vanthro=1 for chimere<=2017
#                        vanthro=2 with 'nsectors' dimension in the output files
[$vanthro]                               : 1

# noutsnap: # of activity sectors in output (the others are added in one fake sector)
# 0: no activity sector, only one emission flux
# 1 (or 2, 3, 4): one (or 2, 3,4) activity sector(s), all the rest in another one noted "0"
[$noutsnap]  : 0

# List of output snap sectors (must be consistent with ${noutsnap} above)
# Possible values are:
[$list_outsnap]                        : '2 7'
```

- *[$vanthro]*: the old format is used with no dimension in the netCDF output dedicated to the emission sectors
- *[$noutsnap]*: The number of separate sectors for whom emissions will be written in the output file
- *[$list_outsnap]*: The list of these sectors (the correspondance to the SNAP sectors)

Anthropogenic emission inventories, provide emission fluxes for several activity sectors. Deferent classifications are used depending the inventory. The emiSURF program will make the correspondance between the input inventory's classification and the 11 SNAP sectors (Table 9.3).

9.2.3 Emission inventories

The different inventories that may be processed with the emiSURF program are shown in Table 9.5.
Depending on the emission inventory a different preprocessing is applied on the input data.

SNAP	Description
S1	Industry combustion
S2	Non-industrial combustion
S3	Combustion in manufacturing industry
S4	Production processes
S5	Extraction/Distribution
S6	Solvents
S7	Road transport
S8	Other transport
S9	Waste
S10	Agriculture
S11	Other sources

Table 9.3: *Description of the 11 SNAP sectors.*

	EMEP	**HTAP**	**CAMS**	**RCP8.5**
Grid Resolution	$0.1^o \times 0.1^o$	$0.1^o \times 0.1^o$	$0.1^o \times 0.1^o$	$0.5^o \times 0.5^o$
Totals	Annual	Annual/Monthly	Monthly	Monthly
Available years	1990-2017	2010	>= 2000	2000,2005,2010 2020, 2030, 2040 2050, 2060, 2070 2080, 2090, 2100
Reported pollutants	CO, NH_3, NMVOC NO_X,SO_X, $PM_{2.5}$ PM_{10},PM_{Coarse}	CO, NH_3, NMVOC NO_X,SO_2, $PM_{2.5}$ PM_{10},BC, OC	CO, NH_3, NMVOC NO_X,SO_2,CO_2 BC, OC	CO, NH_3, NMVOC NO_X,SO_2,CH_4 BC, OC
Spatial coverage	Europe	Global	Global	Global
Activity sectors	GNFR14	Industry, Energy Residential, Transport, Air, Ship Agriculture	Industry, Energy Residential, Road transport Fugitives, Solvents, Off road Transport, Ships, Waste Agriculture, Agriculture livestock	Industry, Energy Residential, Solvents Land Transport, Waste Agriculture Agricultural Waste
Download	https://www.ceip.at/	https://edgar.jrc.ec.europa.eu	https://eccad3.sedoo.fr/	http://www.iiasa.ac.at

Table 9.4: *The different anthropogenic emission inventories processed by the emiSURF model. Level 1 GNFR14 activity sectors are listed in the Table 9.5*

9.2.3.1 Pre-processing the HTAP, CAMS and RCP inventories

In this case, monthly and if necessary annual data are downloaded and stored in a directory indicated in the emisurf.par namelist.
For HTAP it is necessary to download **all data**, monthly and annual, for the year 2010. The user can download the data on this website:
http://edgar.jrc.ec.europa.eu/htap_v2/index.php
The list of available activity sectors is presented in Table 9.5. At the end, you must have a directory called HTAP_01_2010/ including two sub-directories: HTAP_annual_Tons_2010 and HTAP_monthly_Tons_2010. In each sub-directory, you must have the data in ascii format as: edgar_HTAP_BC_emi_ENERGY_2010_1.txt for all species (from 1 to 12) and all SNAP activity sectors.
The preprocessing in these cases is automatic. The only action to take before running the emiSURF program is to correctly fill in the emisurf.par namelist (see Section 9.2.2).

9.2.3.2 Pre-processing the EMEP inventrory

The user needs to go to the annual-EMEP01x01/preproc directory and follow instructions provided in the README files. Scripts to download and preprocess the EMEP annual database for use in the emiSURF program are proposed. The EMEP gridded-data as downloaded from the EMEP website are stored in the *[$EMEPDATABASE]* directory. No need to unzip

the archives, the 1_preproc.sh script will do this automatically. You need to specify the *[$EMEPDATABASE]* directory as an argument to 1_preproc.sh script. You will also have to edit the 1_preproc.sh script to point to your emiSURF simulation *[year]*, the directory where your CHIMERE code is found (i.e. *[chimeredir]*) and the name of your mychimere file (*[mychimere]*).
The speciation profiles between inventory pollutants and "model-species" in emiSURF are defined for the eleven SNAP activity sectors. However, the EMEP inventory data are provided in the GNFR14 activity sectors classification. For this reason, in the case of EMEP processing, the emiSURF model creates and applies a GNFR-to-SNAP mapping table on the inventory data. A first order split of emissions defined in GNFR14 classification to the SNAP sectors is shown in Table 9.5. The emiSURF model provides tools to create a more refined mapping between the two systems based on a Level 2 GNFR14 classification and the year of simulation.

GNFR14	SNAP
A_PublicPower	Industry Combustion
B_Industry	Manufacturing Combustion
C_OtherStationaryComb	Non-industrial Combustion
D_Fugitive	Extraction/Distribution
E_Solvents	Solvents
F_RoadTransport	Road Transport
G_Shipping	Other Transport
H_Aviation	Other Transport
I_Offroad	Road Transport
J_Waste	Waste
K_AgriLivestock	Agriculture
L_AgriOther	Agriculture
N_Natural	Other
O_AviCruise	Other Transport
P_IntShipping	Other Transport

Table 9.5: *Matrix to map Level 1 GNFR14 activity sectors to the SNAP sectors*

The GNFR-to-SNAP matrix is generated in the annual-EMEP01x01/preproc/misc directory by the 1_make_table.sh script. Ideally, this matrix should be updated for each reporting year. To generate it, the user needs to download the officially reported national totals (Level 2) for the year of interest from the EMEP site
https://www.ceip.at/ms/ceip_home1/ceip_home/webdab_emepdatabase/reported_emissiondata/ Emissions for each Level 2 GNFR14 class split in the eleven SNAP activity sectors. The header of the obtained GNFR-to-SNAP correspondence matrix MATRIX_GNFR14_SNAP_${YEAR}.txt reads as follows:

Pollutant	Country	SNAP1 %	SNAP2 %	SNAP3 %	SNAP3 %	SNAP5 %	SNAP5 %	SNAP6 %	SNAP7 %	... ...	SNAP11 %

9.2.4 From inventory pollutants to 'model species'

To prepare emission fluxes for the species corresponding to the CHIMERE simulation chemical mechanisms the user has to properly fill in the following emisurf.par parameters (see Section 9.1).

```
[$meca}]   gas-phase chemical mecanism (melchior/saprc)       : melchior
[$carb]  SOA formation from POA (0-3)  : 1
[$poatype]     SOA chemistry (nop/vbs/h2o)     : nop
```

meca: The same gas-phase chemical mechanism as in the CHIMERE simulation. Depending on the gas-phase chemical mechanism (MELCHIOR or SAPRC) the NMVOC emission splits into a different set of volatile organic compounds ('model species'). For both mechanisms NMVOC emissions first split into 221 "real" compounds of the british PORG speciation depending on the SNAP sector [Passant, 2002]. This speciation is described in *Appendix F, p.283*. Then, a two-step procedure aggregates "real" VOC compounds to a smaller set of VOC classes which correspond directly

to the specific compounds of the MELCHIOR or SAPRC chemical mechanism, following the Middleton methodology [Middleton et al., 1990]. Following this procedure, "real" compounds are lumped together into "model" compounds using a reactivity weighting factor which depends on the relative OH radical rate constants of the "real" and "model" compounds following the equations 9.1 and 9.2. The user should bear in mind that the same reactivity weighting factors are used on emissions for all countries, which represents a significant source of uncertainty in the spatial distribution of VOC emissions.

$$\alpha_i = \frac{1 - exp(-k_i \times INTOH)}{1 - exp(-k_{Model} \times INTOH)} \tag{9.1}$$

$$E_{Model} = \sum_i \alpha_i E_i \tag{9.2}$$

where k_i is the OH rate constant of the real compound and k_{model} the constant of the model compound. These constants are calculate at 300K with an air density of 2.5x10^9 modelc.cm^{-3}. INTOH is the integral of OH concentrations during 2 days (100 ppt.mn^{-1}).
poatype: This flag indicates whether and how primary emitted organic aerosol splits into volatility bins. Its value depends on the *[$soatyp]* of the chimere.par namelist.

- *[$poatype]*=nop: no volatility bins are considered (corresponds to *[$soatyp]*<4). Note that in this case the *[$carb]* flag needs to be set to either 0 or 1.
- *[$poatype]*=h2o: primary organic aerosol splits into low, medium and high volatility bins. This corresponds to *[$soatyp]*=4/5.
- *[$poatype]*=vbs: primary emitted organic aerosol splits into nine volatility bins (*[$soatyp]*=6/7).

carb: The function of this flag depends on the *[$poatype]* flag

- *[$carb=0]*: the non-carbonaceous species PPM are mixed with OCAR and BCAR resulting in the species called TPPM.
- *[$carb=1]*: the anthropogenic species are separated into PPM, OCAR, and BCAR in CHIMERE.
- *[$carb=2]*: the anthropogenic species are separated into PPM, BCAR and POAx (x representing a bin of volatity) in CHIMERE. For *[$poatype]*=vbs, only the functionalization process is taken into account (see [Zhang et al., 2013]).
- *[$carb=3]*: the anthropogenic species are separated into PPM, BCAR and POAx (x representing a bin of volatity) in CHIMERE. For *[$poatype]*=vbs, functionalization, fragmentation and non volatil bin are taken into account. (see [Cholakian et al., 2018]). For *[$poatype]*=h2o, an additional split of POA into low, medium and high volatility bins is done for emissions from wood-burning within the residential sector.

NO_x: For all activity sectors other than road transport, NO_x emissions split into 95%NO, 4.5%NO_2 and 0.5%HONO (the split is 78.5%, 20% and 1.5% respectively for road transport). Depending on the country, NO_x emissions in the EMEP inventory may include soil NO biogenic emissions in the "Agriculture" sector. The user should be carful not to doublecount NO soil emissions when running the biogenic emissions pre-processor embedded in the CHIMERE model.
SO_x: For all activity sectors, 99% of SO_x is assigned to SO_2 and 1% to sulphuric acid (H_2SO_4) in the particulate phase (fine mode).

9.2.5 Horizontal regridding

The source2chimere.e program spatially allocates emission fluxes from the inventory grid to the CHIMERE simulation grid. This allocation is done based on spatial proxies. The common proxy for all inventory databases is a land-use map. The choice of the landuse dataset is in the emisurf.sh script with:

```
# Reggriding landuse : possible Land Use sources: globcover or usgs
lu_source="usgs"
```

- *[lu_source="globcover"]*: GLOBCOVER at 10 arc second (≈300m) resolution developed under the European Space Agency initiative (http://due.esrin.esa.int/page_globcover.php).
- *[lu_source="usgs"]*: USGS data at 1km resolution, with 24 landuse categories, as used in the WRF meteorological model.

In earlier versions of the emisurf preprocessor, emissions for all the activity sectors were performed based on a classification among four landuse types: urban, crop, water or forest. On the new version, several other proxies are proposed

for the spatial allocation of the emissions : population density, road map, agriculture and wood-burning aerosol emissions from the residential sector. However, only the landuse map is a mandatory input for source2chimere. The other proxies are used as described below, and only for SNAP sectors 2 (residential emissions) and 7 (road traffic). Donwscaling using population density as a proxy is activated if pop=1 in emisurf.sh. Downscaling using road map is activated if road=1.

- SNAP sector 2 : if pop=1, emissions from SNAP sector 2 (mostly residential fuel burning) are downscaled using the population density as a proxy for gas species, and a different factor for particulate species, taking into account the fact that wood burning for residential heating occurs predominantly in sparsely populated areas rather than in dense urban areas. If pop=0, then emissions from SNAP sector 2 are downscaled according to the landuse data.
- SNAP sector 7 : if road=1, then emissions from SNAP sector 7 (traffic emissions) are downscaled using the road map data as a proxy. If road=0 but pop=1, then these emissions are dowscaled according to the population density. Otherwise, if road=0 and pop=0, then the emissions from road traffic are downscaled according to the landuse.

9.2.6 Vertical distribution

The vertical distribution is performed to convert the 2D raw fluxes into 3D fluxes. This is achieved by using a typical vertical profile, originating from EMEP data, but modified following [Mailler et al., 2013] and [Terrenoire et al., 2015]. These data are in the file inputdata/HEIGHT_EMEP_update with the informations presented in Table 9.6 .

Injection height (m)	20.	92.	184.	324.	522.	781.	1106.
SNAP 1	0.	0.	0.25	51.	45.3	3.25	0.2
SNAP 2	100.	0.	0.	0.	0.	0.	0.
SNAP 3	6.	16.	75.	3.	0.	0.	0.
SNAP 4	5.	15.	70.	10.	0.	0.	0.
SNAP 5	2.	8.	60.	30.	0.	0.	0.
SNAP 6	100.	0.	0.	0.	0.	0.	0.
SNAP 7	100.	0.	0.	0.	0.	0.	0.
SNAP 8	100.	0.	0.	0.	0.	0.	0.
SNAP 9	0.	0.	41.	57.	2.	0.	0.
SNAP 10	100.	0.	0.	0.	0.	0.	0.
SNAP 11	100.	0.	0.	0.	0.	0.	0.

Table 9.6: *Vertical emissions profiles (%) for each SNAP category, after [Terrenoire et al., 2015].*

9.2.7 Time allocation

The goal of this calculation is to refine the input data and to convert monthly/annual data to hourly data. These variations follow the guidelines provided by the Institute for Energy Economics and Rational Use of Energy, University of Stuttgart; [Ebel et al., 1994].

The seasonal factor: First, for the annual data, a seasonal factor is applied. All required file are in the inputdata directory as inputdata/SEASONAL-FACS_NO_x for NO_x by example. There is one file per "raw" species. In each file, informations are given as:

- First column: the country following the EMEP country code
- Second column: the SNAP code
- Columns from 3 to 14: the twelve months of the year

The weekly factor: The weekly factor has a format close to the seasonal one, with:

- First column: the country following the EMEP country code
- Second column: the SNAP code
- Columns from 3 to 10: the seven days of the week. Days are ordered as Monday (1), ..., Sunday (7).

The hourly factor: This factor is in an unique datafile called inputdata/HOURLY-FACS. This file is read in the sectoremis.F90 program as:

```
nct,itype,ns,(hmod(ihour,ns,nct,itype),ihour=1,nhourpday)
```

with:

- nct: country type number (from 0 to 17)
- itype: the day type (7 days, from Monday to Sunday)
- ns: activity sector number (11 sectors)
- hmod: the hourly factor
- nhourpday: 24 hours from 00:00 LT to 23:00 LT (00:00 is midnight local time).

By default, the values contained in the block corresponding to nct=0 are applied. It means that hourly factors are the same for the whole domain. A refinement may be done using the 17 types of countries classified in [Menut et al., 2012]. These factors are using the EMEP country codes and are thus valid over western Europe only.

Universal Time The time-shift-geia.e program assigns a time-zone to each grid-cell of the simulation domain based on a 0.1^o x 0.1^o time-zone map. Based on this information, the UTC offset and summer-winter time-shifts are estimated for each cell of the simulation domain (Figure 9.2). Information on the summer-winter time-shifts is stored in the emiSURF output files for further application during the CHIMERE simulation (this feature is not yet implemented in the CHIMERE model). Different options are proposed in the emisurf.par file:

```
sumwintime=0 no  (prefer this option at the southern hemisphere or climatic cases to minimize errors)
sumwintime=1 yes (adopt a 1 hour win/sum time from April to October files everywhere)
```

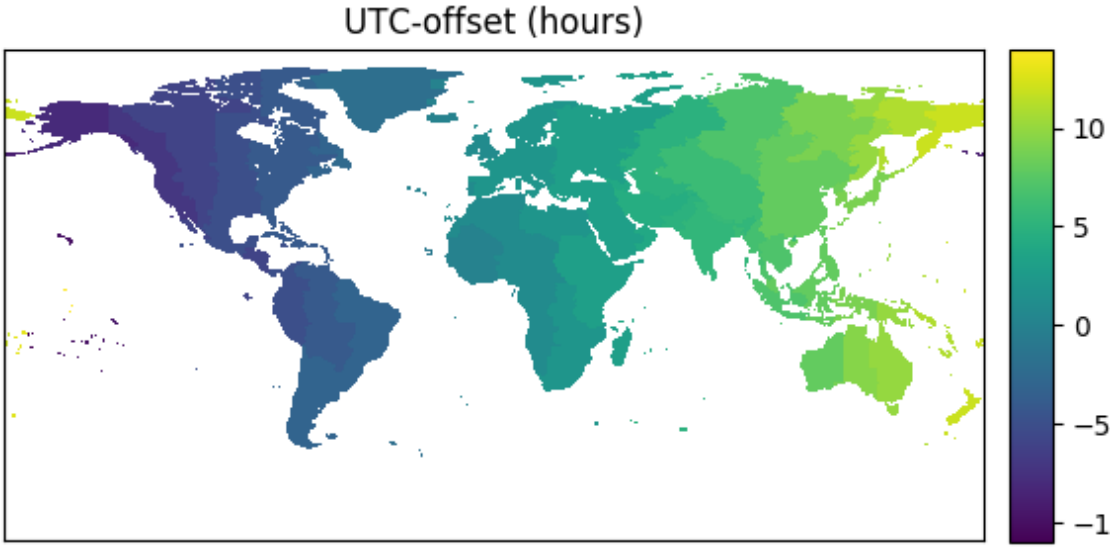


Figure 9.2: *Map of the UTC-offset on the different time-zones.*

9.2.8 User's precompiled emissions

The user can produce his/her own AEMISSIONS.${sim}.nc file. In this case, **this file is an input file**.
There is only one file per period to simulate. The time in this file corresponds exactly to the days and hours to simulate. This file must contain hourly emission fluxes on the CHIMERE 3-D grid for all gases and aerosol species involved in the chemical mechanism of the CHIMERE simulation (MELCHIOR or SAPRC) in molecules·cm^{-2}·s^{-1}. Aerosol fluxes need also to be converted to virtual "molecule-like" units in this anthropogenic emission input file considering a fictitious molar mass equal to 100g·mol^{-1}. The conversion coefficient from mass unit to CHIMERE model unit is 6.022×10^{23} mol^{-1}/(100g·mol^{-1}).

In this case the user should set the *[imakeaemis]* flag at 0 or 2, so that prep_aemis.F90 module subroutines are not invoked, and provide the appropriate *[aemisdir]* and *[fnemisa]* paths pointing to the AEMISSIONS.${sim}.nc file in the chimere.par file.

```
[fnemisa] Anthropic emiss. File (from Stage 2)  : ${datadir}/AEMISSIONS.${sim}.nc
```

The format of the file to provide is as follows:

```
netcdf AEMISSIONS {
dimensions:
      Time = UNLIMITED ; // (24 currently)
      west_east = 79 ;
      south_north = 47 ;
      bottom_top = 8 ;
      SpStrLen = 23 ;
      DateStrLen = 19 ;
      Species = 24 ;

variables:
      char Times(Time, DateStrLen) ;
      char species(Species, SpStrLen) ;
      float lon(south_north, west_east) ;
            lon:units = "degrees_east" ;
            lon:long_name = "Longitude" ;
      float lat(south_north, west_east) ;
            lat:units = "degrees_north" ;
            lat:long_name = "Latitude" ;
      float NO(Time, bottom_top, south_north, west_east) ;
            NO:units = "molecule/cm2/s" ;
            NO:long_name = "NO Emission" ;
      float NO2(Time, bottom_top, south_north, west_east) -
            NO2:units = "molecule/cm2/s" ;
            NO2:long_name = "NO2 Emission" ;

            ....

      float PPM_big(Time, bottom_top, south_north, west_east) ;
            PPM_big:units = "molecule/cm2/s" ;
            PPM_big:long_name = "PPM_big Emission" ;
      float PPM_coa(Time, bottom_top, south_north, west_east) ;
            PPM_coa:units = "molecule/cm2/s" ;
            PPM_coa:long_name = "PPM_coa Emission" ;

            ....
```

An example of complete format for the MELCHIOR2 chemical mechanism is displayed in *Appendix E.2.1, p.253*

9.2.9 Hands-on : a test case with CAMS emissions

This section is a step-by step guide on how to set-up and run the emiSURF model. We chose to use the CAMS anthropogenic emissions inventory for this test-case because of its global coverage. The domain, year and month of the simulation are chosen to correspond to the CHIMERE test-case proposed in Section 2.5.
All case dependent input and desired output are available on the TestCaseEMISURF2020.tar.gz. Once decompressed this archive contains:

- emisur.par.test the configuration file for the emiSURF model simulations. This file has to be properly filled before each model simulation to correspond to the domain, year and month of the CHIMERE simulation (here IDF5, January 2017). You need to carefully fill in the paths to point to the files and directories where necessary input files will be looked for by the code and to an output directory where the user has write access on the computer.
- INPUTS/ This directory has two sub-directories. In the **CAMS** directory are stored the CAMS inventory emission fluxes for January of year 2017 and for all the inventory species. In the **geog** directory you will fine the geo_em.d01.nc which provides the definition of the geographical coordinates (lon,lat in degrees) of the centers of the grid-cells of the simulation domain (in this cas the IDF5 domain over the Ile-de-France region with 5km spacing between each model grid-cell in both west-east and south-north dimensions).

- DESIRED_OUTPUTS/ The subdirectory EMI_2017 contains the output of the emiSURF simulation. You may want to compare the obtained results with the content of this directory to be sure that your run was successful.

To run the emiSURF test-case you need to edit the emisur.par.test configuration file accordingly. As you may have already done if you run the CHIMERE test-case proposed in Section 2.5, you have to search the file for the __FILL__ signs and replace them with the appropriate values for your simulation. The lines that need to be changed are shown here:

```
[$chimeredir]                    : __FILL_
[$outdir]                        : __FILL__
[$geogfile]                      : __FILL__
[$camsdir]                       : __FILL__
[$bigfiles]                      : __FILL__
```

For each line you need to provide the full path indicating the location of the required files and directories in you file system (typically, where you stored them when decompressing the TestCaseEMISURF2020.tar.gz archive.

- chimeredir : is the path towards you CHIMERE directory. The emiSURF program will look in there for the mychimere.par file indicating library paths in your system. This is necessary for the emiSURF model compilation.
- outdir : Here you will indicate the path where you want to save your output files.
- geogfile: The full path to the geog file (here provided in the TestCaseEMISURF2020/INPUTS/geog directory). This file provides the coordinates of the simulation mesh.
- camsdir: The path towards the global anthropogenic emission database. Data for the simulaiton period corresponding to this case-study have been extracted in the TestCaseEMISURF2020/INPUTS/CAMS directory.
- bigfiles: The path towards the BIGFILES directory (same as for the CHIMERE simulations). Here emiSURF will look for spatial proxies to refine the spatial allocation of 0.1 x 0.1^o emissions over the simulation mesh.

At the end of the emiSURF simulation you will find netcdf files for each species in the outdir directory. You may compare these results with the file provided in this directory: TestCaseEMISURF2020/DESIRED_OUTPUT/EMI_2017 to be sure the model run as expected.

9.3 A subgrid scale model to account for heterogeneous surface emissions

Regional scale chemistry transport models such as CHIMERE do not resolve physical and chemical processes occurring at spatial scales bellow $1km^2$. If such effects are to be accounted for, they must be parametrized. Heterogeneous surface emissions at the unresolved subgrid scale (i.e. at spatial scales smaller than the model resolution) have been shown to have a large impact on grid-averaged pollutant concentrations [Galmarini et al., 2008, Valari and Menut, 2010, Korsakissok and Mallet, 2010, Cassiani et al., 2010]. Especially over urban areas with large population density, very contrasted emission sources, such as roads and buildings co-exist within small distance. Each one of these sources, emit different substances with different life-times in the atmosphere and at different rates. One can imagine that the emitted species react fast with each other close to the source before getting perfectly mixed within the volume of the grid-cell. Averaging spatially the products of these local reactions over the grid cell volumes will not result in the same mean value as it would have under the assumption of a perfect mixing before chemistry took place (i.e by averaging emissions within the grid cells before simulation). To illustrate this, Figure 9.3 shows emission fluxes over the road network of central Paris, France (colored lines), as well as grid-averaged emission fluxes over the corresponding 2km x 2km resolution CHIMERE grid cells (colored squares). We see in this figure, that the variability in traffic emission fluxes over the road network is not represented in the CHIMERE simulation, where emissions from all road segments within each model cell are spatially averaged before the simulation.
To account for this subgrid scale effect CHIMERE provides the option of an alternative calculation. The CHIMERE model grid cell area (A) is divided into several subgrid areas A_i, each representing a different emission source. We assume that over each subgrid area (A_i) a unique emission sector (E_i) dominates (Figure 9.4). During the CHIMERE simulation, at each model time-step, over grid-cells where the subgrid scale option is activated, the calculation of chemistry splits into the different subgrid volumes. Chemistry acts separately within each compartment changing pollutant concentrations before they are mixed to the entire volume of the grid-cell. Once local concentrations are estimated the different components are averaged spatially to proceed to the next model time-step.

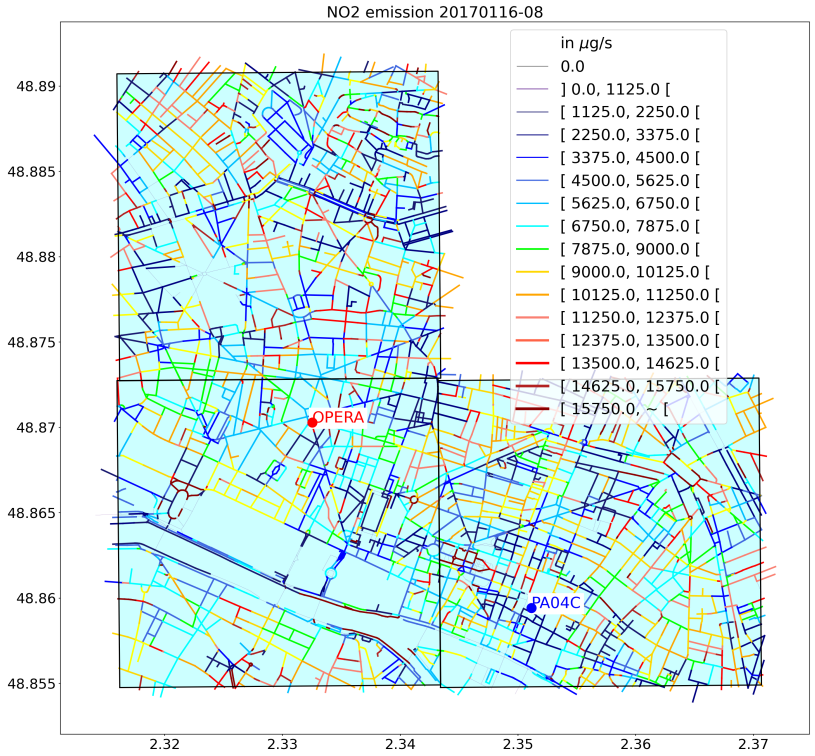


Figure 9.3: *Emission fluxes over the road network within three CHIMERE model grid-cells at 2km x 2km horizontal resolution. Grid averaged emission fluxes at each model grid-cell are also represented with the same color code.*

The subgrid-scale model produces surface concentrations for all model species at each model time-step and for each anthropogenic emission sector accounted for in the simulation. Figure 9.5 compares model results with the usual CHIMERE calculation and with surface measurements of the Paris air-quality network AIRPARIF. We chose two different kind of monitor sites, one close to the road network (OPERA on the top panel) and one located at some distance from the road network, designed to measure urban background pollutant levels (PA04C bottom panel). The location of these monitor sites is shown in Figure 9.3. We see in the time-series of Figure 9.5 that the 'on-road' component of the subgrid-scale simulation models significantly higher NO_2 levels, especially during rash hour. In several cases this excess in modeled NO_2 reaches measured levels at the traffic monitor site OPERA that the grid-averaged CHIMERE concentration would have missed. On the contrary, as expected, over the urban background monitor site PA04C the grid-averaged concentration is found to be closer to measured levels.

9.3.1 Technical note on the activation of the subgrid-scale model

As discussed in Section 3.11, to activate the subgrid-scale model you need to set the *[sgmodel]* to 1.

```
# CHIMERE Subgrid Processes
  [$sgmodel]      Subgrid scale model for anthropogenic emi   : 1
```

At this stage, the subgrid-model outputs concentrations for five pollutants (NO, NO_2, O_3, PM10 and PM2.5). If the *[sgmodel]* flag is activated, you will find in the main CHIMERE output file out.${sim}.nc concentrations estimated at subgrid-scale for the different emission sectors. The five subgrid-scale pollutant concentrations in the output file are named after the pollutant name followed by the postfix '_sg' as shown bellow. The structure of these variables is different from the general structure of CHIMERE output concentration fields. Since the sugrid-scale calculation is not activated on all grid-cells of the CHIMERE mesh, output concentrations for each pollutant is not a 2D field over each point of the

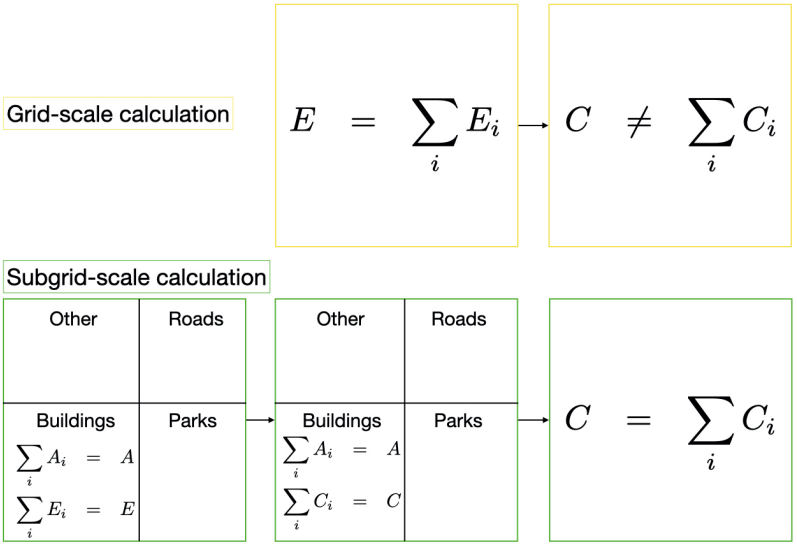


Figure 9.4: *Flow chart showing the concept of (top) usual simulation of pollutant concentrations where emissions are averaged spatially before the simulation and (bottom) the subgrid-scale calculation, where chemistry acts locally before pollutants get mixed in model grid-cells.*

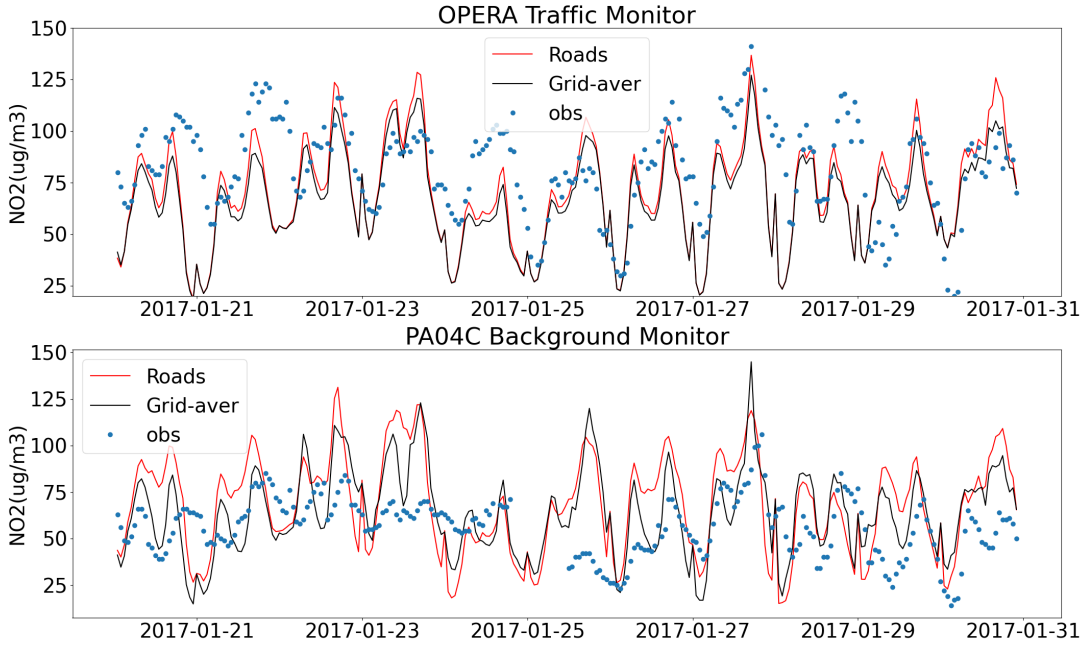


Figure 9.5: *Ten-day hourly time series of modelled and measured NO_2 ($\mu g/m^3$) surface concentrations. The top panel compares simulations with observations over a traffic monitor site while the bottom panel over a background monitor site of the AIRPARIF air-quality network. Concentrations are simulated with the usual CHIMERE calculation where emissions are homogeneous over the grid-cell volumes (black lines) and with the 'on-road' component of the subgrid-scale model.*

horizontal mesh in the west-east and south-north dimensions but rather a 1D vector over the selected grid-cells. Therefore, there are two new dimensions in the output file:

- nsectors: The length of this dimension is the number of activity sectors accounted for in the subgrid scale calculation plus one for the grid-avraged value. So let us say you want to simulate concentrations over the road-network, over residential zones and over all other type of surfaces (BE CAREFUL! the sum of the area fractions must be equal to 1), then Sectors=3+1=4 as in the example shown below
- num_sgcells: The length of this dimension is equal to the number of cells where the subgrid-scale calculattion is activated (20 in the example bellow)

BE CAREFUL IN THE ORDER OF SECTORS: The order of sectors ('nsectors' dimension) in the output subgrid-scale concentration arrays is the same you chose in the compilation of the AEMISSIONS.nc file and the ascii subgrid_LU_$dom.dat file (see following paragraphs). The grid-averaged value corresponds to the last index of the 'nsectors' dimension.
The correspondance between the index in the 'num_sgcells' dimension of species concentrations in the output file and the 2D CHIMERE horizontal mesh is done with two new variables:

- sgcells_zonal_id: This vector of 'num_sgcells' elements maps subgrid-scale model cells to the zonal index (west-east dimension) of the CHIMERE horizontal grid
- sgcells_merid_id: This vector of 'num_sgcells' elements maps subgrid-scale model cells to the meridional index (south-north dimension) of the CHIMERE horizontal grid

```
netcdf out.2017012100_24_nest-IDF2 {
dimensions:
        Time = UNLIMITED ; // (25 currently)
        DateStrLen = 19 ;
        west_east = 80 ;
        south_north = 64 ;
        bottom_top = 15 ;
        vcoord_dim = 15 ;
        wavelength_dim = 5 ;
        nbins = 10 ;
        ninterv = 11 ;
        num_sgcells = 20 ;
        nsectors = 3 ;
variables:
        int sgcells_zonal_id(num_sgcells) ;
                sgcells_zonal_id:units = "we_id" ;
                sgcells_zonal_id:long_name = "we_id_count_from_1" ;
        int sgcells_merid_id(num_sgcells) ;
                sgcells_merid_id:units = "sn_id" ;
                sgcells_merid_id:long_name = "sn_id_count_from_1" ;
        ....
        float NO_sg(Time, nsectors, bottom_top, num_sgcells) ;
                NO_sg:units = "ppb vol" ;
                NO_sg:long_name = "NO_sgConcentration" ;
        float NO2_sg(Time, nsectors, bottom_top, num_sgcells) ;
                NO2_sg:units = "ppb vol" ;
                NO2_sg:long_name = "NO2_sgConcentration" ;
        float O3_sg(Time, nsectors, bottom_top, num_sgcells) ;
                O3_sg:units = "ppb vol" ;
                O3_sg:long_name = "O3_sgConcentration" ;
        float PM25_sg(Time, nsectors, bottom_top, num_sgcells) ;
                PM25_sg:units = "ug/m3" ;
                PM25_sg:long_name = "PM25_sgConcentration" ;
        float PM10_sg(Time, nsectors, bottom_top, num_sgcells) ;
                PM10_sg:units = "ug/m3" ;
                PM10_sg:long_name = "PM10_sgConcentration" ;
        ....
```

At the current state of development of the subgrid-scale model, the simple activation of the *[sgmodel]* flag is not enough to produce subgrid-scale concentrations. The user needs to provide two more pieces of information compiled by his/her own means.

- You need to compile the AEMISSIONS.${sim}.nc file as presented in the Section 9.2.8 but with modifications to include emissions by activity sectors
- You need to provide an ascii file with the percentages of area occupation of each domains' grid-cell area by each of the corresponding source. This file includes a flag to indicate the grid-cells where the subgrid scale calculation will be activated.

9.3.1.0.1 Emissions per sector

The information on emission fluxes per activity sector is provided in the AEMISSIONS.${sim}.nc file, where one nw dimension 'Sectors' is added. The length of this dimension is the number of activity sectors accounted for in the subgrid scale calculation plus one for the grid-avraged value. So let us say you want to simulate concentrations over the road-network, over residential zones and over all other type of surfaces (BE CAREFUL! the sum of the area fractions must be equal to 1), then Sectors=3+1=4 as in the example shown below. You also need to know that in the arrays of species' emission fluxes the first index over the Sectors dimensions is reserved for the grid-averaged flux and the following Sectors-1 for the sectoral emissions. **BE CAREFUL: This order is implicitly inherited to the ascii file providing area fractions of the emitting sources (cf. next paragraph)**

```
netcdf AEMISSIONS {
dimensions:
        Time = UNLIMITED ; // (25 currently)
        Sectors = 4 ;
        bottom_top = 15 ;
        south_north = 64 ;
        west_east = 80 ;
        DateStrLen = 19 ;
        Species = 36 ;
        SpStrLen = 23 ;
variables:
        float APINEN(Time, Sectors, bottom_top, south_north, west_east) ;
                APINEN:units = "molecule/cm2/s" ;
                APINEN:long_name = "APINEN                Emission" ;
        float BCAR_coa(Time, Sectors, bottom_top, south_north, west_east) ;
        ....
```

9.3.1.0.2 Grid-cell area occupation by anthropogenic emission sources

The second piece of information you need to provide in order to run a subgrid scale simulation is the grid-cell area occupation fractions of each of the emitting sources. This is an ascii file that has to be placed in your domains/$dom directory and named subgrid_LU_$dom.dat. The file has one line for each model cell, starting from the lower-left corner of the mesh first spanning all zonal points from west to east before moving one row upwards in latitudes in the south-north dimension. The number of columns in the file is equal to the number of Sectors to simulate plus one for the mask whether to activate the subgrid scale calculation or not on the specific grid-cell. The first column is the mask 0 (not activated) or 1 (activated). The next columns are grid-cell area fractions for each source. **BE CAREFUL: This order must correspond to the order emission fluxes per sector are given in the netcdf file** AEMISSIONS.${sim}.nc (cf. previous paragraph).

```
....
0   0.109   0.891
0   0.107   0.893
1   0.115   0.885
1   0.12     0.88
1   0.104   0.896
1   0.13     0.87
1   0.101   0.899
0   0.016   0.984
....
```

10 Natural emissions

10.1 Biogenic emissions

The biogenic emissions fluxes are calculated if the flag iusebemis is one in the chimere.par file as:

```
[iusebemis]   Need biogenic emissions? (1/0)  : 1
```

The biogenic emissions are estimated using the MEGAN model after [Guenther et al., 2006]. They are calculated on-line in the model and are hourly written in the dedicated output file BEMISSIONS.nc by the src/write_bemis.F90 routine. Its format is described in Appendix E.2.2, p.254.

Model species name	Real species
NO	Nitrogen monoxide
C5H8	Isoprene
APINEN	α-pinene
BPINEN	β-pinene
LIMONE	Limonene
TERPEN	Terpene (Lumped class)
OCIMEN	Ocimene (Lumped class)
HUMULE	Humulene (Lumped class)

Table 10.1: *List of the biogenic emitted species with the MEGAN parameterization.*

In the complete version of the model, the model needs as input the species listed in Table 10.1. The MEGAN model (v. 2.04) exploits most recent measurements in a gridded and canopy scale approach, more appropriate for use in CTMs since it estimates the effective burden of gases that mix and react in the boundary layer. Estimates of biogenic VOCs from vegetation and NO emissions are calculated as :

$$ER_i = EF_i \times \gamma_i(T, PPFD, LAI) \times \rho_i \tag{10.1}$$

where ER_i ($\mu g.m^{-2}.h^{-1}$) is the emission rate of species i, EF_i ($\mu g.m^{-2}.h^{-1}$) is an emission factor at canopy standard conditions, γ_i (unitless) is an emission activity factor that accounts for deviations from canopy standard conditions, and ρ_i is a factor that accounts for production/loss within canopy. The canopy standard conditions relevant for this study are defined as: air temperature (T) of 303 K, photosynthetic photon flux density (PPFD) of 1500 $\mu mol.m^{-2}.s^{-1}$ at the top of the canopy, leaf area index (LAI) of 5 $m^2.m^{-2}$ and a canopy with 80% mature, 10% growing and 10% old foliage. The MEGAN model parameterizes the bulk effect of changing environmental conditions using three time-dependent input variables specified at top of the canopy: temperature (T), radiation (PPFD), and foliage density (LAI). The production/loss term within canopy is assumed to be unity ($\rho = 1$). The equation can then be expanded as:

$$ER_i = EF_i \times \gamma_{T,i} \times \gamma_{PPFD} \times \gamma_{LAI} \tag{10.2}$$

The MEGAN model provides input EF and LAI data over a global grid, herefater projected on the CHIMERE model grid. The current available choice for EFs is restricted to following species: isoprene, α-pinene, β-pinene, myrcene, sabinene, limonene, δ^3-carene, ocimene, and nitrogen oxide. EF's are static and refer to years 2000-2001. They are obtained summing up over several plant functional types (e.g. broadleaf and needle trees, shrubs, etc...). LAI database is given as a monthly mean product derived from MODIS observations, referred to base year 2000. Hourly emissions are calculated using 2-m temperature and short-wave radiation from a meteorological model output. The optimal choice for this work is the 150" resolution products proposed in the MEGAN inventory. The daily PPFD and temperature, which are representative values of the simulation, are calculated over the last 10 days (240 hours). In case of chained simulation, these mean averaged values are written in the ouput files and read for the next period to simulate.

The biogenic emissions are calculated following two steps:

1. **Interpolation of the raw global data on the CHIMERE grid:**
   - The original database is constituted of:
     (a) The emission factors data are netcdf files called <spec>all200021.nc.
     Those files must be stored in $bigfilesdir/MEGAN/NETCDF_150sec/EF/EFMT21b.
     (b) LAI data are 12 netcdf files of the format laivyyyymm.nc (one per month).
     The files must be stored in $bigfilesdir/MEGAN/NETCDF_150sec/LAI/LAI20.
   - The program remap_ef.F90 called by prep_megan.sh interpolates original global MEGAN emission factors and LAI data on the CHIMERE grid for the required domain. prep_megan.sh is called by make-chemgeom.sh, so these calculations can be done only once for a given domain.
   - The output file is called EFMAP_LAI_<domain>.nc and is put to the domains directory. It contains emission factors for each MEGAN species, as well as 12 LAIs for each month as 2D variables on the Chimere grid for the given domain.
2. **On-line hourly calculation of biogenic fluxes:**
   Module diagbio_megan in src/diagbio_megan.F90 contains subroutine megan_bioemis called from src/-worker_bio.F90 that calculates emission rates for the nine afore-mentioned MEGAN species (redistributed into the six CHIMERE species) using parameterizations of the original MEGAN code. The module also contains the data used for the calculations.

10.2 Sea salt emissions

Sea salt emission parameterizations compute the flux of sea salt particle number in particles m^{-2} s^{-1} μm^{-1} as a function of r the particle radius and of U_{10} the 10m-wind speed in m s^{-1}.
The sea salt emissions fluxes are calculated if the flag iusesalt is greater than zero in the chimere.par file as:

```
[iusesalt]          Sea salts emissions? (0/1/2/3/4)      : 1
```

Several schemes are implemented and the flag possible values are:

- iusesalt=0: no sea salt emissions
- iusesalt=1: [Monahan, 1986]
- iusesalt=2: [Monahan, 1986] completed by the parameterization of [Martensson et al., 2003] for the size distribution
- iusesalt=3: [Martensson et al., 2003]
- iusesalt=4: [Grythe et al., 2014]

The simple scheme of [Monahan, 1986] is:

$$\frac{dF}{dr} = 1.373 U_{10}^{3.41} r^{-3} (1 + 0.057 r^{1.05}) 10^{1.19 e^{-B^2}} \tag{10.3}$$

$$B = \frac{0.38 - log(r)}{0.65} \tag{10.4}$$

The sea salt flux is calculated with several limitations: only for the percentage of the model cell covered by sea (and not inland water or sea ice), and only for 10-m wind speed less or equal than 30 m s^{-1} (i.e $U_{10} = min(30., U_{10})$).
The modelled species are NA_coa for the Sodium fraction, HCL_coa the Chloride fraction, H2SO4_coa the Sulfate fraction and WATER_coa the Water fraction. The sum of all these species, except water, is also calculated and written as an output as SALT_coa, the total dry sea salts.

10.3 DMS emissions

DMS emission parameterizations compute the flux of Dimethyl sulfide as a function of $\mathbf{U_{10}}$ the wind speed at 10 m in m s^{-1} and DMS sea urface concentration (nM) (5.2.4).
The DMS emissions fluxes are calculated if the flag *[iusedms]* in the chimere.par file as:

```
[iusedms]   Use DMS marine emissions? ([1-4]/0) : 0
```

DMS emissions can be calculated with several parameterizations according to the value of *[iusedms]*:

- iusedms=0: no DMS emissions
- iusedms=1: scheme of [Liss and Merlivat, 1986]
- iusedms=2: scheme of [Wanninkhof, 1992]
- iusedms=3: scheme of [Nightingale et al., 2000]
- iusedms=4: scheme of [Elliott, 2009]

Following [Liss and Merlivat, 1986], DMS emissions fluxes are driven by the concentration of DMS in the surface ocean, the wind speed at the sea surface and the temperature :

$$F = K_w \Delta C \tag{10.5}$$

K_w is the piston velocity and ΔC the difference of concentration between the air-sea interface
ΔC is expressed by:

$$\Delta C = \frac{C_a}{H} - C_w \tag{10.6}$$

Where C_a is the DMS concentration in the atmosphere, C_w is the DMS concentration in the surface sea ans H the Henryś constant.
In the case of DMS, the atmospheric concentration is negligible compared to what is available in the sea surface (and is at equilibrium with the underlying ocean), therefore:

$$\Delta C = -C_w \tag{10.7}$$

And thus:

$$F = -K_w C_w \tag{10.8}$$

In this equation, the piston velocity is calculated using wind speed and temperature, and the concentration of DMS in the sea can be approximated. In the parts that follow, different methods for the approximation of sea surface concentration of DMS and the calculation of piston velocity will be explained in more detail.
While this resolves the issue of concentration fields for the in-water DMS availability, the flux of DMS into the atmosphere needs to be calculated as well. As mentioned before, this flux depends on the wind speed and temperature. These two parameters are used in the calculation of the piston velocity and the Schmidt number respectively. The Schmidt number is calculated quite easily, [Kettle et al., 1999]:

$$Sc = 2674.0 - 147.12T + 3.726T^2 - 0.038T^3 \tag{10.9}$$

Where, Sc is the Schmidt number, T is the 2m temperature (°C).
However, the piston velocity equation differs depending on the intensity of the wind speed. Several parameterizations exist for the estimation of the piston velocity, four of them are presented in Table 10.2 .

10.4 Mineral dust emissions

The mineral dust emissions are calculated on-line and hourly fluxes are written as an output. Calculations may be done in every part of the world, the required input databases being global. The soil and surface properties are estimated using the USGS database. The use of this database is independent from the landuse used for other parts of the model.
The calculation of mineral dust emissions is done following several steps:

1. Activation of the options in the chimere.par parameter file,
2. Preparation of the required surface and soil informations from global databases and
3. Calculation of the mineral dust fluxes in CHIMERE.

Wind speed range (m.s^{-1})	[Liss and Merlivat, 1986]	[Wanninkhof, 1992]	[Nightingale et al., 2000]	[Elliott, 2009]
$U_{10} > 3.6$	$e_1 U_{10}(Dc_r/Sc)^{2/3}$	$e_1(U_{10})^2(Sc_r/Sc)^{1/2}$	$e_1 U_{10}(Sc_r/Sc)^{1/2}$ $+e_2 U_{10}^2(Sc_r/Sc)^{1/2}$	$e_1(U_{10})^2(Sc_r/Sc)^{1/2}$
$3.6 \leq U_{10} < 5.6$	$e_2(U_{10}-3.6)(Sc_r/Sc)^{1/2}$ $+e_3(Sc_r/Sc)^{2/3}$	$e_1(U_{10})^2(Sc_r/Sc)^{1/2}$	$e_1 U_{10}(Sc_r/Sc)^{1/2}$ $+e_2 U_{10}^2(Sc_r/Sc)^{1/2}$	$e_2(5.6-U_{10})/2.0$ $+e_3(U_{10}-3.6)/2.0$
$5.6 \leq U_{10} < 13$	$e_2(U_{10}-3.6)(Sc_r/Sc)^{1/2}$ $+e_3(Sc_r/Sc)^{2/3}$	$e_1(U_{10})^2(Sc_r/Sc)^{1/2}$	$e_1 U_{10}(Sc_r/Sc)^{1/2}$ $+e_2 U_{10}^2(Sc_r/Sc)^{1/2}$	$e_4(U_{10}-3.6)(Sc_r/Sc)^{1/2}$ $+e_5(Sc_r/Sc)^{2/3}$
$U_{10} \geq 13$	$e_4(U_{10}-13)(Sc_r/Sc)^{1/2}$ $+e_5(Sc_r/Sc)^{1/2}$ $e_6(Sc_r/Sc)^{2/3}$	$e_1(U_{10})^2(Sc_r/Sc)^{1/2}$	$e_1 U_{10}(Sc_r/Sc)^{1/2}$ $+e_2 U_{10}^2(Sc_r/Sc)^{1/2}$	$e_4(U_{10}-5.6)(Sc_r/Sc)^{1/2}$ $+e_5(Sc_r/Sc)^{2/3}$
Constants	$e_1 = 0.170$ $e_2 = 2.850$ $e_3 = 0.612$ $e_4 = 5.900$ $e_5 = 26.79$ $e_6 = 0.612$ $Sc_r = 600$	$e_1 = 0.31$ $Sc_r = 660$	$e_1 = 0.333$ $e_2 = 0.222$ $Sc_r = 600$	$e_1 = 0.310$ $e_2 = 0.310$ $e_3 = 2.850$ $e_4 = 2.850$ $e_5 = 0.612$ $Sc_r = 600$

Table 10.2: *Four parameterizations for the calculation of piston velocity, all depending on wind speed and temperature. The reference for each configuration is given under the table*

10.4.1 Option activation

To activate the calculation of dust emissions, some options have to be selected in the chimere.par file:

```
[iusedust]   Mineral dust emissions? (0/1/2)       : 1
[icuth]      u* threshold scheme                   : 1
[ifluxv]     vertical flux scheme                  : 3
[ifecan]     Use of soil moisture [1] or not [0]   : 1
```

- *[iusedust]*: when this flag is set to 1, mineral dust emissions are calculated.
  - iusedust=0 → No mineral dust emissions
  - iusedust=1 → Mineral dust emissions with only one species called DUST
  - iusedust=2 → Mineral dust emissions with mineralogy. Emissions are calculated for 8 chemical elements: magnesium (mg), phosphorus (P), calcium (Ca), manganese (Mn), iron (Fe), aluminium (Al), silicon (Si) and potassium (K).
  - iusedust=3 → Mineral dust emissions with mineralogy. Emissions are calculated for 12 mineral species such as Smectite, Illite, Hematite etc.
- *[icuth]* is for the u* threshold estimation with:
  - icuth=0 → [Iversen and White, 1982] scheme
  - icuth=1 → [Shao and Lu, 2000] scheme
- *[ifluxv]* is for dust production model for saltation and sandblasting, with:
  - ifluxv=0 → [Marticorena and Bergametti, 1995] scheme
  - ifluxv=1 → [Alfaro and Gomes, 2001]
  - ifluxv=2 → [Alfaro and Gomes, 2001] + optimization following [Menut et al., 2005]
  - ifluxv=3 → [Kok et al., 2014] scheme
- *[ifecan]* is for the use of soil moisture to inhibit mineral dust emissions, following the [Fecan et al., 1999] scheme.
  - ifecan=0 → no use
  - ifecan=1 → use

Dust emissions are managed and calculated in mainly three programs in CHIMERE:

1. src/chimere_params.F90: constant parameters needed for the dust production schemes

2. src/ini_demis.F90: initialization of tables, read input data

3. src/worker_dust.F90: calculation of dust fluxes

10.4.2 The datasets and input parameters

Before a simulation, some datasets are needed on the CHIMERE simulation grid. These datasets are explained in Table 10.3 . Note that explanations about the use of these datasets are also in [Menut et al., 2013b].

Parameters	Units	Time variability
USGS soil texture	% clay, % silt, % sand	constant
USGS landuse types	%	constant
Aeolian roughness length z_0	cm	constant
Vegetation fraction	%	variable (monthly)
10m wind speed	m/s	variable (hourly, on-line)

Table 10.3: *Mandatory informations for the mineral dust emissions fluxes calculations*

10.4.2.1 The USGS soil texture

The soil is represented by relative percentages of sand, silt and clay for each model cell. The USGS database, called STATSGO-FAO, is used and 19 different soil types are recorded in the global database. This is a global dataset used in meteorological modeling and which may be used for global and regional dust transport modelling and the data are available at *http://soils.usda.gov/survey/geography/*.
It provides soil texture characteristics over longitude/latitude regular grids. The native resolution of the dataset is 0.0083 $^o \times 0.0083\ ^o$. In order to have homogeneous datasets, the STATSGO-FAO data are regridded into the CHIMERE simulation grid, retaining the dominant feature as presented in Figure 10.1 .

The soil types in the two datasets are listed in Table 10.4 together with their relative percentage of clay, silt, salt and sand (coarse and fine-medium). Note that STATSGO-FAO does not contain salts and that fine-medium sand and coarse sand are distinguished according to criteria detailed in [Tegen et al., 2002]. With this approach clay loams are considered as highly unlikely to contain coarse sand and sandy clay loam can contain both coarse and medium fine sand. The textural triangle is based on measurements performed by wet sedimentation techniques which break the soil aggregates leading to high amounts of loose clay particles that generally form aggregates of larger size and that may not be encountered in natural soils. Following [Tegen et al., 2002] clay is in the form of aggregate in loamy sands and it is reassigned to the silt fraction.

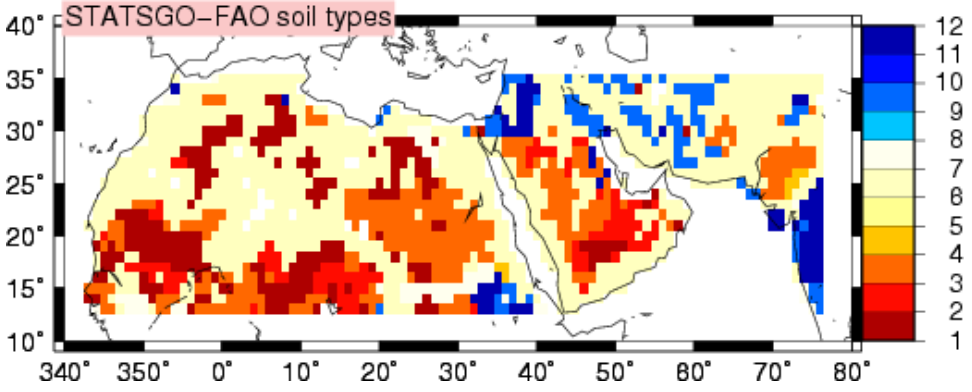


Figure 10.1: *The STATSGO-FAO soil types data regridded to $1^o \times 1^o$ resolution. The values correspond to the surface types explained in the Table 10.4 . For each cell, only the most important relative surface is displayed.*

Category	% CS	% FMS	% silt	% clay	% salt	α	Soiltype
1	46	46	5	3	0	2.e-6	sand
2	41	41	18	0	0	2.e-6	loamy sand
3	29	29	32	10	0	2.e-6	sandy loam
4	0	17	70	13	0	2.e-6	silt loam
5	0	10	85	5	0	2.e-6	silt
6	0	43	39	18	0	2.e-6	loam
7	29	29	15	27	0	2.e-6	sandy clay loam
8	0	10	56	34	0	2.e-6	silty clay loam
9	0	32	34	34	0	2.e-6	clay loam
10	0	52	6	42	0	2.e-6	sandy clay
11	0	6	47	47	0	2.e-6	silty clay
12	0	22	20	58	0	2.e-6	clay
13	0	0	0	0	0	2.e-6	organic material
14	0	0	0	0	0	2.e-6	water
15	0	0	0	0	0	2.e-6	bedrock
16	0	0	0	0	0	2.e-6	other (land-ice)
17	0	0	0	0	0	2.e-6	playa
18	0	0	0	0	0	2.e-6	lava
19	0	0	0	0	0	2.e-6	white sand
99	0	0	0	0	0	2.e-6	ocean

Table 10.4: *Description of the USGS soil types. CS = Coarse sand; FMS = Fine medium sand. Note that only the first 12 soiltype are really used in the mineral dust fluxes calculation. The α value is not used in this model version (but here just for consistency with older model version).*

10.4.2.2 The USGS landuse types

Landuse data describe the surface occupation (wood, mountains, lakes, etc.) and is typically used to prescribe some properties in meteorological models, such as the roughness length, z_0. For each land use type, a z_0, is typically prescribed. Dynamical roughness lengths based on land use types are not appropriate for use in dust emission parameterizations.
For mineral dust emission calculations, the landuse is typically used to provide a desert mask specifying what surface is potentially erodible. Global or regional models typically consider arid and semiarid categories from a landuse database to identify potential dust source areas. Here, to complement the STATSGO-FAO soil dataset, we use the USGS landuse dataset with a high horizontal resolution ($0.0083\,^o \times 0.0083\,^o$). As in the case of the STATSGO-FAO dataset, we calculated the relative surface of the landuse types. The erodible land use types considered were scrublands (8) and barren soil (19). This may appear as a limiting choice but this corresponds to the most important landuse classes able to produce dust in Northern Africa. This constitutes the desert mask as presented in Figure 10.2 .

10.4.2.3 The aeolian roughness length z_{0s}

The aeolian roughness length used in CHIMERE is the GARLAP (Global Aeolian Roughness Lengths from ASCAT and PARASOL) dataset. [Prigent et al., 2012] uses a bi-linear regression between in situ measurements on one hand and visible/near-infrared observations (PARASOL: Polarization and Anisotropy of Reflectances for Atmospheric Sciences coupled with Observations from a Lidar) and satellite microwave backscattering (ASCAT: Advanced SCATterometer) data on the other hand in order to create a new dataset of aeolian roughness length with a 6 km spatial resolution: GARLAP (Global Aeolian Roughness Lengths from ASCAT and PARASOL).

10.4.2.4 Erodibility

For erodibility, the database proposed by [Beegum et al., 2016] is used to calculate a factor related to arid areas. For all model cells considered as 'desert', the MODIS erodibility is used while for all other cells, a constant erodibility factor is applied depending on the USGS landuse, as in [Menut et al., 2013b]. This enables to have a more realistic description of the erodibility in arid areas.

10.4.2.5 User input parameters

Several input parameters are available before a simulation. These parameters are listed in Table 10.5 . They are considered as constant during the whole simulation. It is possible (but not recommended) to change these values in the

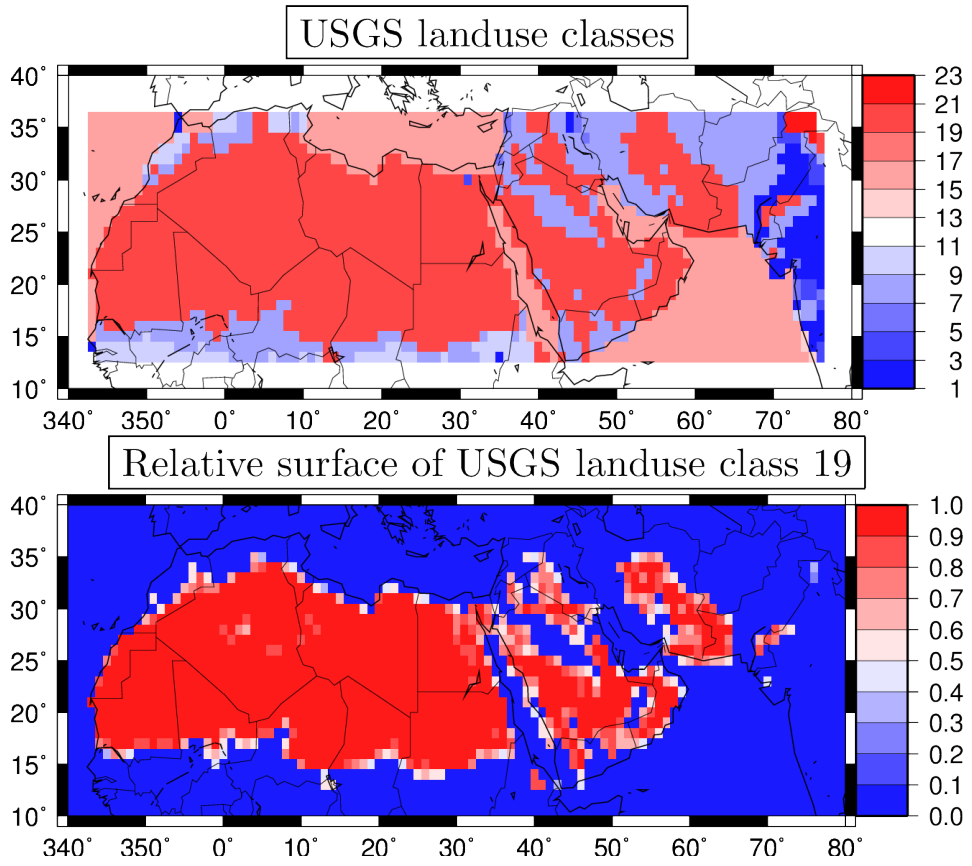


Figure 10.2: *[top] the USGS landuse classes (from 1 to 23) over the domain: (1) urban and built-up land, (2) dryland cropland and pasture, (3) irrigated cropland and pasture, (4) mixed dryland/irrigated cropland and pasture, (5) cropland/grassland mosaic, (6) cropland/woodland mosaic, (7) grassland, (8) shrubland, (9) mixed shrubland/grassland, (10) savanna, (11) deciduous broadleaf forest, (12) deciduous needleleaf forest, (13) evergreen broadleaf forest, (14) evergreen needleleaf forest, (15) mixed forest, (16) water bodies, (17) herbaceous wetland, (18) wooded wetland, (19) barren or sparsely vegetated, (20) herbaceous tundra, (21) wooded tundra, (22) mixed tundra, (23) bare ground tundra. [bottom] The relative percentage of USGS class 19 surface (for barren soils). This class is used as a desert mask for the dust fluxes calculation when using the USGS data.*

src/chimere_params.F90. It is then necessary to recompile the model.

Table 10.5: List of tunable parameters for the mineral dust emissions fluxes

Name	Type	Value
nmode	number of emissions modes for mineral dust	3
nmineral	number of emitted soil particles	5
nsoildmax	number of bins for the soil size distribution	20000
soil_dmin	soil size distribution min (cm)	10^{-4}
soil_dmax	soil size distribution max (cm)	2.10^{-1}
rop	particle density (g cm^{-3}) i.e quartz	2.65
umin	Min wind speed for the 'Owen' effect ($cm.s^{-1}$)	21.0
woff	Wind offset to smooth Richardson numbers (m/s)	0.5
Mineralogy of soil types		
mineral_dp	Mean Mass Median diameter of	
	coarse sand (cm)	0.0690
	fine medium sand (cm)	0.021
	silt (cm)	0.0125
	clay (cm)	0.0002

Continued on next page

Table 10.5 – continued from previous page

Name	Type	Value
	salt (cm)	0.052
mineral_sig	σ of	
	coarse sand	1.6
	fine medium sand	1.8
	silt	1.6
	clay	2.
	salt	1.5
Weibull distribution		
nwb	Number of values for the distribution	12
kref	Width of the distribution	3
Erodibility		
perod_veg	vegetalized areas	1.0
perod_urb	urbanized areas	1.0
ifit	reduce z0 in Europe or not	0
Snow and water in soil coverage		
snow_th	Snow cover if more that [snow_th] cm	0.1
minsurf_th	No flux if more than (minsurf_th x 100) % of water	0.
% of flux in the three modes (only for [Marticorena and Bergametti, 1995] scheme)		
cad1	% in the fine mode	0.2
cad2	% in the coarse mode	0.6
cad3	% in the big mode	0.2
Parameters for the [Alfaro and Gomes, 2001] scheme		
e1	Binding energies (g.cm^2/s^2)	3.61
e2	Binding energies (g.cm^2/s^2)	3.52
e3	Binding energies (g.cm^2/s^2)	3.46
div1	Factor for binding energies (g.cm^2/s^2)	2.0
div2	Factor for binding energies (g.cm^2/s^2)	2.0
div3	Factor for binding energies (g.cm^2/s^2)	2.0
Parameters for the [Kok et al., 2014] scheme		
kok_ds	Constant (μm)	3.4
kok_sigmas	Constant (ad)	3.0
kok_cv	Constant (μm)	12.62
kok_lambda	Constant (μm)	12.0

10.4.3 Mineral dust fluxes parameterizations

Mineral dust emissions strongly depends on input parameters such as the wind speed, the precipitation, the soil moisture etc. In order to have a realistic modelling of these fluxes, with the lowest as possible dependency to the horizontal resolution, many sub-grid scale parameterizations are implemented.

1. A Weibull distribution for the 10-m wind speed
2. The possibility to take into account the effect of rain on erodible surfaces
3. The calculation of the threshold friction velocities and the drag efficiency
4. The possibility to take into account the soil moisture effect (independently of precipitation)

10.4.3.1 Weibull distribution for 10m wind speed

The interest of the specific Weibull distribution to represent wind speed variability was extensively discussed by numerous authors such as [Pavia and O'Brien, 1986], [Pryor et al., 2005] and [Cakmur et al., 2004]. In order to build a distribution for a chosen wind speed value, representing the mean value for a specific area and period, the probability density function is expressed as:

$$p(|U|) = \frac{k}{A}\left(\frac{|U|}{A}\right)^{k-1} exp\left[-\left(\frac{|U|}{A}\right)^{k}\right] \tag{10.10}$$

where k is a dimensionless shape parameter and A is a scale parameter related to the mean of the distribution (in our case the modelled wind speed for each cell and each modelled hour).

10.4.3.2 The sporadic effect of rain

In this model version, we include the possibility to immediately inhibit dust erosion in case of sudden rainfall in western Europe: the modelled soil moisture will actually increase, thereby inhibiting dust erosion but, sometimes, this increase occurs only many hours after the start of precipitations. For dust erosion, two additional processes have to be taken into account:

- The dust erosion must be set to zero during a precipitation (convective or stratiform) event,
- The dust erosion must restart very slowly after a precipitation event, the soil being potentially crusted.

A simple function describing a factor f_p is applied to moderate the dust emissions fluxes when a precipitation is diagnosed as:

$$f_p = E_{dust} \times \left(1 - exp\left(\frac{-\alpha \times \Delta t_p}{\tau}\right)\right) \tag{10.11}$$

with α a constant factor set to α=6, Δt_p the time since the last precipitation event and τ the period after which the surface mineral dust fluxes E_{dust} is fully taken into account, considering that the inhibiting effect of precipitation is finished. For this study, t is in hours and τ=24. This function is displayed in Figure 10.3 .

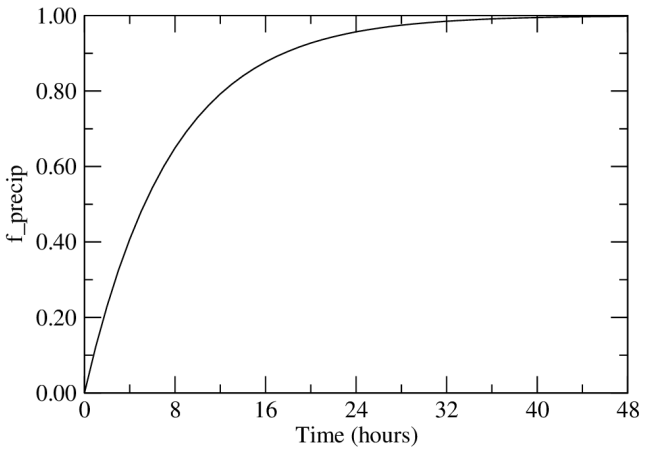


Figure 10.3: *Function defined to moderate the mineral dust emissions fluxes after a precipitation event.*

10.4.3.3 The threshold friction velocities and the drag efficiency

The saltation flux F_h is non-zero only if $u_* > u_*^T(D_p)$ for a given soil particle diameter D_p. u_*^T is the threshold friction velocity depending on D_p, the soil particle diameter, z_0, the 'aeolian' roughness length and z_{0s}, the 'smooth' roughness length.
The 'smooth' threshold friction velocity u_*^{Ts} is estimated following [Shao and Lu, 2000]:

$$u_*^{Ts}(D_p) = \sqrt{a_n\left(\frac{\rho_p g D_p}{\rho_{air}} + \frac{\gamma}{\rho_{air} D_p}\right)} \tag{10.12}$$

with the constant parameters $a_n = 0.0123$ and γ=300 kg m^{-2}. The particle density, ρ_p=2.65 10^3 kg m^{-3} is chosen to be representative of quartz grains.
The threshold friction velocity, u_*^T, is expressed as:

$$u_*^T(D_p) = \frac{u_*^{Ts}(D_p) \times q}{f_{\text{eff}}(z_0, z_{0s})} \tag{10.13}$$

with q a soil moisture correction. f_{eff} represents the drag efficiency, i.e a function depending on z_0, the 'aeolian' roughness length and z_{0s}, the 'smooth' roughness length which is expressed as:

$$f_{\text{eff}} = \left[1 - \left(\frac{ln(z_0/z_{0s})}{ln(0.35[(0.1/z_{0s})^{0.8}])}\right)\right] \tag{10.14}$$

This function takes into account the non erodible elements dissipating a part of the wind momentum that will not be available for saltation.

10.4.3.4 The soil moisture effect

The soil moisture effect is mainly taken into account following the [Fecan et al., 1999] parameterization. This scheme considers that soil moisture will increase the treshold friction velocity, u_*^T, used to determine if erosion occurs or not. To distinguish between soil conditions, the dry and wet treshold friction velocities are defined, and noted u_*^{Td} and u_*^{Tw}, respectively. u_*^{Tw} is estimated as a possible increase of u_*^{Td} as:

$$u_*^{Tw} = f(w) \times u_*^{Td} \tag{10.15}$$

In the model, the dry treshold friction velocity, u_*^{Td} is calculated following the scheme of [Shao and Lu, 2000]. The $f(w)$ factor is estimated as:

$$\begin{cases} f(w) = 1 & \text{for } w < w' \\ f(w) = \left[1 + A\,(w - w')^{b'}\right]^{0.5} & \text{for } w > w' \end{cases} \tag{10.16}$$

where A and b' are constant to estimate. w (in kg.kg^{-1}) is the gravimetric soil moisture and w' corresponds to the minimum soil moisture from which the threshold velocity increases. The values of A, b' and w' are dependent on the soil texture. Using measurements data, [Fecan et al., 1999] showed that the value of w' is mainly dependent on the clay content of the soil and proposed the following fit:

$$w' = 0.0014(\%clay)^2 + 0.17(\%clay) \tag{10.17}$$

The constants are fixed as A=1.21 and b'=0.68. Note that in Equation 10.16, the gravimetric soil moisture w has to be expressed in %, w' being in % in Equation 10.17.

10.4.4 The several mineral dust production schemes

Three different Dust Production Models are implemented:

1. The [Marticorena and Bergametti, 1995] scheme: it uses the [White, 1986]'s equation and distribution of emitted mass fluxes into the three emissions modes is constant
2. The [Alfaro and Gomes, 2001] scheme: it uses the [Marticorena and Bergametti, 1995] scheme for saltation and the vertical flux is estimated using cohesion kinetic energies scheme. A numerically optimized version presented in [Menut et al., 2005] is proposed.
3. The [Kok et al., 2014] scheme: the vertical dust flux is directly diagnosed for one emissions mode.

10.4.4.1 The [Marticorena and Bergametti, 1995] scheme

The vertically integrated saltation flux is estimated using the [White, 1986]'s equation:

$$F_h(D_p) = \frac{K\rho_{air}u_*^3}{g}\left(1 - \frac{u_*^T}{u_*}\right)\left(1 + \frac{u_*^T}{u_*}\right)^2 \tag{10.18}$$

where u_* is the "saltation" friction velocity, i.e the friction velocity calculated using the roughness length, z_0, following [Marticorena and Bergametti, 1995]. u_*^T is the threshold friction velocity depending on the soil particle diameter size D_p and z_0. Following [Marticorena and Bergametti, 1995], we consider the air density as constant with ρ_{air}=1.227 kg.m^{-3}, and used the constant value $K = 1$.

The scheme is designed to estimate one saltation flux. The corresponding vertical flux is estimated by using constant α values displayed in Table 10.4 for each soil type. The vertical flux is then projected into three modes using constant percentages between fine, coarse and big modes.

10.4.4.2 The [Alfaro and Gomes, 2001] scheme

In order to estimate size resolved vertical dust fluxes, i.e sandblasting fluxes, the [Alfaro and Gomes, 2001] DPM is used with the numerical optimization described in [Menut et al., 2005]. The sandblasting flux is computed based on the partitioning of the kinetic energy of individual saltating aggregates and the cohesion energy of the populations of dust particles. This model assumes that dust emitted by sandblasting is characterized by three modes whose proportion depends on the wind friction velocity. From wind tunnel measurements performed on two natural soils from semi arid regions, [Alfaro et al., 1998] consider these three modes as independent of the soil types. They described the three modes using log-normal distributions with diameters d_1=1.5 10^{-6}m, d_2=6.7 10^{-6}m and d_3=14.2 10^{-6}m and their associated standard deviation, respectively σ_1=1.7, σ_2=1.6 and σ_3=1.5. Based on this model, as soil aggregate size or wind speed increases, kinetic energy becomes able to release first particles of the coarsest mode that are associated with the lowest cohesion energy, then particles from the intermediate population, and finally the finest particles. It also implies that for a specific wind speed and soil size distribution, the dust flux may be zero even if the saltation process occurs. In order to apportion the available kinetic energy between the three modes, a constant cohesion energy e_i is associated to each mode values. The numerical values of e_i were determined by adjusting the predicted aerosols size distribution to those measured in wind tunnel under different wind conditions, using an iterative least square routine. In this study, the values recommended by [Alfaro and Gomes, 2001] are used: e_1=0.376, e_2=0.366 and e_3=0.346 kg m^2 s^{-2}.
The kinetic energy is expressed as a function of the soil particle diameter after [Alfaro et al., 1997] and [Shao and Lu, 2000]:

$$e_c = \rho_p \frac{100\pi}{3} D_p^3 \left(u_*\right)^2 \tag{10.19}$$

It is compared to the cohesion energy of the three aerosol modes in order to compute the proportion $p_i(D_p)$ of these three modes in the total dust size-distribution. By integrating equation 10.19 over the three aerosols modes, the total sandblasting flux may be written:

$$F_{v,m,i}(D_p) = \int_{D_p^{min}}^{D_p^{max}} \frac{\pi}{6} \rho_p \, \beta \, \frac{p_i(D_{p,i}) d_{m,i}^3}{e_i} dF_h(D_{p,i}) \tag{10.20}$$

where i=1,N_{class} is the number of intervals discretizing the soil size distribution in the range $[D_p^{min} : D_p^{max}]$ and $d_{m,i}$ the mean mass diameter, [Menut et al., 2005].

10.4.4.3 The [Kok 2014] scheme

In this model version, the Kok mineral dust emissions parameterization is proposed, in addition to the [Marticorena and Bergametti, 1995] and [Alfaro and Gomes, 2001] schemes. To activate this scheme, simply select the flag ifluxv=3 in the chimere.par file.
The Kok scheme is fully described in [Kok et al., 2014]. The vertical dust flux is calculated as:

$$F_d = C_d \times f_{bare} \times f_{clay} \times \frac{\rho_a \left(u_*^2 - u_{*t}^2\right)}{u_{*st}} \times \left(\frac{u_*}{u_{*t}}\right)^{C_\alpha \frac{u_{*st} - u_{*st0}}{u_{*st0}}} \tag{10.21}$$

The flux is calculated only if $u_* > u_{*t}$. The threshold friction velocity, u_{*t}, is calculated using the [Iversen and White, 1982] or the [Shao and Lu, 2000] scheme. The corresponding u_{*st} is this friction velocity but for a standard atmospheric density ρ_{a0}=1.225 kg m^{-3}:

$$u_{*st} = u_{*t} \times \sqrt{\frac{\rho_a}{\rho_{a0}}} \tag{10.22}$$

u_{*st0} represents u_{*st} for an optimally erodible soil and was chosen as u_{*st0}=0.16 m s^{-1} in [Kok et al., 2014]. The dimensionless coefficient C_α is chosen as C_α=2.7.
The dust emission coefficient C_d represents the soil erodibily as:

$$C_d = C_{d0} \times exp\left(-C_e \frac{u_{*st} - u_{*st0}}{u_{*st0}}\right) \tag{10.23}$$

with the constant dimensionless coefficients C_e=2.0 and C_{d0}=4.4 10^{-5}.
The vertical dust flux is integrated over the whole size distribution. This flux is thus redistributed into the model dust size distribution as:

$$\frac{dV_d}{dlnD_d} = \frac{D_d}{c_v}\left[1 + erf\left(\frac{ln(D_d/D_s)}{\sqrt{2}\,ln\sigma_s}\right)\right] exp\left[-\left(\frac{D_d}{\lambda}\right)^3\right] \tag{10.24}$$

with V_d the normalized volume of mineral dust aerosols for each mean mass median diamter D_d, C_v=12.62 μm, σ_s=3.0, D_s=3.4 μm and λ=12.0 μm.

10.4.5 Vegetation variability

The density of the vegetation cover has an impact on the dust emissions flux [Wolfe and Nickling, 1993] and the seasonal variation of vegetation affects the aeolian roughness length [Prigent et al., 2005]. As the z_0 dataset is representative of northern winter months only [Prigent et al., 2012], it does not take into account vegetation variations. Therefore, we weight the z_0 by the green fraction coefficient of the current month, provided by NCAR, divided by the green fraction coefficient of January. This is a simple correction that induce a monthly variability in mineral dust emissions.

10.4.6 Mineralogy

If the user's option is activated, the output flux of mineral dust may be splitted into several minerals or chemical elements. All informations about this calculations are presented in [Menut et al., 2020b]. For this calculation, several additional databases are read during the initialization phase.

- Soil databases describing the relative part of each mineral and each chemical element in each model grid cell
- The relative part of clay and silt in each grid cell and for each mineral
- For each mineral, its density and refractive index
- The solubility of each chemical elements as a function of each mineralogical species

In order to have information on the dust mineralogical composition, the global databases presented in [Journet et al., 2014] are used. These databases are delivered with an horizontal resolution of $0.5^o \times 0.5^o$ and are then interpolated on the model grid used in this study. Data are provided for 12 different species, listed in Table 10.6. For each mineral, we indicate if it can be found in the clay or silt fraction of the soils. The density of each is also provided: this information comes from several references, including [Perlwitz et al., 2015]. These values for densities have a non-negligible but not quantified uncertainty. For each mineral, the reference (peer-reviewed publication or internet database) is specified in the Table. Note that for the Mica density, being a group of numerous minerals, the density of muscovite is used.
For each mineral species, we estimate the chemical elements composition for the following 8 elements: magnesium (mg), phosphorus (P), calcium (Ca), manganese (Mn), iron (Fe), aluminium (Al), silicon (Si) and potassium (K). This information was collected from the following previous studies, [Kandler et al., 2007], [Journet et al., 2014] and [Zhang et al., 2015]. The solubility is also provided as a percentage of each chemical element in each mineral.

10.5 Fire emissions

10.5.1 Option activation

To account for fire emissions in the CHIMERE simulations, the following lines need to informed in chimere.par:

```
[$iuse_firemis]     Fire emissions?   (1/0)              : 1
[$ifirevprof]       Fire vertical profile shape (1/2/3)  : 1
[$imakefemis]  Build fire emissions? (0-2)               : 1
[$fire_emissdir] Fire emission Dir (stage 1)             : <path>
```

Number	Mineral	Silt	Clay	Density
1	calcite	√	√	2.71 (1)
2	chlorite	√	√	2.42 (3)
3	feldspar	√	√	2.68 (1)
4	goethite	√		4.18 (2)
5	gypsum	√		2.30 (2)
6	hematite		√	5.25 (2)
7	illite		√	2.57 (1)
8	kaolinite		√	2.63 (1)
9	mica	√		2.81 (3)
10	quartz	√	√	2.67 (1)
11	smectite		√	2.57 (1)
12	vermiculite		√	2.30 (3)

Table 10.6: *List of mineral species used in this study (in alphabetical order). Their possible presence in the soil fraction that consists of silt and clay is also indicated. Densities in g.cm^{-3}, after (1) [Perlwitz et al., 2015], (2) http://www.mindat.org, (3) http://www.engineeringtoolbox.com/mineral-density-d_1555.html. For the Mica density, being a group of numerous minerals, the density of muscovite is used.*

Species list is the same as the anthropogenic one in the current version.
Fire emissions have to be provided in the following file format:

- One netcdf file per species and per month;
- Hourly emissions gridded on the simulation domain;
- Emissions are only provided for grid cells with non zero values, as a list of sources
- List of variables:
  - lon: Longitude grid (2D)
  - lat: latitude grid (2D)
  - ilatsources: latitude index of non-zero sources (1D=sources)
  - ilonsources: longitude index of non-zero sources (1D=sources)
  - SPEC: hourly emissions for species SPEC (3D=hours,days,sources) in molecule/cm2/s

Emission fluxes are precalculated and processed in CHIMERE during the simulation (prep_femis.F90). In addition to reading the emissions, a vertical profile of injection is estimated (cf. Section 10.5.2.3). Example netcdf file header for CO is given in **Appendix E.2.3**.

10.5.2 How to calculate biomass burning emission fluxes?

Biomass burning emission fluxes may be included in CHIMERE model simulations using two different sources:

- The CAMS model fluxes with an horizontal resolution of 0.1 × 0.1 degrees. These fluxes are reprojected on the CHIMERE grid and with the used chemical model species.
- The APIFLAME model fluxes, with an adaptable horizontal resolution, until the maximum resolution of 1km × 1km. This model reads satellite data, global model fuel load and calculates emissions fluxes. This model is more dedicated to research and enables to develop new biomass burning parameterizations, to develop scenarios, to change hypothesis on the emitted chemical species, etc.

10.5.2.1 The CAMS pre-processor

The pre-processor for ECMWF CAMS data formatting is a program distributed with the new CHIMERE version. The goal is to download the ECMWF model outputs of [Kaiser et al., 2012] and adapt them to the CHIMERE format. For this formatting, several conversions are needed: project the data onto the CHIMERE simulation grid, convert the proposed chemical model species to the CHIMERE chemical mechanism species, temporally refine the data by shifting the time resolution from daily to hourly fluxes. All available chemical species are downloaded and used to create the CHIMERE emitted species. Temporally, having no realistic information to move from daily to hourly emissions fluxes, an academic diurnal profile is applied on the daily data as:

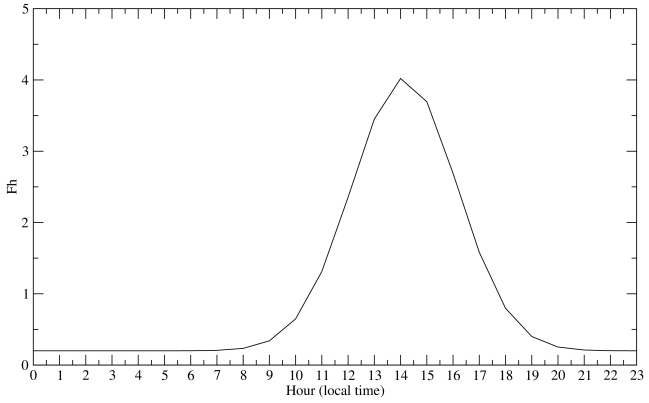


Figure 10.4: *Academic hourly profile applied to the constant daily wildfire fluxes.*

$$\begin{aligned} F_h &= d_w + (1 - d_w)\frac{24}{d_s\sqrt{2\pi}} \\ &\times exp\left(-0.5\left(\frac{h + 0.5 + \delta - t_0}{2d_s}\right)^2\right) \end{aligned} \tag{10.25}$$

with the constant values: h the current GMT hour, d_w=0.2, d_s=2.0, t_0=13.5 and δ=longitude/360x24. An example is displayed in Figure 10.4 .

10.5.2.2 APIFLAME v2

APIFLAME was designed to calculate emissions from biomass burning at high spatial and temporal resolution with flexibility in terms of input data used. While the CAMS approach is based on fire radiative power (FRP) detection (active fires), APIFLAME calculates emissions based on the burned area and resulting biomass consumed (in terms of carbon), as described in [Menut et al., 2013a, Turquety et al., 2014]. Maps of burned area and FRP on the user-defined grid are derived (with corresponding *geog* file) from the MODIS satellite observations (MOD14 and MCD64A1, respectively, version v006), combined with information of vegetation type (satellite observations or landuse databases). Depending on the type of vegetation burned, the amount of carbon consumed is calculated by multiplying burned area with fuel consumption derived from the ORCHIDEE land surface model [Turquety et al., 2014]. Emissions for inventory species are then calculated using a list of emission factors, and converted to emissions fluxes for CHIMERE species using aggregation matrices accounting for reactivity to OH (either SAPRC or MELCHIOR chemical scheme).
The main improvement included in version 2 [Turquety et al., 2019] is related to temporal variability, with the possibility to: (i) redistribute total monthly burned area (MCD64 data) according to the FRP value (MOD14 data) so that higher emissions will correspond to enhanced fire intensity; (ii) redistribute daily emissions using the diurnal variability of FRP in the geostationary observations (SEVIRI above Europe and Africa). Small fires that may be observed with the active fire detection (FRP) but that are not present in the burned area product may also be included.
The flexibility in the use of input datasets for fire activity or vegetation type burned allows the calculation of ensembles that provide information on uncertainties related to these choices. The analysis of a case study in Portugal during the summer of 2016 shows an average dispersion of $\sim$75 % on daily carbon emissions. The list of species included may also be modified without modifying the core of the model (input datasets).

10.5.2.3 The height and shape of the vertical injection

The CAMS or APIFLAME data provide a biomass burning surface flux. It is necessary to redistribute vertically the flux along a profile representing the pyroconvection. The injection height, H_p, estimation is based on the [Sofiev et al., 2012] scheme, as:

$$H_p = \alpha\, H_{abl} + \beta \left(\frac{FRP}{P_{f0}}\right)^{\gamma} exp\left(-\delta N_{FT}^2 / N_0^2\right) \tag{10.26}$$

with α=0.24, β=170m, γ=0.35 and δ=0.6, P_{f0}=10^6W and N_0^2=2.5 $\times$ 10^{-4} s^{-2}. N_{FT} is the Brünt-Vaisala frequency and is calculated between H_{abl} and the level pressure correponding to 300hPa (to ensure to remains in the free troposphere). The Fire Radiative Power (FRP) is expressed in [W] and provided by the emissions fluxes data. In case of 'huge' fires, i.e if the H_p value is greater than 1500m AGL, the calculation is redone following [Veira et al., 2015] and after replacing FRP by $FRP^* = FRP\sqrt{H_p/H_{deep}}$, with $H_{deep} = 1500$m.
After the estimation of the height of the mass injection, it is possible to estimate the shape of the injection profile. Three user's options are implemented in the model, Figure 10.5 .

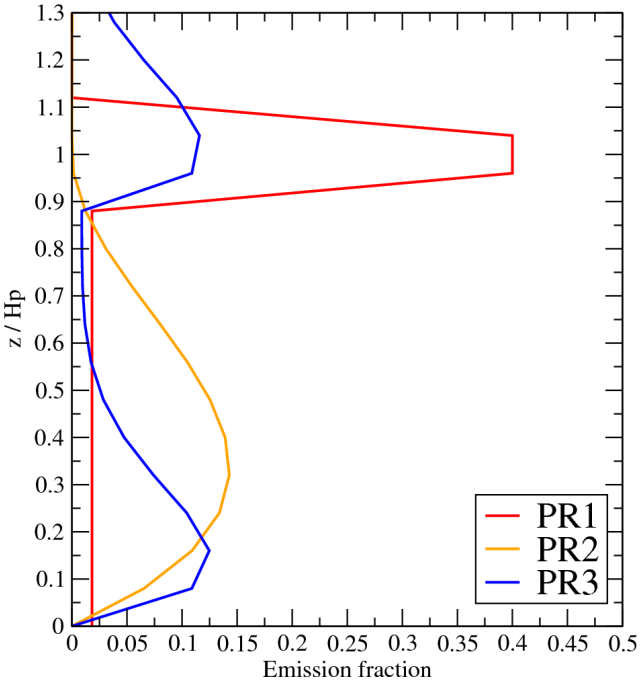


Figure 10.5: *Vertical profiles of factors used for the vertical redistribution of the biomass burning emissions fluxes.*

- PR1: The mass flux is splitted in two parts: 20% is injected between surface and $0.9 \times H_p$ and 80% between $0.9 \times H_p$ and $1.1 \times H_p$. Inside these two intervals, the flux is constant in altitude. This profile is close to the shape of convective profile obtained using thermal parameterizations.

- PR2: The emissions are injected between the surface and H_p, following a profile close to the one used in [Veira et al., 2015] with a K_z-like shape:

$$\begin{cases} \text{if } z_n \leq 1 & EF(z) = H_p\; z_n \left(1 - z_n\right)^2 \\ \text{if } z_n > 1 & EF(z) = 0 \end{cases} \tag{10.27}$$

  with EF the emission flux and $z_n = z\,/\,H_p$.

- PR3: This profile also uses K_z-like shape, but with two maxima: one in the surface layer and one around H_p.

These profiles were tested in [Menut et al., 2018], with the simulation of biomass burning episodes in the center of Africa. It was shown that, after long range transport, the atmospheric concentration of pollutants is not very sensitive to the shape of the vertical profile. It is mainly due to the fact that the pyroconvection is with higher temperature and more turbulence than the environment: the emitted species are quickly and homogeneously mixed in the whole column between the surface and H_p.

10.6 The urban resuspension

The urban resuspension is an emission process and is estimated using the surface wind speed. This emission flux could be activated or not the the chimere.par file:

```
[resusp]      Resuspension [1] or not [0]   : 0
```

The resuspension process is important for particulate matter and may induce a large increase of the emission flux in case of dry soils, for locations where traffic and industries produce available particles. This resuspension process is considered as different and complementary of the aeolian erosion process (leading to mineral dust emissions). In this model version, the resuspension flux is active only for cells containing an urbanized surface. The formulation is derived from the bulk formulation originally proposed by [Loosmore, 2003]. The resuspension rate λ, in s^{-1}, is expressed as:

$$\lambda = 0.01 \frac{u_*^{1.43}}{\tau^{1.03}} \tag{10.28}$$

where τ is the time after the start of resuspension. This time is taken into account considering the particles are first deposited then resuspended. The detail of the processes leading to resuspension are essentially unknown, and we assume here that the available concentration of particulate matter does only depend on the wetness of the surface. In this empirical view, the resuspension flux is assumed to be:

$$F = P \times f(w) \times u_*^{1.43} \tag{10.29}$$

where $f(w)$ is a function of the soil water content and P is a constant tuned in order to approximately close the PM10 mass over Europe. The value of P was estimated in [Vautard et al., 2005a] and was found to give a correct amount of additionnal PM_{10}. In this model version, P is approximated as P=170 $\mu g m^{-2} h^{-1}$ if we consider European mean conditions with a soil water content of 25% and a friction velocity of u_*=0.5 $m.s^{-1}$.
The soil water function $f(w)$ is estimated as:

$$f(w) = \frac{soim_s - g_{soim}}{soim_s - soim_t} \tag{10.30}$$

where $soim_t$ is a soil moisture treshold chosen as $soim_t$=0.1. $soim_s$ is the maximum of soil moisture ponderated by the ratio of water and soil densities, as:

$$soim_s = soim_{max} \times \frac{D_{water}}{D_{soil}} \tag{10.31}$$

with $soim_{max}$ a constant, representing the maximum sol moisture value and, here, chosen as $soim_{max}$=0.3. D_{water} is the water density (assumed to be unity) and D_{soil} is the dry porous soil density. D_{soil} is itself estimated as:

$$D_{soil} = (1 - satsm) \times D_{mine} \tag{10.32}$$

with $satsm$=0.4 the saturation volumetric moisture content and D_{mine}=2.5, the non porous soil density. Those global resuspended flux has to be projected into the aerosol size distribution.
In the absence of any information, this resuspension flux is calculated only for model cells having a non-zero urban landuse. This flux is thus ponderated in the whole cell by considering the relative surface of the urban area. Another hypothis is that the flux is affected into the PPM emissions, considering that this contribution is mainly primary particulate matter emissions (such as industries). Finally, the flux is projected into the model size distribution considering that 2/3 of the flux is in the fine mode, 1/3 in the coarse mode and zero in the big mode. This assumption may be changed bu the user directly in the src/resuspension.F90 program as:

```
rflux(1) = 2d0/3d0*rcumul
rflux(2) = 1d0/3d0*rcumul
rflux(3) = dzero
```

11 [chemprep] the chemical preprocessor

11.1 Preparation of new chemical data

11.1.1 Available options for the chemistry

CHIMERE uses a chemical pre-processor to prepare all necessary informations about the chemistry, both for gaseous species and aerosols. This allows to have a very flexible code where the chemistry has not to be hardcoded in Fortran and in the model itself.
The preparation of the chemistry is done before a simulation and only if the data directory is not existing yet:

- When the script chimere.sh is launched, the availability of chemistry input files corresponding to the specific simulation is checked. If the data already exist, the run continues.
- If not, the script scripts/make-chemistry.sh creates all prerequesite files in a new directory called something like:

```
chemprep/inputdata.211100210121.10
```

  In this new directory, all input files required by CHIMERE for a simulation are stored. The arguments called labchem (following the string inputdata) correspond to the simulation options having an impact on the files preparation. labchem is designed as:

```
labchem=${mecachim}${aero}${iusesalt}${carb}${brc}${trc}${soatyp}
        ${iusedust}${pollen}${iusedms}${iadv}${iadvv}.${nbins}
```

  For example and for a simulation with only gas-phase chemistry and no aerosols, but with a passive gaseous tracer, the directory created will have the label *"100000100000.0"*. This is with this code that the model is able to detect if the chemical files are already available or not. All these parameters are described in the 'chimere.par' file in section 3, p.23.
- Note that in this model version, the 'pollen' is not implemented and have to be zero.
- If a new simulation with the same options is launched or in case of a restart, the files will be used as it: the code will check automatically the existenz of the directory. If the main run options change, the script will automatically be launched to create new files.

If the chemistry directory already exists but the user wants to reprocess it, he can force the building of the whole chemistry by changing the option imakechemprep in the chimere.par parameter file. It is also possible to launch the model to only make the 'chemprep' stage and stop the code by changing istopchemprep. This allows to manually modify the created files (but of course this is not recommended). The possible options are (with in red, the recommended option to save in your chimere.par file):

imakechemprep = 0 : No preparation. We consider the directory already exists and is OK
= 1 : Force building of data
= 2 : Check is directory exists. If yes, continue.
istopchemprep = 0 : Prepare the data if necessary and continue.
= 1 : Prepare and forced stop.

11.1.2 The make-chemistry.sh script

If imakechemprep is 1 or 2, the model will launch automatically the make-chemistry.sh script (contained in the chimere.sh main model script). The main role of make-chemistry.sh is to launch the following scripts:

1. scripts/chemprep-check_consistency.sh:
   check consistency between the several user parameters

2. scripts/chemprep-prep-chemistry-data.sh:
   make the new inputdata directory and all files to create. This script is the main chemprep script.

3. scripts/chemprep-prep-boun-data.sh:
   prepare some data files linking the selected model species for the simulation and the available data contained in the climatological files for the boundary conditions.

4. scripts/chemprep-copy-chemresults.sh:
   at the end, copy the new data files in the inputdata directory.

The first script, scripts/chemprep-check_consistency.sh is very simple and only designed to check is the user's options are logical. The last script, scripts/chemprep-copy-chemresults.sh is also simple and just dedicated to copy all the new files into the new inputdata directory.
The main part of the work is done by the scripts/chemprep-prep-chemistry-data.sh scripts, which makes concatenations of already existing files and calls programs. This script is fully described in the next section 11.2.
The script chemprep-prep-tracer-data.sh adds parameters of gaseous and particulate tracers to the data files describing parameters of the chemistry mechanism, depositions, and output.This script is fully described in the next section 11.4.

11.2 The preparation of gaseous and aerosol chemistry

The script scripts/chemprep-prep-chemistry-data.sh is the one making all new required files. There is three main steps in this scripts:

1. Concatenate already existing ascii files

2. Run the src/chemprep-distrib.F90 program for the calculation of the aerosol size distribution

3. Run the src/chemprep-families.F90 programs to create model species families, following the user choices.

11.2.1 The input chemistry data files

All necessary chemistry files are already prepared. The main goal of the scripts/chemprep-prep-chemistry-data.sh is to concatenate the files depending on the user's options. If the user wants to make a simulation without aerosol for example, only the gaseous files will be used.
The complete list of input ascii files is in the following Table 11.1 .
Note that some files are more detailed in other sections of this documentation:

- The FastJ files are detailed in the section 14, p.149.
- The dry deposition files DEPO_SPEC.$meca depends on the chemical mechanism used, and, thus, the list of active model species. The constants are necessary for the calculation of the quasi-laminar layer resistance r_b for gases. The details are given in Table 15.1 , p.157.
- The DEPO_PARS.univ is not dependent on the chemical mechanim used and is described in Table 15.3 , p.159.
- The wet deposition file WETD_SPEC.univ is detailed in Table 15.4 , p.166.

In Table 11.1 , there are two kinds of data files:

- The data files with the extension "univ" are not dependent on the chemical mechanism used
- The datafiles "ANTHROPIC", "REACTIONS", "DEPO_SPEC.$meca" and "MAKEPRI" directly depend on the chemical mechanism used and thus of the chemical model species.

The BIOGENIC.univ file contains the list of the model species considered as biogenic, i.e with possible emissions calculated using the MEGAN model. More details about these species are given in section 10.1, p.105.
The SEMIVOL.univ file contains parameters for the semi-volatil species (saturated vapor pressure, vaporisation enthalpy, temperature). This datafile is read in the program src/iniaero.F90 and it contains the informations described in Table 11.2 .

File name	Content	Format	Changeable
AEROMIN.nc	Constants for ISORROPIA	NetCDF	no
AEROORG.nc	Constants for ISORROPIA	NetCDF	no
FJX_spec_$meca.dat	Parameters for FastJX	ascii	no
FJX_scat-cld.dat	Cloud scattering data for FastJX	ascii	no
FJX_scat-UMa.dat	Parameters for FastJX	ascii	no
input-fastj.txt	Parameters for FastJX	ascii	yes
atmos_std.dat	Temperature and ozone climatology for FastJX	ascii	no
DEPO_SPEC.$meca	Dry deposited species	ascii	yes
WETD_SPEC.univ	Constants for the wet deposition	ascii	yes
BIOGENIC.univ	List of biogenic active species	ascii	yes
DEPO_PARS.univ	Constants for the dry deposition calculation	ascii	yes
SEMIVOL.univ	Constant for the semi-volatils species	ascii	yes
ANTHROPIC.*	List of anthropogenic species	ascii	yes
AEROSOL.*	Characteristics of aerosols	ascii	yes
MAKEPRI.*	List of mean mass median diameters for emitted aerosols	ascii	yes
REACTIONS.*	List of chemical reactions	ascii	yes

Table 11.1: *List of primary data files in the chemprep/data directory.*

Name	p0L	a0L	t0L
AnA0D	$1.0.10^{-10}$	0.	298.
AnA1D	$1.0.10^{-10}$	0.	298.
AnA2D	$1.0.10^{-10}$	0.	298.
AnBmP	2.10^{-3}	18772.	298.
AnBIP	$2.6.10^{-7}$	18772.	298.
BiA0D	$1.0.10^{-10}$	0.	298.
BiA1D	$1.0.10^{-10}$	0.	298.
BiA2D	$1.0.10^{-10}$	0.	298.
BiBmP	$6.6.10^{-4}$	18772.	298.
ISOPA1	$1.9.10^{-3}$	18772.	295.
ISOPA2	$1.0.10^{-5}$	18772.	295.
BiBhP	$1.0.10^{-10}$	0.	298.
BiBIP	$1.0.10^{-10}$	0.	298.
SOA	$1.00.10^{-10}$	0.	298.00
BaP	$9.34.10^{-7}$	11488.	283.15
BbF	$2.05.10^{-7}$	10541.	283.15
BkF	$7.00.10^{-7}$	10194.	283.15

Table 11.2: *Parameters needed for the semi-volatil compounds and the Secondary Organic Aerosols (SOA) formation. These species are not dependent on the gazeous chemical mechanism, their reactions being added with the secondary organic aerosol scheme chosen.*

11.2.1.1 The chemical mechanisms

Three mechanisms are available in *v2020r1*:

1. MELCHIOR1: the reduced version of the MELCHIOR mechanism, described in section G.1, p.291
2. MELCHIOR2: the complete version of the MELCHIOR mechanism, described in section G.2, p.298
3. SAPRC: The SAPRC-07-A chemical mechanism, described in section G.3, p.301

These files must be modified by the user before the run of chemprep. Their format is detailed in the next sections.

The main principle of the chemprep pre-processor is to concatenate these files, depending on the user's options. For example:

- A simulation with only the gaseous species and the SAPRC mechanism:
  ANTHROPIC=ANTHROPIC.saprc
- A simulation with gas and aerosols, including SOA calculation:
  ANTHROPIC=ANTHROPIC.saprc + ANTHROPIC.aero + ANTHROPIC.med

11.2.1.2 The ANTHROPIC data files

This file is only a list of model species names. For gaseous species, the model species name may be 'APINEN', 'C2H4' etc. The important point is to have these species as in the chemical mechanism file.
For aerosols, the model species are always considered with three modes:"_fin", "_coa" and "_big". For example, OCAR will be defined in the ANTHROPIC file as OCAR_fin and OCAR_coa. This specific management of the aerosol model species name is due to the emissions inventories often delivered with these three modes. This is also to allow to have a flexible aerosol size distribution, with a number of bins changeable by the user.

11.2.1.3 The AEROSOL data files

These files are also a list of model species names but with more informations for each species. For example, the data file AEROSOL.aero.med contains the informations:

```
PPM         1       100.      1
AnA1D       2       168.      1
AnBmP       2       152.      1
BiA1D       2       170.      0
BiBmP       2       236.      0
ISOPA1      2       177.      0
H2SO4       3       96.       1
HNO3        3       62.       1
NH3         3       18.       1
WATER       4       18.       1
```

with the columns:

1. The model species name (without the extension, being here representative of the aerosol without any consideration on its size distribution). This is the table aeroname(nkc) in the model.

2. The type of aerosol. Four types are currently defined in the model:

   (a) ntypesp = 1: the primary species

   (b) ntypesp = 2: the semi-volatile organic species

   (c) ntypesp = 3: the semi-volatile mineral species

   (d) ntypesp = 4: the similar water species

   This is the table ntypesp(nkc) in the model.

3. The molar mass. When this molar mass is not known (as for PPM), the value of 100 is put in this file. This is the table conmw(nkc) in the model.

4. The "origin" of the chemical species. Three types are defined and used in the model core:

   (a) 0 = biogenic

   (b) 1 = anthropogenic

   (c) 2 = mineral

   This is the table origin(nkc) in the model.

11.2.1.4 The MAKEPRI data files

All aerosols are considered with a size distribution. This size distribution is always considered as tri-modal. The MAKEPRI are designed to give informations about these three modes. The user can directly change these values.
For example, the mineral dust are defined with the following constants in MAKEPRI.dust:

```
DUST_fin  DUST  1 1.5e-6 1.7
DUST_coa  DUST  1 6.7e-6  1.6
DUST_big  DUST  1 14.2e-6 1.5
```

where:

1. *DUST_fin*: The species name and its extension to know if we are defining the fine, coarse or big mode.
2. *DUST*: The model species name finally used in CHIMERE
3. *1*: An indicator to recognize:
   (a) 1 = the aerosols with a classical log-normal distribution
   (b) 2 = a specific distribution for sea salt (following the recommendations of [Monahan, 1986])
4. *1.5e-6*: the mean mass median diameter (in meters)
5. *1.7*: the width of the distribution σ (no unit)

All MAKEPRI files are concatened and then read by the src/chemprep-distrib.F90 program.

11.2.1.5 The REACTIONS data files

All lot of REACTIONS files are available, each one listing chemical reactions. These files represent (i) the gaseous mechanisms, (ii) the types of aerosols the user wants in its simulation.
The choice was made to have a chemical mechanim in *human readable format*. This allows the user to simply modify a REACTION file or add a new one. But a specific syntax was defined and has to be respected.
For example, the first lines of the REACTIONS.univ.melchior2 data file are:

```
#    Inorganic chemistry    #
#############################

O3+NO->NO2             k(T)=Aexp(-B/T),A=1.8e-12,B=1370
O3+NO2->NO3            k(T)=Aexp(-B/T),A=1.2e-13,B=2450
O3+OH->HO2             k(T)=Aexp(-B/T),A=1.9e-12,B=1000
O3+HO2->OH             k(T)=Aexp(-B/T),A=1.4e-14,B=600
NO+HO2->OH+NO2         k(T)=Aexp(-B/T),A=3.7e-12,B=-240
NO2+OH+M->HNO3         k(T,M)=mtroe(3.4e-30,0,3.2,4.77e-11,0,1.4,0.30)
NO3+HO2->NO2+OH        k=4e-12
NO3+H2O2->HNO3+HO2     k=2e-15
```

where:

1. The **first column** is the chemical reaction. This reaction is written as:
   A1 + A2 -> α1 B1 + α2 B2
   The A_i species are the reactants (max=4), the B_i species are the products and the α_i are the stoechiometric coefficients of each B_i. The A_i must be active species, and can be repeated as many times as desired.
2. The **second column** is the reaction rate. The chemprep.awk program will interpret automatically the form of the reactions rate, considering its formulation follows the rule described in Table 11.3 . The mathematical expressions for reaction rates (the K's and J's) can take more than 20 forms.

Note there are other rules and possibilities:

- Each line is considered as a reaction except blank lines or lines starting with a "#" sign (left for comments).
- Species names must contain at most 15 characters.
- There are special species whose concentration is only prescribed in the model. These are:
  1. H2O (water vapor),

2. N2 (molecular nitrogen),
3. O2 (molecular oxygen),
4. M (80% N2 + 20% O2).

Their values are given by the meteo data. Reaction with these species is possible with no constraint. Changes in these "prescribed species" imply several modifications in the model and are not an easy issue.

- Reactants can also be one (or several) of the prescribed species (H2O, N2, O2 or M).
- Syntax like A1+A2-A3 is NOT allowed
- Only species which are *consumed* at least once in the mechanism will be considered as **active species** in the model. If the user intends to use a species which results of chemical reactions but is not a reactant in any reaction, she (he) should add an artificial reaction to the mechanism, for instance A->X (A is the concerned product), with a negligible reaction rate, i. e. k=1e-10 (s-1).
- The mathematical expressions for reaction rates (the K's and J's) can take 20 forms, but here only a limited number is available. The table below shows the available possibilities, together with associated syntax.
- For the reaction rate #5 (the photolysis reactions), *label* is a label standing for recognition of chemprep in the FJX_j2j.dat file.
- The reaction rate type #13 is the aggregation of the ozone photolysis with reactions with atomic oxygen. The photolysis rates must be those of ozone only, hence the budget reaction is O3->2*OH.
- The reaction rate #14 corresponds to a specific Fall-off expression to apply to the reaction NO2+OH+M $\rightarrow$ HNO3

This is possible to change or add reactions. This is also possible to add a new reaction rate type. In this case, the user must change several programs:

1. Add this new type description in the code chemprep.awk
2. Define the number of constant ltabrate in the program src/inichem.F90
3. Add the new mathematical expression in the program src/rates.F90

Table 11.3: List of possible chemical reactions

Type number	**Name**	**Mathematical expression**	**syntax in REACTIONS file**
1	Constant K	K	k=*value*
2	Simplified Arrhenius	K=Aexp(-B/T)	k(T)=Aexp(-B/T), A=*value*, B=*value*
3	Arrhenius	K=Aexp(-B/T)(300/T)**N	k(T)=Aexp(-B/T)(300/T)**N, A=*value*, B=*value*, N=*value*
4	Troe or Falloff	K=K_0*f**Z/(1+M*(K_0/K_∞)) K_0=A_0exp(-B_0/T)(300/T)**N_0 K_∞=A_∞exp(-B_∞/T)(300/T)**N_∞ Z=1/(1+(Log10(M*K_0/K_∞))**2) M=air-density	K(T,M)=troe(A_0,B_0,N_0,A_∞,B_∞,N_∞,f)
5	Photolysis	J=J(Z)	J(Z)=photorate(*label*)
6	Special type	K=K_1*K_2/(1+K_2) K_1=A_1exp(-B_1/T) K_2=A_2exp(-B_2/T)	k(T)=SPECIAL_1(A_1, B_1, A_2, B_2)
7	Special type	K=K_1/(1+K_2) K_1=A_1exp(-B_1/T) K_2=A_2exp(-B_2/T)	k(T)=SPECIAL_2(A_1, B_1, A_2, B_2)

Continued on next page

Table 11.3 – continued from previous page

Type number	Name	Mathematical expression	syntax in REACTIONS file
8	Special type	K=2*sqrt(f1*f2*f3*f4/((1+f3)*(1+f4))) f1=A_1exp(-B_1/T) f2=A_2exp(-B_2/T) f3=A_3exp(-B_3/T) f4=A_4exp(-B_4/T)	k=SPECIAL_3(A_1,B_1,A_2,B_2,A_3,B_3,A_4,B_4)
9	Special type	K=2*sqrt(f1*f2)*(1-sqrt(f3*f4))*(1-f4)/(2-f3-f4) f1=A_1exp(-B_1/T) f2=A_2exp(-B_2/T) f3=A_3exp(-B_3/T) f4=A_4exp(-B_4/T)	k=SPECIAL_4(A_1, B_1, A_2, B_2, A_3, B_3, A_4, B_4)
10 to 12	Not used		
13	Aggregated ozone photolysis	J=J0*H2O/(H2O+M*(0.02909*exp(70/T)+0.06545*exp(110d0/T))) H2O=water vapor (molec/cm3) M=air density (molec/cm3)	J(T,Z,H2O)=photorate(O3)
14	Modified Troe	Like Type #4 (Troe) but: Z=1/(1+((Log10(M*K_0/K_∞)-0.12)/1.2)**2)	K(T,M)=troe(A_0,B_0,N_0,A_∞,B_∞,N_∞,f)
15	NO2/HONO Heterogeneous reaction	ks=0.5*depo(NO2) at level 1	ks=0.5*depo(NO2)
16 to 20	Not used		
26	Special type	k(T)=k_0(T)+k_3(T)[M]/(1+k_3(T)[M]/k_2) k_0(T)=A1exp(-B_0/T) k_2(T)=A1exp(-B_2/T) k_3(T)=A1exp(-B_3/T) M=air-density	k(T)=saprc_2(A_0, B_0, A_2, B_2, A_3, B_3)
27	Special type	k(T)=k_1(T)+k_2(T)[M] k_1(T)=A1exp(-B_1/T) k_2(T)=A1exp(-B_2/T) M=air-density	k(T)=saprc_1(A_1, B_1, A_2, B_2)
28	Troe or Falloff for SAPRC	K=K0*f**Z/(1+M*(K0/K_∞)) K0=A0exp(-B0/T)(300/T)**N0 K_∞=A_∞exp(-B_∞/T)(300/T)**N_∞ Z=1/(1+(Log10(M*K0/K_∞)/N)**2) M=air-density	K(T,M)=troe_saprc(A_0,B_0,N_0,A_∞,B_∞,N_∞,f,N)
29	Special type	K=K_1/(1+K_2) K_1=A_1exp(-B_1/T)*M K_2=A_2exp(-B_2/T)*M	k(T)=SPECIAL_5(A_1, B_1, A_2, B_2)

11.2.1.6 The chemprep.awk program for the gaseous reactions

The scripts/chemprep.awk program is the interpreter able to convert the 'human readable' chemistry files into the CHIMERE chemistry files.
The characteristics of this program are:

- Input data: the REACTIONS ascii file
- Output data:
  1. the .species file: This file lists the gaeseous active species. This list is merged with the aerosol list in scripts/chemprep-prep-chemistry-data.sh to produce the ACTIVE_SPECIES file.
  2. the CHEMISTRY file: This is the original REACTIONS but with only the reactants and products and the code number of the reaction.
  3. the STOICHIOMETRY file: Same as CHEMISTRY, but for the stoechiometric coefficients values
  4. The REACTION_RATES file: Same as CHEMISTRY, but for the reactions rates
  5. The FJX_j2j.dat file: Information for the FastJX model for photolysis
  6. The FAMILIES file: List of species in families.

If the user wants to add a new reaction type, it is necessary to define a new syntax (as explained in Table 11.3) and to add a piece of code in this program. The way is to write the test to recognize the reaction type. For example and for the reaction reate #3, corresponding to the "complete Arrhenius" rate, the code is:

```
# Type 3: Complete Arrhenius
$2 ~ /k\(T\)=Aexp\(-B\/T\)\(300\/T\)\*\*N,/ {
 split($2,w,/(=|,)/)
 printf(nofreac" 3 "w[4]" "w[6]" "w[8]"\n") >> "REACTION_RATES"
}
```

11.2.1.7 The ACTIVE_SPECIES and OUTPUT_SPECIES data files

The output files are directly in the model to know the list of chemically active species (including the gases and aerosols) and the species to write in the out.[label].nc results file.
The ACTIVE_SPECIES contains the following informations:

```
1 APINEN 2 1
2 AnA1D 2 1
3 AnA1DAQ 2 1
...
207 p10ISOPA1 2 1
208 p10H2SO4 2 1
209 p10HNO3 2 1
210 p10NH3 2 1
211 p10WATER 2 1
```

where the beginning of the file lists the gaseous species and the end lists the aerosols species. The model considering that one bin of one aerosol is a specific species to transport, all aerosols are defined with the syntax: p<number of the bin><species name>.
This file is read by the program src/inichem.F90 and the four columns correspond to:

1. The number of the species
2. The name of the species
3. The number of the horizontal transport scheme to use for this species (the flag iadv)
4. The number of the vertical transport scheme to use for this species (the flag iadvv)

For the definition of the species to write as output at the end of a simulation, two data files are created:

- OUTPUT_SPECIES.full: contains all chemical species used in the model for the simulation
- OUTPUT_SPECIES.low: contains a reduced list of the chemical species (not all gaseous species and not the binned aerosols)

The choice between these two options is made in the chimere.par file with the option:

```
[$accur]    Output species detailed (low/full)             : full
```

The syntax of the OUTPUT_SPECIES files is:

```
O3      0.
NO2     0.
NO      0.
PAN     0.
...
p01BCAR 100.
p01DUST 100.
p01OCAR 100.
p01PPM 100.
p01SALT 58.5
p01AnA1D 168.
p01AnBmP 152.
...
```

with:

1. The first column is the model species name
2. The second column is the molar mass of the species
   - For gases:
     - ◇ with 0, the result in output is expressed in ppb
     - ◇ with a prescribed molar mass value, the result is automatically converted and written in μg m^{-3}
   - For aerosols, the results are always in μg m^{-3}.
     - ◇ If the molar mass is known (for inorganic species), its value is used,
     - ◇ else a fake value of 100 is used.

11.2.1.8 The FAMILY data file

There is a possibility of defining families. These are groups of species whose concentration sum can be output in the model. However these families are only used in outputs, not in the chemistry itself.

- The **gaseous species families** have to be manually defined in the chemprep/data/FAMILIES* files, corresponding to the chemical mechanism used.
- The **aerosol species families** are automatically calculated by the src/chemprep-distrib.F90 program. The program is designed to produce families for:
  - ◇ All aerosols in order to have one species for the whole size distribution. For example, for a simulation with ten bins:

    ```
    10 pDUST p01DUST p02DUST p03DUST p04DUST p05DUST p06DUST
     p07DUST p08DUST p09DUST p10DUST
    ```

    where pDUST is the sum of all p*DUST species (one concentration per bin).
  - ◇ PM25: all aerosols with a mean mass median diameter less than 2.5 μm
  - ◇ PM10: all aerosols with a mean mass median diameter less than 10 μm
  - ◇ The relative parts of anthropogenic and biogenic PM25 and PM10, called PM25ant, PM25bio, PM10ant and PM25bio.

Many families are already defined, corresponding to the most often use in chemical composition studies. For example, the parameter file FAMILIES.gasp.melchior2 contains the following families for the gaseous species:

```
TOTPAN=PAN+toPAN
NOX=NO+NO2
OX=NO2+O3
HOX=OH+HO2+CH3O2+CH3COO+oRO2+oPAN+oRN1+obio
NOY=NO+NO2+NO3+N2O5+N2O5+HNO3+PAN+oPAN+oRN1+ISNI+CARNIT+HONO
ROOH=CH3O2H+obioH+PPA+oROOH
HCNM=C2H6+C2H6+NC4H10+NC4H10+NC4H10+NC4H10+C2H4+C2H4+C3H6+C3H6
    +C3H6+OXYL+OXYL+OXYL+OXYL+OXYL+OXYL+OXYL+OXYL+C5H8+C5H8+C5H8+C5H8+C5H8
    +APINEN+APINEN+APINEN+APINEN+APINEN+APINEN+APINEN+APINEN+APINEN
```

Other families are automatically defined for aerosols, namely PM25 and PM10. For example, with 10 bins, there are 48 aerosol sections to cumulate to finally have the concentration of the anthropogenic part of PM25.

	N= 3	N= 4	N= 5	N= 6	N= 7	N= 8	N= 9	N= 10	N= 11	N= 12	N= 13	N= 14	N= 15	N= 16
1	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
2	2.50	0.16	0.06	0.04	0.03	0.03	0.03	0.02	0.02	0.02	0.02	0.02	0.02	0.02
3	10.00	2.50	0.40	0.16	0.09	0.09	0.06	0.05	0.05	0.04	0.03	0.03	0.03	0.03
4	40.00	10.00	2.50	0.63	0.27	0.27	0.16	0.11	0.11	0.08	0.06	0.06	0.05	0.05
5	/	40.00	10.00	2.50	0.83	0.83	0.40	0.23	0.23	0.16	0.12	0.12	0.09	0.07
6	/	/	40.00	10.00	2.50	2.50	1.00	0.52	0.52	0.32	0.21	0.21	0.16	0.12
7	/	/	/	40.00	10.00	5.00	2.50	1.14	1.14	0.63	0.40	0.40	0.27	0.20
8	/	/	/	/	40.00	10.00	5.00	2.50	2.50	1.25	0.73	0.73	0.48	0.34
9	/	/	/	/	/	40.00	10.00	5.00	5.00	2.50	1.35	1.35	0.83	0.55
10	/	/	/	/	/	/	40.00	10.00	10.00	5.00	2.50	2.50	1.44	0.92
11	/	/	/	/	/	/	/	40.00	20.00	10.00	5.00	3.97	2.50	1.51
12	/	/	/	/	/	/	/	/	40.00	20.00	10.00	6.30	3.97	2.50
13	/	/	/	/	/	/	/	/	/	40.00	20.00	10.00	6.30	3.97
14	/	/	/	/	/	/	/	/	/	/	40.00	20.00	10.00	6.30
15	/	/	/	/	/	/	/	/	/	/	/	40.00	20.00	10.00
16	/	/	/	/	/	/	/	/	/	/	/	/	40.00	20.00
17	/	/	/	/	/	/	/	/	/	/	/	/	/	40.00

Table 11.4: *Values of the diameter bins obtained for $D_{min} = 0.01\,\mu$m, $D_{max} = 40\,\mu$m, and twelve different values of N.*

```
48 PM25ant p01BCAR p02BCAR p03BCAR p04BCAR p05BCAR p06BCAR p01OCAR p02OCAR
   p03OCAR p04OCAR p05OCAR p06OCAR p01PPM p02PPM p03PPM p04PPM p05PPM p06PPM
   p01AnA1D p02AnA1D p03AnA1D p04AnA1D p05AnA1D p06AnA1D p01AnBmP p02AnBmP
   p03AnBmP p04AnBmP p05AnBmP p06AnBmP p01H2SO4 p02H2SO4 p03H2SO4
   p04H2SO4 p05H2SO4 p06H2SO4 p01HNO3 p02HNO3 p03HNO3 p04HNO3 p05HNO3
   p06HNO3 p01NH3 p02NH3 p03NH3 p04NH3 p05NH3 p06NH3
```

In the CHIMERE output file call out.<label>.nc, the hourly three-dimensional fields of O_x and NO_x, for example, are directly available. For the aerosols, the results are then written for each bin (i.e p01DUST p02DUST p03DUST...) and for the total mass of this aerosols pDUST.

11.3 The aerosols size distribution calculations

11.3.1 Discretization of the size distribution

The CHIMERE model accounts for the size distribution of the aerosols using a size-bin approach: the aerosol particles for each of the model species are distributed in N size bins, covering a diameter range from D_{min} to D_{max}. Given these three user-defined parameters, a preprocessor compute a sequence $(d_i)_{i=1,N+1}$ of cutoff diameters that meets the following requirements:

- $2.5\,\mu$m and $10\,\mu$m are retained as cutoff diameters: two indices $i1$ and $i2$ such that $d_{i1} = 2.5\,\mu$m and $d_{i2} = 10\,\mu$m must exist
- The sequence of the cutoff diameters covers exactly the size interval requested by the user: $d_1 = D_{min}$ and $d_{N+1} = D_{max}$

The first requirement is set to allow a meaningful evaluation of $PM_{2.5}$ and PM_{10} in the model, since these quantities are typically available from routine measurements.
The default (and recommended) values of the extreme diameters are $D_{min} = 0.01\,\mu$m and $D_{max} = 40\,\mu$m. Using these values, the produced size distributions for various values of the number of intervals N are shown in Tab. 11.4 according to the requested number of bins, N. If $N \geq 12$, then the ratio of two successive cut-off diameters is always such as $d_{i+1}/d_i \leq 2$: all particles within a single size bin have comparable diameters at least within a factor 2, which is a good way to ensure that all the size-depending processes affecting the aerosols (sedimentation, coalescence etc.) are treated in a realistic way. However, when calculation speed is a critical requirement, for example for operational prevision, the number of size bins could be lowered to $N = 6$, still ensuring that $d_{i+1}/d_i \leq 4$

11.3.2 Implementation

The following calculation steps are performed in the src/chemprep-distrib.F90 program:

1. The size distribution depending on the distribution range and the number of bins, $nbins$
2. The log-normal distribution to project the three emitted modes into the $nbins$ bins

This program is launched by the scripts/chemprep-prep-chemistry-data.sh and some parameters are available in the script: the user can change some values but it is strongly recommended to keep the values as it. These values are:

```
# cut1 and cut2 forced the size distribution
cut1=2.5
cut2=10.

# dmin and dmax (micrometers) are used as tuner parameter to adjust the aerosol
# distribution (they are not the min and max cut off diameter !!)
dmin=0.01
dmax=40.
```

where:

- `cut1` and `cut2` ensures to have a size distribution with an exact cut-off for $PM_{2.5}$ and PM_{10}.
- `dmin=0.05` and `dmax=40` are the boundaries of the size distribution, in μm (and not the first and last mean mass median diameters).

This program has the following specifications:

1. **Input:** the MAKEPRI file and the options cut1, cut2, dmin and dmax
2. **Output:**
   (a) The AEROSOL file: The definition of the bins sizes and cut-off.
   (b) The PRIMARY file: The definition of the log-normal distribution to reaffect the three modes (fine, coarse and big) into the models bins.

This size distribution is used by the model with the AEROSOL file. An example is given below for a distribution of 10 bins.

```
# Number of sections D(2.5) D(10.0)
          10           6           8
 0.3906250E-07  0.7812500E-07  0.1562500E-06  0.3125000E-06
 0.6250000E-06  0.1250000E-05  0.2500000E-05  0.5000000E-05
 0.1000000E-04  0.2000000E-04  0.4000000E-04
# Name of aerosol components, properties (no blank line a the end!)
# NAME          TYPE   Molar mass     Origin [0=bio,1=anth,2=mineral]
BCAR            1      100.               1
DUST            1      100.               2
OCAR            1      100.               1
PPM             1      100.               1
SALT            1      58.5               0
AnA1D           2      168.               1
AnBmP           2      152.               1
BiA1D           2      170.               0
BiBmP           2      236.               0
ISOPA1          2      177.               0
H2SO4           3      96.                1
HNO3            3      62.                1
NH3             3      18.                1
WATER           4      18.                1
```

An example of an aerosol size distribution is presented in Figure 11.1 .

11.3.3 The log-normal distribution calculation

The second output of the src/chemprep-distrib.F90 program is the log-normal distribution coefficients to estimate the aerosols emitted fluxes from their three main modes to the model bins. This distribution is calculated as:

$$n(D_p) = \frac{N}{\sqrt{2\pi} D_p ln\sigma_c} exp\left(-\frac{(lnD_p - lnD_c)^2}{2ln^2\sigma_c}\right) \tag{11.1}$$

where:

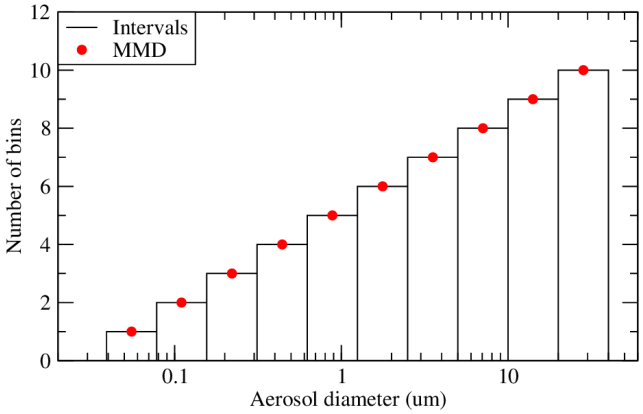


Figure 11.1: *Example of an aerosol size distribution with ten bins. The intervals represent the boundaries of the distribution (nbins+1 values) and the red dots represent the Mean Mass Median Diameter (MMD) of the distribution (nbins values).*

Aerosol	Name	Fine D_p (σ)	Coarse D_p (σ)	Big D_p (σ)
	Anthropogenic emissions			
H2SO4	Primary sulfuric acid	0.2 (2.5)		
OCAR	Primary organic carbon	0.11 (1.6)	4.0 (1.1)	
BCAR	Primary black carbon	0.11 (1.6)	4.0 (1.1)	
PPM	Primary particulate matter	0.2 (2.5)	4.0 (1.1)	
	Sea salt emissions			
SALT	Total dry sea salts		M86	
H2SO4	Sulfate fraction of sea salts		M86	
NA	Sodium fraction of sea salts		M86	
HCL	Chloride fraction of sea salts		M86	
WATER	Water fraction of sea salts		M86	
	Mineral dust emissions			
DUST	Mineral dust	1.5 (1.7)	6.7 (1.6)	14.2 (1.5)

Table 11.5: *Aerosols emissions with the three modes describing their size distribution: fine, coarse and big. The mean mass median diameter D_p is expressed in μm, σ is unitless.*

- D_p the mean mass median diameter of the CHIMERE size distribution (e.g. 10 bins)
- D_c the mean mass median diameter of the raw modes (i.e fine, coarse and big)
- σ_c the standard deviation of each D_c

For a simulation with the parameters already defined in the model, the aerosols are currently defined as in the Table 11.5 .
The results of the $n(D_p)$ values are displayed in Figure 11.2 .
These data are written as output in the PRIMARY file as:

```
11
H2SO4_fin       H2SO4           1          0.20000E-06          0.25000E+01
SALT_coa        SALT            2          0.10000E+01          0.10000E+01
DUST_fin        DUST            1          0.15000E-05          0.17000E+01
DUST_coa        DUST            1          0.67000E-05          0.16000E+01
DUST_big        DUST            1          0.14200E-04          0.15000E+01
OCAR_fin        OCAR            1          0.11000E-06          0.16000E+01
BCAR_fin        BCAR            1          0.11000E-06          0.16000E+01
PPM_fin         PPM             1          0.20000E-06          0.25000E+01
OCAR_coa        OCAR            1          0.40000E-05          0.11000E+01
BCAR_coa        BCAR            1          0.40000E-05          0.11000E+01
PPM_coa         PPM             1          0.40000E-05          0.11000E+01
```

where the first line is the number of aerosols needing a redistribution, the second block represents the aerosols.

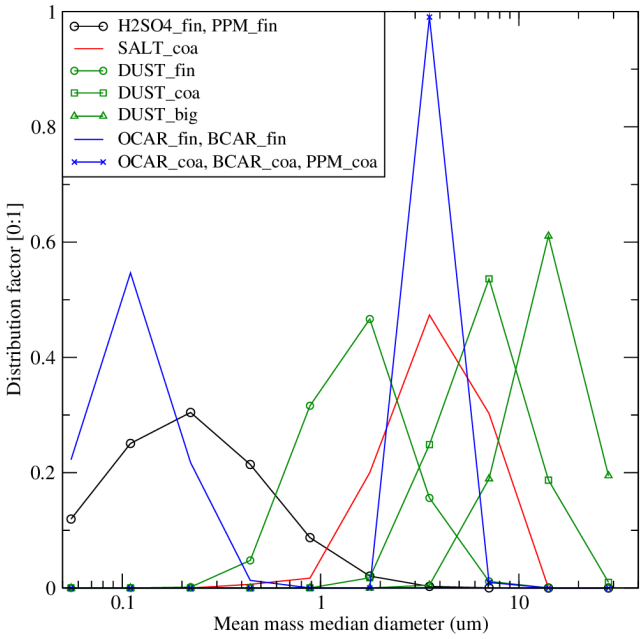


Figure 11.2: *Distribution factors used to project the three aerosol emitted modes on the CHIMERE bins size distribution.*

11.4 The preparation of tracers

11.4.1 Tracers activation and datafile

It is possible to add passive tracers in the simulation. The user has to set to one in the chimere.par file as:

```
   [trc]       Include tracers?     : 1
```

If *trc=1*, the model will read the chemprep/data/TRACERS.data ascii parameter file. It contains the following informations:

```
# Name Type    Lon.   Lat.   First date  Last date    Mass(ton)   Zmin(m)  Zmax(m)   MolMass
GTRC1   1    12.631  35.516  2009010101    2009010312   1000.       0.      2000.    18.
PTRC1   2     9.416  42.833  2009010101    2009010312   1000.       0.      1000.    100.
PTRC2   2     9.416  42.833  2009010101    2009010312   1000.    1000.      2000.    100.
```

where all informations are given line by line. There is one line per tracer. If it is necessary to emit the same tracer at the same location but at two different time periods, the user have to write the line two times. Each line must contain the following informations:

1. **Name**: The name of the tracer. In this example, there is one gaseous tracer (GTRC1) and two particulate tracers (PTRC1 and PTRC2)
2. **Type**: The tracer type: 1 = gaseous species; 2 = aerosol
3. **Lon**: The longitude of the emission (in decimal degrees)
4. **Lat**: The latitude of the emission (in decimal degrees)
5. **First date**: The starting date of the emission (format YYYYMMDDHH)
6. **Last date**: The ending date of the emission (format YYYYMMDDHH)
7. **Mass(ton)**: The mass of the emission (integrated over the whole period and the atmospheric column)
8. **Zmin(m)**: The starting altitude of the emission (in meters)

9. **Zmax(m)**: The ending altitude of the emissions (in meters)
10. **MolMass**: The molar mass of the tracer. For aerosol, a fake molar mass of 100 has to be written.

11.4.2 The chemprep-prep-tracer-data.sh script

This script reads chemprep/data/TRACERS.data and adds parameters of gaseous and particulate tracers to the data files describing parameters of the chemistry mechanism, depositions, and output. At the end the new files created are: FAMILIES.trcgas, ANTHROPIC.trc, REACTIONS.trc, OUTPUT_SPECIES.trc, AEROSOL.trc, MAKEPRI.trc, DEPO_SPEC.trc and WETD_SPEC.trc
Some parameters are available in the script:

```
# The fake aerosols molar mass in g/mol
aeromolmass=100.

# Mean mass median diameter (m) of the three aerosols modes: fine, coarse and big
d_fin=1.5e-6
d_coa=6.7e-6
d_big=14.2e-6
sig_fin=1.7
sig_coa=1.5
sig_big=1.6
```

By default, we consider that the aerosol tracers have the same size distribution than the mineral dust. This may be changed by the user. A change in these values will impact the size distribution of the tracers, then their transport and deposition.
Finally, this script creates or concatenate the necessary parameters files for the chemistry. Even if thse tracers are not chemically active, it is necessary to put their characteristics into all chemical files in order to consider they are active species in the model. To emulate this, a fake reaction is added in the CHEMISTRY file as:

```
    echo {trc_spec}"->X    k=1e-18"                >> REACTIONS.trc
```

For the dry and wet deposition of geaseous tracers, it is considered that the tracer has the same characteristics as SO2 (the tracers being often used for volcano transport studies).

11.4.3 Calculation of emissions fluxes

The emissions fluxes are hourly calculated in the program src/prep_temis.F90. The tabulated mass is linearly splitted between the several emissions time period and over the vertical interval. The three modes are projected into the model size distribution bins considering the following percentages of mass: 30% in the fine mode, 40% in the coarse mode and 30% in the big mode. This may be changed in the model code with the following lines:

```
  mdistrib(1)=0.3d0
  mdistrib(2)=0.4d0
  mdistrib(3)=0.3d0
```

In previous CHIMERE versions, the tracers were considered as anthropogenic species for the numerical management (list of species, calculation of emitted fluxes). In the current version, the tracers are completely independent and can be use alone in the model.

12 Gas-phase chemistry

12.1 Option activation

CHIMERE offers the option to include different gas phase chemical mechanisms. The choice of the mechanism is managed in the chimere.par file as:

```
[mecachim]    Chemistry mechanism (0..3)          : 3
```

with the following possibilities:

1. mecachim=0: no gas phase chemistry
2. mecachim=1: The complete MELCHIOR mechanism called MELCHIOR1
3. mecachim=2: The reduced MELCHIOR mechanism called MELCHIOR2
4. mecachim=3: The SAPRC-07-A mechanism

- **MELCHIOR1**:
  The original, complete MELCHIOR scheme ([Lattuati, 1997]) describes more than 300 reactions of 80 gaseous species. The hydrocarbon degradation is fairly similar to the EMEP gas phase mechanism ([Simpson, 1992]). Adaptations are made in particular for low NO_x conditions and NO_x-nitrate chemistry. All rate constants are updated according to [Atkinson et al., 1997] and [De Moore et al., 1994]. Heterogeneous formation of HONO from deposition of NO2 on wet surfaces is now considered, using the formulation of [Aumont et al., 2003].
- **MELCHIOR2**:
  In order to reduce the computing time a reduced mechanism, **MELCHIOR2**, with less than species and about 120 reactions is derived from MELCHIOR ([Derognat et al., 2003]), following the concept of "chemical operators" ([Carter, 1990]).
- **SAPRC**:
  The gas phase chemical mechanism SAPRC-07-A is also available, [Carter, 2010] [1]. It describes more than 275 reactions of 85 species and it is the most recent mechanism proposed in CHIMERE.

CHIMERE offers the option to include DMS chemistry which it is not included in the chemical schemes described above. This choice is managed in the chimere.par file as:

```
[iusedms]    Use DMS marine emissions? ([1-4]/0)          : 1
```

12.2 The MELCHIOR mechanims

12.2.1 Species list of the reduced MELCHIOR1 gas-phase chemical mechanism

Species	Definition
Inorganic compounds	
O3	ozone
H2O2	hydrogen peroxide
OH	hydroxy radical
HO2	hydroperoxy radical
NO	nitric oxide
NO2	nitrogen dioxide
NO3	nitrogen trioxide
N2O5	dinitrogen pentoxide
HONO	nitrous acide
HNO3	nitric acide
HNO4	pernitric acide
CO	carbon monoxide
SO2	sulfur dioxide

[1] http://www.engr.ucr.edu/~carter/SAPRC/

[2] Hydrocarbon model species (except for CH4 and C2H4) represent groups of hydrocarbons with similar reactivity. As “definition” a typical repre-

H2SO4	sulfuric acid
Hydrocarbon species[2]	
CH4	methane
C2H6	ethane
NC4H10	n-butane
C2H4	ethene
C3H6	propene
OXYL	o-xylene
C5H8	isoprene
APINEN	α-pinene
BPINEN	β-pinene
LIMONE	limonene
TERPEN	terpenes (lumped class)
HUMULE	Humulene (lumped class)
OCIMEN	Ocimene (lumped class)
Alcohol species	
C2H5OH	ethanol
CH3OH	methanol
Carbonyls	
HCHO	formaldehyde
CH3CHO	acetaldehyde
CH3COE	methyl ethyl ketone
CH3COY	dimethyl glyoxal
MEMALD	4-oxo-2-pentenal
MAC	methacroleine
MVK	methyl vinyl ketone
GLYOX	glyoxal
MGLYOX	methyl glyoxal
OFURAN	5-methyle-3H-furan-2-one
Organic nitrates	
CH3NO3	methyl nitrate
ETNO3	ehtyl nitrate
BUNO3	n-butyl-2-nitrate
ISNI	unsaturated nitrate from RO2P1
CARNIT	nitrate carbonyl taken as α-nitrooxy acetone
ORNIT1	α-hydroperox nitrate
ORNIT2	α-hydroperoxy nitrate from C5H8 and APINEN
(Per)oxy radicals	
C2H5O2	perxoy ethyl
CH2O2C	ethyl hydroperoxy
CH3O2	methyl peroxy
CH3COO	peroxy acetyl radical
CH3CHX	propyl hydroperoxy
CH3COX	peroxy from butan-2-one
CH3COZ	peroxy from MACO2+NO reaction
SECBUO	2-oxy butyl
SECC4H	peroxyl butyl
OXYL1	peroxy radical from OXYL
RO2IP1	peroxy radical from C5H8+OH reaction
MEMAL1	adition of OH on MEMALD
MEMAL2	formed from the abstraction of an hydrogen in alpha position
MACO2	peroxy radical from MAC+OH reaction
MVKO2	peroxy radical from MVK
APIOH	Peroxy radical from APINEN+OH reaction
Peroxy nitrates	
RNC2H4	peroxy ethyl nitrate
RNC3H6	peroxy propyl nitrate
RNC5H8	peroxy from C5H8+NO3 reaction
ROXYL1	peroxy from OXYL+NO3 reaction
ISNIR	peroxy from ISNI+OH reaction
APINO3	Peroxy radical from APINEN+NO3 reaction
Pernitrates	
PAN	peroxyacetyl nitrate
MACPAN	peroxy acyl nitrate from MAC
CH3NO4	methyl pernitrate
MEMPAN	peroxy acyl nitrate from MEMAL2
Peroxide	
APIO2H	hydroperoxyde from APINENOH+HO2 reaction
BUO2H	Butyl hydroperoxyde
CH3O2C	hydroxyethyl hydroperoxide
CH3O2H	methyl hydroperoxide
CH4CHX	hydroxypropyl hydroperoxide
CH4COX	butan-2-one-3-hydroperoxide
CH4COZ	hydroperoxide from CH3COZ+HO2 reaction
ETO2H	ethyl hydroperoxide
MACO2H	hydroperoxy from MACO2+HO2 reaction
MEMAH2	hydroperoxy from MEMAL2+HO2 reaction
MEMALH	hydroperoxy from MEMAL1+HO2 reaction
MVKO2H	hydroperoxy from MVKO2+HO2 reaction
OXYLH1	hydroperoxy from OXYL1+HO2 reaction
PEROX	hydroperoxy from RO2P1+HO2 reaction
PPA	peroxy acetyl acid

The reaction list is described in Appendix G.1, p.291

12.2.2 Species list of the reduced MELCHIOR2 gas-phase chemical mechanism

Species	**Definition**
Inorganic compounds	
O3	ozone
H2O2	hydrogen peroxide
OH	hydroxy radical
HO2	hydroperoxy radical
NO	nitric oxide
NO2	nitrogen dioxide
NO3	nitrogen trioxide
N2O5	dinitrogen pentoxide
HONO	nitrous acide
HNO3	nitric acide
CO	carbon monoxide
SO2	sulfur dioxide
H2SO4	sulfuric acid
Hydrocarbon species[3]	
CH4	methane
C2H6	ethane
NC4H10	n-butane
C2H4	ethene
C3H6	propene
OXYL	o-xylene
C5H8	isoprene
APINEN	α-pinene
BPINEN	β-pinene
LIMONE	limonene
TERPEN	terpenes (lumped class)
HUMULE	Humulene (lumped class)
OCIMEN	Ocimene (lumped class)
Carbonyls	

sentative of this group is given. This is also true for the degradation products.

HCHO ..... formaldehyde
CH3CHO ..... acetaldehyde
CH3COE ..... methyl ethyl ketone
GLYOX ..... glyoxal
MGLYOX ..... methyl glyoxal
CH3COY ..... dimethyl glyoxal
MEMALD ..... unsaturated dicarbonyls, reacting like 4-oxo-2-pentenal
MVK ..... methyl vinyl ketone
MAC ..... methacroleine

Organic nitrates

PAN ..... peroxyacetyl nitrate
CARNIT ..... nitrate carbonyl taken as α-nitrooxy acetone
ISNI ..... unsaturated nitrate from isoprene degradation

Organic peroxides

CH3O2H ..... methyl hydroperoxide
PPA ..... peroxy acetyl acid

(Per)oxy radicals

CH3O2 ..... methyl peroxy radical
CH3COO ..... peroxy acetyl radical

Operators [4]

oRO2 ..representing peroxy radicals from OH attack to C2H5, NCHH10, C2H4, C3H6, OXYL, CH3COE, MEMALD, and MVK
oROOH ..... representing organic peroxides from oRO2+HO2 reactions
obio representing peroxy radicals produced by C5H8 and APINEN + OH reaction
obioH ..... representing biogenic organic peroxides from obio+HO2 and obio+obio reactions
oPAN ..... representing PAN homologue compounds (except PAN)
PANH ..... representing results from oPAN+HO2 reaction
toPAN ..... representing results from oPAN+NO2 reaction
oRN1 ..... representing organic nitrate peroxy radicals from NO3 attack to C2H4, C3H6, C5H8, APINEN, BPINEN, LIMONE, TERPEN, OCIMEN, HUMULE and OH attack to ISNI

The reaction list is described in Appendix G.2, p.298

12.3 The SAPRC mechanims

12.3.1 Species list of the SAPRC-07-A gas-phase chemical mechanism

Species ..... **Definition**

Inorganic compounds

O3 ..... ozone
H2O2 ..... hydrogen peroxide
OH ..... hydroxy radical
HO2 ..... hydroperoxy radical
NO ..... nitric oxide
NO2 ..... nitrogen dioxide
NO3 ..... nitrogen trioxide
N2O5 ..... dinitrogen pentoxide
HONO ..... nitrous acide
HNO3 ..... nitric acide
CO ..... carbon monoxide
SO2 ..... sulfur dioxide

Hydrocarbon species

CH4 ..... Methane
C2H4 ..... Ethen
ACYE ..... Acetylen
BENZ ..... Benzen
C5H8 ..... Isoprene
APINEN ..... α-pinene
BPINEN ..... β-pinene
LIMONE ..... limonene

lumped class

TERPEN ..... terpenes
HUMULE ..... Humulene
OCIMEN ..... Ocimene
ALK1 ..... Ethane
ALK2 ..... Propane
ALK3 ..... Alkans 2.5x10^3 <k_{OH}< 5x10^3
ALK4 ..... Alkans 5x10^3<k_{OH}<1x10^4
ALK5 ..... Alkans 1x10^4<k_{OH}
ARO1 ..... Aromatiques k_{OH}<2x10^4
ARO2 ..... Aromatiques k_{OH}>2x10^4
OLE1 ..... Alkens k_{OH}<7x10^4
OLE2 ..... Alkens k_{OH}>7x10^3
AFG1 ..... Lumped photoreactive monounsaturated dicarbonyl aromatic fragmentation products that photolyze to form radicals
AFG2 Lumped photoreactive monounsaturated dicarbonyl aromatic fragmentation products that photolyze to form non-radical products
AFG3 Lumped diunsaturatred dicarbonyl aromatic fragmentation product
RNO3 ..... Lumped Organic Nitrates

PAN and PAN analogues

PAN ..... Peroxy Acetyl Nitrate
PAN2 ..... PPN and other higher alkyl PAN analogues
PBZN ..... PAN analogues formed from Aromatic Aldehydes
MAPAN ..... PAN analogues formed from Methacrolein

Carbonyls

HCHO ..... Formaldehyde
MEK ..... Ketones and other non-aldehyde oxygenated products that react with OH radicals faster than 5x10^{-13} but slower than 5x10^{-12} cm^3.molec^{-2}.s^{-1}
GLY ..... Glyoxal
MACR ..... Methacrolein
MVK ..... Methyl Vinyl Ketone
CCHO ..... Acetaldehyde
RCHO ..... Lumped C3+ Aldehydes
ACET ..... Acetone
BACL ..... Biacetyl
BALD ..... Aromatic aldehydes
CRES ..... Phenols and Cresols
PRD2 ..... Ketones (k_{OH}>0.7x10^4 ppm^{-1}.mn^{-1})
NPHE ..... Nitrophenols
IPRD ..... Lumped isoprene product species
MEOH ..... Methanol

Organic peroxides

XOOH ...Lumped organic hydroperoxide structure, used to represent the effect of photolysis of the hydroperoxide group in the SAPRC-99 peroxy radical representation (Carter, 2000).
COOH ..... Methyl Hydroperoxide
FACD ..... Formic Acid
PACD ..... Higher organic acids

[4] For the concept of "chemical operators" see Carter (1990) and Aumont et al. (1997)

AACD ..... Acetic Acid

(Per)oxy radicals

TBUO ..... t-Butoxy Radicals
RCO3 ..... Peroxy Propionyl and higher peroxy acyl Radicals
MEO2 ..... Methyl Peroxy Radicals
MECO3 ..... Acetyl Peroxy Radicals
MACO3 ..... Peroxyacyl radicals formed from methacrolein and other acroleins
BZCO3 ..... Peroxyacyl radical formed from Aromatic Aldehydes
BZO ..... Phenoxy Radicals

Operators [1]

RO2N ..... Peroxy Radical Operator representing NO consumption with organic nitrate formation
RO2R Peroxy Radical Operator representing NO to NO2 conversion with HO2 formation
R2O2 ..... Peroxy Radical Operator representing NO to NO_2 conversion without HO_2 formation

The reaction list is described in Appendix G.3, p.301

12.3.2 Species list of the chlorine SAPRC-07 gas-phase chemical mechanism

Species ..... **Definition**

Active Inorganic Species

CL2 ..... Chlorine molecules
CLNO ..... ClNO
CLONO ..... ClONO
CLNO2 ..... ClNO2
CLONO2 ..... ClONO2
HOCL ..... HOCl

Active Radical Species and Operators

CL ..... Chlorine atoms
CLO ..... ClO. Radicals

Steady state operators

xCL .... Formation of Cl radicals from alkoxy radicals formed in peroxy radical reactions with NO and NO3 (100% yields) and RO2 (50% yields)
xCLCCHO ..... As above, but for CLCCHO
xCLACET ..... As above, but for CLACET

Active Organic Product Species

CLCCHO Chloroacetaldehyde (and other alpha-chloro aldehydes that are assumed to be similarly photoreactive)
CLACET ..... Chloroacetone (and other alpha-chloro ketones that are assumed to be similarly photoreactive)

Low Reactivity Compounds Represented as Unreactive

HCL ..... Hydrochloric acid
CLCHO ..... Formyl Chloride (assumed to be unreactive)

The reaction list is described in Appendix G.4, p.308

12.4 The DMS chemical scheme

The complete chemistry of DMS is extremely complicated, therefore a reduced version of this chemistry has to be used in CHIMERE in order to express the fate of DMS emissions in the atmosphere and their contribution to SO_2 and sulfate. For this purpose, the chemical scheme suggested by [Mihalopoulos et al., 2007] has been used here, which contains oxidation reaction of DMS with OH by the means of H abstraction, its oxidation by OH addition and its reaction with NO_3. It includes DMSO (Dimethyl sulfoxide) and MSIA (Methanesulfinic acid) as intermediate gas-phase chemistry. While the aqueous reactions are also available in this scheme, they were not implemented in CHIMERE, since their importance compared to the gas-phase DMS chemistry is small. The reactions added to CHIMERE for this purpose are presented below.

Reactions	kinetic rates ($molec.cm^{-3}.s^{-1}$)
DMS + OH → 0.997(SO2 + CH3O2 + HCHO) + 0.003(MSA+HCHO)	k(T)=Aexp(-B/T),A=1.13e-11,B=253
DMS + OH → DMSO	k(T)=SPECIAL_5(1.7e-42,-7810.,5.5e-31,7460)
DMS + NO3 → SO2 + HNO3 + CH3O2 + HCHO	k(T)=Aexp(-B/T),A=1.9e-13,B=-500
DMSO + OH → MSIA + CH3O2	k=9e-11
MSIA + OH → CH3O2 + SO2 + H2O	k=9e-11

[1] For the concept of "chemical operators" see Carter (1990) and Aumont et al. (1997)

13 Aerosol chemistry

13.1 The aerosols species

CHIMERE includes a module to calculate aerosol chemistry. It is optional and, to use it, the following flags have to be selected in the chimere.par file as:

```
[aero]      Chemically-active aerosols (1/0)   : 1
[nbins]     Number of aerosol size sections    : 10
```

Seven aerosol species are defined and are listed in Table 13.1 :

Model species	Species	Type
pPPM	Anthropogenic primary species EC, OC, and other industrial dusts	primary
pSOA	Anthropogenic and Biogenic secondary organic aerosol (ASOA+BSOA)	secondary
pH2SO4	Equivalent Sulfate(*)	secondary
pHNO3	Equivalent Nitrate(*)	secondary
pNH3	Equivalent Ammonium(*)	primary emitted, secondary transferred
pWATER	Water	

Table 13.1: *List of aerosol species. (*): ions, molecules, crystals. Potentially, Chloride et Sodium can be included (high computing time).*

In the model and in the output files, the table notation is: pAERO for the aerosol AERO in mass concentration and for the whole size distribution and bAERO for the same mass concentration but in each bin.
Sulfate is formed through gaseous and aqueous oxidation of SO_2. Nitric acid is produced in the gas phase by NOx oxidation. N_2O_5 is converted into nitric acid via heterogeneous pathways by oxidation on aqueous aerosols. Ammonia is a primary emitted base converted in the aerosol phase by neutralization with nitric and sulfuric acids. Ammonia, nitrate and sulfate exist in aqueous, gaseous and particulate phases in the model. As an example, in the particulate phase the model species pNH_3 represents an equivalent ammonium as the sum of NH_4^+ ion, NH_3 liquid, NH_4NO_3 solid, etc.

13.2 Main processes

The aerosol life cycle is affected by nucleation, coagulation and absorption. These processes are applied to each aerosol and its own size distribution.

13.2.1 Nucleation

The parameterization of [Kulmala et al., 1998] for sulfuric acid nucleation is used. This process, favoured by cold humid atmospheric conditions, affects the number of ultrafine particles. The nucleated flux is added to the smallest bin in the sectional distribution. Nucleation of condensable organic species has been clearly identified in many experimental studies ([Kavouras et al., 1998], there is no available parameterization. Since, the sulfuric acid nucleation process competes with absorption processes, it is expected to occur in weakly particle polluted conditions.

13.2.2 Coagulation

The flux of coagulation $J^{bin}_{coag,i}$ a compound i inside a bin b is computed with the size binning method:

$$J^b_{coag,i} = \sum_{j=1}^{b}\sum_{k=1}^{b} f^b_{j,k} K_{j,l} A^j_{p,i} Number^k - A^b_{p,i} \sum K_{bin,j} Number^k \tag{13.1}$$

with $K_{j,l}$ the coagulation kernel coefficient between bins i and j and $f^b_{j,k}$ the partition coefficient (the fraction of the particle created from the coagulation of bins j and k which is redistributed inside bin b). The coagulation kernel and the partition coefficient are calculated as in [Debry et al., 2007].

13.2.3 Absorption

Absorption is described by the "bulk equilibrium" approach of [Pandis et al., 1993]. In this approach, all the bins for which condensation is very fast are merged into a "bulk particulate phase". Following [Debry et al., 2007], a cutting diameter of 1.25 μm is used to separate bins which are inside the "bulk particle" (with a diameter lower than the cutting diameter). Thermodynamic models are used to compute the partitioning between the gas phase and the particle phase and estimate the gas-phase concentrations at equilibrium.
For semi-volatile inorganic species (sulfate, nitrate, ammonium), the equilibrium concentration G_{eq} is calculated using the thermodynamic module ISORROPIA ([Nenes et al., 1998]). This model also determines the water content of particles. Interactions between inorganic and organic species are not taken into account, the thermodynamic of such mixtures still being poorly understood. Equilibrium concentrations for the semivolatile organic species k are related to particle concentrations through a temperature dependent partition coefficient K^p (in $m^3 \mu g^{-1}$) ([Pankow, 1994]):

$$G^k_{l,eq} = \frac{Q^k_l}{OM_l K^p_k} \tag{13.2}$$

with OM ($\mu g m^{-3}$) is the absorbent organic material concentration. Considering the thermodynamic equilibrium between the gas and particulate phases, this coefficient is given by

$$K^p_k = \frac{10^{-6} RT}{MW_{om} \zeta_k p^0_k} \tag{13.3}$$

with R is the ideal gas constant (8.206 10^{-5}m3 atm $mol^{-1}K^{-1}$), T the temperature (K), MW_{om} the mean molecular weight (g mol^{-1}), p^0_i the vapor pressure of product i as a pure liquid (atm) and ζ the activity coefficient of species in the bulk aerosol phase. The coefficient ζ, difficult to calculate, is assumed constant and equal to unity.
The mass of compounds condensing into particles ΔA_p is redistributed over bins according to the kinetic of condensation into each bins whereas the mass of compounds evaporating from each bins is proportional to the amount of the compounds in the bins.
If the variation of particulate bulk concentration of compound i $\Delta A_{p,i} > 0$:

$$\Delta A^{bin}_{p,i} = \frac{k^{bin}}{\sum_j k^j_i} DeltaA_{p,i} \tag{13.4}$$

with k^{bin}_i the kinetic of condensation given by [Seinfeld and Pandis, 1998]:

$$k^{bin}_i = Number^{bin} \frac{2\pi D^{bin}_p D_i M_i}{RT} f(Kn, \alpha) \tag{13.5}$$

with $Number^{bin}$ the number of particles inside the bin, D^{bin}_p the mean diameter of the bin, D_i the diffusion coefficient for species i in air, M_i its molecular weight and f(Kn,α) is the correction dut to noncontinuum effects and imperfect surface accomodation.
If the variation of particulate bulk concentration of compound i $\Delta A_{p,i} < 0$:

$$\Delta A^{bin}_{p,i} = \frac{A^{bin}_{p,i}}{\sum_j A^j_{p,i}} DeltaA_{p,i} \tag{13.6}$$

The absorption flux J ($\mu g m^{-3} s^{-1}$) of a semi-volatile inorganic or organic species onto a bin computed with:

$$J = \frac{1}{\tau} \Delta A_{p,i}^{bin} \tag{13.7}$$

with τ the time to reach equilibrium. A time of 2 hours is used based on the results of [Aan de Brugh et al., 2012].
If the variation of particulate bulk concentration of compound i $\Delta A_{p,i} > 0$:

$$\Delta A_{p,i}^{bin} = \frac{k^{bin}}{\sum_j k_i^j} DeltaA_{p,i} \tag{13.8}$$

with k_i^{bin} the kinetic of condensation given by [Seinfeld and Pandis, 1998]:

$$k_i^{bin} = Number^{bin} \frac{2\pi D_p^{bin} D_i M_i}{RT} f(Kn, \alpha) \tag{13.9}$$

with $Number^{bin}$ the number of particles inside the bin, D_p^{bin} the mean diameter of the bin, D_i the diffusion coefficient for species i in air, M_i its molecular weight and f(Kn,α) is the correction dut to noncontinuum effects and imperfect surface accomodation.
If the variation of particulate bulk concentration of compound i $\Delta A_{p,i} < 0$:

$$\Delta A_{p,i}^{bin} = \frac{A_{p,i}^{bin}}{\sum_j A_{p,i}^j} DeltaA_{p,i} \tag{13.10}$$

The absorption flux J ($\mu g m^{-3} s^{-1}$) of a semi-volatile inorganic or organic species onto a bin computed with:

$$J = \frac{1}{\tau} \Delta A_{p,i}^{bin} \tag{13.11}$$

with τ the time to reach equilibrium. A time of 2 hours is used based on the results of [Aan de Brugh et al., 2012].

13.3 Thermodynamic equilibrium of inorganic compounds

The inorganic part mainly containing ammonium, sulfates and nitrates is the major part of the particulate matter. This is particularly true in the fine mode (D < 2.5 μm). The thermodynamic equilibrium model ISORROPIA [Nenes et al., 1998], is used to determine the particle/gas partitioning of semi-volatil inorganic species. The model calculates the thermodynamical equilibrium of the system: sulfate-nitrate-ammonium-sodium-chloride-water at a given temperature and relative humidity.
There is two ways to use this model: online or offline. In offline mode, CHIMERE read look-up tables already computed and proposed in the code distribution. In online mode, the calculation is performed at each chemical time-step and each cell of the domain.

```
[$nequil]    ISORROPIA coupling (1) / tabulation (0)        : 0
[$npeq]      - equilibrium calc freq (phys steps, < 5)      : 1
```

The aerosol species involved in this equilibrium are:

Gas phase :	NH_3, HNO_3, HCl, H_2O.
Liquid phase :	NH_4^+, Na^+, H^+, Cl^-, NO_3^-, SO_4^{2-}, HSO_4^-, OH^-, H_2O, $HNO_{3(aq)}$, $HCl_{(aq)}$, $NH_{3(aq)}$, $H_2SO_{4(aq)}^2$.
Solid phase :	$(NH_4)_2SO_4$, NH_4HSO_4, $(NH_4)_3H(SO_4)_2$, NH_4NO_3, NH_4Cl, NaCl, $NaNO_3$, $NaHSO_4$, Na_2SO_4

Due to its low vapor pressure, sulfuric acid is entirely transfered in the condensed phase. Also, the Sodium is located in the particulate phase. Inorganic aerosols can be a mix of solid and liquid species depending on temperature and humidity. The solid/liquid phase transition is solved by ISORROPIA by computing the deliquescent relative humidities (transition relative humidity between the phases).

If the user selects the offline mode, the pre-calculated look-up table stored in the NetCDF file AEROMIN.nc is used. Characteristics of this look-up table are explained in Table 13.2): the partitioning coefficient for the nitrates and ammonium, and the aerosol water content has been calculated for a range of temperature from 260 to 312K, for relative humidity from 0.3 to 0.99 and for concentrations from 10^{-2} to 65 $\mu.g.m^{-3}$. Because of numerical limits, Sodium and Chloride are not accounted for in this table. The active seasalt version needs to use the on-line coupling.

Variable	Value		Increment
	Min.	Max.	
Temperature (K)	260	312	2.5
Relative Humidity	0.3	0.99	0.05
H_2SO_4, HNO_3, NH_3 ($\mu.g.m^{-3}$)	10^{-2}	65	x 1,5

Table 13.2: *Characteristics of the look-up table used for the calculation of the thermodynamic equilibrium with ISORROPIA.*

The differences obtained when comparing the on-line coupling and the look-up table are small enough compare to the uncertainties in the modelling of the other processes. The look-up table of ISORROPIA is able to determine the partition of NH_4/NH_3 and HNO_3/NO_3.

13.3.1 Humid diameter and density of aerosols

In many processes, the diameter and the density of aerosols are used (deposition, absorption, coagulation, etc...). These processes have to take into account that the diameter and the density of aerosols change with humidity due to the amount of water absorbed into the particles. The amount of water in each bins is computed with the "reverse mode" of ISORROPIA by using the composition of particles. The density of the aqueous phase of particles is computed according to composition following the method of [Semmler et al., 2006]. The density of the aqueous phase and the amount of water of each of the bins are used to wet diameters.

13.4 Thermodynamic equilibrium of organic compounds

For the [Pun] mechanisms, the gas-particle partitioning formulation is computed as in [Pun et al., 2006]. The overall approach consists in differentiating between hydrophilic SOA that are most likely to dissolve into aqueous inorganic particles and hydrophobic SOA that are most likely to absorb into organic particles. The dissolution of hydrophilic SOA is governed by Henry's law whereas the absorption of hydrophobic particles is governed by Raoult's law. The large number of condensable organic compounds is represented by a set of surrogate compounds that cover the range of physico-chemical properties relevant for aerosol formation, i.e., water solubility and acid dissociation for hydrophilic compounds and saturation vapor pressure for hydrophobic compounds. These surrogate compounds were selected by grouping identified particulate-phase molecular products with similar properties. The molecular weight of each surrogate compound is determined based on its structure and functional groups. The Henry's law constant or the saturation vapor pressure of the surrogate species is derived from the average properties of the group. Other properties are estimated using the structure of each surrogate compound.
For the H^2O mechanism, the gas-particle partitioning is computed with the Secondary Organic Aerosol Processor (SOAP) model. This model is presented into details in [Couvidat and Sartelet, 2015]. A small presentation of SOAP is done in section 13.6.3.2.
The absorption process in CHIMERE is implemented as in [Bowman et al., 1997]. A dynamical approach is adopted to describe the gas/particle conversion. A typical time-scale for mass transfer being more than 20 mn, the calculation is performed at the chemical time-step ($\approx$ 5 mn) in the model.

$$J_i = \frac{1}{\tau_i}(G_i - G_i^{eq}) \tag{13.12}$$

J_i ($\mu g.m^{-3}.s^{-1}$) is the absorption or desorption flux of species i, τ_i (s) is a characteristic time of the mass transfer that depends on particle size and the chemical properties of species i, G_i is the bulk gas-phase concentration of species i and G_i^{eq} is the gas-phase concentration of species i at thermodynamic equilibrium (i.e., at the surface of the particle). The

equilibrium gas-phase concentrations are functions of the particle chemical composition, temperature and, for hydrophilic species, relative humidity, as described by [Pun et al., 2006].

13.5 Multiphase chemistry

13.5.1 Sulfur aqueous chemistry

If the aerosol option of the model is selected, Aqueous sulphate chemistry is considered; Sulfate is produced in the following aqueous reactions ([Berge, 1993]; [Hoffmann and Calvert, 1985]; [Lee and Schwartz, 1983]):

$$\begin{aligned} SO_2^{aq} + O_3^{aq} &\rightarrow SO_4^{2-} \\ HSO_3^- + O_3^{aq} &\rightarrow SO_4^{2-} \\ SO_3^{2-} + O_3^{aq} &\rightarrow SO_4^{2-} \\ SO_2^{aq} + H_2O_2^{aq} &\rightarrow SO_4^{2-} \\ SO_2^{aq} + NO_2^{aq} &\rightarrow SO_4^{2-} \\ SO_3^{2-}(Fe^{3+}) &\rightarrow SO_4^{2-} \\ HSO_3^-(Mn^{2+}) &\rightarrow SO_4^{2-} \end{aligned} \tag{13.13}$$

SO_2, H_2O_2 and O_3 in the aqueous phase are in equilibrium with the concentrations in the gas phase. Moreover, aqueous SO_2 is dissociated into HSO_3^- and SO_3^{2-}. Catalyzed oxidation reactions of sulfur dioxide in aqueous droplets with iron and manganese are considered, following [Hoffmann and Calvert, 1985] among others. Henry's law coefficient and other aqueous equilibrium constants are used ([Seinfeld and Pandis, 1998]).
Sulfur chemistry is very pH sensitive. pH is computed according to concentrations of CO_2, SO_2, H_2SO_4, HNO_3, HCl, Na and NH_3.

13.5.2 Heterogeneous chemistry

Again with the aerosol option of the model, a few heterogeneous reactions are also considered. Nitric acid is produced onto existing particles and fog droplets. Although aerosol particles and cloud droplets occupy a very small fraction of the atmosphere, it is now well established that reactions involving gas species onto their surfaces may significantly contribute to atmospheric chemistry cycles. [Jacob, 2000] recommends for ozone model to include a minimal set of reactions with associated uptake coefficients given by [Harrison and Kito, 1990], and other references in [Jacob, 2000]:

$$\begin{aligned} HO_2 &\rightarrow 0.5H_2O_2 & \gamma &= 0.2 \\ NO_3 &\rightarrow HNO_3 & \gamma &= 0.001 \\ NO_2 &\rightarrow 0.5HNO_3 + 0.5HONO & \gamma &= 0.0001 \\ N_2O_5 &\rightarrow 2HNO_3 & \gamma &= 0.01 - 1 \end{aligned}$$

The first-order rate constant k for gas heterogeneous loss onto particles is given by:

$$k = \sum_l \left(\frac{d_l}{2D_g} + \frac{4}{V\gamma} \right)^{-1} A_l \tag{13.14}$$

with d_l, the particle diameter (m), D_g, the reacting gas molecular diffusivity (m2 s^{-1}), n, the mean molecular velocity (m s-1), A_l, the total surface area in the particle bin l and gamma the uptake coefficient of reactive species. The uptake coefficient for N_2O_5 is assumed to be temperature-dependent in the range 0.01 - 1 ([De Moore et al., 1994]) with increasing values for decreasing temperatures.
Finally, the study of [Aumont et al., 2003] suggested that NO_2 reactions on wet surfaces could be an important source for HONO production during wintertime smog episodes. An additional reaction is added in the gas-phase mechanism,.

13.6 Secondary organic aerosol (SOA)

13.6.1 Option activation

The SOA formation from Primary Organic Aerosols (POA) are taken into account if the flag carb is greater than one in the chimere.par file as:

```
[carb]   Include carboneceous species? (0...4)   : 3
```

The values of the flag carb are:

- carb=0: no carboneceous species
- carb=1: carboneceous species (OCAR and BCAR) are used.
- carb=2: carboneceous species (POA and BCAR) are used. POA (POA+IVOC+SVOC) are splitted in 9 volatility bins. Only the functionalization process is taken into account. See [Zhang et al., 2013].
- carb=3: carboneceous species (POA and BCAR) are used. POA (POA+IVOC+SVOC) are splitted in 9 volatility bins. Functionalization, fragmentation and non volatil bin are taken into account. See [Cholakian et al., 2018].

The SOA formation from VOC are taken into account if the flag soatyp is non zero in the chimere.par file as:

```
[soatyp] secondary organic aerosol scheme (0..7) : 2
```

The values of the flag soatype are:

- soatyp=0: no SOA chemistry
- soatyp=1: single-step oxydation (simple)
- soatyp=2: single-step oxydation (medium)
- soatyp=3: single-step oxydation (complex)
- soatyp=4: H^2O (Hydrophilic/Hydrophobic Organics) mechanism [Couvidat et al., 2018]
- soatyp=5: H^2O reduced mechanism [Couvidat et al., 2018]
- soatyp=6: VBS scheme [Zhang et al., 2013]
- soatyp=7: VBS scheme with fragmentation process [Cholakian et al., 2018]

All shemas used in CHIMERE take into account the biogenic VOC precursors emitted by MEGAN (10.1, p.105
Biogenic precursors include: APINEN (α-pinene and sabinene), BPINEN (β-pinene and δ^3-carene), LIMONE (limonene), OCI (myrcene and ocimene) C5H8 (isoprene).

13.6.2 The SOA schemes whith single-step oxidation

For soatyp equal 1, 2 or 3, the formation of SOA is taken into account with single-step oxidation schemes. Anthropogenic precursors include:

- TOL (benzene, toluene and other mono-substituted aromatics),
- TMB (Trimethylbenzene and other poly-substituted aromatics),
- NC4H10 (higher alkanes).

Secondary organic aerosols calculated in the model are:
The complete chemical scheme implemented in CHIMERE includes biogenic and anthropogenic precursors is described in Appendix G.9, p.315 for the complex scehme, Appendix G.8, p.314 for the medium and Appendix G.7, p.313 for the simple one.
SOA formation is represented according to a single-step oxidation of the relevant precursors and gas-particle partitioning of the condensable oxidation products.
The base SOA module was tested against the smog chamber data of [Odum et al., 1997] for anthropogenic compounds and those of [Griffin et al., 1999] for biogenic compounds and was shown to satisfactorily reproduce SOA formation for those compounds [Pun et al., 2006]. Higher alkanes and isoprene were added to the original chemical mechanism of [Pun et al., 2006]. The formation of SOA from higher alkanes follows the formulation of [Zhang et al., 2007] for the stoichiometric SOA yield and it is assumed that the SOA species can be represented by a hydrophobic surrogate compound with a moderate saturation vapor pressure. The formation of SOA from the oxidation of isoprene by hydroxyl radicals is represented with two surrogate products and follows the formulation of [Kroll et al., 2006, Zhang et al., 2007].

Model species	Type	Hydro	Enthalpy of vaporization ($kJ.mol^{-1}$)
AnA0D	anthropogenic nondissociative species	hydrophilic	88
AnA1D	anthropogenic once-dissociative species	hydrophilic	88
AnA2D	anthropogenic twice-dissociative species	hydrophilic	88
BiA0D	biogenic non dissociative species	hydrophilic	88
BiA1D	biogenic once-dissociative species	hydrophilic	88
BiA2D	biogenic twice-dissociative species	hydrophilic	109
AnBmP	anthropogenic species with moderate saturation vapor pressure	hydrophobic	88
AnBlP	anthropogenic species with low saturation vapor pressure	hydrophobic	88
BiBmP	biogenic species with moderate saturation vapor pressure	hydrophobic	175
ISOPA1, ISOPA2	isoprene oxidation products		

Table 13.3: *List and characteristics of the surrogate SOA compounds*

13.6.3 The H^{2}O mechanism

The H^{2}O mechanism development is presented into details in [Couvidat et al., 2012] and its performance was evaluated over Europe in [Couvidat et al., 2018]. This mechanism is an evolution of the [Pun] mechanism. It uses the molecular surrogate, into which semi-volatile organic compounds with similar thermodynamic properties are lumped together and are represented with a single molecular structure. By attaching a molecular structure to the surrogate structure, several processes can be represented that are often not-taken into account. For example, absorption of organic compounds onto the aqueous phase of aerosols, the acidic dissociation and non-ideality (interactions between organic compounds and with inorganics compounds) can be estimated. The model distinguishes two types of compounds: hydrophilic compounds (compounds with a high affinity with water that can condense onto the aqueous phase of particles) and hydrophobic compounds (compounds with a low affinity with water that condense onto an organic-rich phase).
The mechanism takes into account SOA formation from the oxydation of isoprene, monoterpene, sesquisterpenes and aromatic compounds. The properties of the surrogate compounds is described in the species.cxx file in the src directory and are summarized in Table 13.4 . It should be noted that some coumpounds (like AnClP or POA) do not a molecular structure attached. In that case, an hydrophobic default structure is used.
The complete meachnism (soatyp=4) has 15 species and has been reduced to 7 species (soatyp=5).
Contrary to the other SOA mechanism, equilibrium is computed with SOAP to account for non-ideality and aqueous-phase absorption. ithermo in chimere.par must be above 1.

13.6.3.1 Semi-volatile properties of POA

As for the VBS, POA can be considered semi-volatile within H^{2}O with the option carb equal to 2 or 3 in chimere.par. With carb equal to 2, POA are split into three compounds POAlP (25% of emissions), POAmP (32% of emissions) and POAhP (43% of emissions, having respectively a low, medium and high volatility, to follow the dilution curve of POA in [Robinson et al., 2007]. With carb equal to 3, the same split is done except for POA from biomass burning which are split into three other compounds BOAlP (26%), BOAmP (31%) and BOAhP (43%) to follow the dilution curve for biomass burning of [May et al., 2013]. The two dilution curves are similar. However, BOAlP is almost non-volatile contrary to POAlP. The oxidation of POAlP, POAmP, POAhP, BOAlP, BOAmP and BOAhP is assumed to lead to the compounds SOAlP, SOAmP, SOAhP, BSOAlP, BSOAmP and BSOAhP, respectively. Based on [Grieshop et al., 2009], those compounds are assumed to be less volatile by a factor 100 and the oxidation rate constant equal to 2×10^{-11} molecules^{-1}.cm^3.s^{-1}. The properties of the primary semi-volatile compounds are shown in Table 13.5 .

13.6.3.2 Computation of partitioning of SVOC with SOAP

SOAP can compute the partitioning of SVOC with either an equilibrium (assumig instataneous partitioning) or a dynamic approach (solving the differential equations involved in the formation of SOA). In the current state, only

Table 13.4: *Properties of the surrogate SOA species of the H^2O mechanism.*

Surrogate	Type [a]	H [b]	P^0 [c]	ΔH_{vap} [d]	Comments
BiMT	hydrophilic	0.805	1.45×10^{-6}	38.4	-
BiPER	hydrophilic	0.111	2.61×10^{-6}	38.4	-
BiDER	hydrohilic	2.80	4.10×10^{-7}	38.4	-
BiMGA	hydrophilic	1.13×10^{-2}	1.4×10^{-5}	43.2	pK_a = 4.0
BiNGA	hydrophobic	-	1.4×10^{-5}	43.2	$K_{p,eff}=K_p(1+K_{oligo})$ [e]
BiNIT3	hydrophobic	-	1.45×10^{-6}	38.4	-
BiA0D	hydrophilic	4.82×10^{-5}	2.70×10^{-4}	50	Account for oligomerization [f]
BiA1D	hydrophilic	2.73×10^{-3}	2.17×10^{-7}	50	pK_a = 3.2
BiA2D	hydrophilic	6.52×10^{-3}	1.43×10^{-7}	50	pK_{a1} = 3.4, pK_{a2} = 5.1
BiNIT	hydrophobic	-	2.5×10^{-6}	109	-
BiBlP	hydrophobic	-	6.0×10^{-10}	175	-
BiBmP	hydrophobic	-	3.0×10^{-7}	175	-
AnBlP	hydrophobic	-	6.8×10^{-8}	50	-
AnBmP	hydrophobic	-	8.4×10^{-6}	50	-
AnClP	hydrophobic	-	-	non volatile	-

[a] Type A: hydrophilic species, type B: hydrophobic species, type C: hydrophobic non-volatile species, which is not used to compute activity coefficients

[b] Henry's law constant [(μg/μg water)/(μg/m^3)]

[c] Saturation vapor pressure [torr]

[d] Enthalpy of vaporization [kJ.mol^{-1}]

[e] K_{oligo} (equal to 64.2) is used to take into account the formation of oligomers [Couvidat et al., 2012]. $K_{p,eff}$ is the effective partitioning constant and K_p is the partitioning constant calculated as in [Pankow, 1994].

[f] An effective Henry's law is computed as a function of pH to account for oligomerization of aldehydes. with the $H_{eff}=H\left(1+0.1\left(\frac{a(H^+)}{10^{-6}}\right)^{1.91}\right)$ with H the Henry's law and $a(H^+)$, the activity of ion H^+.

the equilibrium approach has been implemented into CHIMERE. The two methods are described extensively in [Couvidat and Sartelet, 2015].

SOAP computes the gas-particle partitioning into an organic phase is based on Raoult's law by taking into account the non-ideality (interactions between organic compounds) of the organic phase. The partitioning is computed with the following algorithm.

$$\frac{A_{p,i}}{A_{g,i}} = K_{p,i} M_o \tag{13.15}$$

with $A_{p,i}$ the concentration of i in the organic phase (in μg m$^-3$), $A_{g,i}$ the concentration of i in the gas phase (in μg m$^-3$), M_o the concentration of the organic phase (in μg m$^-3$) and $K_{p,i}$ the organic-phase partitioning coefficient (in m^3/μg) which is computed with this equation:

$$K_{p,i} = \frac{760 \times 8.202 \times 10^{-5} \times T}{M_{ow} \gamma_{i,org} P_i^0 \times 10^6} \tag{13.16}$$

with T the temperature (in K), M_{ow} the mean molar mass of the organic phase (in g/mol) and the saturation vapour pressure P_i^0 in torr. In SOAP, activity coefficients $\gamma_{i,org}$ are computed with the thermodynamic model UNIFAC (UNIversal Functional group Activity Coefficient [Fredenslund et al., 1975]).

Similarly, the partitioning onto the aqueous phase of particles (due to water absorbed by inorganics) is computed with a non-ideal Henry's law:

$$K_{aq,i} = \frac{H_i RT}{\rho_{water} \zeta_i \times 1.013 \times 10^{11}} \times \frac{18}{M_{aq}} \tag{13.17}$$

with H_i in (M.atm^{-1}), ρ_{water} the density of the aqueous phase (in kg m^{-3}), M_{aq} the molar mass of the aqueous phase (in g/mol) which can be slightly different from the molar mass of water due to the presence of other compounds and ζ_i the

Table 13.5: Properties of primary and aged SVOC the H^2O mechanism.

Surrogate	MW[a]	K_p [b]	ΔH_{vap} [c]
POAlP	280	1.1	106
POAmP	280	0.0116	91
POAhP	280	0.00031	79
SOAlP	392	110	106
SOAmP	392	1.16	91
SOAhP	392	0.031	79
BOAlP	280	18.3	106
BOAmP	280	0.040	91
BOAhP	280	0.00023	79
BSOAlP	392	1830	106
BSOAmP	392	4.0	91
BSOAhP	392	0.023	79

[a] Molecular weight [g.mol^{-1}]
[b] Partitioning constant [m^3.μg^{-1}]
[c] Enthalpy of vaporization [kJ.mol^{-1}]

activity coefficient by reference to infinite dilution. ζ_i is computed with:

$$\zeta_i = \frac{\gamma_{i,aq}}{\gamma^{\infty}_{i,aq}} \tag{13.18}$$

where $\gamma^{\infty}_{i,aq}$ is the activity coefficient at infinite dilution in water, which is computed with UNIFAC. However, UNIFAC only compute the activity coefficients due to short-range interactions and does not take into account medium and long range interactions due to the presence of electrolytes in the aqueous phase. In the aqueous phase, activity coefficients are computed with:

$$\gamma_{i,aq} = \gamma_{LR}\gamma_{MR}\gamma_{SR} \tag{13.19}$$

γ_{LR}, γ_{MR} and γ_{SR} are respectively the activity coefficients at long, medium and short range interactions. γ_{SR} is computed with UNIFAC whereas γ_{LR} and γ_{MR} are computed with the AIOMFAC method and parameters [Zuend et al., 2008, Zuend et al., 2011, Zuend and Seinfeld, 2012] and therefore the influence of inorganic ions on the partitioning of organic compounds is taken into account.
SOAP can be used with 3 ithermo options. For ithermo equal to 1 in chimere.par, ideality is assumed by reference to Raoult's law for hydrophobic partitioning ($\gamma_{i,org}$=1) and Henry's law for hydrophobic partitioning (ζ_i=1). For ithermo equal to 2 in chimere.par, non-ideality is computed with UNIFAC but long and medium range interactions are not taken into account (i.e. no interactions with electrolytes). Hygroscopicity of organics is also taken into account. For ithermo equal to 3 in chimere.par, long and medium range interactions are taken into account. The effect of non-ideality on SOA formation has been studied by [Couvidat et al., 2012] and [Kim et al., 2019].

13.6.4 The VBS mechanisms

13.6.4.1 SOA from Semi and Intermediate volatile compounds

For carb equal to 2 or 3 in chimere.par, the POA ared distributed in bins covering the range of $10^{-2}\mu.m^{-3}$ to $10^{6}\mu.m^{-3}$. These volatility bins cover three larger groups: non-volatile organic compounds with C* $< 10^{-2}\mu.m^{-3}$, Semi Volatile Organic Compounds (SVOC) covering the range of $10^{-2}\mu.m^{-3} < C^* < 10^{3}\mu.m^{-3}$ and Intermediate-volatility Organic Compounds (IVOCs) covering the range of $10^{3}\mu.m^{-3} < C^* < 10^{6}\mu.m^{-3}$ [Robinson et al., 2007]. The POA compounds are lited in the Table 13.6 .

- VBS scheme with functionalization process

For carb equal 2, the fraction of POA in the gazeous phase can be oxidized by OH radical to formed an oxydized POA noted OPOAx (x represents the bin of volatility defined in Table 13.6). The chemical mechanism used in [Zhang et al., 2013] is is described in Appendix G.5, p.311.

Model species	C*($molec.cm^{-3}$)	Molar Mass ($g.mol^{-1}$)
POA1	10^{-2}	250
POA2	10^{-1}	250
POA3	10^{0}	250
POA4	10^{1}	250
POA5	10^{2}	250
POA6	10^{3}	250
POA7	10^{4}	250
POA8	10^{5}	250
POA9	10^{6}	250

Table 13.6: *List and characteristics of the POA compounds*

- VBS scheme with functionalization and fragmentation process

For carb equal 3 corresponds to the scheme used in [Cholakian et al., 2018]. Two processes were added to the VBS scheme presented above. Fragmentation processes, corresponding to the breakup of oxidized OA compounds in the atmosphere into smaller and thus more volatile molecules [Shrivastava et al., 2011] and formation of nonvolatile SOA, where SOA can become nonvolatile after formation [Shrivastava et al., 2015]. The chemical mechanism is described in Appendix G.6, p.312

13.6.5 SOA from VOC oxydation

- VBS scheme with functionalization process

For soatyp equal 6, the formation of SOA from VOC is taken into account with VBS scheme with functionnalization procces. Anthropogenic precursors include:
OLE1 (Alkens k_{OH}<7x10^4) OLE2 (Alkens k_{OH}>7x10^3) ALK4 (Alkans 5x10^3<k_{OH}<1x10^4) ALK5 (Alkans 1x10^4<k_{OH}) ARO1 (Aromatiques k_{OH}<2x10^4) ARO2 (Aromatiques k_{OH}>2x10^4)
SOA yields were kept as in the standard VBS scheme desribed in [Zhang et al., 2013]; however, instead of using the low-NO_x or the high-NO_x regimes, an interpolation between the yields of these two regimes was added to the model. For this purpose, a parameter is added to the scheme, which calculates the ratio of the reaction rate of RO_2 radicals with NO (high-NO_x regime) with respect to the sum of reaction rates of the reactions with HO_2 and RO_2 (low-NO_x regime). For this purpose, a parameter (α) is added to the scheme, which calculates the ratio of the reaction rate of RO_2 radicals with NO (ν_{NO}; high-NO_x regime) with respect to the sum of reaction rates of the reactions with HO_2 (ν_{HO_2}) and RO_2 (ν_{RO_2}; low-NO_x regime). The parameter α is expressed as follows:

$$\alpha = \frac{\nu_{NO}}{\nu_{NO} + \nu_{HO_2} + \nu_{RO_2}} \tag{13.20}$$

This α value represents the part of RO_2 radicals reacting with NO (which leads to applying "high NO_x yields"). It is calculated for each grid cell by using the instantaneous NO, HO_2, and RO_2 concentrations in the model. Then, the following equation is used to calculate an adjusted SOA yield using this α value [Carlton et al., 2009].

$$Y = \alpha \times Y_{highNO_x} + (1 - \alpha) \times Y_{lowNO_x} \tag{13.21}$$

The complete chemical scheme implemented in CHIMERE includes biogenic and anthropogenic precursors is described in Appendix G.10, p.316.

- VBS scheme with functionalization and fragmentation process

For soatyp equal 7, the formation of SOA from VOC is taken into account with VBS scheme with functionalization and fragmentation process. The fragmentation processes for the SVOC start after thethird generation of oxidation because fragmentation is fa-vored with respect to functionalization for more oxidizedcompounds. Therefore, three series of species in different volatility bins were added to present each generation, similar to the approach setup in

[Shrivastava et al., 2013]. For bio-genic VOC, fragmentation processes come into effect starting from the first generation, as in [Shrivastava et al., 2013], because the intermediate species are considered to be moreoxidized. A fragmentation rate of 75% (with 25% left for functionalization) is used here for each oxidation step following [Shrivastava et al., 2015]. The formation of non-volatile SOA is performed by moving a part of each aerosol bin to nonvolatile bins with a reaction constant corresponding to a lifetime of 1h, similar to [Shrivastava et al., 2015].
Anthropogenic precursors include:
OLE1 (Alkens k_{OH}<7x10^4) OLE2 (Alkens k_{OH}>7x10^3) ALK4 (Alkans 5x10^3<k_{OH}<1x10^4) ALK5 (Alkans 1x10^4<k_{OH}) ARO1 (Aromatiques k_{OH}<2x10^4) ARO2 (Aromatiques k_{OH}>2x10^4)
The complete chemical scheme implemented in CHIMERE includes biogenic and anthropogenic precursors is described in Appendix G.11, p.316.

14 [FastJX] Radiation, AOD and photosynthesis

14.1 Radiative processes in the atmosphere

For radiative processes in the atmosphere, all calculations are done online using the Fast-JX model for radiative transfer (version 7.0b, see [Wild et al., 2000, Bian et al., 2002]).
This model is a model for radiative transfer and photochemical rates that has been designed with the purpose of its inclusions in CTMs, and is already used in models such as Polair3D, Geos-Chem, UKCA and PHOTOMCAT (e.g. [Voulgarakis et al., 2009, Real and Sartelet, 2011, Telford et al., 2013]). The photolysis rates calculated by fast-JX have been validated in the boundary layer ([Barnars et al., 2004]) as well as in the free troposphere ([Voulgarakis et al., 2009]).
The following effects on photolysis are taken into account:

- Effect of ice- and liquid-water clouds on photolysis rates through Mie diffusion
- Effect of aerosols on photolysis rates through absorption and Mie diffusion
- Effect of absorption by ozone on photolysis rates, using real-time ozone concentrations within the model domain instead of climatologies

From a computational point of view, the calculation of photochemical reaction rates is performed in three steps (Figure 14.1):

1. prep_mie: Preprocessing of aerosol optical properties (Mie scattering and absorption) for each aerosol species and bin, generate input files for Fast-JX: necessary only when changing aerosol species or their optical properties
2. inifastj: Initialisation of Fast-JX model (read input files): at the beginning of each run
3. photorates_fastj: Online calculation of photochemical reaction rates (every physical time step)

14.2 Optical preprocessing using prep_mie

The Mie calculations for each aerosol species and bins is performed by a short standalone program src/prep_mie.F90. This program is called by chimere-step1.sh and runs in the temporary execution directory. The name of its input and output files is stored in the prep_mie.nml file, in the same directory. src/prep_mie.F90 needs three input files:

1. AEROSOL:
   number of bins, their diameters and the list of the AEROSOL species,
2. input-fastj.txt:
   optical properties of each aerosol species (real and imaginary part of the refraction indices for 5 wavelength - 200 nm, 300 nm, 400 nm, 600 nm and 999 nm), as well as the density and Hanel size growth coefficient. This file needs to include these quantities for ALL aerosol species listed in the AEROSOL file. If the user adds a new aerosol specie, he will need to put its optical and physical properties in the input-fastj.txt file by editing it manually. Species in input-fastj.txt do not need to be in the same order as in the AEROSOL file, but it is requested that all aerosol species listed in the AEROSOL file shall be present in the input-fastj.txt file, with exactly the same name. It is permitted that input-fastj.txt contains species that are not actually needed for the present run: input-fastj is used as a dictionnary of "all" the possible aerosol species and their physical and optical properties.
3. ACTIVE_SPECIES:
   list of the active species. This file is used only to check whether prep_mie writes its outputs with all the aerosol species in the right order.

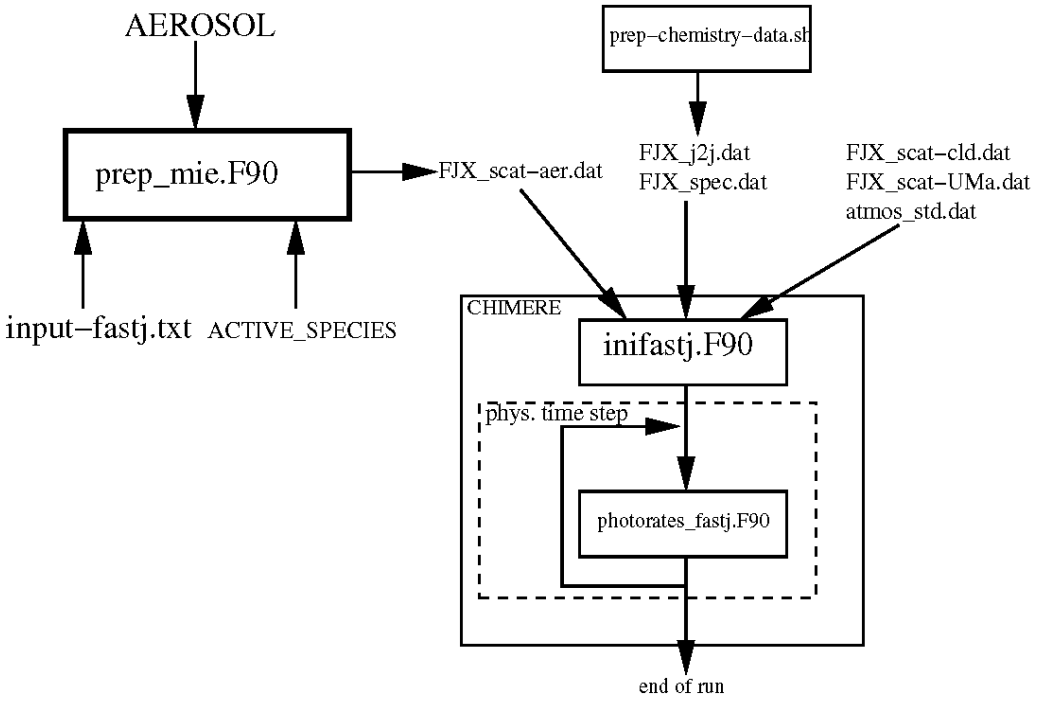


Figure 14.1: *Routines, files and scripts involved in the calculation of radiative transfers*

It is worth noting that Fast-JX (as well as CHIMERE) treats distinct bins of a given aerosol species as distinct species. Therefore, prep_mie will perform the Mie calculations for $nbins \times naerospec$ species. The output file of src/prep_mie is stored in FJX_scat-aer.dat. This file includes, for each of the $nbins \times naerospec$ species, the effective radius, density, the efficiency factor Q, the single-scattering albedo and the first 7 terms of the Taylor expansion of the scattering phase function. These values are written in the exact same order as the order of the aerosol species in the ACTIVE_SPECIES file.
The Mie calculation is based on Michael Mischenko's spher.f code (which is distributed with CHIMERE as well). As it is a relatively long calculation (up to a couple of minutes depending on the computer and on the number of aerosol species and bins), it is performed only if the FJX_scat-aer.dat is absent in the inputdata directory. Therefore, if the user modifies the optical or physical properties of the aerosols in input-fastj.txt, he will need to remove the FJX_scat-aer.dat file from the inputdata directory in order to force recalculation of the scattering phase functions.

14.3 Fast-JX initialisation

Fast-JX initialisation is performed by the inifastj.F90 routine. This routine is a wrapper for the INIT_FJX routine of the module fjx_init_mod.F90. INIT_FJX reads the following input files:

1. FJX_spec.dat : cross-sections for photochemical reaction rates.
2. FJX_scat-cld.dat : cloud scattering data
3. FJX_scat-UMa.dat : University of Michigan aerosol scattering data (not used)
4. atmos_std.dat : temperature and ozone climatology - used for radiative transfers above the simulation domain
5. FJX_j2j.dat : file containing the information needed to reorder Fast-JX photochemical reaction rates in the order needed by the CTM - in this case, CHIMERE.

The values read from these input files are stored in arrays that are used for the online calculation of photochemical reaction rates at each physical time step.

14.4 Online calculation of photochemical reaction rates

The calculation of the photochemical reaction rates is performed by the routine photorates_fastj.F90, which calls the fast-jx PHOTO_JX routine (in fjx_sub_mod.f90) for each atmospheric column, after filling the input arrays of Fast-JX with the values from the meteorological model (for clouds) and from CHIMERE (for ozone and aerosols). photorates_fastj.F90 finally takes into account the clouds above model top by a simple calculation of the attenuation factor - whereas clouds within the CHIMERE domain are taken into account in details, separating water- and ice-clouds and performing an explicit Mie calculation.

14.5 Aerosol and Cloud Optical Depth

The optical depth of aerosols, optdaero, and clouds, optdcloud, is a direct output of the fastJX model. They are written in the CHIMERE output file as vertically integrated variables and for each fastJX wavelength (the wavelength table: 200, 300, 400, 600, 999nm). Other FastJX outputs are the Single Scattering Albedo (SSA) and the Aerosols asymmetry factor (ASA). They are not written in CHIMERE output but send to WRF in case of online coupling.

14.6 Surface albedo

In previous model versions, the surface albedo in the near UV band was assumed equal to 0.1, constant in space and time. However, it has been shown that the surface albedo in this wavelength band, which is the determinant band for the calculation of photolytic rates, is highly variable, according to the landuse and to the presence of snow. The surface albedo in the UV band is evaluated according to [Laepple et al., 2005] in the absence of snow, and from [Tanskannen and Manninen, 2007] in the presence of snow (Table 14.1).

	Landuse	**Albedo**	
#	Description	(snow $<$ 1cm)	(snow $>$ 10 cm)
1	Agricultural land / crops	0.035	0.376
2	GrasslandLanduse type	0.04	0.720
3	Barren land/bare ground	0.10	0.836
4	Inland Water	0.07	-
5	Urban	0.035	0.3
6	Shrubs	0.05	0.558
7	Needleaf forest	0.025	0.278
8	Broadleaf forest	0.025	0.558
9	Ocean	0.07	-

Table 14.1: *Tabulated values from [Laepple et al., 2005] (for snow $<$ 1cm) and [Tanskannen and Manninen, 2007] (for snow $>$ 10 cm) used for the calculation of the albedo in the UV band*

The snow-depth is read from the meteorological inputs. In this version, this feature has been coded only for WRF inputs. If any other model is used, the snow cover will be assumed inexistent. If the snow-cover is thinner than 1cm in the model, the albedo is assumed to be that of dry land. If the snow-cover is thicker than 10cm, the albedo is assumed to be that of snow-covered land. In-between, a linear interpolation is performed. The calculated value of the albedo is available in the CHIMERE outputs (variable "albedo"). Keep in mind that this albedo represents the albedo in the near-UV band, which is very different from the albedo in the visible band.

14.7 The lidar vertical profiles

14.7.1 Methodology

A general overview of the lidar signal modeling is displayed in Figure 14.2 . The first column represents a model column, where aerosol concentrations (c_i) are available in grid cells for several model levels (z_i). This leads to a vertical concentration profile, where each c_i concentration represent the mean value between z_{i-1} and z_i (z_0 representing the ground).

Based on this concentration profile, we simulate the lidar signal that would be observed by a lidar in space (third column) or by a ground based lidar (fourth column).

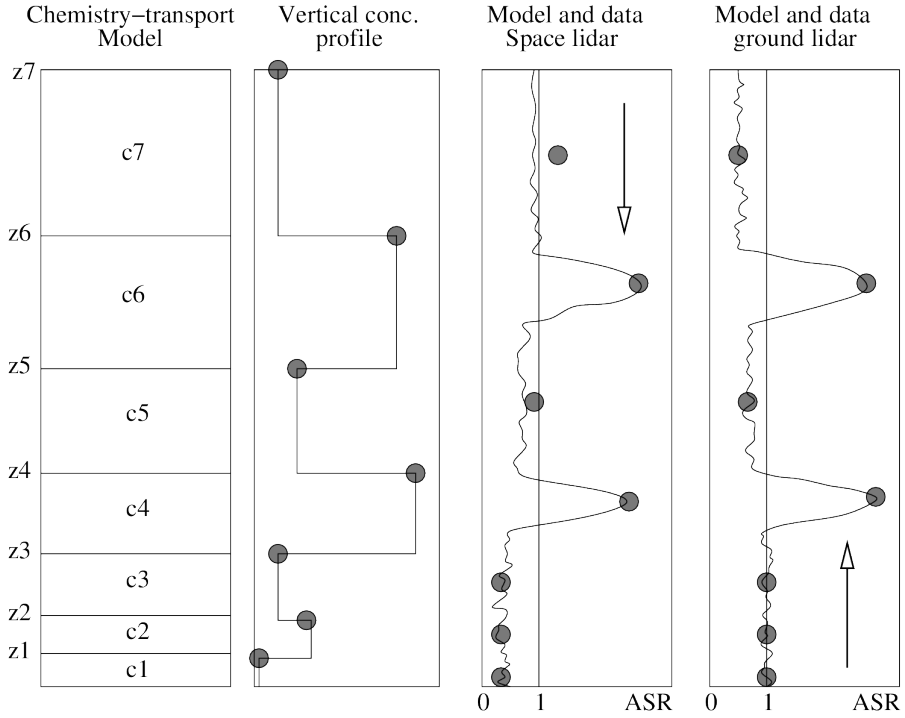


Figure 14.2: *Overview of the comparison methodology: example of the modelled Attenuated Scattering Ratio (ASR or $R'(z)$) estimation from initial concentration profiles (middle-left panel) on a specific model vertical grid (left panel) in the case of a space lidar (middle-right panel) or a ground lidar (right panel). The grey dots correspond to the value reported in the model simulation. After [Stromatas et al., 2012]*

The calculation of the lidar signal from simulated aerosol concentration fields requires additional parameters than those used for the AOD.
In order to efficiently compare modeled and measured lidar profiles, the simulator is designed to calculate the Attenuated Scattering Ratio, $R'(z)$. By definition, $R'(z)$ is equal to 1 in absence of aerosols/clouds and when the signal is not attenuated. In the presence of aerosol, $R'(z)$ would be greater than one. Following [Winker et al., 2009], this ratio is expressed as:

$$R'(z) = \frac{\beta'(z)}{\beta'_{\mathrm{m}}(z)} \tag{14.1}$$

where

$$\beta'(z,\lambda) = \left[\frac{\sigma_{\mathrm{m}}^{\mathrm{sca}}(z,\lambda)}{S_{\mathrm{m}}(z,\lambda)} + \frac{\sigma_{\mathrm{p}}^{\mathrm{sca}}(z,\lambda)}{S_{\mathrm{p}}(z,\lambda)}\right] \cdot \exp\left(-2\left[\int_z^{\mathrm{TOA}} \sigma_{\mathrm{m}}^{\mathrm{ext}}(z',\lambda)\mathrm{d}z' + \eta' \int_z^{\mathrm{TOA}} \sigma_{\mathrm{p}}^{\mathrm{ext}}(z',\lambda)\mathrm{d}z'\right]\right) \tag{14.2}$$

and

$$\beta'_{\mathrm{m}}(z,\lambda) = \frac{\sigma_{\mathrm{m}}^{\mathrm{sca}}(z,\lambda)}{S_{\mathrm{m}}(z,\lambda)} \cdot \exp\left(-2\int_z^{\mathrm{TOA}} \sigma_{\mathrm{m}}^{\mathrm{ext}}(z',\lambda)\mathrm{d}z'\right) \tag{14.3}$$

$\beta'(z,\lambda)$ and $\beta'_{\mathrm{m}}(z,\lambda)$ are, respectively the total and molecular attenuated backscatter signal. $\sigma_{\mathrm{p}}^{\mathrm{sca/ext}}(z,\lambda)$ and $\sigma_{\mathrm{m}}^{\mathrm{sca/ext}}(z,\lambda)$ are the extinction/scattering coefficients for particles and molecules (in km^{-1}). S_{m} (respectively S_{p}) is the molecular (respectively particular) extinction-to-backscatter ratio (in sr).

Finally, $\eta'(z)$ represents the particles multiple scattering and z represents the distance between the emitter and the studied point. Note that for the case of a space lidar the integration begins from the top of the atmosphere (TOA) while for a ground lidar the integration begins from 0 (ground level) to z.
The molecular contribution (σ_{m} and S_{m}) is calculated theoretically. In order to remain consistent with the observations, the same equation as the one used to derive the molecular backscattering coefficient for producing the CALIOP attenuated backscatter profiles is implemented [Hostetler et al., 2006]. When the vertical distribution of pressure (P) and temperature (T) is known, the molecular backscattering coefficient can be expressed as:

$$\sigma_{\mathrm{m}}^{\mathrm{sca}} = \frac{P}{k_{\mathrm{B}}T} \cdot \frac{s_{\mathrm{sca,mol}}(\pi)}{S_{\mathrm{m}}} \tag{14.4}$$

with k_{B} being the Boltzmann constant and $s_{\mathrm{sca,mol}}$ the molecular scattering cross section (in m^2), given by:

$$s_{\mathrm{sca,mol}} = 4.5102 \cdot 10^{-31} \cdot \left(\frac{\lambda}{0.55}\right)^{-4.025-0.05627\cdot(\lambda/550)-1.647} \tag{14.5}$$

Note that $S_{\mathrm{m}} = (8\pi/3)k_{bw}(\lambda)$ where k_{bw} represents the dispersion of the refractive index and King factor of air.
The molecular extinction coefficient $\sigma_{\mathrm{m}}^{\mathrm{ext}}$ is given as a function of $\sigma_{\mathrm{m}}^{\mathrm{sca}}$:

$$\sigma_{\mathrm{m}}^{\mathrm{ext}} = \frac{8\pi}{3} \cdot \sigma_{\mathrm{m}}^{\mathrm{sca}} \tag{14.6}$$

The particles contribution (σ_{p} and S_{p}) can be written as a function of the particle concentration (N_{p} in m^{-3}) and on the particle scattering/extinction efficiency ($Q_{\mathrm{sca/ext}}$) which depends on the refractive index, the size of particles and the wavelength (λ).
Note that multiple scattering effects are not taken into account here (η is set to 1 in Eq. 14.2). The single scattering approximation is adequate for small optical depths and non-absorbing aerosols [Gordon, 1997]. However, large scattering particles such as mineral dust could lead to non-negligible multiple scattering effects [Wandinger et al., 2010].
Finally, we also simulate the color ratio (χ') which corresponds to the ratio between two lidar profiles observed simultaneously at two different wavelengths ($\lambda_1 = 1064$ nm and $\lambda_2 = 532$ nm) :

$$\chi'(z) = \frac{R'_{\lambda_1}(z)}{R'_{\lambda_2}(z)} == \frac{\beta'_{\lambda_1}(z)}{\beta'_{\lambda_2}(z)} \tag{14.7}$$

Since scattering is more efficient when the wavelength is of the same order of magnitude as the particle diameter, this ratio provides information of the size of the particles in the backscattering layers and hence on their nature (cloud droplets, dust or pollution aerosols for instance). χ' is expected to be lower than 1 for small particles compared to wavelengths. It allows the qualitative identification of large/small particles but since there is a significant overlap between the distributions of χ' for different aerosol types, it cannot be used directly as an aerosol type identification tool [Omar et al., 2010].

14.7.2 The lidar outputs

All calculations are done in the src/lidar.F90 routine. This routine is called hourly and the results are hourly written in the out result file. Note that the lidar profile are calculated using the same wavelengths as those used for fastJX.
The input variables are mainly extracted from the FastJX routines or from the meteorology and are:

- thlaylocm: current cell thickness (in meters)
- CSCAtab: scattering cross section [m^2]
- CEXTtab: extinction cross section [m^2]
- PPItab: Backward scattering ratio [ad.]
- wavel(nwl): wavelengths (meters)
- temp: temperature in Kelvin
- presloc: Pressure field (Pa)
- numconc: the concentration in number of all aerosols in a bin and a cell (m^{-3})

The output variables are tables named:

- Nadir:

  - atb_mol_nad: Molecular attenuated backscatter signal ($m^{-1}.sr^{-1}$)
  - atb_tot_nad: Total attenuated backscatter signal (molecular + particles) ($m^{-1}.sr^{-1}$)
  - scat_ratio_nad: Attenuated scattering ratio
- Zenith:
  - atb_mol_zen: Molecular attenuated backscatter signal ($m^{-1}.sr^{-1}$)
  - atb_tot_zen: Total attenuated backscatter signal (molecular + particles) ($m^{-1}.sr^{-1}$)
  - scat_ratio_zen: Attenuated scattering ratio

15 Depositions

15.1 Dry deposition

15.1.1 The deposition velocity for gas and aerosols

The dry deposition process is commonly described through a resistance analogy proposed by [Wesely, 1989]. These resistances are not expressed in the same way, considering gas or particles as displayed in Figure 15.1 .

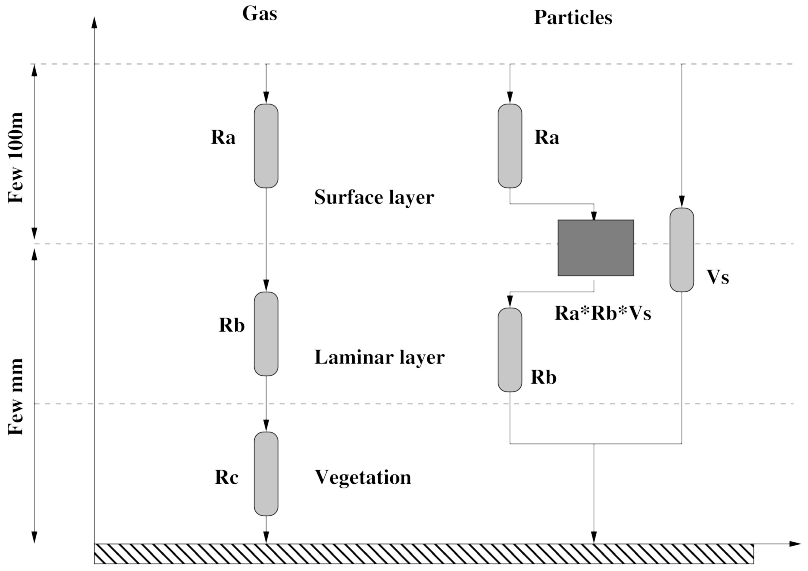


Figure 15.1: *Main principle of dry deposition for gas and particles. For each model species, three resistances are estimated: deposition occurs if the sum of all these resistances is low. For particles, the settling velocity is added.*

The deposition process is a sink and only acts on a concentration c along the vertical, as:

$$\Delta c = \frac{\partial}{\partial z}(v_d.c) \tag{15.1}$$

with v_d the dry deposition velocity. For each gaseous species or particles, the deposition is due to three different processes. First, the turbulent diffusivity is needed to estimate the aerodynamical resistance, r_a. Second, the diffusivity near the ground, in the "laminar" layer, is needed to estimate the surface resistance r_b. Third, for gaseous species only, the species water solubility is needed to estimate the canopy resistance r_c. For particles, there is no solubility, the particle is subject to gravity, falling with a settling velocity v_s.
The dry deposition velocity (in m s^{-1}) for gaseous species is expressed as:

$$v_d = \frac{1}{r_a + r_b + r_c} \tag{15.2}$$

For aerosols, two formulations are implemented in the model:
Following [Wesely, 1989]:

$$v_d = v_s + \frac{1}{r_a + r_b + r_a r_b v_s} \tag{15.3}$$

Following [Zhang et al., 2001]:

$$v_d = v_s + \frac{1}{r_a + r_s} \tag{15.4}$$

with r_s the surface resistance, depending on the vegetation type.
In CHIMERE, all dry deposition calculations are done in:

- Input parameters are initialized in routine inidepo.F90.
- At the coarse time step frequency, deposition velocities are calculated in routine depvel.F90 for gases and depaero.F90 for aerosols.
- Deposition fluxes are calculated at the fine time step frequency in routine deposition.F90.

15.1.2 The aerodynamical resistance, r_a

The aerodynamical resistance r_a depends on several turbulent parameters, such as the Monin-Obukhov length L, the friction velocity u_*, and the dynamical roughness length z_{0m}. This resistance is calculated in the calc_turb.F90 routine based on the Monin-Obukhov similarity theory with the Businger's formulation of the stability-dependent temperature profile function $\phi_T(\zeta)$. The reference height z_r is taken in the middle of the first vertical model layer of thickness H_1: $z_r = H_1/2$. Depending on the atmospheric surface layer stability,

$$\begin{cases} r_a &= \frac{1}{ku_*}\left[\ln\left(\frac{z}{z_{0m}}\right) + 4.7(\zeta_r - \zeta_0)\right] & (\text{stable, L} \geq 0) \\ &= \frac{1}{ku_*}\left[\ln\left(\frac{z}{z_{0m}}\right) + \ln\left(\frac{(\eta_0^2+1)(\eta_0+1)^2}{(\eta_r^2+1)(\eta_r+1)^2}\right) + 2\left(\tan^{-1}\eta_r - \tan^{-1}\eta_0\right)\right] & (\text{unstable, L} < 0) \end{cases} \tag{15.5}$$

where $\eta_0 = (1-15\zeta_0)^{1/4}$, $\eta_r = (1-15\zeta_r)^{1/4}$ and $\zeta_0 = z_0/L$, $\zeta_r = z_r/L$, k=0.41 the von Karman constant. z_0 depends on the fraction of land-use for each category and on the season.

15.1.3 The quasi-laminar layer resistance r_b for gases

The quasi-laminary boundary layer resistance r_b for gases is calculated in depvel.F90. It is estimated following [Hicks et al., 1987], as:

$$r_b = \frac{2}{ku_*}\left(\frac{S_c}{P_r}\right)^{2/3} \tag{15.6}$$

where k the Karman number (here k=0.41), S_c and P_r are the Schmidt and turbulent Prandtl numbers, respectively. The Prandl number is constant and set to Pr=0.72. The Schmidt number is defined as:

$$S_c = \frac{\nu}{D_i} \tag{15.7}$$

where ν is the kinematic viscosity of air. Its value is fixed as ν=0.15 cm^{2}s^{-1} for a reference atmosphere of T=20^{o}C and P=1 atm. D_i is the molecular diffusivity of the species i. This molecular diffusivity is estimated as the diffusion coefficient ratio of water to the pollutant i, following [Erisman et al., 1994].

$$D_i = \frac{D_{H2O}}{D_{gx}} \tag{15.8}$$

with D_{H2O}=0.25, the molecular diffusivity of water. For gaseous species, the molecular diffusivity is expressed as:

$$D_{gx} = \sqrt{\frac{dMx}{18}} \tag{15.9}$$

with dMx the molar mass of the model species. Being 'model species' dependent, one file per chemical mechanism is defined for these parameters. They are listed in the chemprep/data/DEPO_SPEC.meca ascii file. As an example, and for the MELCHIOR1 mechanism, the Table 15.1 presents these parameters values. This file is read in the routine inidepo.F90.

Species	dMx	dHx	$df0$	$dRwat$	Species	dMx	dHx	$df0$	$dRwat$
'charspec'	'rMx'	'rHx'	'rf0'	'rRwat'	'charspec'	'rMx'	'rHx'	'rf0'	'rRwat'
O3	48	0.01	1	2000	BUO2H*	64	1e5	0	1e-20
SO2	64	1e5	0	1.e-20	CH4COX*	64	1e5	0	1e-20
NO2	46	0.01	0.1	2000	CH3O2C*	64	1e5	0	1e-20
NO	30	2e-3	0	2000	CH4CHX*	64	1e5	0	1e-20
NH3	17	1e+5	0	1e-20	OXYLH1*	64	1e5	0	1e-20
HCL	36.5	2.05e+6	0	1e-20	MEMALH*	64	1e5	0	1e-20
HCHO	30	6e3	0	1e-20	MEMAH2*	64	1e5	0	1e-20
CH3CHO	44	14	0	1e-20	PEROX*	64	1e5	0	1e-20
CH3COE	72	13	0	1e-20	MACO2H*	64	1e5	0	1e-20
CH3COY	86	74	0	1e-20	CH4COZ*	64	1e5	0	1e-20
GLYOX	58	3.6e5	0	1e-20	APIO2H*	64	1e5	0	1e-20
MGLYOX	72	3.2e4	0	1e-20	obioH*	64	1e5	0	1e-20
MEMALD	98	70	0	1e-20	oROOH*	64	1e5	0	1e-20
MVK	70	70	0	1e-20	PANH*	64	1e5	0	1e-20
MAC	70	70	0	1e-20	AnA2D	186.	2.6e16	0	1e-20
PAN	109	3.6	0.1	1e-20	AnA1D	168.	1.5e10	0	1e-20
MACPAN	161	1.7	0.1	1e-20	AnA0D	142.	3.5e5	0	1e-20
MEMPAN	161	1.7	0.1	1e-20	AnBIP	167.	0.01	0	1e-20
toPAN	161	1.7	0.1	1e-20	AnBmP	152.	0.01	0	1e-20
HONO	47	1e5	0.1	1e-20	BiA2D	186.	1.6e11	0	1e-20
HNO3	63	1e14	0	1e-20	BiA1D	170.	2.9e10	0	1e-20
H2O2	34	1e5	1	1e-20	BiA0D	168.	1.9e6	0	1e-20
CH3O2H*	64	1e5	0	1e-20	BiBmP	236.	0.01	0	1e-20
PPA*	64	1e5	0	1e-20	ISOPA1	177.	1e5	0	1e-20
ETO2H*	64	1e5	0	1e-20	ISOPA2	177.	1e5	0	1e-20

Table 15.1: *The (chemprep/data/DEPO_SPEC.meca) data file. The main characteristics of the dry deposited species with their names, dMx the molar mass of the model species, dHx the effective Henry's law constant (M atm^{-1}) for the gas, df0 the normalised (0 to 1) reactivity factor for the dissolved gas. Note that all peroxides (with *) are treated as SO_2. $dRwat$ is a resistance to the solution of gases in water', en s m^{-1} and is not used in the current model version.*

15.1.4 The surface resistance for gases, r_c

This resistance acts only for gaseous species. Its formulation follows [Erisman et al., 1994] and the developments made in the EMEP model ([Emberson et al., 2000], [Simpson et al., 2003], [Simpson et al., 2012]). It uses a number of different other resistances accounting mainly for stomatal and surface processes which are again dependent on the land use type and season. Necessary chemical parameters for the calculation of r_c are also taken from [Erisman et al., 1994] except for carbonyls ([Sander et al., 1999]; [Baer and Nester, 1992]) and peroxide species ([Hall et al., 1999]). dHx and $df0$, presented in Table 15.1 , are used for the mesophyllic resistance value as described in [Seinfeld and Pandis, 1998]).
The general formulation for this surface resistance is proposed by [Erisman et al., 1994]:

$$r_c = \left[\frac{1}{r_{sto} + r_m} + \frac{1}{r_{inc} + r_{soil}} + \frac{1}{r_{ext}} \right]^{-1} \tag{15.10}$$

where:

- r_{sto} the canopy stomatal resistance
- r_m the mesophyll resistance
- r_{soil} the soil resistance
- r_{inc} the in-canopy aerodynamic resistance
- r_{ext} the canopy cuticle or external leaf resistance

Some specific cases are defined:

- Over water surfaces: r_c=r_{wat}
- Over bare soils: r_c=r_{soil}
- Over snow cover: r_c=r_{snow}

In CHIMERE this resistance is formulated following [Simpson et al., 2003] using a distinction between the 'bulk canopy resistance' R (and the associated conductance $G_c = 1/R$) and the 'unit-leaf-area' resistance r (and the associated conductance $g = 1/r_{sto}$). Using these notations, the bulk canopy conductance is expressed as:

$$G_c = LAI \times g_{sto} + G_{ns} \tag{15.11}$$

where LAI is the Leaf Area Index (in $m^2.m^{-2}$), g_{sto} the stomatal conductance, G_{ns} the bulk non-stomatal conductance. Two special cases are defined:

- For sub-zero surface temperature, the surface resistance reduces to:

$$r_c = r_{low} = 1000 \exp(-(T_s + 4)) \tag{15.12}$$

  Note that CHIMERE uses the 2m temperature as a 'surface temperature'. The temperature in Equation 15.12 is in ^{o}C.
- Under classical meteorological conditions, the surface resistance for HNO_3 is zero. To ensure model calculations stability, a minimum value of r_c=1 $s.m^{-1}$ is defined:

$$r_c^{HNO_3} = \max(1, r_{low}) \tag{15.13}$$

15.1.4.1 The stomatal conductance, g_{sto}

The stomatal conductance g_{sto} is estimated using the model of [Emberson et al., 2000]:

$$g_{sto} = g_{max} \times f_{phen} \times f_{light} \times \max(g_{min}, (f_{temp} \times fVPD \times f_{SWP})) \tag{15.14}$$

The parameters are:

- g_{max} is the maximum daytime stomatal conductance in mmol O_3 m^{-2} s^{-1}. To use it as a conductance and with the appropriate units, the unit conversion is made: g_{max}=$g^m_{max} RT/P$. At standard temperature and pressure, RT/P is assumed to be 1/41000. With the main units in CHIMERE being centimeters, the factor equals 410.
- g_{min} is the minimum daytime stomatal conductance. It depends only on the vegetation type.
- f_{phen} is the relative conductance determined by leaf/needle age
- f_{light} is the relative conductance determined by irradiance
- f_{temp} is the relative conductance determined by temperature
- f_{VPD} is the relative conductance determined by the leaf-to-air vapour pressure deficit (VPD)
- f_{SWP} is the relative conductance determined by the soil water potential (SWP). Note that in CHIMERE, the value of this parameter is the one chosen in [Emberson et al., 2000], i.e. f_{SWP}=1.

All the relative factors are comprised between 0 and 1. The g_{min} and g_{max} factors are constant and depend only on the vegetation type. They are specified in the model file chemprep/data/DEPO_PARS.univ as a function of the vegetation type. This file uses the 16 EMEP vegetation types (Table 15.3). Note that only the first 10 vegetation types are really used, since the last 6 categories have $g_{max} = 0$.

1	CF	T/B conif	9	GR	Grass
2	DF	T/B decid	10	MS	Med. scrub
3	NF	Med. needle	11	WE	wetlands
4	BF	Med. Broadleaf	12	TU	Tundra
5	TC	T/B crop	13	DE	Desert
6	MC	Med. crop	14	W	Water
7	RC	Root crop	15	ICE	Ice
8	SNL	Moorland	16	U	Urban

Table 15.2: *List of acronyms used in the deposition parameters data file. T/B stands for "temperate or boreal", Med for Mediterranean.*

The stomatal conductance was estimated for ozone only in [Emberson et al., 2000]. In order to derive a value for all other gaseous model species, the conductance value for ozone is scaled using the ratio of the diffusivities in the air of ozone and of another model species:

$$g^x_{sto} = g^{O_3}_{sto} \times \sqrt{\frac{dMO_3}{dMx}} \tag{15.15}$$

Here the ratio of diffusivities denoted 'facD' in the code and calculated in inidepo.F90 follows the Graham's Law of effusion:

$$\frac{\text{Rate of effusion of gas 1}}{\text{Rate of effusion of gas 2}} = \sqrt{\frac{M_2}{M_1}} \tag{15.16}$$

	CF	DF	NF	BF	TC	MC	RC	SNL	GR	MS	WE	TU	DE	W	I	U
Gmax [1]	160	134	180	200	300	156	360	60	270	213	0	0	0	0	0	0
Gmin [1]	0.1	0.13	0.13	0.03	0.01	0.019	0.02	0.01	0.01	0.014	0	0	0	0	0	0
Tmin [1]	1	6	4	4	12	0	8	1	12	4	0	0	0	0	0	0
Topt [1]	18	20	20	20	26	25	24	18	26	20	20	20	20	20	20	20
Tmax [1]	36	34	37	37	40	51	50	36	40	37	40	40	40	40	40	40
alpha [1]	83e-4	6e-3	13e-3	9e-3	9e-3	48e-4	23e-4	9e-3	9e-3	12e-3	0	0	0	0	0	0
VPDmax [1]	0.6	0.93	0.4	1.8	0.9	1.0	0.31	88.8	1.3	1.3	88.8	88.8	88.8	88.8	88.8	88.8
VPDmin [1]	3.3	3.4	1.6	2.8	2.8	2.5	2.7	99.9	3.0	3.0	99.9	99.9	99.9	99.9	99.9	99.9
SGS [2]	0	90	0	0	105	150	130	0	0	0	0	0	0	0	0	0
EGS [2]	366	300	366	366	197	300	250	366	366	366	366	366	366	366	366	366
SGS length [2]	192	30	180	180	70	70	35	192	180	1	1	1	1	1	1	1
EGS length [2]	96	90	180	180	22	44	65	96	180	1	1	1	1	1	1	1
LAI min [1]	3.4	3.5	3.5	3.5	0	0	0	2.0	2.0	2.5	0	0	0	0	0	0
LAI max [1]	4.5	5.0	3.5	3.5	3.5	3.0	4.2	3.0	3.5	2.5	0	0	0	0	0	0
PHENa [1]	0.2	0.3	0.3	0.3	0.1	0.1	0.2	0.1	1	0.2	1	1	1	1	1	1
PHENb [1]	130	50	110	110	0	0	20	0	0	130	0	0	0	0	0	0
PHENc [1]	130	50	150	150	45	45	45	45	0	130	0	0	0	0	0	0
Zcanopy [0]	20	20	20	20	1	2	0.5	1	1	2	0	0	0	0	0	0
RGSO3 [1]	200	200	200	200	200	200	200	400	1000	200	2000	400	2000	2000	2000	400
RGSSO2 [1]	0	0	0	0	0	0	0	0	0	0	1	500	1000	1	100	400
so2RH [1]	0.85	0.85	0.85	0.85	0.75	0.75	0.75	0.75	0.75	0.75	0	0	0	0	0	0

Table 15.3: *Dry deposition constants set in chemprep/data/DEPO_PARS.univ dependingon the vegetation type: (0) estimated for CHIMERE; (1) taken from the EMEP model documentation (EMEP website); (2) EMEP data with corrections applied for the growing season for the DF type: EGS=300 instead of 270, SGS_len=30 instead of 56.*

The tabulated parameters listed in Table 15.3 are:

- gmax: Maximum stomatal conductance (mmol O_3 m^{-2} s^{-1})
- gmin: Minimum daytime stomatal conductance observed under field conditions
- deptmin, deptopt and deptmax: Temperature boundaries to define the temperature factor. In this formulation, deptmax is not used.
- depalph: factor used with the PAR to moderate the light on leaves.
- depvpd1 and depvpd2: Threshold values (in kPa) for the minimum and maximum values of the leaf-to-air vapour pressure deficit (VPD)
- depsgs and depegs: (defined as SGS50 and EGS50 in [Simpson et al., 2012]). Start and end of the growing season at latitude 50^{o}N
- depsgl and depegl: duration of the start and end of the growing season
- deplai1 and deplai2: LAI min and max
- depphe0, depphe1, depphe2: phenological factors
- zcanopy: Canopy height (m)
- RGSO3 and RGSSO2: Ground surface resistance for O_3 and SO_2.
- so2rh: a threshold to estimate if the conditions are wet or dry for the SO_2 deposition.

15.1.4.1.1 The LAI, SAI and phenological factor. To take into account the vegetation growth variability in the deposition velocity calculation, the Leaf Area Index (LAI), the Surface Area Index (SAI) and the phenological factor f_{phen} are estimated.
There are two ways to estimate these parameters in the model. If the MEGAN scheme is used, the LAI is simply read from its database. The phenological factor is thus interpreted as the green fraction factor provided in the landuse file (from WRF or a database such as GLOBCOVER).
If these data are not available in the datasets used with CHIMERE, they are estimated using simple functions as described in [Emberson et al., 2000]. For the LAI, this fit uses minimum and maximum values of the growing season for each

vegetation type. Between these values, a simple function is applied as explained in Figure 15.2 (left).The phenology factor is also a simple function, depending only on three constant parameters (for each vegetation type): depphe0, depphe1, depphe2. The function is shown in Figure 15.2 (right): it reaches its maximum of 1 between PHENb and PHENc, while its minimum value PHENa depends on the vegetation type and varies from 0.1 to 0.3.
The SAI (surface area index) is defined in terms of LAI: SAI is set to the LAI value within the growing season and 1 outside the growing season.

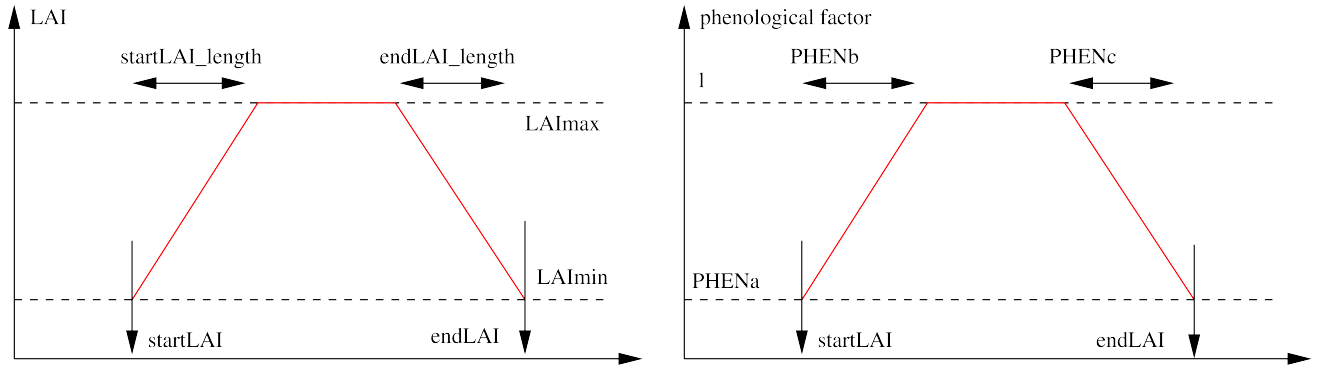


Figure 15.2: *Function used to define the hourly variability of the LAI and the phenological function. The required parameters are shown using their names in the model: SGS and EGS for the start and end of the growing season, LAImin and LAImax for the minimum and maximum of LAI value, SGSlength and EGSlength for the length of the start and end of the growing season. The phenological parameters are PHENa, PHENb, and PHENc to define the minimum value, the length, the start, and the end of the season.*

15.1.4.1.2 The light factor, f_{light}. The light factor, g_{light}, depends on the vegetation type. It is first needed to calculate the direct, I_{dir}, and diffuse, I_{diff}, solar irradiance (W m^{-2}), and the solar elevation, $\sin\beta$. In CHIMERE, the beam radiation, b_{rad}, (*beamrad*) is calculated as:

$$b_{rad} = 4.57 \times A_r \times 1069 \times \exp\left(\frac{-0.21}{cz}\right) \tag{15.17}$$

where cz is the cosinus of the zenithal angle ($\cos\theta$), 4.57 is a conversion factor to convert the irradiances from W m^{-2} to μ mol m^{-2} s^{-1}, A_r (*atte*) is the radiation attenuation calculated as:

$$A_r = \exp(-0.11 \times AOD^{0.67}) \tag{15.18}$$

The solar irradiances are thus estimated as:

$$I_{dir} = A_r \times cz \tag{15.19}$$

$$I_{diff} = I_{dir} \times C \tag{15.20}$$

C being a constant as C=0.13. This constant could probably be variable, this parameter being the impact of the water content in the whole atmospheric column.
Using these irradiances, the LAI and PAR are estimated as:

$$LAI_{sun} = \left(1 - \exp\left(\frac{-0.5 \times LAI}{cz}\right)\right) \times \frac{cz}{\cos(\alpha)} \tag{15.21}$$

In the CHIMERE model, we assume that $\cos(\alpha)$=0.5, corresponding to a constant inclination of leaves in the canopy of 60^o. The relative part of the shaded LAI is estimated as a fraction of LAI_{sun} as:

$$LAI_{shade} = LAI - LAI_{sun} \tag{15.22}$$

For the PAR, the coefficients PAR_{sun} and PAR_{shade} are calculated as:

$$\begin{aligned} PAR_{sun} &= \frac{\cos(\alpha)}{\cos(\theta)} \times I_{dir} + PAR_{shade} \\ PAR_{shade} &= I_{diff} \times \exp\left(-0.5 \times LAI^{0.7}\right) \\ &+ 0.07 \times I_{dir} \times (1.1 - 0.1 \times LAI) \\ &\times \exp(\sin\beta) \end{aligned} \tag{15.23}$$

with $\sin\beta = \cos\theta$. Finally, the f_{light} factors for the sun and shade light conditions are estimated as:

$$\begin{aligned} g_{light,shade} &= 1 - \exp\left(-\alpha \times PAR_{shade}\right) \\ g_{light,sun} &= 1 - \exp\left(-\alpha \times PAR_{sun}\right) \end{aligned} \tag{15.24}$$

with α a factor for each vegetation type. This factor (depalph in the model) is tabulated (see Table 15.3).
The complete f_{light} factor is finally:

$$f_{light} = \frac{g_{light,sun} \times LAI_{sun} + g_{light,shade} \times LAI_{shade}}{LAI} \tag{15.25}$$

To avoid division by zero, the LAI is here chosen as $LAI = \max(LAI, 1.e-20)$.

15.1.4.1.3 The temperature factor, f_{temp}. The temperature factor is estimated between two extrema of temperatures, T_{min} and T_{max}, or between T_{min} and a temperature of maximum conductance, T_{opt}. For all vegetation types, this temperature factor is always between 0 and 40^{o}C. For temperature below 0^{o}C or above 40^{o}C, the stomata are considered to be closed, and r_{sto} is set to a large value to avoid the deposition this way.
In [Emberson et al., 2000], this factor is estimated as:

$$f_{temp} = 1 - \left[\frac{T - T_{opt}}{T_{opt} - T_{min}}\right]^2 \tag{15.26}$$

with T the 2m temperature in ^{o}C. To ensure correct values of r_{sto} at various simulated T_{2m}, T is estimated as:

$$\begin{aligned} T_1 &= \max(T_{min} + 0.1, T_{2m}) \\ T &= \min(T_1, 2 \times T_{opt} - T_{min} - 0.1) \end{aligned} \tag{15.27}$$

15.1.4.1.4 The leaf to air vapour pressure deficit factor f_{vpd}. The leaf-to-air vapour pressure deficit (VPD) describes the difference between the vapor pressure inside the leaf compared to that of the air. This factor varies between 1 (high deficit, VPD$_{max}$, denoted depvpd1 in the code) and 0.1 (low deficit, VPD$_{min}$, denoted depvpd2 in the code). Between VPD$_{max}$ and VPD$_{min}$, a linear dependence is assumed. The current VPD value (vpdkPa in the model) is calculated hourly as:

$$VPD = (1 - rh/100) \times es \tag{15.28}$$

with rh the relative humidity in %, and es, in mbar (denoted vapp in the model) is calculated as:

$$es = 100. \times 6.11 \times \exp\left[\frac{17.27 \times T_{2m}}{T_{2m} - 35.86}\right] \tag{15.29}$$

The factor of 100 is to convert es to Pa (instead of hPa=mbar). In this equation, the 2m temperature is in K. The surface relative humidity is between 0 and 1 in the model and is estimated in calc_turb.F90.
Depending on the relative values of VPD and VPD$_{min}$, VPD$_{max}$ (in kPa, a factor 10^3 is thus applied), the f_{vpd} factor is estimated as:

$$\begin{cases} f_{vpd} = 1 & \text{if } VPD \geq VPD_{max} \\ f_{vpd} = 0.1 & \text{if } VPD \leq VPD_{min} \\ f_{vpd} = 1 - \dfrac{(VPD_{max} - VPD)(1 - 0.1)}{VPD_{max} - VPD_{min}} & \text{if } VPD_{max} \geq VPD \geq VPD_{min} \end{cases} \tag{15.30}$$

15.1.4.1.5 The soil water potential factor, f_{SWP}. The f_{SWP} represents the evolution of the water in soil. The soil water potential (in MPa) can be deduced from soil water content values. In the current scheme, its value is set to the constant f_{SWP}=1.

15.1.4.1.6 The mesophyllic resistance, r_m. The mesophyllic resistance r_m is calculated for each deposited species and follows the expression in [Seinfeld and Pandis, 1998] as:

$$r_m = \frac{1}{3.3 \times 10^{-4} \times dHx + df0 \times 100} \tag{15.31}$$

with dHx the effective Henry's law constant (M atm^{-1}) for the gas and $df0$ the normalised (0 to 1) reactivity factor for the dissolved gas (values are displayed in Table 15.1). Over vegetation canopies, corrections have been implemented according to [Zhang et al., 2001], [Giorgi, 1986] and [Peters and Eiden, 1992].

15.1.4.2 The non stomatal bulk conductance, G_{ns}

Non stomatal conductances are explicitly calculated for O_3, SO_2 and NH_3. For all other gaseous species, the conductance value is estimated by interpolating the values found for O_3 and SO_2 as:

$$\left\{ \begin{array}{lcl} G^i_{ns,dry} & = & 10^{-5} \times dH \times G^{SO_2}_{ns,dry} + df0 \times G^{O_3}_{ns} \\ G^i_{ns,wet} & = & 10^{-5} \times dH \times G^{SO_2}_{ns,wet} + df0 \times G^{O_3}_{ns} \end{array} \right. \tag{15.32}$$

where dH and $df0$ are the effective Henry's law constant and the normalised reactivity factor, respectively. For all model species, these values are displayed in Table 15.1 .

15.1.4.2.1 Non stomatal bulk conductance for ozone. After [Emberson et al., 2000], this conductance is expressed as:

$$G^{O_3}_{ns} = SAI \times g_{ext} + \frac{1}{R_{inc} + R^{O3}_{gs}} \tag{15.33}$$

where: g_{ext} is the external leaf uptake conductance, R_{inc} the in-canopy resistance and R^{O3}_{gs} the ground surface resistance (tabulated as a function of the vegetation type in Table 15.3).
Low temperature conditions are taken into account to modify the R^{O3}_{gs} values as:

$$R^{O3}_{gs} = R^{O3}_{gs} + R_{low} + 2000\delta_{snow} \tag{15.34}$$

In the current model version, the presence of snow is not taken into account and a fixed value of δ_{snow}=0 is implemented. This may change in a future version.

15.1.4.2.2 The in-canopy resistance, R_{inc}. The in-canopy aerodynamic resistance R_{inc} represents the deposition to soils and under canopy. [Erisman et al., 1994] explains that it can represent $\approx$ 20% of O_3 and SO_2 dry deposition. In this model version, this resistance is calculated only for ozone. This resistance uses the friction velocity and the canopy height z_c (in meters). z_c is tabulated independently of the season and latitude in the current model version, as displayed in Table 15.3 .

$$R_{inc} = 100. \times \frac{b \times SAI \times z_c}{u_*} \tag{15.35}$$

where 100 is a factor to convert m to cm, b=14 m^{-1}.

15.1.4.2.3 The external leaf uptake resistance, R_{ext}. Several formulations are proposed in the litterature for this resistance. A complete scheme is proposed by [Erisman et al., 1994] to represent the deposition of soluble gases at wet surfaces, for example for SO_2.
For SO_2, this resistance, in s.m^{-1}, is thus derived as:

$$\begin{cases} R_{ext} & = 1 & \text{during or just after precipitation} \\ & = 25000\ \exp(-0.0693\,rh) & \text{at rh} < 81.3\% \\ & = 0.58 \times 10^{12}\ \exp(-0.278\,rh) & \text{at rh} > 81.3\% \end{cases} \tag{15.36}$$

In CHIMERE, this resistance is defined in a simpler way after [Emberson et al., 2000]:

$$R_{ext} = 2500 s.m^{-1} \tag{15.37}$$

Expressed in terms of conductance, using the SAI dependency and converted to cm.s^{-1}, this leads to:

$$g_{ext} = \frac{1}{25} \tag{15.38}$$

15.1.4.2.4 Non stomatal bulk conductance for SO_2. This conductance for sulphur dioxide is controlled by wetness and NH_3 surface concentrations. This estimation uses the 'acidity ratio' (denoted 'acraloc' in CHIMERE) as:

$$a_{SN} = 0.6 \times \frac{[SO_2]}{[NH_3]} \tag{15.39}$$

Using this ratio, the F_{SN} is defined as:

$$\begin{cases} F_{SN} & = \exp(-2 + a_{SN}) & \text{if } a_{SN} < 2 \\ & = 1 & \text{if } a_{SN} \geq 2 \end{cases} \tag{15.40}$$

and used in the following relationship to estimate $R_{ns}^{SO_2}$:

$$\begin{cases} R_{ns,dry}^{SO_2} & = & R_d \times F_{SN} + R_{low} + 2000 \times \delta_{snow} \\ R_{ns,wet}^{SO_2} & = & R_w \times F_{SN} + R_{low} + 2000 \times \delta_{snow} \end{cases} \tag{15.41}$$

with R_d=180 s.m^{-1} and R_w=100 s.m^{-1}.

15.1.4.2.5 Non stomatal bulk conductance for NH_3. The acidity ratio is also used to estimate this conductance as:

$$\begin{cases} R_{ns}^{NH_3} & = \beta \times F_1 \times F_2 & \text{for } T_s > 0 \\ & = 200 & \text{for } -5 < T_s \leq 0 \\ & = 1000 & \text{for } T_s \leq -5 \end{cases} \tag{15.42}$$

β is a constant as β=0.0455 and the two functions are:

$$\begin{cases} F_1 & = & 10 \times log(T_s + 2) \times \exp\left(\dfrac{100 - RH}{7}\right) \\ F_2 & = & 10^{-1.1099 a_{SN} + 1.6769} \end{cases} \tag{15.43}$$

where RH is here the surface relative humidity in %. For this chemical species, there is no dependency on the wet or dry atmospheric conditions and the final resistance is:

$$R_{ns} = \frac{1}{G_{ns}^{NH3}} \tag{15.44}$$

15.1.5 The quasi-laminar layer resistance r_b for aerosols

For aerosols, the quasi-laminar resistance completely differs from that of gases. This resistance depends on the particle diameter and represents the effect of three parallel resistances: Brownian diffusion, impaction and interception, [Kouznetsov and Sofiev, 2012], [Seinfeld and Pandis, 2006], as:

$$r_b = \frac{1}{u_* \left(S_c^{-2/3} + 10^{-3/S_t} \right)} \tag{15.45}$$

with S_t the Stokes number and S_c the Schmidt number (dimensionless) as:

$$S_c = \frac{\nu_{air}}{D_i} \tag{15.46}$$

with ν_{air} the kinematic viscosity of air (in $m^2.s^{-1}$) and D_i the diffusivity in air of the particle. In case of aerosols, this diffusivity is considered as a Brownian diffusion and is estimated using the Stokes-Einstein relation as:

$$D_i = \frac{k_B \times T \times C_c}{3 \times \pi \times \mu_{air} \times D_p} \tag{15.47}$$

with k_B the Boltzmann constant, $k_B = 1.38065 \times 10^{-23}$ J.K^{-1} (with 1 J = 1 kg.m^2.s^{-1}).
The kinematic viscosity ν_{air} is calculated using the value of the dynamic viscosity μ_{air} (in kg.m^{-1}.s^{-1}):

$$\nu_{air} = \frac{\mu_{air}}{\rho_{air}} \tag{15.48}$$

The calculation of the dynamic viscosity is possible using a large number of fitted data. In the model, the Sutherland relation is used.

$$\mu = \mu_0 \frac{T_0 + C}{T + C} \left(\frac{T}{T_0} \right)^{3/2} \tag{15.49}$$

with the references values for air: T_0=291.15 K, C=120 K and μ_0=18.27 10^{-6} Pa.s (i.e. kg.m^{-1}.s^{-1}).
The dimensionless Stokes number is calculated as:

$$S_t = \frac{v_s \times u_*^2}{g \times \nu_{air}} \tag{15.50}$$

with v_s the settling velocity, itself depending on the aerosol mass median diameter, D_p. It is calculated in the depaero.F90 routine

15.1.6 The surface resistance for aerosols, r_s

This formulation follows the scheme of [Zhang et al., 2001]. This surface resistance is calculated as:

$$r_s = \frac{1}{\epsilon_0 \times u_* \times R_1 \times (E_B + E_{IM} + E_{IN})} \tag{15.51}$$

with:

- ϵ_0 is a constant set to ϵ_0=3, for all landuse categories.
- R_1 a correction factor describing the relative amount of aerosols sticking at the surface
- E_B collection efficiency from Brownian diffusion
- E_{IM} collection efficiency from impaction
- E_{IN} collection efficiency from interception

The R_1 factor is estimated following [Slinn, 1982]:

$$R_1 = \exp(-St^{1/2}), \tag{15.52}$$

where this factor is applied only for particles with $D_p > 5\mu m$.

For Brownian diffusion, the resistance is estimated as:

$$E_B = Sc^{-\gamma} \tag{15.53}$$

where γ is a constant depending on the landuse type. In the model, this constant varies between 0.54 and 0.58.
For the impaction, the resistance can have a lot of definitions, depending on the landuse. By default, the resistance value used is:

$$E_{IM} = \left(\frac{St}{\alpha + St}\right)^2 \tag{15.54}$$

The α values are landuse dependent and are tabulated following [Zhang et al., 2001]. In this model version, a distinction is made between northern and southern hemisphere, in order to use the correct α for a specific modelled day.
For specific vegetation types, the formulation is changed. For high vegetation (such as forests), the resistance proposed by [Giorgi, 1986] is used:

$$E_{IM} = \left(\frac{St}{0.6 + St}\right)^{3.2} \tag{15.55}$$

For grassland vegetation, a parameterization is proposed by [Davidson et al., 1982]:

$$E_{IM} = \frac{St^3}{St^3 + 0.753\ St^2 + 2.796\ St - 0.202} \tag{15.56}$$

Finally, the collection efficiency from interception is calculated as:

$$E_{IM} = \frac{1}{2}\left(\frac{D_p}{A}\right)^2 \tag{15.57}$$

with A a characteristic diameter given for landuse and seasonal categories.

15.1.7 The settling velocity v_s

The settling velocity represents the effect of gravity on particles and is calculated in depaero.F90 as:

$$v_s = \frac{1}{18}\frac{D_p^2\ \rho_p\ g\ C_c}{\mu} \tag{15.58}$$

with ρ_p, the particle density (chosen as ρ_p=2.65 g cm^{-3} for mineral dust), D_p the mass median diameter of particles, C_c a slip correction factor accouting for the non-continuum effects when D_p becomes smaller and of the same order of magnitude as the mean free path of air, λ, [Seinfeld and Pandis, 1998]. g is the gravitational acceleration with g=9.81 m s^{-2}, μ the dynamic viscosity (here the air dynamic viscosity is set to μ_{air}=1.8 $\times$ 10^{-5} kg m^{-1} s^{-1}).
The slip correction factor C_c is estimated as:

$$C_c = 1 + \frac{2\lambda}{D_p}\left[1.257 + 0.4\ \exp\left(-\frac{1.1 D_p}{2\lambda}\right)\right] \tag{15.59}$$

with λ the mean free path of air, in meters, estimated as:

$$\lambda = \frac{2\mu_{\text{air}}}{p\sqrt{\frac{8M_{\text{air}}}{\pi RT}}} \tag{15.60}$$

where M_{air} is the molecular mass of dry air (here 28.8 g mol^{-1}), T the temperature (K), p the pressure (Pa), μ the air dynamic viscosity and R the universal gas constant.

15.2 Wet scavenging

When the aerosol option is selected in chimere.par, scavenging for gas/aerosols in clouds or rain and snow droplets is taken into account. The calculation is done in several routines:

1. inidepo.F90: Read the input parameters
2. depwet.F90: Calculation of the deposition fluxes

15.2.1 Input parameters

In the inidepo.F90, parameters are read for the wet scavenging of gas and particles. These parameters are stored in the chemprep/inputdata/WETD_SPEC.univ. This parameter file is explained in Table 15.4 .

Species	**Type**	**In cloud scavenging constants**			Molar mass
'charspec'	'itywet'		'wetdk'		'dMx_wet'
Gaseous species					
HNO3	1	2.0e-3	0.	0.10	63.0
NH3	1	2.0e-3	0.	0.10	17.0
H2O2	1	2.0e-3	0.	0.18	34.0
HCL	1	2.0e-3	0.	0.10	36.5
Aerosol species					
H2SO4AQ	2	0.	1.	0.	96.9
HNO3AQ	2	0.	1.	0.	62.0
NH3AQ	2	0.	1.	0.	18.0
SOAAQ	2	0.	1.	0.	250.0
AnA0DAQ	2	0.	1.	0.	142.0
AnA1DAQ	2	0.	1.	0.	168.0
AnA2DAQ	2	0.	1.	0.	186.0
AnBmPAQ	2	0.	1.	0.	152.0
AnBIPAQ	2	0.	1.	0.	167.0
BiA0DAQ	2	0.	1.	0.	168.0
BiA1DAQ	2	0.	1.	0.	170.0
BiA2DAQ	2	0.	1.	0.	186.0
BiBmPAQ	2	0.	1.	0.	236.0
ISOPA1AQ	2	0.	1.	0.	177.0
ISOPA2AQ	2	0.	1.	0.	177.0
PPMAQ	2	0.	1.	0.	199.0
BaPAQ	2	0.	1.	0.	252.0
BbFAQ	2	0.	1.	0.	252.0
BkFAQ	2	0.	1.	0.	252.0
SALTAQ	2	0.	1.	0.	58.5
OCARAQ	2	0.	1.	0.	100.0
BCARAQ	2	0.	1.	0.	100.0
NAAQ	2	0.	1.	0.	23.0
HCLAQ	2	0.	1.	0.	25.5
H2O2	2	1.	7.45e4	0.	38.0
SO2	2	1.	1.51e-2	1.	64.0

Table 15.4: *Parameters for the wet deposition with their model species names, charspec, the type of the species, in cloud scavenging constants and the molar mass.*

In the depwet.F90, some constants are hardcoded and are explained in Table 15.5 .
CHIMERE separate scavenging due to snow and due to rain.

15.2.2 Wet scavenging for aerosol

For **below-cloud scavenging of particles by rain droplets**: Particles are scavenged by raining drops. Following [Willis and Tattelman, 1989], a polydisperse distribution of raining drops is applied:

$$N(R) = 1.06 \times 10^{14} \times P^{-0.0295d0} (2R)^{2.16d0} \exp\left(-5679 P^{-0.153d0} 2R\right) \tag{15.61}$$

Constant	Significance	Value
diff	Molecular diffusivity of the species	0.25 10^{-4} (m^2 s^{-1})
Rg	Universal gas constant	8.205 10^{-2} (m^3.atm.$kmol^{-1}$.K^{-1})
wated	Water density	1.0
diami	Minimal diameter for droplets	0.03 10^{-3} (cm)
diamf	Maximal diameter for droplets	1.2 10^{-2} (cm)
ccloud	volume fraction of cloud in the cell (for cumulus)	0.3 (g/m3)
Fc	Surface of the cell covered by clouds	1.0
pcpth	Precipitation threshold	10^{-8} (g/cm2/s)
cwth	Cloud liquid water content threshold	10^{-11} (kg/kg)
Eff	Default scavenging efficiency	1.0

Table 15.5: *Constants used for the calculation of the wet scavenging fluxes for gas and aerosols*

with P the precipitation rate in mm/h and R the radius of the droplet (in m).
The flux of deposition is calculated with:

$$\left[\frac{dQ_l^k}{dt}\right] = -Q_l^k \int_R \pi R^2 u_g(R) E(R, r_l) N(R) \mathrm{d}R \tag{15.62}$$

with R, the radius of the raindrop (in m), r_l the radius of the particle (in m), u_g the terminal drop velocity (in m/s), E(R,r_l) the collision efficiency of a particle with a raindrop, N(R) (in m^{-4}) the raindrop size distribution.
For **below-cloud scavenging of particles by snow**:
Particles are scavenged by appling the parameterization of [Wang et al., 2014]. A scavenging coefficient λ_{snow} is computed with:

$$log\left(\lambda_{snow}\right) = logA + B \tag{15.63}$$

For particles with a diameter dp below 1.44 μm

$$logA = -4.4260 + 1.3940 \times ldp - 1.2020 \times ldp^2 - 3.2942 \times ldp^3 - 1.9521 \times ldp^4 - 0.49040 ldp^5 - 4.5700d - 2 \times ldp^6 \tag{15.64}$$

$$B = 0.56640 + 8.5000d - 3 \times ldp - 0.19480 \times ldp^2 - 0.65320 \times ldp^3 - 0.5462 \times ldp^4 - 0.17780 \times ldp^5 - 2.0100d - 2 \times ldp^6 \tag{15.65}$$

Else:

$$logA = -4.3531 - 0.78280 \times ldp + 12.768 \times ldp^2 - 19.864 \times ldp^3 + 13.618 \times ldp^4 - 4.4350 \times ldp^5 + 0.55510d0 \times ldp^6 \tag{15.66}$$

$$B = 0.56890 - 9.2300d - 2 \times ldp + 4.0200d - 2 \times ldp^2 + 1.4523 \times ldp^3 - 2.0780 \times ldp^4 + 1.0500 \times ldp^5 - 0.18210 \times ldp^6 \tag{15.67}$$

with ldp = log(dp).
The flux of deposition is calculated with:

$$\left[\frac{dQ_l^k}{dt}\right] = -\lambda_{snow} Q_l^k \tag{15.68}$$

For **in-cloud scavenging of particles** : Deposition of particles is assumed to occur when a precipitating cloud encounter particles. The rate of deposition is computed by calculating the rate of impaction between hydrometeors and cloud droplets (assumed to have a diameter of 10 μm). The rate of scavenging is computed with equations 15.70 and 15.68 for dp=10μm.

15.2.3 Wet scavenging for gases

The wet scavenging of gases is divided into two processes: in-cloud scavenging (absorption by clouds of gases followed by deposition when precipitation occur) and below-cloud scavenging by rain and snow.
For the **gases in clouds**, the fraction of compounds inside the cloud f_{cl} is assumed to be scavenged by hydrometeors with the rate of cloud droplet scavenging computed with equations 15.70 and 15.68 for dp=10μm. The absorption of compounds by clouds is assumed to be at equilibrium and is computed according to Henry's law as a function of temperature, liquid water content and pH (for acidic compounds and ammonia).

For the **below-cloud scavenging of gases by rain droplets**, the dissolution of gases in precipitating drops is assumed to be irrevertible. The scavenging coefficient is computed with the polydisperse distribution of rain drops of [Willis and Tattelman, 1989] shown by Eq. 15.61.

$$\left[\frac{dQ}{dt}\right] = -Q \int_R 2\pi D_g (2 + 0.6 Re^{0.5} Sc^{0.333}) R \times N(R) \mathrm{d}R \tag{15.69}$$

with Q the concentration of the gas, D_g the molecular diffusion coefficient, Re the Reynolds number and Sc the Schimdt number.
For the **below-cloud scavenging of gases by snow**, the absorption of gases by snow is assumed to be irrevertible. The scavenging coefficient is computed with the parameterization of [Chang, 1984]:

$$\left[\frac{dQ}{dt}\right] = -Q \int_R 1.5\pi D_g A \left(1 + 206(2R)^{0.65}\right) R \times N(R) \mathrm{d}R \tag{15.70}$$

with R the melted snowflake radius and A the maximun dimension of the snowflake. A is computed a function of R by assuming that snowflakes are dendrites. The polydisperse distribution of [Sekhon and Srivastava, 1970] is used :

$$N(R) = 2.5 \times 10^6 P^{-0.94} \exp\left(-22.9 P^{-0.45} 200 R\right) \tag{15.71}$$

with P the precipitation rate in mm/h and R the radius of the droplet (in m).

Bibliography

[Aan de Brugh et al., 2012] Aan de Brugh, J. M. J., Henzing, J. S., Schaap, M., Morgan, W. T., van Heerwaarden, C. C., Weijers, E. P., Coe, H., and Krol, M. C. (2012). Modelling the partitioning of ammonium nitrate in the convective boundary layer. *Atmos. Chem. Phys.*, 12:3005–3023.

[Abdul-Razzak and Ghan, 2002] Abdul-Razzak, H. and Ghan, S. J. (2002). A parameterization of aerosol activation 3. sectional representation. *Journal of Geophysical Research: Atmospheres*, 107(D3):AAC 1–1–AAC 1–6.

[Alfaro et al., 1997] Alfaro, S. C., Gaudichet, A., Gomes, L., and Maillé, M. (1997). Modeling the size distribution of a soil aerosol produced by sandblasting. *J of Geophysical Research*, 102:11,239–11,249.

[Alfaro et al., 1998] Alfaro, S. C., Gaudichet, A., Gomes, L., and Maillé, M. (1998). Mineral aerosol production by wind erosion: aerosol particle sizes and binding energies. *Geophys. Res. Lett*, 25:991–994.

[Alfaro and Gomes, 2001] Alfaro, S. C. and Gomes, L. (2001). Modeling mineral aerosol production by wind erosion: Emission intensities and aerosol size distribution in source areas. *J of Geophysical Research*, 106:18,075–18,084.

[Atkinson et al., 1997] Atkinson, R., Baulsch, D. L., Cox, R. A., Hampton, R. F., Kerr, J. A., Rossi, M. J., and Troe, J. (1997). Evaluated kinetics, photochemical and heterogeneous data. *J. Phys. Chem.*, 26(3):521–1012.

[Aumont et al., 2003] Aumont, B., Chervier, F., and Laval, S. (2003). Contribution of HONO sources to the NOx/HOx/O3 chemistry in the polluted boundary layer. *Atmospheric Environment*, 37(4):487 – 498.

[Baer and Nester, 1992] Baer, M. and Nester, K. (1992). Parameterization of trace gas dry deposition velocities for a regional mesoscale diffusion model. *Ann. Geophys.*, 10:912–923.

[Balkanski et al., 2007] Balkanski, Y., Schulz, M., Claquin, T., and Guibert, S. (2007). Reevaluation of mineral aerosol radiative forcings suggests a better agreement with satellite and aeronet data. *Atmospheric Chemistry and Physics*, 7(1):81–95.

[Barnars et al., 2004] Barnars, J. C., Chapman, E. G., Fast, J. D., Schmelzer, J. R., Slusser, J. R., and Shetter, R. E. (2004). An evaluation of the FAST-J photolysis algorithm for predicting nitrogen dioxide photolysis rates under clear and cloudy sky conditions. *Atmos Environ*, 38:3393–3403.

[Bauer et al., 2004] Bauer, S., Balkanski, Y., Schulz, M., Hauglustaine, D. A., , and Dentener, F. (2004). Heterogeneous chemistry on mineral aerosol surfacesă: a global modelling study on the influence on tropospheric ozone chemistry and comparison to observations. *J Geophys Res*, 109:D02304.

[Beegum et al., 2016] Beegum, S., I.Gherboudj, N.Chaouch, F.Couvidat, L.Menut, and H.Ghedira (2016). Simulating Aerosols over Arabian Peninsula with CHIMERE: Sensitivity to soil, surface parameters and anthropogenic emission inventories. *Atmospheric Environment*, 128:185–197.

[Berge, 1993] Berge, E. (1993). Coupling of wet scavenging of sulphur to clouds in a numerical weather prediction model. *Tellus*, 45B:1–22.

[Bessagnet et al., 2004] Bessagnet, B., Hodzic, A., Vautard, R., Beekmann, M., Cheinet, S., Honoré, C., Liousse, C., and Rouil, L. (2004). Aerosol modeling with CHIMERE: preliminary evaluation at the continental scale. *Atmospheric Environment*, 38:2803–2817.

[Bessagnet et al., 2008] Bessagnet, B., Menut, L., Aymoz, G., Chepfer, H., and Vautard, R. (2008). Modelling dust emissions and transport within Europe: the Ukraine March 2007 event. *Journal of Geophysical Research*, 113:D15202.

[Bessagnet et al., 2009] Bessagnet, B., Menut, L., Curci, G., Hodzic, A., Guillaume, B., Liousse, C., Moukhtar, S., Pun, B., Seigneur, C., and Schulz, M. (2009). Regional modeling of carbonaceous aerosols over Europe - Focus on Secondary Organic Aerosols. *Journal of of Atmospheric Chemistry*, 61:175–202.

[Bian et al., 2002] Bian, H., , and Prather, M. (2002). Fast-J2: accurate simulation of stratospheric photolysis in global chemical models. *J. Atmos. Chem.*, 41:281–296.

[Bicheron et al., 2011] Bicheron, P., Amberg, V., Bourg, L., Petit, D., Huc, M., Miras, B., Brockmann, C., Hagolle, O., Delwart, S., Ranera, F., Leroy, M., and Arino, O. (2011). Geolocation assessment of meris globcover orthorectified products. *IEEE Transactions on geoscience and remote sensing*, 49(8):2972–2982.

[Bowman et al., 1997] Bowman, F. m., Odum, J. R., Seinfeld, J. H., and Pandis, S. N. (1997). Mathematical model for gas-particle parttioning of secondary organic aerosols. *Atmos. Environ.*, 31:3921–3931.

[Briant et al., 2017] Briant, R., Tuccella, P., Deroubaix, A., Khvorostyanov, D., Menut, L., Mailler, S., and Turquety, S. (2017). Aerosol–radiation interaction modelling using online coupling between the WRF 3.7.1 meteorological model and the CHIMERE 2016 chemistry-transport model, through the OASIS3-MCT coupler. *Geoscientific Model Development*, 10(2):927–944.

[Byun et al., 1999] Byun, D. W., Young, J., Pleim, J., Odman, M. T., and Alapaty, K. (1999). Numerical transport algorithms for the community multiscale air quality (cmaq) chemical transport model in generalized coordinates. In *Science Algorithms of the EPA Models-3 Community Multiscale Air Quality (CMAQ) Modeling System*. US-EPA Office of Research and Development Washington, DC 20460, EPA/600/R-99/030.

[Cakmur et al., 2004] Cakmur, R. V., Miller, R. L., and Torres, O. (2004). Incorporating the effect of small-scale circulations upon dust emission in an atmospheric general circulation model. *J. Geophys. Res.*, 109:D07201.

[Carlton et al., 2009] Carlton, A. G., Wiedinmyer, C., and Kroll, J. H. (2009). A review of secondary organic aerosol (soa) formation from isoprene. *Atmospheric Chemistry and Physics*, 9(14):4987–5005.

[Carter, 2010] Carter, W. (2010). Development of the saprc-07 chemical mechanism. *Atmos Environ*, 44(40):5324 – 5335.

[Carter, 1990] Carter, W. P. L. (1990). A detail mechanism for the gas-phase atmospheric reactions of organic compounds. *Atmos Environ*, 24:481–518.

[Cassiani et al., 2010] Cassiani, M., Vinuesa, J., Galmarini, S., and Denby, B. (2010). Stochastic fields method for sub-grid scale emission heterogeneity in mesoscale atmospheric dispersion models. *Atmospheric Chemistry and Physics*, 10:267–277.

[Chang, 1984] Chang, T. (1984). Rain and snow scavenging of hno3 vapor in the atmosphere. *Atmos. Enviro.*, 18:191 – 197.

[Cheinet and Teixeira, 2003] Cheinet, S. and Teixeira, J. (2003). A simple formulation for the eddy-diffusivity parameterization of cloud-topped boundary layers. *GEOPHYSICAL RESEARCH LETTERS*, 30(18).

[Cholakian et al., 2018] Cholakian, A., Beekmann, M., Colette, A., Coll, I., Siour, G., Sciare, J.and Marchand, N., Couvidat, F., Pey, J., Gros, V., Sauvage, S., Michoud, V., Sellegri, K., Colomb, A., Sartelet, K.and Langley DeWitt, H., Elser, M., PrÃl'vot, A. S. H., Szidat, S., and Dulac, F. (2018). Simulation of fine organic aerosols in the western mediterranean area during the charmex 2013 summer campaign. *Atmos. Chem. Phys.*, 18(10):7287–7312.

[Colella and Woodward, 1984] Colella, P. and Woodward, P. R. (1984). The piecewise parabolic method (PPM) for gas-dynamical simulations. *Journal of Computational Physics*, 11:38–39.

[Courant et al., 1952] Courant, R., Issacson, E., and Reees, M. (1952). On the solution of nonlinear hyperbolic differential equations by finite differences. *Comm. Pure Appl. Math.*, 5:243–255.

[Couvidat et al., 2018] Couvidat, F., Bessagnet, B., Garcia-Vivanco, M., Real, E., Menut, L., and Colette, A. (2018). Development of an inorganic and organic aerosol model (CHIMERE 2017β v1.0): seasonal and spatial evaluation over Europe. *Geoscientific Model Development*, 11(1):165–194.

[Couvidat et al., 2012] Couvidat, F., Debry, E., Sartelet, K., and Seigneur, C. (2012). A hydrophilic/hydrophobic organic (H2O) aerosol model: Development, evaluation and sensitivity analysis. *Journal of Geophysical Research Atmospheres*, 117:D10304.

[Couvidat and Sartelet, 2015] Couvidat, F. and Sartelet, K. (2015). The Secondary Organic Aerosol Processor (SOAP v1.0) model: a unified model with different ranges of complexity based on the molecular surrogate approach. *Geosci. Model Dev.*, 8:1111–1138.

[Craig et al., 2017] Craig, A., Valcke, S., and Coquart, L. (2017). Development and performance of a new version of the OASIS coupler, OASIS3-MCT_3.0. *Geoscientific Model Development*, 10(9):3297–3308.

[Davidson et al., 1982] Davidson, C., Miller, J., and Pleskow, M. (1982). The influence of surface structure on predicted particle air deposition to natural grass canopies. *Water, Air and Soil Pollution*, 18:25–44.

[De Moore et al., 1994] De Moore, W. B., Sandetr, S. P., Golden, D. M., Hampton, R. F., Kurylo, M. J., Howard, C. J., Ravishankara, A. R., Kolb, C. E., and Molina, M. J. (1994). Chemical kinetics and photochimical data for use in stratospheric modelling evaluation. *JPL publication*, 94, 26, JPL, Pasadena, US.

[Debry et al., 2007] Debry, E., Fahey, K., Sartelet, K., Sportisse, B., and Tombette, M. (2007). Technical Note: A new SIze REsolved Aerosol Model (SIREAM). *Atmos. Chem. Phys.*, 7:1537–1547.

[DeMott et al., 2015] DeMott, P. J., Prenni, A. J., McMeeking, G. R., Sullivan, R. C., Petters, M. D., Tobo, Y., Niemand, M., Möhler, O., Snider, J. R., Wang, Z., and Kreidenweis, S. M. (2015). Integrating laboratory and field data to quantify the immersion freezing ice nucleation activity of mineral dust particles. *Atmospheric Chemistry and Physics*, 15(1):393–409.

[Derognat et al., 2003] Derognat, C., Beekmann, M., Baeumle, M., Martin, D., and Schmidt, H. (2003). Effect of biogenic volatile organic compound emissions on tropospheric chemistry during the atmospheric pollution over the paris area (esquif) campaign in the ile-de-france region. *JOURNAL OF GEOPHYSICAL RESEARCH-ATMOSPHERES*, 108(D17).

[Després and Lagoutière, 1999] Després, B. and Lagoutière, F. (1999). Un schÃl'ma non linéaire anti-dissipatif pour l'équation d'advection linéaire. *Comptes Rendus de l'Académie des Sciences - Series I - Mathematics*, 328(10):939 – 943.

[Ebel et al., 1994] Ebel, A., Friedrich, R., and Rodhe, H. (1994). Tropospheric modelling and emission estimation: Generation of european emission data for episodes (genemis) project, eurotrac annual report 1993, part 5.

[Elliott, 2009] Elliott, S. (2009). Dependence of dms global sea-air flux distribution on transfer velocity and concentration field type. *Journal of Geophysical Research: Biogeosciences*, 114(G2).

[Emberson et al., 2000] Emberson, L.-D., Simpson, D., Tuovinen, J.-P., Ashmore, M.-R., and Cambridge, H.-M. (2000). Towards a model of ozone deposition and stomatal uptake over Europe. Technical report. EMEP/MSC-W 2000.

[Erisman et al., 1994] Erisman, J. W., van Pul, A., and Wyers, P. (1994). Parameterization of surface resistance for the quantification of atmospheric deposition of acidifying pollutants and ozone. *Atmospheric Environemnt*, 28:2595–2607.

[Fecan et al., 1999] Fecan, F., Marticorena, B., and Bergametti, G. (1999). Parameterization of the increase of aeolian erosion threshold wind friction velocity due to soil moisture for arid and semi-arid areas. *Annals of Geophysics*, 17:149–157.

[Folberth et al., 2006] Folberth, G. A., Hauglustaine, D. A., Lathière, J., and Brocheton, F. (2006). Interactive chemistry in the laboratoire de météorologie dynamique general circulation model: model description and impact analysis of biogenic hydrocarbons on tropospheric chemistry. *Atmos. Chem. Phys.*, 6:2273–2319.

[Fredenslund et al., 1975] Fredenslund, A., Jones, R. L., and Prausnitz, J. M. (1975). Group-contribution estimation of activity-coefficients in nonideal liquid-mixtures. *AIChE J.*, 21:1086–1099.

[Galmarini et al., 2008] Galmarini, S., Vinuesa, J.-F., and Martilli, A. (2008). Modeling the impact of sub-grid scale emission variability on upper-air concentration. *Atmospheric Chemistry and Physics*, 8(2):141–158. Publisher: Copernicus GmbH.

[Ghan et al., 1997] Ghan, S. J., Leung, L. R., Easter, R. C., and Abdul-Razzak, H. (1997). Prediction of cloud droplet number in a general circulation model. *Journal of Geophysical Research: Atmospheres*, 102(D18):21777–21794.

[Ginoux et al., 2001] Ginoux, P., Chin, M., Tegen, I., Prospero, J. M., Holben, B., Dubovik, O., and Lin, S. J. (2001). Sources and distributions of dust aerosols simulated with the GOCART model. *Journal of Geophysical Research*, 106:20255–20273.

[Giorgi, 1986] Giorgi, F. (1986). A particle dry deposition scheme for use in tracer transport models'. *J. Geophys. Res.*, 91:9794–9806.

[Gordon, 1997] Gordon, H. (1997). Atmospheric correction of ocean color imagery in the earth observing system era. *J. Geophys. Res.*, 102(D14):17081–17106.

[Grieshop et al., 2009] Grieshop, A. P., Logue, J. M., Donahue, N. M., and Robinson, A. L. (2009). Laboratory investigation of photochemical oxidation if organic aerosol from wood fires. 1: measurement and simulation of organic aerosol evolution. *Atmos. Chem. Phys.*, 9:1263–1277.

[Griffin et al., 1999] Griffin, R. J., Cocker, E. R., Flagan, R. C., and Seinfeld, J. H. (1999). Organic aerosol formation from the oxidation of biogenic hydrocarbons. *J of Geophysical Research*, 104:3555–3567.

[Grythe et al., 2014] Grythe, H., Ström, J., Krejci, R., Quinn, P., and Stohl, A. (2014). A review of sea-spray aerosol source functions using a large global set of sea salt aerosol concentration measurements. *Atmospheric Chemistry and Physics*, 14(3):1277–1297.

[Guenther et al., 2006] Guenther, A., Karl, T., Harley, P., Wiedinmyer, C., Palmer, P., and Geron, C. (2006). Estimates of global terrestrial isoprene emissions using MEGAN (Model of Emissions of Gases and Aerosols from Nature). *Atmos. Chem. Phys.*, 6:3181–3210.

[Hall et al., 1999] Hall, S., Khudaish, E., and Hart, A. (1999). Electrochemical oxidation of hydrogen peroxide at platinum electrodes. part iii: Effect of temperature. electrochim. *Electrochim. Acta*, 44:2455–2462.

[Harrison and Kito, 1990] Harrison, R. and Kito, A. (1990). Field intercomparison of filter pack and denuder sampling methods for reactive gaseous and particulate pollutants. *Atmos. Environ.*, 24:2633–2640.

[Hauglustaine et al., 2014] Hauglustaine, D. A., Balkanski, Y., and Schulz, M. (2014). A global model simulation of present and future nitrate aerosols and their direct radiative forcing of climate. *Atmos Chem Phys*, 14:6863–6949.

[Hauglustaine et al., 2004] Hauglustaine, D. A., Hourdin, F., Jourdain, L., Filiberti, M.-A., Walters, S., Lamarque, J.-F., and Holland, E. A. (2004). Interactive chemistry in the Laboratoire de Meteorologie Dynamique general circulation model: Description and background tropospheric chemistry evaluation. *Journal of Geophysical Research*, 109(D04314). doi:10.1029/2003JD003,957.

[Hicks et al., 1987] Hicks, B. B., Baldocchi, D. D., Meyers, T. P., Hosker Jr., R. P., and Matt, D. R. (1987). A preliminary multiple resistance routine for deriving dry deposition velocities from measured quantities. *Water, Air, Soil Pollut.*, 36.

[Hodzic et al., 2006a] Hodzic, A., Bessagnet, B., and Vautard, R. (2006a). A model evaluation of coarse-mode nitrate heterogeneous formation on dust particles. *Atmos. Environ.*, 40:4158–4171.

[Hodzic et al., 2005] Hodzic, A., Vautard, R., Bessagnet, B., Lattuati, M., and Moreto, F. (2005). Long-term urban aerosol simulation versus routine particulate matter observations. *Atmospheric Environemnt*, 39:5851–5864.

[Hodzic et al., 2006b] Hodzic, A., Vautard, R., Chazette, P., Menut, L., and B.Bessagnet (2006b). Aerosol chemical and optical properties over the Paris area within ESQUIF project. *Atmospheric Chemistry and Physics*, 6:3257–3280.

[Hoffmann and Calvert, 1985] Hoffmann, M. R. and Calvert, J. G. (1985). Chemical transformation modules for eulerian acid deposition models,. EPA/600/3-85/017.

[Honoré et al., 2008] Honoré, C., Rouïl, L., Vautard, R., Beekmann, M., Bessagnet, B., Dufour, A., Elichegaray, C., Flaud, J., Malherbe, L., Meleux, F., Menut, L., Martin, D., Peuch, A., Peuch, V., and Poisson, N. (2008). Predictability of European air quality: The assessment of three years of operational forecasts and analyses by the PREV'AIR system. *Journal of Geophysical Research*, 113:D04301.

[Honoré and Vautard, 2000] Honoré, C. and Vautard, R. (2000). Photochemical regimes in urban atmospheres: The influence of dispersion. *Geophys. Res. Letters*, 27:1895–1898.

[Horowitz, 2003] Horowitz, L. W., e. a. (2003). A global simulation of tropospheric ozone and related tracers: Description and evaluation of mozart, version 2. *J Geophys Res*, 108(D24):4784.

[Hostetler et al., 2006] Hostetler, C. A., Liu, Z., Reagan, J., Vaughan, M., Winker, D., Osborn, M., Hunt, W. H., Powell, K. A., , and Trepte, C. (2006). Caliop algorithm theoretical basis document, calibration and level 1 data products. *PC-SCI-201*, (NASA Langley Research Center):Hampton, VA 23681.

[Hourdin and Armengaud, 1999] Hourdin, F. and Armengaud, A. (1999). On the use of finite volume methods for atmospheric advection of trace species. Part I: Test of various formulations in a general circulation models. *Monthly Weather Review*, 127:822–837.

[Iacono et al., 2008] Iacono, M. J., Delamere, J. S., Mlawer, E. J., Shephard, M. W., Clough, S. A., and Collins, W. D. (2008). Radiative forcing by long-lived greenhouse gases: Calculations with the aer radiative transfer models. *Journal of Geophysical Research: Atmospheres*, 113(D13).

[Irvine, 1976] Irvine, M. (1976). The asymmetry of the scattering diagram of a spherical particle. *Bulletin of the Astronomical Institutes of the Netherlands*, 17:176.

[Iversen and White, 1982] Iversen, J. D. and White, B. R. (1982). Saltation threshold on earth, mars and venus. *Sedimentology*, 29:111–119.

[Jacob, 2000] Jacob, D. J. (2000). Heterogeneous chemistry and tropospheric ozone. *Atmos Environ*, 34(34):2131–2159.

[Jacobson, 1999] Jacobson, M. Z. (1999). *Fundamentals of atmospheric modeling*. Cambridge University Press.

[Journet et al., 2014] Journet, E., Balkanski, Y., and Harrison, S. P. (2014). A new data set of soil mineralogy for dust-cycle modeling. *Atmospheric Chemistry and Physics*, 14(8):3801–3816.

[Kaiser et al., 2012] Kaiser, J. W., Heil, A., Andreae, M. O., Benedetti, A., Chubarova, N., Jones, L., Morcrette, J.-J., Razinger, M., Schultz, M. G., Suttie, M., and van der Werf, G. R. (2012). Biomass burning emissions estimated with a global fire assimilation system based on observed fire radiative power. *Biogeosciences*, 9(1):527–554.

[Kandler et al., 2007] Kandler, K., Benker, N., Bundke, U., Cuevas, E., Ebert, M., Knippertz, P., Rodriguez, S., Schutz, L., and Weinbruch, S. (2007). Chemical composition and complex refractive index of saharan mineral dust at izana, tenerife (spain) derived by electron microscopy. *Atmos Environ*, 41:8058–8074.

[Kavouras et al., 1998] Kavouras, I., Mihalopoulos, N., and Stephanou, E. (1998). Formation of atmospheric particles from organic acids produced by forests. *Nature*, 395:683–686.

[Kettle et al., 1999] Kettle, A. J., Andreae, M. O., Amouroux, D., Andreae, T. W., Bates, T. S., Berresheim, H., Bingemer, H., Boniforti, R., Curran, M. A. J., DiTullio, G. R., Helas, G., Jones, G. B., Keller, M. D., Kiene, R. P., Leck, C., Levasseur, M., Malin, G., Maspero, M., Matrai, P., McTaggart, A. R., Mihalopoulos, N., Nguyen, B. C., Novo, A., Putaud, J. P., Rapsomanikis, S., Roberts, G., Schebeske, G., Sharma, S., Simo, R., Staubes, R., Turner, S., and Uher, G. (1999). A global database of sea surface dimethylsulfide (DMS) measurements and a procedure to predict sea surface DMS as a function of latitude, longitude, and month. *Global Biogeochemical Cycles*, 13(2):399–444.

[Kim et al., 2019] Kim, Y., Sartelet, K., and Couvidat, F. (2019). Modeling the effect of non-ideality, dynamic mass transfer and viscosity on soa formation in a 3-d air quality model. *Atmos. Chem. Phys.*, 19:1241–1261.

[Kok et al., 2014] Kok, J. F., Mahowald, N. M., Fratini, G., Gillies, J. A., Ishizuka, M., Leys, J. F., Mikami, M., Park, M.-S., Park, S.-U., Van Pelt, R. S., and Zobeck, T. M. (2014). An improved dust emission model ̆2013 part 1: Model description and comparison against measurements. *Atmospheric Chemistry and Physics*, 14(23):13023–13041.

[Koop et al., 2001] Koop, T., Luo, B., Tsias, A., and Peter, T. (2001). Water activity as the determinant for homogeneous ice nucleation in aqueous solutions. *Nature*, (406):611–614.

[Korsakissok and Mallet, 2010] Korsakissok, I. and Mallet, V. (2010). Development and application of a reactive plume-in-grid model: evaluation over Greater Paris. *Atmospheric Chemistry and Physics*, 10(18).

[Kouznetsov and Sofiev, 2012] Kouznetsov, R. and Sofiev, M. (2012). A methodology for evaluation of vertical dispersion and dry deposition of atmospheric aerosols. *Journal of Geophysical Research: Atmospheres*, 117(D1).

[Kroll et al., 2006] Kroll, J. H., Ng, N. L., Murphy, S. M., Flagan, R. C., and Seinfeld, J. H. (2006). Secondary organic aerosol formation from isoprene photooxidation. *Environ. Sci. Technol.*, 40:1869–1877.

[Kulmala et al., 1998] Kulmala, M., A., L., and Pirjola, L. (1998). Parameterization for sulfuric acid / water nucleation rates. *J. Geophys. Res.*, 103(No D7):8301–8307.

[Lachatre et al., 2020] Lachatre, M., Mailler, S., Menut, L., Turquety, S., Sellitto, P., Guermazi, H., Salerno, G., , Caltabiano, T., and Carboni, E. (2020). New strategies for vertical transport in chemistry-transport models: application to the case of the mount etna eruption on march 18, 2012. *Geosci. Model Dev. Discuss.*

[Laepple et al., 2005] Laepple, T., Schultz, M. G., Lamarque, J. F., Madronich, S., Shetter, R. E., Lefer, B. L., and Atlas, E. (2005). Improved albedo formulation for chemistry transport models based on satellite observations and assimilated snow data and its impact on tropospheric photochemistry. *J. Geophys. Res.*, 110:D11308.

[Lana et al., 2011] Lana, A., Bell, T. G., Simo, R., Vallina, S. M., Ballabrera-Poy, J., Kettle, A. J., Dachs, J., Bopp, L., Saltzman, E. S., Stefels, J., Johnson, J. E., and Liss, P. S. (2011). An updated climatology of surface dimethlysulfide concentrations and emission fluxes in the global ocean. *Global Biogeochemical Cycles*, 25(1).

[Lattuati, 1997] Lattuati, M. (1997). *Contribution à l'étude du bilan de l'ozone troposphérique à l'interface de l'Europe et de l'Atlantique Nord: modélisation lagrangienne et mesures en altitude*. Phd thesis, Université P.M.Curie, Paris, France.

[Lee and Schwartz, 1983] Lee, Y. N. and Schwartz, S. E. (1983). *Precipitation scavenging, dry deposition and resuspension, vol. 1.*, chapter Kinetics of oxidation of aqueous sulfur (IV) by nitrogen dioxide. Elsevier, New York.

[Liss and Merlivat, 1986] Liss, P. S. and Merlivat, L. (1986). *Air-Sea Gas Exchange Rates: Introduction and Synthesis*, pages 113–127. Springer Netherlands, Dordrecht.

[Loosmore, 2003] Loosmore, G. (2003). Evaluation and development of models for resuspension of aerosols at short times after deposition. *Atmospheric Environment*, 37:639–647.

[Louis et al., 1982] Louis, J., Tiedke, M., and Geleyn, J. (1982). A short history of the pbl parametrization at ecmwf. In *ECMWF Workshop on Planetary Boundary Layer parametrization*, pages 59–80, University of Reading.

[Loveland et al., 2000] Loveland, T., Reed, B., Brown, J., Ohlen, D., Zhu, J., Yang, L., and Merchant, J. (2000). Development of a global land cover characteristics database and igbp discover from 1-km avhrr data. *International Journal of Remote Sensing*, 21(6/7):1303–1330.

[Mailler et al., 2013] Mailler, S., Khvorostyanov, D., , and Menut, L. (2013). Impact of the vertical emission profiles on ground-level gas-phase pollution simulated from the EMEP emissions over Europe. *Atmospheric Chemistry and Physics*, 13:5987–5998.

[Mailler et al., 2016] Mailler, S., Menut, L., di Sarra, A. G., Becagli, S., Di Iorio, T., Bessagnet, B., Briant, R., Formenti, P., Doussin, J.-F., Gómez-Amo, J. L., Mallet, M., Rea, G., Siour, G., Sferlazzo, D. M., Traversi, R., Udisti, R., and Turquety, S. (2016). On the radiative impact of aerosols on photolysis rates: comparison of simulations and observations in the lampedusa island during the charmex/adrimed campaign. *Atmospheric Chemistry and Physics*, 16(3):1219–1244.

[Mailler et al., 2017] Mailler, S., Menut, L., Khvorostyanov, D., Valari, M., Couvidat, F., Siour, G., Turquety, S., Briant, R., Tuccella, P., Bessagnet, B., Colette, A., Létinois, L., Markakis, K., and Meleux, F. (2017). CHIMERE-2017: from urban to hemispheric chemistry-transport modeling. *Geoscientific Model Development*, 10(6):2397–2423.

[Martensson et al., 2003] Martensson, E. M., Nilsson, E. D., de Leeuw, G., Cohen, L. H., and Hansson, H.-C. (2003). Laboratory simulations and parameterization of the primary marine aerosol production. *Journal of Geophysical Research: Atmospheres*, 108(D9).

[Marticorena and Bergametti, 1995] Marticorena, B. and Bergametti, G. (1995). Modeling the atmospheric dust cycle: 1 Design of a soil derived dust production scheme. *J. Geophys. Res.*, 100:16415–16430.

[May et al., 2013] May, A. A., Levin, E. J. T., Hennigan, C. J., Riipinen, I., Lee, T., Collett Jr., J. L., Jimenez, J. L., Kreidenweis, S. M., and Robinson, A. L. (2013). Gas-particle partitioning of primary organic aerosol emissions: 3. biomass burning. *J. Geophys. Res. (Atmospheres)*, 118:11,327–11,338.

[Menut, 2008] Menut, L. (2008). Sensitivity of hourly Saharan dust emissions to NCEP and ECMWF modelled wind speed. *J Geophys Res*, 113:D16201.

[Menut et al., 2013a] Menut, L., Bessagnet, B., Khvorostyanov, D., Beekmann, M., Blond, N., Colette, A., Coll, I., Curci, G., Foret, F., Hodzic, A., Mailler, S., Meleux, F., Monge, J., Pison, I., Siour, G., Turquety, S., Valari, M., Vautard, R., and Vivanco, M. (2013a). CHIMERE 2013: a model for regional atmospheric composition modelling. *Geoscientific Model Development*, 6:981–1028.

[Menut et al., 2020a] Menut, L., Bessagnet, B., Mailler, S., Pennel, R., and Siour, G. (2020a). Impact of lightning nox emissions on atmospheric composition and meteorology in africa and europe. *Atmosphere*, 11(10).

[Menut et al., 2005] Menut, L., C.Schmechtig, and B.Marticorena (2005). Sensitivity of the sandblasting fluxes calculations to the soil size distribution accuracy. *Journal of Atmospheric and Oceanic Technology*, 22:1875–1884.

[Menut et al., 2018] Menut, L., Flamant, C., Turquety, S., Deroubaix, A., Chazette, P., and Meynadier, R. (2018). Impact of biomass burning on pollutant surface concentrations in megacities of the Gulf of Guinea. *Atmospheric Chemistry and Physics*, 18(4):2687–2707.

[Menut et al., 2007] Menut, L., G.Foret, and G.Bergametti (2007). Sensitivity of mineral dust concentrations to the model size distribution accuracy. *Journal of Geophysical Research, Atmospheres*, 112:D10210.

[Menut et al., 2012] Menut, L., Goussebaile, A., Bessagnet, B., Khvorostiyanov, D., and Ung, A. (2012). Impact of realistic hourly emissions profiles on modelled air pollutants concentrations. *Atmos Environ*, 49:233–244.

[Menut et al., 2013b] Menut, L., Perez Garcia-Pando, C., Haustein, K., Bessagnet, B., Prigent, C., and Alfaro, S. (2013b). Relative impact of roughness and soil texture on mineral dust emission fluxes modeling. *J Geophys Res*, 118:6505–6520.

[Menut et al., 2020b] Menut, L., Siour, G., Bessagnet, B., Couvidat, F., Journet, E., Balkanski, Y., and Desboeufs, K. (2020b). Modelling the mineralogical composition and solubility of mineral dust in the Mediterranean area with CHIMERE 2017r4. *Geoscientific Model Development*, 13(4):2051–2071.

[Menut et al., 2000a] Menut, L., Vautard, R., Beekmann, M., and Honoré, C. (2000a). Sensitivity of Photochemical Pollution using the Adjoint of a Simplified Chemistry-Transport Model. *J. Geophys. Res.*, 105(D12):15379–15402.

[Menut et al., 2000b] Menut, L., Vautard, R., Flamant, C., Abonnel, C., Beekmann, M., Chazette, P., Flamant, P., Gombert, D., Guédalia, D., Lefebvre, M., Lossec, B., D., M., Mégie, G., Perros, P., Sicard, M., and Toupance, G. (2000b). Measurements and modelling of atmospheric pollution over the Paris area: an overview of the ESQUIF Project. *Annales Geophysicae*, 18(11):1467–1481.

[Middleton et al., 1990] Middleton, P., Stockwell, W. R., and Carter, W. P. (1990). Agregation and analysis of volatile organic compound emissions for regional modelling. *Atmos Environ*, 24:1107–1133.

[Mihalopoulos et al., 2007] Mihalopoulos, N., Kerminen, V., Kanakidou, M., Berresheim, H., and Sciare, J. (2007). Formation of particulate sulfur species (sulfate and methanesulfonate) during summer over the eastern mediterranean: A modelling approach. *Atmospheric Environment*, 41(32):6860 – 6871.

[Monahan, 1986] Monahan, E. C. (1986). *In The Role of Air-Sea Exchange in Geochemical Cycling*, chapter The ocean as a source of atmospheric particles, pages 129–163. Kluwer Academic Publishers, Dordrecht, Holland.

[Nenes et al., 1998] Nenes, A., Pilinis, C., and Pandis, S. (1998). ISORROPIA: A new thermodynamic model for inorganic multicomponent atmospheric aerosols. *Aquatic Geochem.*, 4:123–152.

[Nightingale et al., 2000] Nightingale, P. D., Malin, G., Law, C. S., Watson, A. J., Liss, P. S., Liddicoat, M. I., Boutin, J., and Upstill-Goddard, R. C. (2000). In situ evaluation of air-sea gas exchange parameterizations using novel conservative and volatile tracers. *Global Biogeochemical Cycles*, 14(1):373–387.

[Odum et al., 1997] Odum, J. R., Jungkamp, T. P. W., Griffin, R. J., Forster, H. J. L., Flagan, R. C., and Seinfeld, J. H. (1997). Aromatics, reformulated gasoline and atmospheric organic aerosol formation. *Environ. Sci. Technol.*, 31:1890–1897.

[Omar et al., 2010] Omar, A., Winker, D. M., Vaughan, M. A., Hu, Y., Trepte, C. R., Ferrare, R. A., Lee, K.-P., Hostetler, C. A., Kittaka, C., Rogers, R. R., Kuehn, R. E., and Liu, Z. (2010). The calipso automated aerosol classification and lidar ratio selection algorithm. *Journal of Atmospheric and Oceanic Technology*, 26(10):1994–2014.

[Pandis et al., 1993] Pandis, S., Wexler, A., and Seinfeld, J. (1993). Secondary organic aerosol formation and transport -ii. predicting the ambient secondary organic aerosol size distribution. *Atmos. Environ.*, 27A:2403–2416.

[Pankow, 1994] Pankow, J. F. (1994). An absorption model of gas/aerosol partition involved in the formation of secondary organic aerosol. *Atmos. Environ.*, 28:189–193.

[Passant, 2002] Passant, N. (2002). Speciation of uk emissions of nmvoc. (AEAT/ENV/R/0545).

[Pavia and O'Brien, 1986] Pavia, E. and O'Brien, J. (1986). Weibull statistics speed over the ocean. *Journal of Climate and Applied Meteorology*, 25:324–332.

[Péré et al., 2014] Péré, J., Bessagnet, B., Mallet, M., Waquet, F., Chiapello, I., Minvielle, F., Pont, V., and Menut, L. (2014). Direct radiative effect of the Russian wildfires and their impact on air temperature and atmospheric dynamics during August 2010. *Atmospheric Chemistry and Physics*, 14:1999–2013.

[Perlwitz et al., 2015] Perlwitz, J. P., Pérez García-Pando, C., and Miller, R. L. (2015). Predicting the mineral composition of dust aerosols - part 1: Representing key processes. *Atmospheric Chemistry and Physics*, 15(20):11593–11627.

[Peters and Eiden, 1992] Peters, K. and Eiden, R. (1992). Modelling the dry deposition velocity of aerosol particles to a spruce forest. *Atmos Environ*, 26A(14):2555–2564.

[Petters and Kreidenweis, 2007] Petters, M. D. and Kreidenweis, S. M. (2007). A single parameter representation of hygroscopic growth and cloud condensation nucleus activity. *Atmospheric Chemistry and Physics*, 7(8):1961–1971.

[Priestley, 1949] Priestley, C. (1949). Heat transport and zonal stress between latitudes. *Quaterly Journal of the Royal Meteorology*, 75:28–40.

[Prigent et al., 2012] Prigent, C., Jiménez, C., and Catherinot, J. (2012). Comparison of satellite microwave backscattering (ASCAT) and visible/near-infrared reflectances (PARASOL) for the estimation of aeolian aerodynamic roughness length in arid and semi-arid regions. *Atmos. Meas. Tech.*, 5:2703–2712.

[Prigent et al., 2005] Prigent, C., Tegen, I., Aires, F., Marticorena, B., and Zribi, M. (2005). Estimation of the aerodynamic roughness length in arid and semi-arid regions over the globe with the ERS scatterometer. *J. Geophys. Res.*, 110:D09205.

[Pryor et al., 2005] Pryor, S., Schoof, J., and Barthelmie, R. (2005). Empirical downscaling of wind speed probability distributions. *J. Geophys. Res.*, 110:D19109.

[Pun et al., 2006] Pun, B. K., Seigneur, C., and Lohman, K. (2006). Modeling secondary organic aerosol formation via multiphase partitioning with molecular data. *Environ. Sci. Technol.*, 40:4722–4731.

[Real and Sartelet, 2011] Real, E. and Sartelet, K. (2011). Modeling of photolysis rates over europe: impact on chemical gaseous species and aerosols. *Atmos. Chem. Phys*, 11:1711–1727.

[Robinson et al., 2007] Robinson, A., Donahue, N., Shrivastava, M. K., Weitkamp, E. A., Sage, A. M., Grieshop, A. P., Lane, T. E., Pierce, J. R., and Pandis, S. N. (2007). Rethinking Organic Aerosols: Semivolatile Emissions and Photochemical Aging. *Science*, 315(5816):1259–1262.

[Rouïl et al., 2009] Rouïl, L., Honoré, C., Vautard, R., Beekmann, M., Bessagnet, B., Malherbe, L., Meleux, F., Dufour, A., Elichegaray, C., Flaud, J., Menut, L., Martin, D., Peuch, A., Peuch, V., and Poisson, N. (2009). PREV'AIR : an operational forecasting and mapping system for air quality in Europe. *BAMS*, 90:73–83.

[Sander et al., 1999] Sander, G.C. and. Parlange, J.-Y., Smith, R.E.and Haverkamp, R., and Hogarth, W. (1999). Estimation of ponding time for constant surface flux. *Hydrology days Ed. AGU Pub. 19, San Francisco*, pages 402–410.

[Schmidt et al., 2001] Schmidt, H., Derognat, C., Vautard, R., and Beekmann, M. (2001). A comparison of simulated and observed ozone mixing ratios for the summer of 1998 in western Europe. *Atmospheric Environment*, 35:6277–6297.

[Schulz et al., 2006] Schulz, M., Textor, C., Kinne, S., Balkanski, Y., Bauer, S., Berntsen, T., Berglen, T., Boucher, O., Dentener, F., Guibert, S., Isaksen, I. S. A., Iversen, T., Koch, D., Kirkevag, A., Liu, X., Montanaro, V., Myhre, G., Penner, J. E., Pitari, G., Reddy, S., Seland, O., Stier, P., and Takemura, T. (2006). Radiative forcing by aerosols as derived from the aerocom present-day and pre-industrial simulations. *Atmos Chem Phys*, 6:5225–5246.

[Seinfeld and Pandis, 2006] Seinfeld, J. and Pandis, S. (2006). *Atmospheric Chemistry and Physics: From Air Pollution to Climate Change*. Wiley-Interscience, Hoboken, 2nd edition.

[Seinfeld and Pandis, 1998] Seinfeld, J. H. and Pandis, S. N. (1998). *Atmospheric chemistry and physics: From air pollution to climate change*. Wiley-Interscience, J.Wiley, New York.

[Sekhon and Srivastava, 1970] Sekhon, K. S. and Srivastava, R. C. (1970). Snow size spectra and radar reflectivity. *J. Atmos. Sci.*, 27:299–307.

[Semmler et al., 2006] Semmler, M., Luo, B. P., and Koop, T. (2006). Densities of liquid h+/nh4+/so42-/no3-/h2o solutions at tropospheric temperatures. *Atmos. Environ.*, 40:467–483.

[Shao and Lu, 2000] Shao, Y. and Lu, I. (2000). A simple expression for wind erosion threshold friction velocity. *J. Geophys. Res.*, 105:22,437–22,443.

[Shrivastava et al., 2015] Shrivastava, M., Easter, R. C., Liu, X., Zelenyuk, A., Singh, B., Zhang, K., Ma, P.-l., Chand, D., Ghan, S., Jimenez, J. L., Zhang, Q., Fast, J., Rasch, P. J., and Tiitta, P. (2015). Global transformation and fate of SOA: Implications of low-volatility SOA and gas-phase fragmentation reactions. *Journal of Geophysical Research : Atmospheres*, 120(9):4169–4195.

[Shrivastava et al., 2011] Shrivastava, M., Fast, J., Easter, R., Gustafson, W. I., Zaveri, R. A., Jimenez, J. L., Saide, P., and Hodzic, A. (2011). Modeling organic aerosols in a megacity: Comparison of simple and complex representations of the volatility basis set approach. *Atmospheric Chemistry and Physics*, 11(13):6639–6662.

[Shrivastava et al., 2013] Shrivastava, M., Zelenyuk, A., Imre, D., Easter, R., Beranek, J., Zaveri, R. A., and Fast, J. (2013). Implications of low volatility SOA and gas-phase fragmentation reactions on SOA loadings and their spatial and temporal evolution in the atmosphere. *Journal of Geophysical Research Atmospheres*, 118(8):3328–3342.

[Simpson, 1992] Simpson, D. (1992). Long period modeling of photochemical oxidants in Europe, Calculations for July 1985. *Atmos. Environ.*, 26:1609–1634.

[Simpson et al., 2012] Simpson, D., Benedictow, A., Berge, H., Bergström, R., Emberson, L. D., Fagerli, H., Flechard, C. R., Hayman, G. D., Gauss, M., Jonson, J. E., Jenkin, M. E., Nyíri, A., Richter, C., Semeena, V. S., Tsyro, S., Tuovinen, J.-P., Valdebenito, A., and Wind, P. (2012). The emep msc-w chemical transport model - technical description. *Atmospheric Chemistry and Physics*, 12(16):7825–7865.

[Simpson et al., 2003] Simpson, D., Fagerli, H., Jonson, J.-E., Tsyro, S., Wind, P., and Tuovinen, J.-P. (2003). Transboundary Acidification, Eutrophication and Ground Level Ozone in Europe. Part I: Unified EMEP Model Description. Technical report. EMEP/ Status Report 2003.

[Skamarock et al., 2007] Skamarock, W., Klemp, J., Dudhia, J., Gill, D., Barker, D., Wang, W., and Powers, J. (2007). A description of the advanced research wrf version 2. *NCAR Technical Note, Boulder, Colorado, USA*, pages NCAR/TN–468+STR.

[Slinn, 1982] Slinn, W. (1982). Predictions for particle deposition to vegetation surfaces. *Atmospheric Environment*, 23:1293–1304.

[Sofiev et al., 2012] Sofiev, M., Ermakova, T., and Vankevich, R. (2012). Evaluation of the smoke-injection height from wild-land fires using remote-sensing data. *Atmospheric Chemistry and Physics*, 12(4):1995–2006.

[Stromatas et al., 2012] Stromatas, S., Turquety, S., Menut, L., Chepfer, H., Cesana, G., Pere, J., and Bessagnet, B. (2012). Lidar signal simulation for the evaluation of aerosols in chemistry-transport models. *Geoscientific Model Development*, 5.

[Tanskannen and Manninen, 2007] Tanskannen, A. and Manninen, T. (2007). Effective UV surface albedo of seasonally snow-covered lands. *Atmos. Chem. Phys.*, 7:2759–2764.

[Tegen et al., 2002] Tegen, I., Harrison, S. P., Kohfeld, K., Prentice, I. C., Coe, M., and Heimann, M. (2002). Impact of vegetation and preferential source areas on global dust aerosol: Results from a model study. *J. Geophys. Res.*, 109.

[Telford et al., 2013] Telford, P. J., Abraham, N. L., Archibald, A. T., Braesicke, P., Dalvi, M., Morgenstern, O., OĆonnor, F. M., Richards, N. A. D., and Pyle, J. A. (2013). Implementation of the Fast-JX photolysis scheme (v6.4) into the UKCA component of the MetUM chemistry-climate model (v7.3). *Geoscientific Model Development*, 6:161–177.

[Terrenoire et al., 2015] Terrenoire, E., Bessagnet, B., Rouïl, L., Tognet, F., Pirovano, G., Létinois, L., Beauchamp, M., Colette, A., Thunis, P., Amann, M., and Menut, L. (2015). High-resolution air quality simulation over Europe with the chemistry transport model CHIMERE. *Geoscientific Model Development*, 8(1):21–42.

[Thompson and Eidhammer, 2014] Thompson, G. and Eidhammer, T. (2014). A study of aerosols impatcs on clouds and precipitation development in a large winter cyclone. *J Atmos Sci*, 71:3636–3659.

[Tiedtke, 1989] Tiedtke, M. (1989). A comprehensive mass flux scheme for cumulus parameterization in large-scale models. *Monthly Weather Reviews*, 117:1779–1800.

[Troen and Mahrt, 1986] Troen, I. and Mahrt, L. (1986). A simple model of the atmospheric boundary layer: Sensitivity to surface evaporation. *Boundary-Layer Meteorology*, 37:129–148.

[Tuccella et al., 2019] Tuccella, P., Menut, L., Briant, R., Deroubaix, A., Khvorostyanov, D., Mailler, S., Siour, G., and Turquety, S. (2019). Implementation of Aerosol-Cloud Interaction within WRF-CHIMERE Online Coupled Model: Evaluation and Investigation of the Indirect Radiative Effect from Anthropogenic Emission Reduction on the Benelux Union. *Atmosphere*, 10(20).

[Turquety et al., 2014] Turquety, S., Menut, L., Bessagnet, B., Anav, A., Viovy, N., Maignan, F., and Wooster, M. (2014). APIFLAME v1.0: High resolution fire emission model and application to the Euro-Mediterranean region. *Geoscientific Model Development*, 7:587–612.

[Turquety et al., 2019] Turquety, S., Menut, L., Siour, G., Mailler, S., Hadji-Lazaro, J., George, M., Clerbaux, C., Hurtmans, D., and Coheur, P.-F. (2019). Apiflame v2.0 trace gas and aerosol emissions from biomass burning: application to portugal during the summer of 2016 and evaluation against satellite observations of co (iasi) and aod (modis). *Geoscientific Model Development Discussions*, 2019:1–38.

[Ullrich et al., 2010] Ullrich, P. A., Jablonowski, C., and Leer, B. V. (2010). High-order finite-volume methods for the shallow-water equations on the sphere. *J. Comput. Phys*, 229:6104–6134.

[Valari and Menut, 2010] Valari, M. and Menut, L. (2010). Transferring the heterogeneity of surface emissions to variability in pollutant concentrations over urban areas through a chemistry transport model. *Atmospheric Environment*, 44:3229–3238.

[Van Leer, 1977] Van Leer, B. (1977). Towards the ultimate conservative difference scheme. IV. A new approach to Numerical convection. *J. Computational Phys.*, 23:276–299.

[Vautard et al., 2005a] Vautard, R., B.Bessagnet, M.Chin, and Menut, L. (2005a). On the contribution of natural Aeolian sources to particulate matter concentrations in Europe: testing hypotheses with a modelling approach. *Atmospheric Environment*, 39:3291–3303.

[Vautard et al., 2000] Vautard, R., Beekmann, M., and Menut, L. (2000). Applications of adjoint modelling in atmospheric chemistry: sensitivity and inverse modelling. *Environmental Modelling and Software*, 15:703–709.

[Vautard et al., 2001] Vautard, R., Beekmann, M., Roux, J., and Gombert, D. (2001). Validation of a hybrid forecasting system for the ozone concentrations over the Paris area. *Atmos. Environ.*, 35:2449–2461.

[Vautard et al., 2005b] Vautard, R., Honoré, C., Beekmann, M., and Rouil, L. (2005b). Simulation of ozone during the August 2003 heat wave and emission control scenarios. *Atmospheric Environment*, 39(16):2957–2967.

[Veira et al., 2015] Veira, A., Kloster, S., Wilkenskjeld, S., and Remy, S. (2015). Fire emission heights in the climate system - part 1: Global plume height patterns simulated by echam6-ham2. *Atmospheric Chemistry and Physics*, 15(13):7155–7171.

[Verwer, 1994] Verwer, J. (1994). Gauss-Seidel iteration for stiff odes from chemical kinetics. *Journal on Scientific Computing*, 15:1243–1250.

[Voulgarakis et al., 2009] Voulgarakis, A., Savage, N. H., Wild, O., Carver, G. D., Clemitshaw, K. C., and Pyle, J. A. (2009). Upgrading photolysis in the PHOTOMCAT CTM: model evaluation and assessment of the role of clouds. *Geoscientific Model Development*, 2:59–72.

[Vuolo et al., 2009] Vuolo, M., Menut, L., and Chepfer, H. (2009). Impact of transport schemes accuracy on modelled dust concentrations variability. *Journal of Atmospheric and Oceanic Technology*, 26:1135–1143.

[Wandinger et al., 2010] Wandinger, U., Tesche, M., Seifert, P., Ansmann, A., Müller, D., and Althausen, D. (2010). Size matters: Influence of multiple scattering on calipso lightâĂŘextinction profiling in desert dust. *Geophysical Research Letters*, 37(L10801).

[Wang et al., 2014] Wang, X., Zhang, L., and Moran, M. D. (2014). Development of a new semi-empirical parameterization for below-cloud scavenging of size-resolved aerosol particles by both rain and snow. *Geoscientific Model Development*, 7:799–819.

[Wanninkhof, 1992] Wanninkhof, R. (1992). Relationship between wind speed and gas exchange over the ocean. *Journal of Geophysical Research: Oceans*, 97(C5):7373–7382.

[Wesely, 1989] Wesely, M. (1989). Parameterization of Surface Resistances to Gaseous Dry Deposition in Regional-Scale Numerical Models. *Atmospheric Environment*, 23(23):1293–1304.

[White, 1986] White, B. R. (1986). Encyclopedia of fluid mechanics. *Gulf Publishing, Houston, TX, USA*, pages 239–282.

[Wild et al., 2000] Wild, O., Zhu, X., and Prather, J. (2000). Fast-J: accurate simulation of the in- and below-cloud photolysis in tropospheric chemical models. *J. Atmos. Chem.*, 37:245–282.

[Willis and Tattelman, 1989] Willis, P. and Tattelman, P. (1989). Drop-size distributions associated with intense rainfall. *J. Appl. Meteorol.*, 28:3–15.

[Winker et al., 2009] Winker, D. M., M. A. Vaughan, A. O., Hu, Y., Powell, K. A., Liu, Z., Hunt, W. H., and Young, S. A. (2009). Overview of the calipso mission and caliop data processing algorithms. *Journal of Atmospheric and Oceanic Technology*, 26(2310-2323).

[Wolfe and Nickling, 1993] Wolfe, S. and Nickling, W. (1993). The protective role of sparse vegetation in wind erosion. *Progress in Physical Geography*, 17:50–68.

[Zhang et al., 2001] Zhang, L., Gong, S., Padro, J., and Barrie, L. (2001). A size-segregated particle dry deposition scheme for an atmospheric aerosol module. *Atmospheric Environment*, 35(3):549–560.

[Zhang et al., 2013] Zhang, Q. J., Beekmann, M., Drewnick, F., Freutel, F., Schneider, J.and Crippa, M., Prevot, A. S. H., Baltensperger, U., Poulain, L., Wiedensohler, A., Sciare, J., Gros, V., Borbon, A., Colomb, A., Michoud, V., Doussin, J. F., van der Gon, H. A. C. D., Haeffelin, M., Dupont, J. C., Siour, G., Petetin, H., Bessagnet, B., Pandis, S. N., Hodzic, A., Sanchez, O., Honore, C., and Perrussel, O. (2013). Formation of organic aerosol in the paris region during the megapoli summer campaign: evaluation of the volatility-basis-set approach within the chimere model. *Atmospheric Chemistry and Physics*, 13(11):5767–5790.

[Zhang et al., 2007] Zhang, Y., Huang, J.-P., Henze, D. K., and Seinfeld, J. H. (2007). Role of isoprene in secondary organic aerosol formation on a regional scale. *Journal of Geophysical Research Atmospheres*, 112(D20).

[Zhang et al., 2015] Zhang, Y., Mahowald, N., Scanza, R. A., Journet, E., Desboeufs, K., Albani, S., Kok, J. F., Zhuang, G., Chen, Y., Cohen, D. D., Paytan, A., Patey, M. D., Achterberg, E. P., Engelbrecht, J. P., and Fomba, K. W. (2015). Modeling the global emission, transport and deposition of trace elements associated with mineral dust. *Biogeosciences*, 12(19):5771–5792.

[Zuend et al., 2011] Zuend, A., Marcolli, C., Booth, A. M., Lienhard, D. M., Soonsin, V., Krieger, U. K., Topping, D. O., McFiggans, G., Peter, T., and Seinfeld, J. H. (2011). New and extended parameterization of the thermodynamic model aiomfac: calculation of activity coefficients for organic-inorganic mixtures containing carboxyl, hydroxyl, carbonyl, ether, ester, alkenyl, alkyl, and aromatic functional groups. *Atmos. Chem. Phys.*, 11:9155–9206.

[Zuend et al., 2008] Zuend, A., Marcolli, C., Luo, B. P., and Peter, T. (2008). A thermodynamic model of mixed organic-inorganic aerosols to predict activity coefficients. *Atmos. Chem. Phys.*, 8:4559âĂŞ4593.

[Zuend and Seinfeld, 2012] Zuend, A. and Seinfeld, J. H. (2012). Modeling the gas-particle partitioning of secondary organic aerosol: the importance of liquid-liquid phase separation. *Atmos. Chem. Phys.*, 12:3857–3882.

A References using CHIMERE

A.1 Reference papers

The description reference papers:

- Mailler S., L. Menut, D. Khvorostyanov, M. Valari, F. Couvidat, G. Siour, S. Turquety, R. Briant, P. Tuccella, B. Bessagnet, A. Colette, L. Letinois, and F. Meleux, CHIMERE-2017: from urban to hemispheric chemistry-transport modeling , Geosci. Model Dev., 10, 2397-2423, https://doi.org/10.5194/gmd-10-2397-2017, 2017
- Menut L, B.Bessagnet, D.Khvorostyanov, M.Beekmann, N.Blond, A.Colette, I.Coll, G.Curci, G.Foret, A.Hodzic, S.Mailler, F.Meleux, J.L.Monge, I.Pison, G.Siour, S.Turquety, M.Valari, R.Vautard and M.G.Vivanco, 2013, CHIMERE 2013: a model for regional atmospheric composition modelling, Geoscientific Model Development, 6, 981-1028, doi:10.5194/gmd-6-981-2013

About online coupling:

- Tuccella P., L.Menut, R.Briant, A.Deroubaix, D.Khvorostyanov, S.Mailler, G.Siour, S.Turquety, 2019, Implementation of aerosol-cloud interaction within WRF-CHIMERE online coupled model: evaluation and investigation of the indirect radiative effect from anthropogenic emission reduction on the Benelux, Atmosphere, 10, 20; doi:10.3390/atmos10010020.
- Briant, R., Tuccella, P., Deroubaix, A., Khvorostyanov, D., Menut, L., Mailler, S., and Turquety, S.: Aerosol-radiation interaction modelling using online coupling between the WRF 3.7.1 meteorological model and the CHIMERE 2016 chemistry-transport model, through the OASIS3-MCT coupler, Geosci. Model Dev., 10, 927-944, doi:10.5194/gmd-10-927-2017, 2017.

About real-time forecast:

- Menut L. and B.Bessagnet, 2010, Atmospheric composition forecasting in Europe , Annales Geophysicae, 28, 61-74
- Rouil L., C.Honore, R.Vautard, M.Beekmann, B.Bessagnet, L.Malherbe, F.Meleux, A.Dufour, C.Elichegaray, J-M.Flaud, L.Menut, D.Martin, A.Peuch, V-H.Peuch, N.Poisson, 2009, PREV'AIR : an operational forecasting and mapping system for air quality in Europe, BAMS, DOI: 10.1175/2008BAMS2390.1

About impact of air pollution on health:

- Valari M. and L.Menut, 2010, Transferring the heterogeneity of surface emissions to variability in pollutant concentrations over urban areas through a chemistry transport model, Atmospheric Environment, Volume 44, Issue 27, Pages 3229-3238

About fire emissions:

- Turquety, S., Menut, L., Siour, G., Mailler, S., Hadji-Lazaro, J., George, M., Clerbaux, C., Hurtmans, D., and Coheur, P.-F.: APIFLAME v2.0 biomass burning emissions model: impact of refined input parameters on atmospheric concentration in Portugal in summer 2016, Geosci. Model Dev., 13, 2981-3009, https://doi.org/10.5194/gmd-13-2981-2020, 2020.
- Turquety S., L.Menut, B.Bessagnet, A.Anav, N.Viovy, F.Maignan and M.Wooster, 2014, APIFLAME v1.0: High resolution fire emission model and application to the Euro-Mediterranean region, Geosci. Model. Dev., 7, 587-612, https://doi.org/10.5194/gmd-7-587-2014, 2014.

About mineral dust emissions:

- Menut, L., Siour, G., Bessagnet, B., Couvidat, F., Journet, E., Balkanski, Y., and Desboeufs, K.: Modelling the mineralogical composition and solubility of mineral dust in the Mediterranean area with CHIMERE 2017r4, Geosci. Model Dev. Discuss., 13, 2051-2071, 2020.

- Menut, L., 2018: Modeling of Mineral Dust Emissions with a Weibull Wind Speed Distribution Including Subgrid-Scale Orography Variance. J. Atmos. Oceanic Technol., 35, 1221-1236, https://doi.org/10.1175/JTECH-D-17-0173.1
- Menut L., C.Perez Garcia-Pando, K.Haustein, B.Bessagnet, C.Prigent and S.Alfaro, 2013, Relative impact of roughness and soil texture on mineral dust emission fluxes modeling, Journal of Geophysical Research Atmospheres, 118, 6505-6520, doi:10.1002/jgrd.50313

A.2 List of papers using CHIMERE

A.2.1 Publications of developers

Publications where a CHIMERE developer is first author.

58. Menut, L.; Bessagnet, B.; Siour, G.; Mailler, S.; Pennel, R.; Cholakian A.: Impact of lockdown measures to combat Covid-19 on air quality over western Europe, Science of The Total Environment, 741, 2020.

57. Bessagnet, B.; Menut, L.; Lapere, R.; Couvidat, F.; Jaffrezo, J.-L.; Mailler, S.; Favez, O.; Pennel, R.; Siour, G: High Resolution Chemistry Transport Modeling with the On-Line CHIMERE-WRF Model over the French Alps: Analysis of a Feedback of Surface Particulate Matter Concentrations on Mountain Meteorology, Atmosphere, 11, 565, 2020.

56. Turquety, S., Menut, L., Siour, G., Mailler, S., Hadji-Lazaro, J., George, M., Clerbaux, C., Hurtmans, D., and Coheur, P.-F.: APIFLAME v2.0 biomass burning emissions model: impact of refined input parameters on atmospheric concentration in Portugal in summer 2016, Geosci. Model Dev., 13, 2981-3009, https://doi.org/10.5194/gmd-13-2981-2020, 2020.

55. Valari, M., Markakis, K., Powaga, E., Collignan, B., Perrussel, O., 2020. EXPLUME v1.0: a model for personal exposure to ambient O3 and PM2.5. Geosci. Model Dev. 13, 1075-1094. https://doi.org/10.5194/gmd-13-1075-2020

54. Menut, L., Siour, G., Bessagnet, B., Couvidat, F., Journet, E., Balkanski, Y., and Desboeufs, K.: Modelling the mineralogical composition and solubility of mineral dust in the Mediterranean area with CHIMERE 2017r4, Geosci. Model Dev. Discuss., 13, 2051-2071, 2020.

53. Lannuque, V.; Couvidat, F.; Camredon, M.; Aumont, B.; Bessagnet, B., Modeling organic aerosol over Europe in summer conditions with the VBS-GECKO parameterization: sensitivity to secondary organic compound properties and IVOC (intermediate-volatility organic compound) emissions. Atmos. Chem. Phys. 2020, 20, (8), 4905-4931.

52. Menut, L., Tuccella, P., Flamant, C., Deroubaix, A., and Gaetani, M.: The role of aerosol-cloud interactions in linking anthropogenic pollution over southern West Africa and dust emission over the Sahara, Atmos. Chem. Phys., 19, 14657-14676

51. Bessagnet B., L.Menut, F.Couvidat, F.Meleux, G.Siour, S.Mailler, What can we expect from data assimilation for air quality forecast? Part 2: Analysis with a semi-real case, Journal of Atmospheric and Oceanic Technology, , 36, 1433-1448, https://doi.org/10.1175/JTECH-D-18-0117.1

50. Bessagnet Bertrand, Florian Couvidat, Vincent Lemaire, A statistical physics approach to perform fast highly-resolved air quality simulations - A new step towards the meta-modelling of chemistry transport models, Environmental Modelling and Software, Volume 116, 2019, Pages 100-109, https://doi.org/10.1016/j.envsoft.2019.02.017.

49. Menut L. and B.Bessagnet, 2019, What can we expect from data assimilation for air quality forecast? Part 1: Quantification with academic test cases, Journal of Atmospheric and Oceanic Technology, 36, 269-279, 10.1175/JTECH-D-18-0002.1

48. Couvidat, F., Bessagnet, B., Garcia-Vivanco, M., Real, E., Menut, L., and Colette, A., 2018, Development of an inorganic and organic aerosol model (Chimere2017b2 v1.0): seasonal and spatial evaluation over Europe, Geosci. Model Dev., 11, 165-194, https://doi.org/10.5194/gmd-11-165-2018.

47. Menut, L., Flamant, C., Turquety, S., Deroubaix, A., Chazette, P., and Meynadier, R., 2018, Impact of biomass burning on pollutant surface concentrations in megacities of the Gulf of Guinea , Atmos. Chem. Phys., 18, 2687-2707, https://doi.org/10.5194/acp-18-2687-2018.

46. Menut, L., 2018: Modeling of Mineral Dust Emissions with a Weibull Wind Speed Distribution Including Subgrid-Scale Orography Variance. J. Atmos. Oceanic Technol., 35, 1221-1236, https://doi.org/10.1175/JTECH-D-17-0173.1,

45. Bessagnet B., L.Menut, A.Colette, F.Couvidat, M.Dan, S.Mailler, L.Létinois, V.Pont, L.Rouil, 2017, An evaluation of the CHIMERE chemistry transport model to simulate dust outbreaks across the north Hemisphere in March 2014, Atmospheres,

44. Colette, A., Andersson, C., Manders, A., Mar, K., Mircea, M., Pay, M.-T., Raffort, V., Tsyro, S., Cuvelier, C., Adani, M., Bessagnet, B., Bergstrom, R., Briganti, G., Butler, T., Cappelletti, A., Couvidat, F., D'Isidoro, M., Doumbia, T., Fagerli, H., Granier, C., Heyes, C., Klimont, Z., Ojha, N., Otero, N., Schaap, M., Sindelarova, K., Stegehuis, A. I., Roustan, Y., Vautard, R., van Meijgaard, E., Vivanco, M. G., and Wind, P.: EURODELTA-Trends, a multi-model experiment of air quality hindcast in Europe over 1990-2010, Geosci. Model Dev., 10, 3255-3276, https://doi.org/10.5194/gmd-10-3255-2017, 2017.

43. Mailler S., L. Menut, D. Khvorostyanov, M. Valari, F. Couvidat, G. Siour, S. Turquety, R. Briant, P. Tuccella, B. Bessagnet, A. Colette, L. Letinois, and F. Meleux, 2017, CHIMERE-2017: from urban to hemispheric chemistry-transport modeling , Geosci. Model Dev., 10, 2397-2423, https://doi.org/10.5194/gmd-10-2397-2017, 2017.

42. Menut, L., Mailler, S., Bessagnet, B., Siour, G., Colette, A., Couvidat, F., and Meleux, F., 2017, An alternative way to evaluate chemistry-transport model variability , Geosci. Model Dev., doi:10.5194/gmd-10-1199-2017,

41. Bessagnet B., G. Pirovano, M. Mircea, C. Cuvelier, A. Aulinger, G. Calori, G. Ciarelli, A. Manders, R. Stern, S. Tsyro, M. Garcia Vivanco, P. Thunis, M.-T. Pay, A. Colette, F. Couvidat, F. Meleux, L. Rouil, A. Ung, S. Aksoyoglu, J.-M. Baldasano, J. Bieser, G. Briganti, A. Cappelletti, M. D'Isodoro, S. Finardi, R. Kranenburg, C. Silibello, C. Carnevale, W. Aas, J.-C. Dupont, H. Fagerli, L. Gonzalez, L. Menut, A. S. H. Prevot, P. Roberts, and L. White, 2015, Presentation of the EURODELTA III inter-comparison exercise - Evaluation of the chemistry transport models performance on criteria pollutants and joint analysis with meteorology, Atmos. Chem. Phys., 16, 12667-12701, doi:10.5194/acp-16-12667-2016, 2016.

40. Mailler, S. and Menut, L. and di Sarra, A. G. and Becagli, S. and Di Iorio, T. and Bessagnet, B. and Briant, R. and Formenti, P. and Doussin, J. F. and Gomez-Amo, J. L. and Mallet, M. and Rea, G. and Siour, G. and Sferlazzo, D. M. and Traversi, R. and Udisti, R. and Turquety, S., On the radiative impact of aerosols on photolysis rates: comparison of simulations and observations in the Lampedusa island during the ChArMEx/ADRIMED campaign, Atmospheric Chemistry and Physics, 16, 3, 1219-1244, 2016

39. Menut L., G.Siour, S.Mailler, F.Couvidat and B.Bessagnet, 2016, Observations and regional modeling of aerosol optical proprerties, speciation and size distribution over Northern Africa and western Europe, Atmos. Chem. Phys., 6, 12961-12982,

38. Colette A., C. Andersson, A. Baklanov, B. Bessagnet, J. Brandt, J.H. Christensen, R. Doherty, M. Engardt, C. Geels, C. Giannakopoulos, G.H. Hedegaard, E. Katragkou, J. Langner, H. Lei, A. Manders, D. Melas, F. Meleux, L. Rouil, M. Sofiev, J. Soares, D.S. Stevenson, M. Tombrou-Tzella, K.V. Varotsos and P. Young, Is the ozone climate penalty robust in Europe?, Environmental Research Letters 10(2015), p. 084015.

37. Menut, L. and Mailler, S. and Siour, G. and Bessagnet, B. and Turquety, S. and Rea, G. and Briant, R. and Mallet, M. and Sciare, J. and Formenti, P. and Meleux, F., Ozone and aerosol tropospheric concentrations variability analyzed using the ADRIMED measurements and the WRF and CHIMERE models, Atmospheric Chemistry and Physics, 15, 11, 6159-6182, 2015

36. Menut, L. and Rea, G. and Mailler, S. and Khvorostyanov, D. and Turquety, S., Aerosol forecast over the Mediterranean area during July 2013 (ADRIMED/CHARMEX), Atmospheric Chemistry and Physics, 15, 14, 7897-7911, 2015

35. Bessagnet, Bertrand and Beauchamp, Maxime and Guerreiro, Cristina and de Leeuw, Frank and Tsyro, Svetlana and Colette, Augustin and Meleux, Frederik and Rouil, Laurence and Ruyssenaars, Paul and Sauter, Ferd and Velders, Guus J. M. and Foltescu, Valentin L. and van Aardenne, John, Can further mitigation of ammonia emissions reduce exceedances of particulate matter air quality standards?, Environmental Science and Policy, 44, 149-163, 2014

34. Colette, A. and Bessagnet, B. and Meleux, F. and Terrenoire, E. and Rouil, L., Frontiers in air quality modelling, Geoscientific Model Development, 7, 1, 203-210, 2014

33. Turquety, S. and Menut, L. and Bessagnet, B. and Anav, A. and Viovy, N. and Maignan, F. and Wooster, M., API-FLAME v1.0: high-resolution fire emission model and application to the Euro-Mediterranean region, Geoscientific Model Development, 7, 2, 587-612, 2014

32. Colette, A. and Bessagnet, B. and Vautard, R. and Szopa, S. and Rao, S. and Schucht, S. and Klimont, Z. and Menut, L. and Clain, G. and Meleux, F. and Curci, G. and Rouil, L., European atmosphere in 2050, a regional air quality and climate perspective under CMIP5 scenarios, Atmospheric Chemistry and Physics, 13, 15, 7451-7471, 2013

31. Mailler, S. and Khvorostyanov, D. and Menut, L., Impact of the vertical emission profiles on background gas-phase pollution simulated from the EMEP emissions over Europe, Atmospheric Chemistry and Physics, 13, 12, 5987-5998, 2013

30. Menut, Laurent and Bessagnet, Bertrand and Colette, Augustin and Khvorostiyanov, Dmitry, On the impact of the vertical resolution on chemistry-transport modelling, Atmospheric Environment, 67, 370-384, 2013

29. Menut, L. and Bessagnet, B. and Khvorostyanov, D. and Beekmann, M. and Blond, N. and Colette, A. and Coll, I. and Curci, G. and Foret, G. and Hodzic, A. and Mailler, S. and Meleux, F. and Monge, J. L. and Pison, I. and Siour, G. and Turquety, S. and Valari, M. and Vautard, R. and Vivanco, M. G., CHIMERE 2013: a model for regional atmospheric composition modelling, Geoscientific Model Development, 6, 4, 981-1028, 2013

28. Menut, L. and Perez, C. and Haustein, K. and Bessagnet, B. and Prigent, C. and Alfaro, S., Impact of surface roughness and soil texture on mineral dust emission fluxes modeling, Journal of Geophysical Research-Atmospheres, 118, 12, 6505-6520, 2013

27. Siour, G. and Colette, A. and Menut, L. and Bessagnet, B. and Coll, I. and Meleux, F., Bridging the scales in a eulerian air quality model to assess megacity export of pollution, Environmental Modelling and Software, 46, 271-282, 2013

26. Colette, A., Granier, C., Hodnebrog, Ø., Jakobs, H., Maurizi, A., Nyiri, A., Rao, S., Amann, M., Bessagnet, B., D'Angiola, A., Gauss, M., Heyes, C., Klimont, Z., Meleux, F., Memmesheimer, M., Mieville, A., Rouïl, L., Russo, F., Schucht, S., Simpson, D., Stordal, F., Tampieri, F., and Vrac, M.: Future air quality in Europe: a multi-model assessment of projected exposure to ozone, Atmos. Chem. Phys., 12, 10613-10630, 2012

25. Menut, Laurent and Goussebaile, Arnaud and Bessagnet, Bertrand and Khvorostiyanov, Dmitry and Ung, Anthony, Impact of realistic hourly emissions profiles on air pollutants concentrations modelled with CHIMERE, Atmospheric Environment, 49, 233-244, 2012

24. Menut, Laurent and Tripathi, Om P. and Colette, Augustin and Vautard, Robert and Flaounas, Emmanouil and Bessagnet, Bertrand, Evaluation of regional climate simulations for air quality modelling purposes, Climate Dynamics, 40, 9-10, 2515-2533, 2012

23. Colette, A. and Favez, O. and Meleux, F. and Chiappini, L. and Haeffelin, M. and Morille, Y. and Malherbe, L. and Papin, A. and Bessagnet, B. and Menut, L. and Leoz, E. and Rouil, L., Assessing in near real time the impact of the April 2010 Eyjafjallajokull ash plume on air quality, Atmospheric Environment, 45, 5, 1217-1221, 2011

22. Colette, A. and Granier, C. and Hodnebrog, O. and Jakobs, H. and Maurizi, A. and Nyiri, A. and Bessagnet, B. and D'Angiola, A. and D'Isidoro, M. and Gauss, M. and Meleux, F. and Memmesheimer, M. and Mieville, A. and Rouil, L. and Russo, F. and Solberg, S. and Stordal, F. and Tampieri, F., Air quality trends in Europe over the past decade: a first multi- model assessment, Atmospheric Chemistry and Physics, 11, 22, 11657-11678, 2011

21. Valari, Myrto and Menut, Laurent and Chatignoux, Edouard, Using a Chemistry Transport Model to Account for the Spatial Variability of Exposure Concentrations in Epidemiologic Air Pollution Studies, Journal of the Air and Waste Management Association, 61, 2, 164-179, 2011

20. Bessagnet, Bertrand and Seigneur, Christian and Menut, Laurent, Impact of dry deposition of semi-volatile organic compounds on secondary organic aerosols, Atmospheric Environment, 44, 14, 1781-1787, 2010

19. Menut, L. and Bessagnet, B., Atmospheric composition forecasting in Europe, Annales Geophysicae, 28, 1, 61-74, 2010

18. Valari, Myrto and Menut, Laurent, Transferring the heterogeneity of surface emissions to variability in pollutant concentrations over urban areas through a chemistry-transport model, Atmospheric Environment, 44, 27, 3229-3238, 2010

17. Menut, Laurent and Chiapello, Isabelle and Moulin, Cyril, Predictability of mineral dust concentrations: The African Monsoon Multidisciplinary Analysis first short observation period forecasted with CHIMERE-DUST, Journal of Geophysical Research-Atmospheres, 114, 2009

16. Menut, Laurent and Masson, Olivier and Bessagnet, Bertrand, Contribution of Saharan dust on radionuclide aerosol activity levels in Europe? The 21-22 February 2004 case study, Journal of Geophysical Research-Atmospheres, 114, 2009

15. Bessagnet, Bertrand and Menut, Laurent and Aymoz, Gilles and Chepfer, Helene and Vautard, Robert, Modeling dust emissions and transport within Europe: the Ukraine March 2007 event, Journal of Geophysical Research-Atmospheres, 113, D15, 2008

14. Bessagnet, Bertrand and Menut, Laurent and Curci, Gabriele and Hodzic, Alma and Guillaume, Bruno and Liousse, Catherine and Moukhtar, Sophie and Pun, Betty and Seigneur, Christian and Schulz, Michael, Regional modeling of carbonaceous aerosols over Europe-focus on secondary organic aerosols, Journal of Atmospheric Chemistry, 61, 3, 175-202, 2008

13. Colette, Augustin and Menut, Laurent and Haeffelin, Martial and Morille, Yohann, Impact of the transport of aerosols from the free troposphere towards the boundary layer on the air quality in the Paris area, Atmospheric Environment, 42, 2, 390-402, 2008

12. Menut, Laurent, Sensitivity of hourly Saharan dust emissions to NCEP and ECMWF modeled wind speed, Journal of Geophysical Research-Atmospheres, 113, D16, 2008

11. Valari, Myrto and Menut, Laurent, Does an Increase in Air Quality Models' Resolution Bring Surface Ozone Concentrations Closer to Reality?, Journal of Atmospheric and Oceanic Technology, 25, 11, 1955-1968, 2008

10. Meleux, Frederik and Solmon, Fabien and Giorgi, Filippo, Increase in summer European ozone amounts due to climate change, Atmospheric Environment, 41, 35, 7577-7587, 2007

9. Menut, Laurent and Foret, Gilles and Bergametti, Gilles, Sensitivity of mineral dust concentrations to the model size distribution accuracy, Journal of Geophysical Research-Atmospheres, 112, D10, 2007

8. Colette, A. and Ancellet, G. and Menut, L. and Arnold, S. R., A Lagrangian analysis of the impact of transport and transformation on the ozone stratification observed in the free troposphere during the ESCOMPTE campaign, Atmospheric Chemistry and Physics, 6, 3487-3503, 2006

7. Bessagnet, B. and Hodzic, A. and Blanchard, O. and Lattuati, M. and Le Bihan, O. and Marfaing, H. and Rouil, L., Origin of particulate matter pollution episodes in wintertime over the Paris Basin, Atmospheric Environment, 39, 33, 6159-6174, 2005

6. Bessagnet, B. and Hodzic, A. and Blanchard, O. and Lattuati, M. and Le Bihan, O. and Marfaing, H. and Rouil, L., Origin of particulate matter pollution episodes in wintertime over the Paris Basin, Atmospheric Environment, 39, 33, 6159-6174, 2005

5. Menut, L. and Coll, I. and Cautenet, S., Impact of meteorological data resolution on the forecasted ozone concentrations during the ESCOMPTE IOP2a and IOP2b, Atmospheric Research, 74, 1-4, 139-159, 2005

4. Bessagnet, B. and Hodzic, A. and Vautard, R. and Beekmann, M. and Cheinet, S. and Honore, C. and Liousse, C. and Rouil, L., Aerosol modeling with CHIMERE - preliminary evaluation at the continental scale, Atmospheric Environment, 38, 18, 2803-2817, 2004

3. Menut, L., Adjoint modeling for atmospheric pollution process sensitivity at regional scale, Journal of Geophysical Research-Atmospheres, 108, D17, 2003

2. Menut, L. and Vautard, R. and Beekmann, M. and Honore, C., Sensitivity of photochemical pollution using the adjoint of a simplified chemistry-transport model, Journal of Geophysical Research-Atmospheres, 105, D12, 15379-15402, 2000

1. Menut, L. and Vautard, R. and Flamant, C. and Abonnel, C. and Beekmann, M. and Chazette, P. and Flamant, P. H. and Gombert, D. and Guedalia, D. and Kley, D. and Lefebvre, M. P. and Lossec, B. and Martin, D. and Megie, G. and Perros, P. and Sicard, M. and Toupance, G., Measurements and modelling of atmospheric pollution over the Paris area: an overview of the ESQUIF Project, Annales Geophysicae-Atmospheres Hydrospheres and Space Sciences, 18, 11, 1467-1481, 2000

A.2.2 Publications of users

252. Fortems-Cheiney, A., Dufour, G., Dufossé, K., Couvidat, F., Gilliot, J.-M., Siour, G., Beekmann, M., Foret, G., Meleux, F., Clarisse, L., Coheur, P.-F., Van Damme, M., Clerbaux, C., and Génermont, S.: Do alternative inventories converge on the spatiotemporal representation of spring ammonia emissions in France?, Atmos. Chem. Phys., 20, 13481-13495, 2020.

251. Niramson Azouz, Matthias Beekmann, Guillaume Siour, Pierre Cellier, Jean-Louis Drouet: Changes in spatial patterns of ammonia dry deposition flux and deposition threshold exceedance according to dispersion model formalism and horizontal resolution, Environmental Pollution, 268, 115823, 2020.

250. Mathieu Lachatre, Gilles Foret, Benoit Laurent, Guillaume Siour, Juan Cuesta, Gaelle Dufour, Fan Meng, Wei Tang, Qijie Zhang and Matthias Beekmann:Air Quality Degradation by Mineral Dust over Beijing, Chengdu and Shanghai Chinese Megacities, Atmosphere 11, 708, 2020.

249. Blechschmidt, A.-M., Arteta, J., Coman, A., Curier, L., Eskes, H., Foret, G., Gielen, C., Hendrick, F., Marecal, V., Meleux, F., Parmentier, J., Peters, E., Pinardi, G., Piters, A. J. M., Plu, M., Richter, A., Segers, A., Sofiev, M., Valdebenito, A. M., Van Roozendael, M., Vira, J., Vlemmix, T., and Burrows, J. P.: Comparison of tropospheric NO2 columns from MAX-DOAS retrievals and regional air quality model simulations, Atmos. Chem. Phys., 20, 2795-2823, https://doi.org/10.5194/acp-20-2795-2020, 2020.

248. Belis, C. A.; Pernigotti, D.; Pirovano, G.; Favez, O.; Jaffrezo, J. L.; Kuenen, J.; Denier van Der Gon, H.; Reizer, M.; Riffault, V.; Alleman, L. Y.; Almeida, M.; Amato, F.; Angyal, A.; Argyropoulos, G.; Bande, S.; Beslic, I.; Besombes, J. L.; Bove, M. C.; Brotto, P.; Calori, G.; Cesari, D.; Colombi, C.; Contini, D.; De Gennaro, G.; Di Gilio, A.; Diapouli, E.; El Haddad, I.; Elbern, H.; Eleftheriadis, K.; Ferreira, J.; Vivanco, M. G.; Gilardoni, S.; Golly, B.; Hellebust, S.; Hopke, P. K.; Izadmanesh, Y.; Jorquera, H.; Krajsek, K.; Kranenburg, R.; Lazzeri, P.; Lenartz, F.; Lucarelli, F.; Maciejewska, K.; Manders, A.; Manousakas, M.; Masiol, M.; Mircea, M.; Mooibroek, D.; Nava, S.; Oliveira, D.; Paglione, M.; Pandolfi, M.; Perrone, M.; Petralia, E.; Pietrodangelo, A.; Pillon, S.; Pokorna, P.; Prati, P.; Salameh, D.; Samara, C.; Samek, L.; Saraga, D.; Sauvage, S.; Schaap, M.; Scotto, F.; Sega, K.; Siour, G.; Tauler, R.; Valli, G.; Vecchi, R.; Venturini, E.; Vestenius, M.; Waked, A.; Yubero, E., Evaluation of receptor and chemical transport models for PM10 source apportionment. Atmospheric Environment: X 2019, 5, 100053.

247. Cholakian, A., Beekmann, M., Coll, I., Ciarelli, G., and Colette, A.: Biogenic secondary organic aerosol sensitivity to organic aerosol simulation schemes in climate projections, Atmos. Chem. Phys., 19, 13209-13226, https://doi.org/10.5194/acp-19-13209-2019, 2019.

246. Ciarelli, G., Theobald, M. R., Vivanco, M. G., Beekmann, M., Aas, W., Andersson, C., Bergstrom, R., Manders-Groot, A., Couvidat, F., Mircea, M., Tsyro, S., Fagerli, H., Mar, K., Raffort, V., Roustan, Y., Pay, M.-T., Schaap, M., Kranenburg, R., Adani, M., Briganti, G., Cappelletti, A., D'Isidoro, M., Cuvelier, C., Cholakian, A., Bessagnet, B., Wind, P., and Colette, A.: Trends of inorganic and organic aerosols and precursor gases in Europe: insights from the EURODELTA multi-model experiment over the 1990-2010 period, Geosci. Model Dev., 12, 4923-4954, https://doi.org/10.5194/gmd-12-4923-2019, 2019.

245. Trewhela B., Huneeus N., M.Munizaga, A.Mazzeo, L.Menut, S.Mailler, M.Valari, C.Ordonez, 2019, Analysis of exposure to fine particulate matter using passive data from public transport, Atmospheric Environment, 215, 116878

244. Blechschmidt, A.-M., Arteta, J., Coman, A., Curier, L., Eskes, H., Foret, G., Gielen, C., Hendrick, F., Marecal, V., Meleux, F., Parmentier, J., Peters, E., Pinardi, G., Piters, A. J. M., Plu, M., Richter, A., Segers, A., Sofiev, M., Valdebenito, A. M., Van Roozendael, M., Vira, J., Vlemmix, T., and Burrows, J. P.: Comparison of tropospheric NO2 columns from MAX-DOAS retrievals and regional air quality model simulations, Atmos. Chem. Phys., 20, 2795-2823, https://doi.org/10.5194/acp-20-2795-2020, 2020.

243. Ciarelli, G., Theobald, M. R., Vivanco, M. G., Beekmann, M., Aas, W., Andersson, C., Bergstrom, R., Manders-Groot, A., Couvidat, F., Mircea, M., Tsyro, S., Fagerli, H., Mar, K., Raffort, V., Roustan, Y., Pay, M.-T., Schaap, M., Kranenburg, R., Adani, M., Briganti, G., Cappelletti, A., D'Isidoro, M., Cuvelier, C., Cholakian, A., Bessagnet, B., Wind, P., and Colette, A.: Trends of inorganic and organic aerosols and precursor gases in Europe: insights from the EURODELTA multi-model experiment over the 1990-2010 period, Geosci. Model Dev., 12, 4923-4954, https://doi.org/10.5194/gmd-12-4923-2019, 2019.

242. Trewhela B., Huneeus N., M.Munizaga, A.Mazzeo, L.Menut, S.Mailler, M.Valari, C.Ordonez, 2019, Analysis of exposure to fine particulate matter using passive data from public transport, Atmospheric Environment, 215, 116878

241. Cholakian Arineh, Augustin Colette, Isabelle Coll, Giancarlo Ciarelli and Matthias Beekmann: Future climatic drivers and their effect on PM10 components in Europe and the Mediterranean Sea, Atmos. Chem. Phys.,19,4459-44484, https://doi.org/10.5194/acp-19-4459-2019.

240. Lachatre, M., Fortems-Cheiney, A., Foret, G., Siour, G., Dufour, G., Clarisse, L., Clerbaux, C., Coheur, P.-F., Van Damme, M., and Beekmann, M.: The unintended consequence of SO2 and NO2 regulations over China: increase of ammonia levels and impact on PM2.5 concentrations, Atmos. Chem. Phys., 19, 6701-6716, https://doi.org/10.5194/acp-19-6701-2019, 2019.

239. Azouz Niramson, Jean-Louis Drouet, Matthias Beekmann, Guillaume Siour, Roy Wichink Kruit, Pierre Cellier: Comparison of spatial patterns of ammonia concentration and dry deposition flux between a regional Eulerian chemistry-transport model and a local Gaussian plume model, Air Quality, Atmosphere and Health, 2019,

238. Theobald, M. R., Vivanco, M. G., Aas, W., Andersson, C., Ciarelli, G., Couvidat, F., Cuvelier, K., Manders, A., Mircea, M., Pay, M.-T., Tsyro, S., Adani, M., Bergstrom, R., Bessagnet, B., Briganti, G., Cappelletti, A., D'Isidoro, M., Fagerli, H., Mar, K., Otero, N., Raffort, V., Roustan, Y., Schaap, M., Wind, P., and Colette, A.: An evaluation of European nitrogen and sulfur wet deposition and their trends estimated by six chemistry transport models for the period 1990-2010, Atmos. Chem. Phys., 19, 379-405, https://doi.org/10.5194/acp-19-379-2019, 2019.

237. Brasseur, G. P., Xie, Y., Petersen, A. K., Bouarar, I., Flemming, J., Gauss, M., Jiang, F., Kouznetsov, R., Kranenburg, R., Mijling, B., Peuch, V.-H., Pommier, M., Segers, A., Sofiev, M., Timmermans, R., van der A, R., Walters, S., Xu, J., and Zhou, G.: Ensemble forecasts of air quality in eastern China - Part 1: Model description and implementation of the MarcoPolo-Panda prediction system, version 1, Geosci. Model Dev., 12, 33-67, https://doi.org/10.5194/gmd-12-33-2019, 2019.

236. Curci, G., Alyuz, U., Baro, R., Bianconi, R., Bieser, J., Christensen, J. H., Colette, A., Farrow, A., Francis, X., Jimenez-Guerrero, P., Im, U., Liu, P., Manders, A., Palacios-Pena, L., Prank, M., Pozzoli, L., Sokhi, R., Solazzo, E., Tuccella, P., Unal, A., Vivanco, M. G., Hogrefe, C., and Galmarini, S.: Modelling black carbon absorption of solar radiation: combining external and internal mixing assumptions, Atmos. Chem. Phys., 19, 181-204, https://doi.org/10.5194/acp-19-181-2019, 2019.

235. Tuccella P., L.Menut, R.Briant, A.Deroubaix, D.Khvorostyanov, S.Mailler, G.Siour, S.Turquety, Implementation of aerosol-cloud interaction within WRF-CHIMERE online coupled model: evaluation and investigation of the indirect radiative effect from anthropogenic emission reduction on the Benelux, Atmosphere 2019, 10, 20; doi:10.3390/atmos10010020,

234. Vogel, F. R., Frey, M., Staufer, J., Hase, F., Broquet, G., Xueref-Remy, I., Chevallier, F., Ciais, P., Sha, M. K., Chelin, P., Jeseck, P., Janssen, C., Té, Y., GroSS, J., Blumenstock, T., Tu, Q., and Orphal, J.: XCO2 in an emission hot-spot region: the COCCON Paris campaign 2015, Atmos. Chem. Phys., 19, 3271-3285, https://doi.org/10.5194/acp-19-3271-2019, 2019.

233. Dufour, G.; Eremenko, M.; Beekmann, M.; Cuesta, J.; Foret, G.; Fortems-Cheiney, A.; Lachatre, M.; Lin, W.; Liu, Y.; Xu, X.; Zhang, Y., Lower tropospheric ozone over the North China Plain: variability and trends revealed by IASI satellite observations for 2008-2016. Atmos. Chem. Phys. 2018, 18, (22), 16439-16459.

232. Le Morvan-Quéméner, A.; Coll, I.; Kammer, J.; Lamaud, E.; Loubet, B.; Personne, E.; Stella, P., Impact of parameterization choices on the restitution of ozone deposition over vegetation. Atmospheric Environment 2018, 178, 49-65.

231. Anav, A., Proietti, C., Menut, L., Carnicelli, S., De Marco, A., and Paoletti, E.: Sensitivity of stomatal conductance to soil moisture: implications for tropospheric ozone, Atmos. Chem. Phys., 18, 5747-5763, https://doi.org/10.5194/acp-18-5747-2018, 2018, Menut, L., Flamant, C., Turquety, S., Deroubaix, A., Chazette, P., and Meynadier, R.: Impact of biomass burning on pollutant surface concentrations in megacities of the Gulf of Guinea , Atmos. Chem. Phys., 18, 2687-2707, https://doi.org/10.5194/acp-18-2687-2018, 2018. ,

230. Balkanski, Y., Menut, L., Garnier, E., Wang, R., Evangeliou, N., Jourdain, S., Eschstruth, C., Vrac, M., Yiou, P., 2018, Mortality induced by PM2.5 exposure following the 1783 Laki eruption using reconstructed meteorological fields, Nature Scientific Reports, 8:15896, 10.1038/s41598-018-34228-7.

229. Broquet, G., Breon, F.-M., Renault, E., Buchwitz, M., Reuter, M., Bovensmann, H., Chevallier, F., Wu, L., and Ciais, P.: The potential of satellite spectro-imagery for monitoring CO2 emissions from large cities, Atmos. Meas. Tech., 11, 681-708, https://doi.org/10.5194/amt-11-681-2018, 2018.

228. Cholakian, A., Beekmann, M., Colette, A., Coll, I., Siour, G., Sciare, J., Marchand, N., Couvidat, F., Pey, J., Gros, V., Sauvage, S., Michoud, V., Sellegri, K., Colomb, A., Sartelet, K., Langley DeWitt, H., Elser, M., Prevot, A. S. H., Szidat, S., Dulac, F.: Simulation of fine organic aerosols in the western Mediterranean area during the ChArMEx 2013 summer campaign, Atmos. Chem. Phys., 18, 7287-7312, 2018

227. Deroubaix, A., Menut, L., Flamant, C., Brito, J., Denjean, C., Dreiling, V., Fink, A., Jambert, C., Kalthoff, N., Knippertz, P., Ladkin, R., Mailler, S., Maranan, M., Pacifico, F., Piguet, B., Siour, G., and Turquety, S.: Diurnal cycle of coastal anthropogenic pollutant transport over southern West Africa during the DACCIWA campaign, Atmos. Chem. Phys. Discuss., https://doi.org/10.5194/acp-2018-766, accepted, 2018.

226. Deroubaix, A., Flamant, C., Menut, L., Siour, G., Mailler, S., Turquety, S., Briant, R., Khvorostyanov, D., and Crumeyrolle, S., 2017, Interactions of Atmospheric Gases and Aerosols with the Monsoon Dynamics over the Sudano-Guinean region during AMMA, Atmos. Chem. Phys., 18, 445-465, https://doi.org/10.5194/acp-18-445-2018, 2018.

225. Ding, J., van der A, R. J., Mijling, B., Jalkanen, J.-P., Johansson, L. and Levelt, P. F., Maritime NOx emissions over Chinese seas derived from satellite observations. Geophysical Research Letters, 45. https://doi.org/10.1002/2017GL076788, 2018

224. Flamant, C., Deroubaix, A., Chazette, P., Brito, J., Gaetani, M., Knippertz, P., Fink, A. H., de Coetlogon, G., Menut, L., Colomb, A., Denjean, C., Meynadier, R., Rosenberg, P., Dupuy, R., Schwarzenboeck, A., and Totems, J., 2018: Aerosol distribution in the northern Gulf of Guinea: local anthropogenic sources, long-range transport and the role of coastal shallow circulations, Atmos. Chem. Phys., 18, 12363-12389,

223. Falasca, S. and Curci, G. (2018), High-resolution air quality modeling: Sensitivity tests to horizontal resolution and urban canopy with WRF-CHIMERE, Atmos. Environ., 187, 241-254, doi: 10.1016/j.atmosenv.2018.05.048

222. Falasca, S. and Curci, G. (2018), Impact of Highly Reflective Materials on Meteorology, PM10 and Ozone in Urban Areas: A Modeling Study with WRF-CHIMERE at High Resolution over Milan (Italy), Urban Science, 2(1), 18, doi:10.3390/urbansci2010018

221. Galmarini S., I. Kioutsioukis, E. Solazzo, U. Alyuz, A. Balzarini, R. Bellasio, A.M.K. Benedictow, R. Bianconi, J. Bieser, J. Brandt, J.H. Christensen, A. Colette, G. Curci, Y. Davila, X. Dong, J. Flemming, X. Francis, A. Fraser, J. Fu, D.K. Henze, C. Hogrefe, U. Im, M. Garcia Vivanco, P. Jiménez-Guerrero, J.E. Jonson, N. Kitwiroon, A. Manders, R. Mathur, L. Palacios-Pena, G. Pirovano, L. Pozzoli, M. Prank, M. Schultz, R.S. Sokhi, K. Sudo, P. Tuccella, T. Takemura, T. Sekiya and A. Unal, Two-scale multi-model ensemble: is a hybrid ensemble of opportunity telling us more?, Atmos. Chem. Phys. 18(2018), pp. 8727-8744.

220. Im U., J. Brandt, C. Geels, K.M. Hansen, J.H. Christensen, M.S. Andersen, E. Solazzo, I. Kioutsioukis, U. Alyuz, A. Balzarini, R. Baro, R. Bellasio, R. Bianconi, J. Bieser, A. Colette, G. Curci, A. Farrow, J. Flemming, A. Fraser, P. Jimenez-Guerrero, N. Kitwiroon, C.K. Liang, U. Nopmongcol, G. Pirovano, L. Pozzoli, M. Prank, R. Rose, R. Sokhi, P. Tuccella, A. Unal, M.G. Vivanco, J. West, G. Yarwood, C. Hogrefe and S. Galmarini, Assessment and economic valuation of air pollution impacts on human health over Europe and the United States as calculated by a multi-model ensemble in the framework of AQMEII3, Atmos. Chem. Phys. 18(2018a), pp. 5967-5989.

219. Im U., J.H. Christensen, C. Geels, K.M. Hansen, J. Brandt, E. Solazzo, U. Alyuz, A. Balzarini, R. Baro, R. Bellasio, R. Bianconi, J. Bieser, A. Colette, G. Curci, A. Farrow, J. Flemming, A. Fraser, P. Jimenez-Guerrero, N. Kitwiroon, P. Liu, U. Nopmongcol, L. Palacios-Pena, G. Pirovano, L. Pozzoli, M. Prank, R. Rose, R. Sokhi, P. Tuccella, A. Unal, M.G. Vivanco, G. Yarwood, C. Hogrefe and S. Galmarini, Influence of anthropogenic emissions and boundary conditions on multi-model simulations of major air pollutants over Europe and North America in the framework of AQMEII3, Atmos. Chem. Phys. 18(2018b), pp. 8929-8952.

218. Mazzeo A., Ordonez C., Orfanoz-Cheuquelaf A., Menut L., Mailler S., Valari M., Denier van der Gon H., Gallardo L., Munoz R., Donoso R., Galleguillos M., Osses M., Tolvett S., 2018, Impact of residential combustion and transport emissions on air pollution in Santiago during winter, Atmospheric Environment, vol.190, p.195-208, doi.org/10.1016/j.atmosenv.2018.06.043.

217. Monteiro A., Russo M., Gama C., Borrego C. (2018). How maritime emissions are important to the air quality over Portugal. Environmental Pollution 242 (A), 565-575.https://doi.org/10.1016/j.envpol.2018.07.011

216. Otero N., J. Sillmann, K.A. Mar, H.W. Rust, S. Solberg, C. Andersson, M. Engardt, R. Bergstrom, B. Bessagnet, A. Colette, F. Couvidat, C. Cuvelier, S. Tsyro, H. Fagerli, M. Schaap, A. Manders, M. Mircea, G. Briganti, A. Cappelletti, M. Adani, M. D'Isidoro, M.T. Pay, M. Theobald, M.G. Vivanco, P. Wind, N. Ojha, V. Raffort and T. Butler, A multi-model comparison of meteorological drivers of surface ozone over Europe, Atmos. Chem. Phys. 18(2018), pp. 12269-12288.

215. Vautard R. , A. Colette, E. Van Meijgaard, F. Meleux, G. Jan van Oldenborgh, F. Otto, I. Tobin and P. Yiou, Attribution of Wintertime Anticyclonic Stagnation Contributing to Air Pollution in Western Europe, Bulletin of the American Meteorological Society 99(2018), pp. S70-S75.

214. Pison, I., Berchet, A., Saunois, M., Bousquet, P., Broquet, G., Conil, S., Delmotte, M., Ganesan, A., Laurent, O., Martin, D., O'Doherty, S., Ramonet, M., Spain, T. G., Vermeulen, A., and Yver Kwok, C.: How a European network may help with estimating methane emissions on the French national scale, Atmos. Chem. Phys., 18, 3779-3798, https://doi.org/10.5194/acp-18-3779-2018, 2018.

213. Vivanco M.G. , M.R. Theobald, H. Garcia-Gomez, J.L. Garrido, M. Prank, W. Aas, M. Adani, U. Alyuz, C. Andersson, R. Bellasio, B. Bessagnet, R. Bianconi, J. Bieser, J. Brandt, G. Briganti, A. Cappelletti, G. Curci, J.H. Christensen, A. Colette, F. Couvidat, C. Cuvelier, M. D'Isidoro, J. Flemming, A. Fraser, C. Geels, K.M. Hansen, C. Hogrefe, U. Im, O. Jorba, N. Kitwiroon, A. Manders, M. Mircea, N. Otero, M.T. Pay, L. Pozzoli, E. Solazzo, S. Tsyro, A. Unal, P. Wind and S. Galmarini, Modeled deposition of nitrogen and sulfur in Europe estimated by 14 air quality model systems: evaluation, effects of changes in emissions and implications for habitat protection, Atmos. Chem. Phys. 18(2018), pp. 10199-10218.

212. Briant, R., Tuccella, P., Deroubaix, A., Khvorostyanov, D., Menut, L., Mailler, S., and Turquety, S.: Aerosol-radiation interaction modelling using online coupling between the WRF 3.7.1 meteorological model and the CHIMERE 2016 chemistry-transport model, through the OASIS3-MCT coupler, Geosci. Model Dev., 10, 927-944, doi:10.5194/gmd-10-927-2017, 2017

211. Fortems-Cheiney A., G. Foret, G. Siour, R. Vautard, S. Szopa, G. Dufour, A. Colette, G. Lacressonniere and M. Beekmann, A 3oC global RCP8.5 emission trajectory cancels benefits of European emission reductions on air quality, Nature Communications 8(2017), p. 89.

210. Kalabokas P., J. Hjorth, G. Foret, G. Dufour, M. Eremenko, G. Siour, J. Cuesta, and M. Beekmann: An investigation on the origin of regional spring time ozone episodes in the Western Mediterranean and Central Europe, Atmos. Chem. Phys., 17, 3905-3928, 2017.

209. Konovalov, I. B., Beekmann, M., Berezin, E. V., Formenti, P., and Andreae, M. O.: Probing into the aging dynamics of biomass burning aerosol by using satellite measurements of aerosol optical depth and carbon monoxide, Atmos. Chem. Phys., 17, 4513-4537, doi:10.5194/acp-17-4513-2017, 2017.

208. Petit J.E., T.Amodeo, F.Meleux, B.Bessagnet, L.Menut, D.Grenier, Y.Pellan, A.Ockler, B.Rocq, V.Gros, J.Sciare, O.Favez, 2017, Characterising an intense PM pollution episode in March 2015 in France from multi-site approach and near real time data: climatology, variabilities, geographical origins and model evaluation., Atmospheric Environment, 155, 68-84, http://dx.doi.org/10.1016/j.atmosenv.2017.02.012

207. Shaiganfar, R., Beirle, S., Denier van der Gon, H., Jonkers, S., Kuenen, J., Petetin, H., Zhang, Q., Beekmann, M., and Wagner, T.: Estimation of the Paris NOx emissions from mobile MAX-DOAS observations and CHIMERE model simulations during the MEGAPOLI campaign using the closed integral method, Atmos. Chem. Phys., 17, 7853-7890, https://doi.org/10.5194/acp-17-7853-2017, 2017.

206. Solazzo, E., Hogrefe, C., Colette, A., Garcia-Vivanco, M., and Galmarini, S.: Advanced error diagnostics of the CMAQ and Chimere modelling systems within the AQMEII3 model evaluation framework, Atmos. Chem. Phys., 17, 10435-10465, https://doi.org/10.5194/acp-17-10435-2017, 2017.

205. Solazzo E., R. Bianconi, C. Hogrefe, G. Curci, P. Tuccella, U. Alyuz, A. Balzarini, R. Baro, R. Bellasio, J. Bieser, J. Brandt, J.H. Christensen, A. Colette, X. Francis, A. Fraser, M.G. Vivanco, P. Jimenez-Guerrero, U. Im, A. Manders, U. Nopmongcol, N. Kitwiroon, G. Pirovano, L. Pozzoli, M. Prank, R.S. Sokhi, A. Unal, G. Yarwood and S. Galmarini, Evaluation and error apportionment of an ensemble of atmospheric chemistry transport modeling systems: multivariable temporal and spatial breakdown, Atmos. Chem. Phys. 17(2017a), pp. 3001-3054.

204. Thunis P., B. Degraeuwe, E. Pisoni, F. Meleux, A. Clappier, 2017, Analyzing the efficiency of short-term air quality plans in European cities, using the CHIMERE air quality model, Air Quality, Atmosphere and Health, Volume 10, Issue 2, pp 235-248, https://link.springer.com/article/10.1007/s11869-016-0427-y.

203. Vivanco Marta G., Bertrand Bessagnet; Kees Cuvelier; Mark R Theobald; Svetlana Tsyro; Armin Aulinger; Johannes Bieser; Giuseppe Calori; Giancarlo Ciarelli; Astrid Manders; Mihaela Mircea; Sebnem Aksoyoglu; Gino Briganti; Augustin Colette; Florian COUVIDAT; Andrea Cappelletti; Massimo D'Isidoro; Richard Kranenburg; Frederik Meleux; Laurent Menut; Maria Teresa Pay; Guido Pirovano; Laurence Rouil; Camilo Silibello; Philippe Thunis; Anthony Ung, Joint analysis of deposition fluxes and atmospheric concentrations predicted by six chemistry transport models in the frame of the EURODELTAIII project, 2017, Atmospheric Environment, 151, pp.152-175, http://dx.doi.org/10.1016/j.atmosenv.2016.11.042

202. Wang Y., Gregoire Broquet, Philippe Ciais, Frédéric Chevallier, Felix Vogel, Nikolay Kadygrov, Lin Wu, Yi Yin, Rong Wang, and Shu Tao, Estimation of observation errors for large-scale atmospheric inversion of CO2 emissions from fossil fuel combustion, Tellus B: Chemical and Physical Meteorology Vol. 69 , Iss. 1,2017.

201. Anav, Alessandro and De Marco, Alessandra and Proietti, Chiara and Alessandri, Andrea and Dell'Aquila, Alessandro and Cionni, Irene and Friedlingstein, Pierre and Khvorostyanov, Dmitry and Menut, Laurent and Paoletti, Elena and Sicard, Pierre and Sitch, Stephen and Vitale, Marcello, Comparing concentration-based (AOT40) and stomatal uptake (PODY) metrics for ozone risk assessment to European forests, Global Change Biology, 22, 4, 1608-1627, 2016

200. Beegum S.N., I.Gherboudj, N.Chaouch, F.Couvidat, L.Menut, and H.Ghedira, 2016, Simulating Aerosols over Arabian Peninsula with CHIMERE: Sensitivity to soil, surface parameters and anthropogenic emission inventories, Atmospheric Environment, 128:185-197.

199. Besson L., O.Caumont, L.Goulet, S.Bastin, L.Menut, F.Fabry, J. Parent du Chatelet, 2016, Comparison of real-time refractivity measurement by radar with automatic weather stations, AROME-WMED and WRF forecasts simulations during the SOP1 of HyMeX campaign, Quarterly Journal of the Royal Meteorological Society, 142, 138–152, http://dx.doi.org/10.1002/qj.2799, .

198. Boon, A., Broquet, G., Clifford, D. J., Chevallier, F., Butterfield, D. M., Pison, I., Ramonet, M., Paris, J.-D., and Ciais, P.: Analysis of the potential of near-ground measurements of CO2 and CH4 in London, UK, for the monitoring of city-scale emissions using an atmospheric transport model, Atmos. Chem. Phys., 16, 6735-6756, https://doi.org/10.5194/acp-16-6735-2016, 2016.

197. Garcia Ferreyra M. F., G. Curci and M. Lanfri, "First Implementation of the WRF-CHIMERE-EDGAR Modeling System Over Argentina," in IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 9, no. 12, pp. 5304-5314, Dec. 2016, doi: 10.1109/JSTARS.2016.2588502, http://ieeexplore.ieee.org/abstract/document/7529134/.

196. Fortems-Cheiney, A., G. Dufour, L. Hamaoui-Laguel, G. Foret, G. Siour, M. Van Damme, F. Meleux, P.-F. Coheur, C. Clerbaux, M. Wallasch, V. Novak, and M. Beekmann (2016), Unaccounted variability in NH3 agricultural sources detected by IASI contributing to European spring haze episode, Geophys. Res. Lett., 43, doi:10.1002/2016GL069361.

195. Konovalov, I. B., Berezin, E. V., Ciais, P., Broquet, G., Zhuravlev, R. V., and Janssens-Maenhout, G.: Estimation of fossil-fuel CO2 emissions using satellite measurements of "proxy" species, Atmos. Chem. Phys., 16, 13509-13540, https://doi.org/10.5194/acp-16-13509-2016, 2016.

194. Lemaire V.E.P., A. Colette, and L. Menut, 2016, Using proxies to explore ensemble uncertainty in climate impact studies: the example of air pollution, Atmos. Chem. Phys., 16, 2559-2574, doi:10.5194/acp-16-2559-2016

193. Lacressonniere G., L.Watson, M.Gauss, M.Engardt, C.Andersson, M.Beekmann, A.Colette, G.Foret, B.Josse, V.Marecal, A.Nyiri, G.Siour, S.Sobolowski, R.Vautard: Particulate matter air pollution in Europe in a +2C warming world, Atmospheric Environment, 154, 129-140, 2017

192. Lacressonniere G, G.Foret, M.Beekmann, G.Siour, M.Engardt, M.Gauss, L.Watson, C.Andersson, A.Colette, B.Josse, V.Marecal, A.Nyiri and R.Vautard, 2016, Impacts of regional climate change on air quality projections and associated uncertainties, Climatic Change, DOI 10.1007/s10584-016-1619-z.

191. Lemaire, V. and Coll, I. and Couvidat, F. and Mouchel-Vallon, C. and Seigneur, C. and Siour, G., Oligomer formation in the troposphere: from experimental knowledge to 3-D modeling, Geosci. Model Dev., 9, 4, 1361-1382, 2016

190. Lemaire, V. E. P. and Colette, A. and Menut, L., Using statistical models to explore ensemble uncertainty in climate impact studies: the example of air pollution in Europe, Atmos. Chem. Phys., 16, 4, 2559-2574, 2016

189. Mallet, M. and Dulac, F. and Formenti, P. and Nabat, P. and Sciare, J. and Roberts, G. and Pelon, J. and Ancellet, G. and Tanre, D. and Parol, F. and Denjean, C. and Brogniez, G. and di Sarra, A. and Alados-Arboledas, L. and Arndt, J. and Auriol, F. and Blarel, L. and Bourrianne, T. and Chazette, P. and Chevaillier, S. and Claeys, M. and D'Anna, B. and Derimian, Y. and Desboeufs, K. and Di Iorio, T. and Doussin, J. F. and Durand, P. and Feron, A. and Freney, E. and Gaimoz, C. and Goloub, P. and Gomez-Amo, J. L. and Granados-Munoz, M. J. and Grand, N. and Hamonou, E. and Jankowiak, I. and Jeannot, M. and Leon, J. F. and Maille, M. and Mailler, S. and Meloni, D. and Menut, L. and Momboisse, G. and Nicolas, J. and Podvin, T. and Pont, V. and Rea, G. and Renard, J. B. and Roblou, L. and Schepanski, K. and Schwarzenboeck, A. and Sellegri, K. and Sicard, M. and Solmon, F. and Somot, S. and Torres, B. and Totems, J. and Triquet, S. and Verdier, N. and Verwaerde, C. and Waquet, F. and Wenger, J. and Zapf, P., Overview of the Chemistry-Aerosol Mediterranean Experiment/Aerosol Direct Radiative Forcing on the Mediterranean Climate (ChArMEx/ADRIMED) summer 2013 campaign, Atmospheric Chemistry and Physics, 16, 2, 455-504, 2016

188. Markakis, K., Valari, M., Engardt, M., Lacressonniere, G., Vautard, R., and Andersson, C.: Mid-21st century air quality at the urban scale under the influence of changed climate and emissions Ž013 case studies for Paris and Stockholm, Atmos. Chem. Phys., 16, 1877-1894, https://doi.org/10.5194/acp-16-1877-2016, 2016.

187. Petetin Herve, Jean Sciare, Michael Bressi, Valérie Gros, Amandine Rosso, Olivier Sanchez, Roland Sarda-Estève, Jean-Eudes Petit, Matthias Beekmann, Assessing the ammonium nitrate formation regime in the Paris megacity and its representation in the CHIMERE model, Atmos. Chem. Phys., 16, 10419-1044, 10.5194/acp-16-10419-2016, 2016

186. Rea, Geraldine and Paton-Walsh, Clare and Turquety, Solene and Cope, Martin and Griffith, David, Impact of the New South Wales fires during October 2013 on regional air quality in eastern Australia, Atmospheric Environment, 131, 150-163, 2016

185. Staufer, J., Broquet, G., Breon, F.-M., Puygrenier, V., Chevallier, F., Xueref-Rémy, I., Dieudonné, E., Lopez, M., Schmidt, M., Ramonet, M., Perrussel, O., Lac, C., Wu, L., and Ciais, P.: The first 1-year-long estimate of the Paris region fossil fuel CO2 emissions based on atmospheric inversion, Atmos. Chem. Phys., 16, 14703-14726, doi:10.5194/acp-16-14703-2016, 2016.

184. Watson L., Lacressonnière G., Gauss M., Engardt M., Andersson C., Josse B., Marécal V., Nyiri A., Sobolowski S., Siour G., Szopa S., Vautard R., Impact of emissions and +2oC climate change upon future ozone and nitrogen dioxide over Europe, Atmospheric Environment, 142, 271-285, 2016

183. Wu, L., Broquet, G., Ciais, P., Bellassen, V., Vogel, F., Chevallier, F., Xueref-Remy, I., and Wang, Y.: What would dense atmospheric observation networks bring to the quantification of city CO2 emissions?, Atmos. Chem. Phys., 16, 7743-7771, doi:10.5194/acp-16-7743-2016, 2016.

182. Beauchamp, Maxime and Malherbe, Laure and de Fouquet, Chantal, A pragmatic approach to estimate the number of days in exceedance of PM10 limit value, Atmospheric Environment, 111, 79-93, 2015

181. Beekmann, M. and Prevot, A. S. H. and Drewnick, F. and Sciare, J. and Pandis, S. N. and van der Gon, H. A. C. Denier and Crippa, M. and Freutel, F. and Poulain, L. and Ghersi, V. and Rodriguez, E. and Beirle, S. and Zotter, P. and von der Weiden-Reinmueller, S. L. and Bressi, M. and Fountoukis, C. and Petetin, H. and Szidat, S. and Schneider, J. and Rosso, A. and El Haddad, I. and Megaritis, A. and Zhang, Q. J. and Michoud, V. and Slowik, J. G. and Moukhtar, S. and Kolmonen, P. and Stohl, A. and Eckhardt, S. and Borbon, A. and Gros, V. and Marchand, N. and Jaffrezo, J. L. and Schwarzenboeck, A. and Colomb, A. and Wiedensohler, A. and Borrmann, S. and Lawrence, M. and Baklanov, A. and Baltensperger, U., In situ, satellite measurement and model evidence on the dominant regional contribution to fine particulate matter levels in the Paris megacity, Atmospheric Chemistry and Physics, 15, 16, 9577-9591, 2015

180. Berchet, A. and Pison, I. and Chevallier, F. and Bousquet, P. and Bonne, J. L. and Paris, J. D., Objectified quantification of uncertainties in Bayesian atmospheric inversions, Geoscientific Model Development, 8, 5, 1525-1546, 2015

179. Boichu, M. and Clarisse, L. and Pere, J. C. and Herbin, H. and Goloub, P. and Thieuleux, F. and Ducos, F. and Clerbaux, C. and Tanre, D., Temporal variations of flux and altitude of sulfur dioxide emissions during volcanic eruptions: implications for long-range dispersal of volcanic clouds, Atmospheric Chemistry and Physics, 15, 14, 8381-8400, 2015

178. Breon, F. M. and Broquet, G. and Puygrenier, V. and Chevallier, F. and Xueref-Remy, I. and Ramonet, M. and Dieudonne, E. and Lopez, M. and Schmidt, M. and Perrussel, O. and Ciais, P., An attempt at estimating Paris area CO2 emissions from atmospheric concentration measurements, Atmospheric Chemistry and Physics, 15, 4, 1707-1724, 2015

177. Cuesta, Juan and Eremenko, Maxim and Flamant, Cyrille and Dufour, Gaelle and Laurent, Benoit and Bergametti, Gilles and Hoepfner, Michael and Orphal, Johannes and Zhou, Daniel, Three-dimensional distribution of a major desert dust outbreak over East Asia in March 2008 derived from IASI satellite observations, Journal of Geophysical Research-Atmospheres, 120, 14, 7099-7127, 2015

176. Ding, J. and van der A, R. J. and Mijling, B. and Levelt, P. F. and Hao, N., NOx emission estimates during the 2014 Youth Olympic Games in Nanjing, Atmospheric Chemistry and Physics, 15, 16, 9399-9412, 2015

175. Hamaoui-Laguel, Lynda and Vautard, Robert and Liu, Li and Solmon, Fabien and Viovy, Nicolas and Khvorostyanov, Dmitry and Essl, Franz and Chuine, Isabelle and Colette, Augustin and Semenov, Mikhail A. and Schaffhauser, Alice and Storkey, Jonathan and Thibaudon, Michel and Epstein, Michelle M., Effects of climate change and seed dispersal on airborne ragweed pollen loads in Europe, Nature Climate Change, 5, 8, 766-U186, 2015

174. Kadygrov, N., Broquet, G., Chevallier, F., Rivier, L., Gerbig, C., and Ciais, P.: On the potential of the ICOS atmospheric CO2measurement network for estimating the biogenic CO2 budget of Europe, Atmos. Chem. Phys., 15, 12765-12787, https://doi.org/10.5194/acp-15-12765-2015, 2015.

173. Konovalov, I. B. and Beekmann, M. and Berezin, E. V. and Petetin, H. and Mielonen, T. and Kuznetsova, I. N. and Andreae, M. O., The role of semi-volatile organic compounds in the mesoscale evolution of biomass burning aerosol: a modeling case study of the 2010 mega-fire event in Russia, Atmospheric Chemistry and Physics, 15, 23, 13269-13297, 2015

172. Likhvar, Victoria N. and Pascal, Mathilde and Markakis, Konstantinos and Colette, Augustin and Hauglustaine, Didier and Valari, Myrto and Klimont, Zbigniew and Medina, Sylvia and Kinney, Patrick, A multi-scale health impact assessment of air pollution over the 21st century, Science of the Total Environment, 514, 439-449, 2015

171. Marecal, V. and Peuch, V. H. and Andersson, C. and Andersson, S. and Arteta, J. and Beekmann, M. and Benedictow, A. and Bergstrom, R. and Bessagnet, B. and Cansado, A. and Cheroux, F. and Colette, A. and Coman, A. and Curier, R. L. and van der Gon, H. A. C. Denier and Drouin, A. and Elbern, H. and Emili, E. and Engelen, R. J. and Eskes, H. J. and Foret, G. and Friese, E. and Gauss, M. and Giannaros, C. and Guth, J. and Joly, M. and Jaumouille, E. and Josse, B. and Kadygrov, N. and Kaiser, J. W. and Krajsek, K. and Kuenen, J. and Kumar, U. and Liora, N. and Lopez, E. and Malherbe, L. and Martinez, I. and Melas, D. and Meleux, F. and Menut, L. and Moinat, P. and Morales, T. and Parmentier, J. and Piacentini, A. and Plu, M. and Poupkou, A. and Queguiner, S. and Robertson, L. and Rouil, L. and Schaap, M. and Segers, A. and Sofiev, M. and Tarasson, L. and Thomas, M. and Timmermans, R. and Valdebenito, A. and van Velthoven, P. and van Versendaal, R. and Vira, J. and Ung, A., A regional air quality forecasting system over Europe: the MACC-II daily ensemble production, Geoscientific Model Development, 8, 9, 2777-2813, 2015

170. Markakis, K. and Valari, M. and Perrussel, O. and Sanchez, O. and Honore, C., Climate-forced air-quality modeling at the urban scale: sensitivity to model resolution, emissions and meteorology, Atmospheric Chemistry and Physics, 15, 13, 7703-7723, 2015

169. Monks P.S., A.T. Archibald, A. Colette, O. Cooper, M. Coyle, R. Derwent, D. Fowler, C. Granier, K.S. Law, G.E. Mills, D.S. Stevenson, O. Tarasova, V. Thouret, E. von Schneidemesser, R. Sommariva, O. Wild and M.L. Williams, Tropospheric ozone and its precursors from the urban to the global scale from air quality to short-lived climate forcer, Atmos. Chem. Phys. 15(2015), pp. 8889-8973.<

168. Pere, J. C. and Bessagnet, B. and Pont, V. and Mallet, M. and Minvielle, F., Influence of the aerosol solar extinction on photochemistry during the 2010 Russian wildfires episode, Atmospheric Chemistry and Physics, 15, 19, 10983-10998, 2015

167. Petetin, H. and Beekmann, M. and Colomb, A. and van der Gon, H. A. C. Denier and Dupont, J. C. and Honore, C. and Michoud, V. and Morille, Y. and Perrussel, O. and Schwarzenboeck, A. and Sciare, J. and Wiedensohler, A. and Zhang, Q. J., Evaluating BC and NOx emission inventories for the Paris region from MEGAPOLI aircraft measurements, Atmospheric Chemistry and Physics, 15, 17, 9799-9818, 2015

166. Ratola, Nuno and Jimenez-Guerrero, Pedro, Combined field/modelling approaches to represent the air-vegetation distribution of benzo[a]pyrene using different vegetation species, Atmospheric Environment, 106, 34-42, 2015

165. Rea, G. and Turquety, S. and Menut, L. and Briant, R. and Mailler, S. and Siour, G., Source contributions to 2012 summertime aerosols in the Euro-Mediterranean region, Atmospheric Chemistry and Physics, 15, 14, 8013-8036, 2015

164. Terrenoire, E. and Bessagnet, B. and Rouil, L. and Tognet, F. and Pirovano, G. and Letinois, L. and Beauchamp, M. and Colette, A. and Thunis, P. and Amann, M. and Menut, L., High-resolution air quality simulation over Europe with the chemistry transport model CHIMERE, Geoscientific Model Development, 8, 1, 21-42, 2015

163. Schaap, M. and Cuvelier, C. and Hendriks, C. and Bessagnet, B. and Baldasano, J. M. and Colette, A. and Thunis, P. and Karam, D. and Fagerli, H. and Graff, A. and Kranenburg, R. and Nyiri, A. and Pay, M. T. and Rouil, L. and Schulz, M. and Simpson, D. and Stern, R. and Terrenoire, E. and Wind, P., Performance of European chemistry transport models as function of horizontal resolution, Atmospheric Environment, 112, 90-105, 2015

162. Schucht, Simone and Colette, Augustin and Rao, Shilpa and Holland, Mike and Schoepp, Wolfgang and Kolp, Peter and Klimont, Zbigniew and Bessagnet, Bertrand and Szopa, Sophie and Vautard, Robert and Brignon, Jean-Marc and Rouil, Laurence, Moving towards ambitious climate policies: Monetised health benefits from improved air quality could offset mitigation costs in Europe, Environmental Science and Policy, 50, 252-269, 2015

161. Schaap M., C. Cuvelier, C. Hendriks, B. Bessagnet, J.M. Baldasano, A. Colette, P. Thunis, D. Karam, H. Fagerli, A. Graff, R. Kranenburg, A. Nyiri, M.T. Pay, L. Rouil, M. Schulz, D. Simpson, R. Stern, E. Terrenoire and P. Wind, Performance of European chemistry transport models as function of horizontal resolution, Atmospheric Environment 112(2015), pp. 90-105.

160. Shaiganfar, R. and Beirle, S. and Petetin, H. and Zhang, Q. and Beekmann, M. and Wagner, T., New concepts for the comparison of tropospheric NO2 column densities derived from car-MAX-DOAS observations, OMI satellite observations and the regional model CHIMERE during two MEGAPOLI campaigns in Paris 2009/10, Atmospheric Measurement Techniques, 8, 7, 2827-2852, 2015

159. Sofiev, M. and Berger, U. and Prank, M. and Vira, J. and Arteta, J. and Belmonte, J. and Bergmann, K. C. and Cheroux, F. and Elbern, H. and Friese, E. and Galan, C. and Gehrig, R. and Khvorostyanov, D. and Kranenburg, R. and Kumar, U. and Marecal, V. and Meleux, F. and Menut, L. and Pessi, A. M. and Robertson, L. and Ritenberga, O. and Rodinkova, V. and Saarto, A. and Segers, A. and Severova, E. and Sauliene, I. and Siljamo, P. and Steensen, B. M. and Teinemaa, E. and Thibaudon, M. and Peuch, V. H., MACC regional multi-model ensemble simulations of birch pollen dispersion in Europe, Atmospheric Chemistry and Physics, 15, 14, 8115-8130, 2015

158. Wang, Rong and Tao, Shu and Balkanski, Yves and Ciais, Philippe and Boucher, Olivier and Liu, Junfeng and Piao, Shilong and Shen, Huizhong and Vuolo, Maria Raffaella and Valari, Myrto and Chen, Han and Chen, Yuanchen and Cozic, Anne and Huang, Ye and Li, Bengang and Li, Wei and Shen, Guofeng and Wang, Bin and Zhang, Yanyan, Exposure to ambient black carbon derived from a unique inventory and high-resolution model, Proceedings of the National Academy of Sciences of the United States of America, 111, 7, 2459-2463, 2015

157. Zhang, Q. J. and Beekmann, M. and Freney, E. and Sellegri, K. and Pichon, J. M. and Schwarzenboeck, A. and Colomb, A. and Bourrianne, T. and Michoud, V. and Borbon, A., Formation of secondary organic aerosol in the Paris pollution plume and its impact on surrounding regions, Atmospheric Chemistry and Physics, 15, 24, 13973-13992, 2015

156. Bentayeb, M. and Stempfelet, M. and Wagner, V. and Zins, M. and Bonenfant, S. and Songeur, C. and Sanchez, O. and Rosso, A. and Brulfert, G. and Rios, I. and Chaxel, E. and Virga, J. and Armengaud, A. and Rossello, P. and Riviere, E. and Bernard, M. and Vasbien, F. and Deprost, R., Retrospective modeling outdoor air pollution at a fine spatial scale in France, 1989-2008, Atmospheric Environment, 92, 267-279, 2014

155. Boichu, Marie and Clarisse, Lieven and Khvorostyanov, Dmitry and Clerbaux, Cathy, Improving volcanic sulfur dioxide cloud dispersal forecasts by progressive assimilation of satellite observations, Geophysical Research Letters, 41, 7, 2637-2643, 2014

154. Foret, G. and Eremenko, M. and Cuesta, J. and Sellitto, P. and Barre, J. and Gaubert, B. and Coman, A. and Dufour, G. and Liu, X. and Joly, M. and Doche, C. and Beekmann, M., Ozone pollution: What can we see from space? A case study, Journal of Geophysical Research-Atmospheres, 119, 13, 2014

153. Garcia-Gomez, H. and Garrido, J. L. and Vivanco, M. G. and Lassaletta, L. and Rabago, I. and Avila, A. and Tsyro, S. and Sanchez, G. and Gonzalez Ortiz, A. and Gonzalez-Fernandez, I. and Alonso, R., Nitrogen deposition in Spain: Modeled patterns and threatened habitats within the Natura 2000 network, Science of the Total Environment, 485, 450-460, 2014

152. Gaubert, B. and Coman, A. and Foret, G. and Meleux, F. and Ung, A. and Rouil, L. and Ionescu, A. and Candau, Y. and Beekmann, M., Regional scale ozone data assimilation using an ensemble Kalman filter and the CHIMERE chemical transport model, Geoscientific Model Development, 7, 1, 283-302, 2014

151. Hamaoui-Laguel, Lynda and Meleux, Frederik and Beekmann, Matthias and Bessagnet, Bertrand and Genermont, Sophie and Cellier, Pierre and Letinois, Laurent, Improving ammonia emissions in air quality modelling for France, Atmospheric Environment, 92, 584-595, 2014

150. Jacob, Daniela and Petersen, Juliane and Eggert, Bastian and Alias, Antoinette and Christensen, Ole Bossing and Bouwer, Laurens M. and Braun, Alain and Colette, Augustin and Deque, Michel and Georgievski, Goran and Georgopoulou, Elena and Gobiet, Andreas and Menut, Laurent and Nikulin, Grigory and Haensler, Andreas and Hempelmann, Nils and Jones, Colin and Keuler, Klaus and Kovats, Sari and Kroener, Nico and Kotlarski, Sven and Kriegsmann, Arne and Martin, Eric and van Meijgaard, Erik and Moseley, Christopher and Pfeifer, Susanne and Preuschmann, Swantje and Radermacher, Christine and Radtke, Kai and Rechid, Diana and Rounsevell, Mark and Samuelsson, Patrick and Somot, Samuel and Soussana, Jean-Francois and Teichmann, Claas and Valentini, Riccardo and Vautard, Robert and Weber, Bjorn and Yiou, Pascal, EURO-CORDEX: new high-resolution climate change projections for European impact research, Regional Environmental Change, 14, 2, 563-578, 2014

149. Kiesewetter, G. and Borken-Kleefeld, J. and Schoepp, W. and Heyes, C. and Thunis, P. and Bessagnet, B. and Terrenoire, E. and Gsella, A. and Amann, M., Modelling NO2 concentrations at the street level in the GAINS integrated assessment model: projections under current legislation, Atmospheric Chemistry and Physics, 14, 2, 813-829, 2014

148. Konovalov, I. B. and Berezin, E. V. and Ciais, P. and Broquet, G. and Beekmann, M. and Hadji-Lazaro, J. and Clerbaux, C. and Andreae, M. O. and Kaiser, J. W. and Schulze, E. D., Constraining CO2 emissions from open biomass burning by satellite observations of co-emitted species: a method and its application to wildfires in Siberia, Atmospheric Chemistry and Physics, 14, 19, 10383-10410, 2014

147. Markakis, K. and Valari, M. and Colette, A. and Sanchez, O. and Perrussel, O. and Honore, C. and Vautard, R. and Klimont, Z. and Rao, S., Air quality in the mid-21st century for the city of Paris under two climate scenarios; from the regional to local scale, Atmospheric Chemistry and Physics, 14, 14, 7323-7340, 2014

146. Pere, J. C. and Bessagnet, B. and Mallet, M. and Waquet, F. and Chiapello, I. and Minvielle, F. and Pont, V. and Menut, L., Direct radiative effect of the Russian wildfires and its impact on air temperature and atmospheric dynamics during August 2010, Atmospheric Chemistry and Physics, 14, 4, 1999-2013, 2014

145. Petetin, H. and Beekmann, M. and Sciare, J. and Bressi, M. and Rosso, A. and Sanchez, O. and Ghersi, V., A novel model evaluation approach focusing on local and advected contributions to urban PM2.5 levels - application to Paris, France, Geoscientific Model Development, 7, 4, 1483-1505, 2014

144. Viana M., P. Hammingh, A. Colette, X. Querol, B. Degraeuwe, I.d. Vlieger and J. van Aardenne, Impact of maritime transport emissions on coastal air quality in Europe, Atmospheric Environment 90(2014), pp. 96-105.

143. Berchet, A. and Pison, I. and Chevallier, F. and Bousquet, P. and Conil, S. and Geever, M. and Laurila, T. and Lavric, J. and Lopez, M. and Moncrieff, J. and Necki, J. and Ramonet, M. and Schmidt, M. and Steinbacher, M. and Tarniewicz, J., Towards better error statistics for atmospheric inversions of methane surface fluxes, Atmospheric Chemistry and Physics, 13, 14, 7115-7132, 2013

142. Berezin, E. V. and Konovalov, I. B. and Ciais, P. and Richter, A. and Tao, S. and Janssens-Maenhout, G. and Beekmann, M. and Schulze, E. D., Multiannual changes of CO2 emissions in China: indirect estimates derived from satellite measurements of tropospheric NO2 columns, Atmospheric Chemistry and Physics, 13, 18, 9415-9438, 2013

141. Boichu, M. and Menut, L. and Khvorostyanov, D. and Clarisse, L. and Clerbaux, C. and Turquety, S. and Coheur, P. F., Inverting for volcanic SO2 flux at high temporal resolution using spaceborne plume imagery and chemistry-transport modelling: the 2010 Eyjafjallajokull eruption case study, Atmospheric Chemistry and Physics, 13, 17, 8569-8584, 2013

140. Borrego, C. and Souto, J. A. and Monteiro, A. and Dios, M. and Rodriguez, A. and Ferreira, J. and Saavedra, S. and Casares, J. J. and Miranda, A. I., The role of transboundary air pollution over Galicia and North Portugal area, Environmental Science and Pollution Research, 20, 5, 2924-2936, 2013

139. Broquet, G. and Chevallier, F. and Breon, F. M. and Kadygrov, N. and Alemanno, M. and Apadula, F. and Hammer, S. and Haszpra, L. and Meinhardt, F. and Morgui, J. A. and Necki, J. and Piacentino, S. and Ramonet, M. and Schmidt, M. and Thompson, R. L. and Vermeulen, A. T. and Yver, C. and Ciais, P., Regional inversion of CO2 ecosystem fluxes from atmospheric measurements: reliability of the uncertainty estimates, Atmospheric Chemistry and Physics, 13, 17, 9039-9056, 2013

138. Cuesta, J. and Eremenko, M. and Liu, X. and Dufour, G. and Cai, Z. and Hoepfner, M. and von Clarmann, T. and Sellitto, P. and Foret, G. and Gaubert, B. and Beekmann, M. and Orphal, J. and Chance, K. and Spurr, R. and Flaud, J. M., Satellite observation of lowermost tropospheric ozone by multispectral synergism of IASI thermal infrared and GOME-2 ultraviolet measurements over Europe, Atmospheric Chemistry and Physics, 13, 19, 9675-9693, 2013

137. Dall'Osto, M. and Querol, X. and Alastuey, A. and Minguillon, M. C. and Alier, M. and Amato, F. and Brines, M. and Cusack, M. and Grimalt, J. O. and Karanasiou, A. and Moreno, T. and Pandolfi, M. and Pey, J. and Reche, C. and Ripoll, A. and Tauler, R. and Van Drooge, B. L. and Viana, M. and Harrison, R. M. and Gietl, J. and Beddows, D. and Bloss, W. and O'Dowd, C. and Ceburnis, D. and Martucci, G. and Ng, N. L. and Worsnop, D. and Wenger, J. and Mc Gillicuddy, E. and Sodeau, J. and Healy, R. and Lucarelli, F. and Nava, S. and Jimenez, J. L. and Gomez Moreno, F. and Artinano, B. and Prevot, A. S. H. and Pfaffenberger, L. and Frey, S. and Wilsenack, F. and Casabona, D. and Jimenez-Guerrero, P. and Gross, D. and Cots, N., Presenting SAPUSS: Solving Aerosol Problem by Using Synergistic Strategies in Barcelona, Spain, Atmospheric Chemistry and Physics, 13, 17, 8991-9019, 2013

136. Jerez, S. and Jimenez-Guerrero, P. and Montavez, J. P. and Trigo, R. M., Impact of the North Atlantic Oscillation on European aerosol ground levels through local processes: a seasonal model-based assessment using fixed anthropogenic emissions, Atmospheric Chemistry and Physics, 13, 22, 11195-11207, 2013

135. Jimenez-Guerrero, Pedro and Gomez-Navarro, Juan J. and Baro, Rocio and Lorente, Raquel and Ratola, Nuno and Montavez, Juan P., Is there a common pattern of future gas-phase air pollution in Europe under diverse climate change scenarios?, Climatic Change, 121, 4, 661-671, 2013

134. Jimenez-Guerrero, P. and Jerez, S. and Montavez, J. P. and Trigo, R. M., Uncertainties in future ozone and PM10 projections over Europe from a regional climate multiphysics ensemble, Geophysical Research Letters, 40, 21, 5764-5769, 2013

133. Landi, T. C. and Curci, G. and Carbone, C. and Menut, L. and Bessagnet, B. and Giulianelli, L. and Paglione, M. and Facchini, M. C., Simulation of size-segregated aerosol chemical composition over northern Italy in clear sky and wind calm conditions, Atmospheric Research, 125, 1-11, 2013

132. Mijling, B. and van der A, R. J. and Zhang, Q., Regional nitrogen oxides emission trends in East Asia observed from space, Atmospheric Chemistry and Physics, 13, 23, 12003-12012, 2013

131. Monteiro A., Ribeiro I., Tchepel O., Se E., Ferreira J., Carvalho A., Martins V., Strunk A., Galmarini S., Elbern H., Schaap M., Builtjes P., Miranda A.I., Borrego C. (2013). Bias correction techniques to improve air quality ensemble predictions: focus on O3 and PM over Portugal. Environmental Modelling and Assessment 18 (5), 533-546. https://doi.org/10.1007/s10666-013-9358-2

130. Monteiro A., Ribeiro I., Tchepel O., Carvalho A., Martins H., Se E., Ferreira J., Galmarini S., Miranda A.I., Borrego C. (2013). Ensemble techniques to improve air quality assessment: focus on O3 and PM. Environmental Modelling and Assessment 19 (3), 249-257. https://doi.org/10.1007/s10666-012-9344-0

129. Moorthy, K. Krishna and Beegum, S. Naseema and Srivastava, N. and Satheesh, S. K. and Chin, Mian and Blond, Nadege and Babu, S. Suresh and Singh, S., Performance evaluation of chemistry transport models over India, Atmospheric Environment, 71, 210-225, 2013

128. Zhang, Q. J. and Beekmann, M. and Drewnick, F. and Freutel, F. and Schneider, J. and Crippa, M. and Prevot, A. S. H. and Baltensperger, U. and Poulain, L. and Wiedensohler, A. and Sciare, J. and Gros, V. and Borbon, A. and Colomb, A. and Michoud, V. and Doussin, J. F. and van der Gon, H. A. C. Denier and Haeffelin, M. and Dupont, J. C. and Siour, G. and Petetin, H. and Bessagnet, B. and Pandis, S. N. and Hodzic, A. and Sanchez, O. and Honore, C. and Perrussel, O., Formation of organic aerosol in the Paris region during the MEGAPOLI summer campaign: evaluation of the volatility-basis-set approach within the CHIMERE model, Atmospheric Chemistry and Physics, 13, 11, 5767-5790, 2013

127. Anav, A. and Menut, L. and Khvorostyanov, D. and Viovy, N., A comparison of two canopy conductance parameterizations to quantify the interactions between surface ozone and vegetation over Europe, Journal of Geophysical Research-Biogeosciences, 117, 2012

126. Coman, A. and Foret, G. and Beekmann, M. and Eremenko, M. and Dufour, G. and Gaubert, B. and Ung, A. and Schmechtig, C. and Flaud, J. M. and Bergametti, G., Assimilation of IASI partial tropospheric columns with an Ensemble Kalman Filter over Europe, Atmospheric Chemistry and Physics, 12, 5, 2513-2532, 2012

125. Drobinski, Philippe and Anav, Alesandro and Lebeaupin Brossier, Cindy and Samson, Guillaume and Stefanon, Marc and Bastin, Sophie and Baklouti, Melika and Beranger, Karine and Beuvier, Jonathan and Bourdalle-Badie, Romain and Coquart, Laure and D'Andrea, Fabio and de Noblet-Ducoudre, Nathalie and Diaz, Frederic and Dutay, Jean-Claude and Ethe, Christian and Foujols, Marie-Alice and Khvorostyanov, Dmitry and Madec, Gurvan and Mancip, Martial and Masson, Sebastien and Menut, Laurent and Palmieri, Julien and Polcher, Jan and Turquety, Solene and Valcke, Sophie and Viovy, Nicolas, Model of the Regional Coupled Earth system (MORCE): Application to process and climate studies in vulnerable regions, Environmental Modelling and Software, 35, 1-18, 2012

124. Jimenez-Guerrero, Pedro and Pedro Montavez, Juan and Jose Gomez-Navarro, Juan and Jerez, Sonia and Lorente-Plazas, Raquel, Impacts of climate change on ground level gas-phase pollutants and aerosols in the Iberian Peninsula for the late XXI century, Atmospheric Environment, 55, 483-495, 2012

123. Jimenez-Guerrero, Pedro and Pedro Montavez, Juan and Jose Gomez-Navarro, Juan and Jerez, Sonia and Lorente-Plazas, Raquel, Impacts of climate change on ground level gas-phase pollutants and aerosols in the Iberian Peninsula for the late XXI century, Atmospheric Environment, 55, 483-495, 2012

122. Konovalov, I. B. and Beekmann, M. and D'Anna, B. and George, C., Significant light induced ozone loss on biomass burning aerosol: Evidence from chemistry-transport modeling based on new laboratory studies, Geophysical Research Letters, 39, 2012

121. Mijling, B. and van der A, R. J., Using daily satellite observations to estimate emissions of short-lived air pollutants on a mesoscopic scale, Journal of Geophysical Research-Atmospheres, 117, 2012

120. Pere, J. C. and Colette, A. and Dubuisson, P. and Bessagnet, B. and Mallet, M. and Pont, V., Impacts of future air pollution mitigation strategies on the aerosol direct radiative forcing over Europe, Atmospheric Environment, 62, 451-460, 2012

119. Pernigotti, D. and Georgieva, E. and Thunis, P. and Bessagnet, B., Impact of meteorology on air quality modeling over the Po valley in northern Italy, Atmospheric Environment, 51, 303-310, 2012

118. Pirovano, G. and Balzarini, A. and Bessagnet, B. and Emery, C. and Kallos, G. and Meleux, F. and Mitsakou, C. and Nopmongcol, U. and Riva, G. M. and Yarwood, G., Investigating impacts of chemistry and transport model formulation on model performance at European scale, Atmospheric Environment, 53, 93-109, 2012

117. Schere, Kenneth and Flemming, Johannes and Vautard, Robert and Chemel, Charles and Colette, Augustin and Hogrefe, Christian and Bessagnet, Bertrand and Meleux, Frederik and Mathur, Rohit and Roselle, Shawn and Hu, Rong-Ming and Sokhi, Ranjeet S. and Rao, S. Trivikrama and Galmarini, Stefano, Trace gas/aerosol boundary concentrations and their impacts on continental-scale AQMEII modeling domains, Atmospheric Environment, 53, 38-50, 2012

116. Solazzo, Efisio and Bianconi, Roberto and Pirovano, Guido and Matthias, Volker and Vautard, Robert and Moran, Michael D. and Appel, K. Wyat and Bessagnet, Bertrand and Brandt, Jorgen and Christensen, Jesper H. and Chemel, Charles and Coll, Isabelle and Ferreira, Joana and Forkel, Renate and Francis, Xavier V. and Grell, Georg and Grossi, Paola and Hansen, Ayoe B. and Miranda, Ana Isabel and Nopmongcol, Uarporn and Prank, Marje and Sartelet, Karine N. and Schaap, Martijn and Silver, Jeremy D. and Sokhi, Ranjeet S. and Vira, Julius and Werhahn, Johannes and Wolke, Ralf and Yarwood, Greg and Zhang, Junhua and Rao, S. Trivikrama and Galmarini, Stefano, Operational model evaluation for particulate matter in Europe and North America in the context of AQMEII, Atmospheric Environment, 53, 75-92, 2012

115. Solazzo, Efisio and Bianconi, Roberto and Vautard, Robert and Appel, K. Wyat and Moran, Michael D. and Hogrefe, Christian and Bessagnet, Bertrand and Brandt, Jorgen and Christensen, Jesper H. and Chemel, Charles and Coll, Isabelle and van der Gon, Hugo Denier and Ferreira, Joana and Forkel, Renate and Francis, Xavier V. and Grell, George and Grossi, Paola and Hansen, Ayoe B. and Jericevic, Amela and Kraljevic, Luksa and Miranda, Ana Isabel and Nopmongcol, Uarporn and Pirovano, Guido and Prank, Marje and Riccio, Angelo and Sartelet, Karine N. and Schaap, Martijn and Silver, Jeremy D. and Sokhi, Ranjeet S. and Vira, Julius and Werhahn, Johannes and Wolke, Ralf and Yarwood, Greg and Zhang, Junhua and Rao, S. Trivikrama and Galmarini, Stefano, Model evaluation and ensemble modelling of surface-level ozone in Europe and North America in the context of AQMEII, Atmospheric Environment, 53, 60-74, 2012

114. Stromatas, S. and Turquety, S. and Menut, L. and Chepfer, H. and Pere, J. C. and Cesana, G. and Bessagnet, B., Lidar signal simulation for the evaluation of aerosols in chemistry transport models, Geoscientific Model Development, 5, 6, 1543-1564, 2012

113. Tuccella, Paolo and Curci, Gabriele and Visconti, Guido and Bessagnet, Bertrand and Menut, Laurent and Park, Rokjin J., Modeling of gas and aerosol with WRF/Chem over Europe: Evaluation and sensitivity study, Journal of Geophysical Research-Atmospheres, 117, 2012

112. Vautard, Robert and Moran, Michael D. and Solazzo, Efisio and Gilliam, Robert C. and Matthias, Volker and Bianconi, Roberto and Chemel, Charles and Ferreira, Joana and Geyer, Beate and Hansen, Ayoe B. and Jericevic, Amela and Prank, Marje and Segers, Arjo and Silver, Jeremy D. and Werhahn, Johannes and Wolke, Ralf and Rao, S. T. and Galmarini, Stefano, Evaluation of the meteorological forcing used for the Air Quality Model Evaluation International Initiative (AQMEII) air quality simulations, Atmospheric Environment, 53, 15-37, 2012

111. Watson, Laura and Lacressonniere, Gwendoline and Gauss, Michael and Engardt, Magnuz and Andersson, Camilla and Josse, Béatrice and Marécal, Virginie and Nyiri, Agnes and Sobolowski, Stefan and Siour, Guillaume and Vautard, Robert, The impact of meteorological forcings on gas phase air pollutants over Europe, Atmospheric Environment, 119, 240-257, Wilson, R. C. and Fleming, Z. L. and Monks, P. S. and Clain, G. and Henne, S. and Konovalov, I. B. and Szopa, S. and Menut, L., Have primary emission reduction measures reduced ozone across Europe? An analysis of European rural background ozone trends 1996-2005, Atmospheric Chemistry and Physics, 12, 1, 437-454, 2012

110. Zhang, Q. J. and Laurent, B. and Velay-Lasry, F. and Ngo, R. and Derognat, C. and Marticorena, B. and Albergel, A., An air quality forecasting system in Beijing - Application to the study of dust storm events in China in May 2008, Journal of Environmental Sciences, 24, 1, 102-111, 2012

109. Zyryanov, D. and Foret, G. and Eremenko, M. and Beekmann, M. and Cammas, J. P. and D'Isidoro, M. and Elbern, H. and Flemming, J. and Friese, E. and Kioutsioutkis, I. and Maurizi, A. and Melas, D. and Meleux, F. and Menut, L. and Moinat, P. and Peuch, V. H. and Poupkou, A. and Razinger, M. and Schultz, M. and Stein, O. and Suttie, A. M. and Valdebenito, A. and Zerefos, C. and Dufour, G. and Bergametti, G. and Flaud, J. M., 3-D evaluation of tropospheric ozone simulations by an ensemble of regional Chemistry Transport Model, Atmospheric Chemistry and Physics, 12, 7, 3219-3240, 2012

108. Alonso, Rocio and Vivanco, Marta G. and Gonzalez-Fernandez, Ignacio and Bermejo, Victoria and Palomino, Inmaculada and Luis Garrido, Juan and Elvira, Susana and Salvador, Pedro and Artinano, Begona, Modelling the influence of peri-urban trees in the air quality of Madrid region (Spain), Environmental Pollution, 159, 8-9, 2138-2147, 2011

107. Anav, A. and Menut, L. and Khvorostyanov, D. and Viovy, N., Impact of tropospheric ozone on the Euro-Mediterranean vegetation, Global Change Biology, 17, 7, 2342-2359, 2011

106. Borrego, C. and Monteiro, A. and Pay, M. T. and Ribeiro, I. and Miranda, A. I. and Basart, S. and Baldasano, J. M., How bias-correction can improve air quality forecasts over Portugal, Atmospheric Environment, 45, 37, 6629-6641, 2011

105. Boynard, A. and Beekmann, M. and Foret, G. and Ung, A. and Szopa, S. and Schmechtig, C. and Coman, A., An ensemble assessment of regional ozone model uncertainty with an explicit error representation, Atmospheric Environment, 45, 3, 784-793, 2011

104. Broquet, Gregoire and Chevallier, Frederic and Rayner, Peter and Aulagnier, Celine and Pison, Isabelle and Ramonet, Michel and Schmidt, Martina and Vermeulen, Alex T. and Ciais, Philippe, A European summertime CO2 biogenic flux inversion at mesoscale from continuous in situ mixing ratio measurements, Journal of Geophysical Research-Atmospheres, 116, 2011

103. Carvalho, A. and Monteiro, A. and Flannigan, M. and Solman, S. and Miranda, A. I. and Borrego, C., Forest fires in a changing climate and their impacts on air quality, Atmospheric Environment, 45, 31, 5545-5553, 2011

102. de Fouquet, Chantal and Malherbe, Laure and Ung, Anthony, Geostatistical analysis of the temporal variability of ozone concentrations. Comparison between CHIMERE model and surface observations, Atmospheric Environment, 45, 20, 3434-3446, 2011

101. Hodzic, A. and Jimenez, J. L., Modeling anthropogenically controlled secondary organic aerosols in a megacity: a simplified framework for global and climate models, Geoscientific Model Development, 4, 4, 901-917, 2011

100. Irie, H. and Takashima, H. and Kanaya, Y. and Boersma, K. F. and Gast, L. and Wittrock, F. and Brunner, D. and Zhou, Y. and Van Roozendael, M., Eight-component retrievals from ground-based MAX-DOAS observations, Atmospheric Measurement Techniques, 4, 6, 1027-1044, 2011

99. Jimenez-Guerrero, P. and Jorba, O. and Pay, M. T. and Montavez, J. P. and Jerez, S. and Gomez-Navarro, J. J. and Baldasano, J. M., Comparison of two different sea-salt aerosol schemes as implemented in air quality models applied to the Mediterranean Basin, Atmospheric Chemistry and Physics, 11, 10, 4833-4850, 2011

98. Jimenez-Guerrero, Pedro and Jose Gomez-Navarro, Juan and Jerez, Sonia and Lorente-Plazas, Raquel and Andres Garcia-Valero, Juan and Pedro Montavez, Juan, Isolating the effects of climate change in the variation of secondary inorganic aerosols (SIA) in Europe for the 21st century (1991-2100), Atmospheric Environment, 45, 4, 1059-1063, 2011

97. Konovalov, I. B. and Beekmann, M. and Kuznetsova, I. N. and Yurova, A. and Zvyagintsev, A. M., Atmospheric impacts of the 2010 Russian wildfires: integrating modelling and measurements of an extreme air pollution episode in the Moscow region, Atmospheric Chemistry and Physics, 11, 19, 10031-10056, 2011

96. Kukkonen, J. and Olsson, T. and Schultz, D. M. and Baklanov, A. and Klein, T. and Miranda, A. I. and Monteiro, A. and Hirtl, M. and Tarvainen, V. and Boy, M. and Peuch, V. H. and Poupkou, A. and Kioutsioukis, I. and Finardi, S. and Sofiev, M. and Sokhi, R. and Lehtinen, K. E. J. and Karatzas, K. and San Jose, R. and Astitha, M. and Kallos, G. and Schaap, M. and Reimer, E. and Jakobs, H. and Eben, K., A review of operational, regional-scale, chemical weather forecasting models in Europe, Atmospheric Chemistry and Physics, 12, 1, 1-87, 2011

95. Kuznetsova, I. N. and Konovalov, I. B. and Glazkova, A. A. and Nakhaev, M. I. and Zaripov, R. B. and Lezina, E. A. and Zvyagintsev, A. M. and Beekmann, M., Observed and calculated variability of the particulate matter concentration in Moscow and in Zelenograd, Russian Meteorology and Hydrology, 36, 3, 175-184, 2011

94. Pere, J. C. and Mallet, M. and Pont, V. and Bessagnet, B., Impact of aerosol direct radiative forcing on the radiative budget, surface heat fluxes, and atmospheric dynamics during the heat wave of summer 2003 over western Europe: A modeling study, Journal of Geophysical Research-Atmospheres, 116, 2011

93. Royer, P. and Chazette, P. and Sartelet, K. and Zhang, Q. J. and Beekmann, M. and Raut, J. C., Comparison of lidar-derived PM10 with regional modeling and ground-based observations in the frame of MEGAPOLI experiment, Atmospheric Chemistry and Physics, 11, 20, 10705-10726, 2011

92. Schmechtig, C. and Marticorena, B. and Chatenet, B. and Bergametti, G. and Rajot, J. L. and Coman, A., Simulation of the mineral dust content over Western Africa from the event to the annual scale with the CHIMERE-DUST model, Atmospheric Chemistry and Physics, 11, 14, 7185-7207, 2011

91. Viatte, C. and Gaubert, B. and Eremenko, M. and Hase, F. and Schneider, M. and Blumenstock, T. and Ray, M. and Chelin, P. and Flaud, J. M. and Orphal, J., Tropospheric and total ozone columns over Paris (France) measured using medium-resolution ground-based solar-absorption Fourier-transform infrared spectroscopy, Atmospheric Measurement Techniques, 4, 10, 2323-2331, 2011

90. Xueref-Remy, I. and Bousquet, P. and Carouge, C. and Rivier, L. and Ciais, P., Variability and budget of CO2 in Europe: analysis of the CAATER airborne campaigns - Part 2: Comparison of CO2 vertical variability and fluxes between observations and a modeling framework, Atmospheric Chemistry and Physics, 11, 12, 5673-5684, 2011

89. Aulagnier, C. and Rayner, P. and Ciais, P. and Vautard, R. and Rivier, L. and Ramonet, M., Is the recent build-up of atmospheric CO2 over Europe reproduced by models. Part 2: an overview with the atmospheric mesoscale transport model CHIMERE, Tellus Series B-Chemical and Physical Meteorology, 62, 1, 14-25, 2010

88. Beekmann, M. and Vautard, R., A modelling study of photochemical regimes over Europe: robustness and variability, Atmospheric Chemistry and Physics, 10, 20, 10067-10084, 2010

87. Borrego, C. and Valente, J. and Carvalho, A. and Sa, E. and Lopes, M. and Miranda, A. I., Contribution of residential wood combustion to PM10 levels in Portugal, Atmospheric Environment, 44, 5, 642-651, 2010

86. Carvalho, A. and Monteiro, A. and Solman, S. and Miranda, A. I. and Borrego, C., Climate-driven changes in air quality over Europe by the end of the 21st century, with special reference to Portugal, Environmental Science and Policy, 13, 6, 445-458, 2010

85. Coll, Isabelle and Rousseau, Cecile and Barletta, Barbara and Meinardi, Simone and Blake, Donald R., Evaluation of an urban NMHC emission inventory by measurements and impact on CTM results, Atmospheric Environment, 44, 31, 3843-3855, 2010

84. Curci, G. and Palmer, P. I. and Kurosu, T. P. and Chance, K. and Visconti, G., Estimating European volatile organic compound emissions using satellite observations of formaldehyde from the Ozone Monitoring Instrument, Atmospheric Chemistry and Physics, 10, 23, 11501-11517, 2010
83. Gualtieri, G., Implementing an Operational Ozone Forecasting System Based on WRF/CALMET/CALGRID Models: a 5-Month Case Study over Tuscany, Italy, Water Air and Soil Pollution, 209, 1-4, 269-293, 2010
82. Hodzic, A. and Jimenez, J. L. and Madronich, S. and Canagaratna, M. R. and DeCarlo, P. F. and Kleinman, L. and Fast, J., Modeling organic aerosols in a megacity: potential contribution of semi-volatile and intermediate volatility primary organic compounds to secondary organic aerosol formation, Atmospheric Chemistry and Physics, 10, 12, 5491-5514, 2010
81. Hodzic, A. and Jimenez, J. L. and Prevot, A. S. H. and Szidat, S. and Fast, J. D. and Madronich, S., Can 3-D models explain the observed fractions of fossil and non-fossil carbon in and near Mexico City?, Atmospheric Chemistry and Physics, 10, 22, 10997-11016, 2010
80. Huijnen, V. and Eskes, H. J. and Poupkou, A. and Elbern, H. and Boersma, K. F. and Foret, G. and Sofiev, M. and Valdebenito, A. and Flemming, J. and Stein, O. and Gross, A. and Robertson, L. and D'Isidoro, M. and Kioutsioukis, I. and Friese, E. and Amstrup, B. and Bergstrom, R. and Strunk, A. and Vira, J. and Zyryanov, D. and Maurizi, A. and Melas, D. and Peuch, V. H. and Zerefos, C., Comparison of OMI NO2 tropospheric columns with an ensemble of global and European regional air quality models, Atmospheric Chemistry and Physics, 10, 7, 3273-3296, 2010
79. Konovalov, I. B. and Beekmann, M. and Richter, A. and Burrows, J. P. and Hilboll, A., Multi-annual changes of NOx emissions in megacity regions: nonlinear trend analysis of satellite measurement based estimates, Atmospheric Chemistry and Physics, 10, 17, 8481-8498, 2010
78. Leitao, J. and Richter, A. and Vrekoussis, M. and Kokhanovsky, A. and Zhang, Q. J. and Beekmann, M. and Burrows, J. P., On the improvement of NO2 satellite retrievals - aerosol impact on the airmass factors, Atmospheric Measurement Techniques, 3, 2, 475-493, 2010
77. Pere, J. C. and Mallet, M. and Pont, V. and Bessagnet, B., Evaluation of an aerosol optical scheme in the chemistry-transport model CHIMERE, Atmospheric Environment, 44, 30, 3688-3699, 2010
76. Rolland, M. N. and Gabrielle, B. and Laville, P. and Cellier, P. and Beekmann, M. and Gilliot, J. M. and Michelin, J. and Hadjar, D. and Curci, G., High-resolution inventory of NO emissions from agricultural soils over the Ile-de-France region, Environmental Pollution, 158, 3, 711-722, 2010
75. Sciare, J. and d'Argouges, O. and Zhang, Q. J. and Sarda-Esteve, R. and Gaimoz, C. and Gros, V. and Beekmann, M. and Sanchez, O., Comparison between simulated and observed chemical composition of fine aerosols in Paris (France) during springtime: contribution of regional versus continental emissions, Atmospheric Chemistry and Physics, 10, 24, 11987-12004, 2010
74. Borrego, C. and Sa, E. and Monteiro, A. and Ferreira, J. and Miranda, A. I., Forecasting human exposure to atmospheric pollutants in Portugal - A modelling approach, Atmospheric Environment, 43, 36, 5796-5806, 2009
73. Chaxel, Eric and Chollet, Jean-Pierre, Ozone production from Grenoble city during the August 2003 heat wave, Atmospheric Environment, 43, 31, 4784-4792, 2009
72. Coll, Isabelle and Lasry, Fanny and Fayet, Sylvain and Armengaud, Alexandre and Vautard, Robert, Simulation and evaluation of 2010 emission control scenarios in a Mediterranean area, Atmospheric Environment, 43, 27, 4194-4204, 2009
71. Curci, G. and Beekmann, M. and Vautard, R. and Smiatek, G. and Steinbrecher, R. and Theloke, J. and Friedrich, R., Modelling study of the impact of isoprene and terpene biogenic emissions on European ozone levels, Atmospheric Environment, 43, 7, 1444-1455, 2009
70. de Meij, A. and Gzella, A. and Cuvelier, C. and Thunis, P. and Bessagnet, B. and Vinuesa, J. F. and Menut, L. and Kelder, H. M., The impact of MM5 and WRF meteorology over complex terrain on CHIMERE model calculations, Atmospheric Chemistry and Physics, 9, 17, 6611-6632, 2009
69. de Meij, A. and Thunis, P. and Bessagnet, B. and Cuvelier, C., The sensitivity of the CHIMERE model to emissions reduction scenarios on air quality in Northern Italy, Atmospheric Environment, 43, 11, 1897-1907, 2009
68. Dufour, G. and Wittrock, F. and Camredon, M. and Beekmann, M. and Richter, A. and Aumont, B. and Burrows, J. P., SCIAMACHY formaldehyde observations: constraint for isoprene emission estimates over Europe?, Atmospheric Chemistry and Physics, 9, 5, 1647-1664, 2009
67. Flaounas, E. and Coll, I. and Armengaud, A. and Schmechtig, C., The representation of dust transport and missing urban sources as major issues for the simulation of PM episodes in a Mediterranean area, Atmospheric Chemistry and Physics, 9, 20, 8091-8101, 2009
66. Foret, G. and Hamaoui, L. and Schmechtig, C. and Eremenko, M. and Keim, C. and Dufour, G. and Boynard, A. and Coman, A. and Ung, A. and Beekmann, M., Evaluating the potential of IASI ozone observations to constrain simulated surface ozone concentrations, Atmospheric Chemistry and Physics, 9, 21, 8479-8491, 2009
65. Hodzic, A. and Jimenez, J. L. and Madronich, S. and Aiken, A. C. and Bessagnet, B. and Curci, G. and Fast, J. and Lamarque, J. F. and Onasch, T. B. and Roux, G. and Schauer, J. J. and Stone, E. A. and Ulbrich, I. M., Modeling organic aerosols during MILAGRO: importance of biogenic secondary organic aerosols, Atmospheric Chemistry and Physics, 9, 18, 6949-6981, 2009
64. Konovalov, I. B. and Beekmann, M. and Meleux, F. and Dutot, A. and Foret, G., Combining deterministic and statistical approaches for PM10 forecasting in Europe, Atmospheric Environment, 43, 40, 6425-6434, 2009
63. Konovalov, I. B. and Elanskii, N. F. and Zvyagintsev, A. M. and Belikov, I. B. and Beekmann, M., Validation of chemistry transport model of the lower atmosphere in the central European region of Russia using ground-based and satellite measurement data, Russian Meteorology and Hydrology, 34, 4, 236-242, 2009
62. Mijling, B. and van der A, R. J. and Boersma, K. F. and Van Roozendael, M. and De Smedt, I. and Kelder, H. M., Reductions of NO2 detected from space during the 2008 Beijing Olympic Games, Geophysical Research Letters, 36, 2009
61. Pere, J. C. and Mallet, M. and Bessagnet, B. and Pont, V., Evidence of the aerosol core-shell mixing state over Europe during the heat wave of summer 2003 by using CHIMERE simulations and AERONET inversions, Geophysical Research Letters, 36, 2009
60. Rouil, Laurence and Honore, Cecile and Vautard, Robert and Beekmann, Matthias and Bessagnet, Bertrand and Malherbe, Laure and Meleux, Frederik and Dufour, Anne and Elichegaray, Christian and Flaud, Jean-Marie and Menut, Laurent and Martin, Daniel and Peuch, Aline and Peuch, Vincent-Henri and Poisson, Nathalie, PREV'AIR An Operational Forecasting and Mapping System for Air Quality in Europe, Bulletin of the American Meteorological Society, 90, 1, 73-83, 2009

59. Steinbrecher, Rainer and Smiatek, Gerhard and Koeble, Renate and Seufert, Guenther and Theloke, Jochen and Hauff, Karin and Ciccioli, Paolo and Vautard, Robert and Curci, Gabriele, Intra- and inter-annual variability of VOC emissions from natural and semi-natural vegetation in Europe and neighbouring countries, Atmospheric Environment, 43, 7, 1380-1391, 2009

58. Szopa, S. and Foret, G. and Menut, L. and Cozic, A., Impact of large scale circulation on European summer surface ozone and consequences for modelling forecast, Atmospheric Environment, 43, 6, 1189-1195, 2009

57. Timmermans, Renske M. A. and Schaap, Martijn and Elbern, Hendrik and Siddans, Richard and Tjemkes, Stephen and Vautard, Robert and Builtjes, Peter, An Observing System Simulation Experiment (OSSE) for Aerosol Optical Depth from Satellites, Journal of Atmospheric and Oceanic Technology, 26, 12, 2673-2682, 2009

56. Timmermans, Renske M. A. and Segers, Arjo J. and Builtjes, Peter J. H. and Vautard, Robert and Siddans, Richard and Elbern, Hendrik and Tjemkes, Stephen A. T. and Schaap, Martijn, The Added Value of a Proposed Satellite Imager for Ground Level Particulate Matter Analyses and Forecasts, Ieee Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 2, 4, 271-283, 2009

55. Vautard, R. and Schaap, M. and Bergstrom, R. and Bessagnet, B. and Brandt, J. and Builtjes, P. J. H. and Christensen, J. H. and Cuvelier, C. and Foltescu, V. and Graff, A. and Kerschbaumer, A. and Krol, M. and Roberts, P. and Rouil, L. and Stern, R. and Tarrason, L. and Thunis, P. and Vignati, E. and Wind, P., Skill and uncertainty of a regional air quality model ensemble, Atmospheric Environment, 43, 31, 4822-4832, 2009

54. Vivanco, Marta G. and Palomino, Immaculada and Vautard, Robert and Bessagnet, Bertrand and Martin, Fernando and Menut, Laurent and Jimenez, Santiago, Multi-year assessment of photochemical air quality simulation over Spain, Environmental Modelling and Software, 24, 1, 63-73, 2009

53. Vuolo, Maria Raffaella and Chepfer, Helene and Menut, Laurent and Cesana, Gregory, Comparison of mineral dust layers vertical structures modeled with CHIMERE-DUST and observed with the CALIOP lidar, Journal of Geophysical Research-Atmospheres, 114, 2009

52. Vuolo, Maria Raffaella and Menut, Laurent and Chepfer, Helene, Impact of Transport Schemes on Modeled Dust Concentrations, Journal of Atmospheric and Oceanic Technology, 26, 6, 1135-1143, 2009

51. Deguillaume, L. and Beekmann, M. and Derognat, C., Uncertainty evaluation of ozone production and its sensitivity to emission changes over the Ile-de-France region during summer periods, Journal of Geophysical Research-Atmospheres, 113, D2, 2008

50. Eremenko, M. and Dufour, G. and Foret, G. and Keim, C. and Orphal, J. and Beekmann, M. and Bergametti, G. and Flaud, J. M., Tropospheric ozone distributions over Europe during the heat wave in July 2007 observed from infrared nadir spectra recorded by IASI, Geophysical Research Letters, 35, 18, 2008

49. Honore, Cecile and Rouil, Laurence and Vautard, Robert and Beekmann, Matthias and Bessagnet, Bertrand and Dufour, Anne and Elichegaray, Christian and Flaud, Jean-Marie and Malherbe, Laure and Meleux, Frederik and Menut, Laurent and Martin, Daniel and Peuch, Aline and Peuch, Vincent-Henri and Poisson, Nathalie, Predictability of European air quality: Assessment of 3 years of operational forecasts and analyses by the PREV'AIR system, Journal of Geophysical Research-Atmospheres, 113, D4, 2008

48. Jorba, Oriol and Loridan, Thomas and Jimenez-Guerrero, Pedro and Perez, Carlos and Baldasano, Jose Maria, Linking the advanced research WRF meteorological model with the CHIMERE chemistry-transport model, Environmental Modelling and Software, 23, 8, 1092-1094, 2008

47. Stern, R. and Builtjes, P. and Schaap, M. and Timmermans, R. and Vautard, R. and Hodzic, A. and Memmesheimer, M. and Feldmann, H. and Renner, E. and Wolke, R. and Kerschbaumer, A., A model inter-comparison study focussing on episodes with elevated PM10 concentrations, Atmospheric Environment, 42, 19, 4567-4588, 2008

46. Auger, Ludovic and Legras, Bernard, Chemical segregation by heterogeneous emissions, Atmospheric Environment, 41, 11, 2303-2318, 2007

45. Blond, N. and Boersma, K. F. and Eskes, H. J. and van der A, R. J. and Van Roozendael, M. and De Smedt, I. and Bergametti, G. and Vautard, R., Intercomparison of SCIAMACHY nitrogen dioxide observations, in situ measurements and air quality modeling results over Western Europe, Journal of Geophysical Research-Atmospheres, 112, D10, 2007

44. Deguillaume, L. and Beekmann, M. and Menut, L., Bayesian Monte Carlo analysis applied to regional- scale inverse emission modeling for reactive trace gases, Journal of Geophysical Research-Atmospheres, 112, D2, 2007

43. Drobinski, P. and Said, F. and Arteta, J. and Augustin, P. and Bastin, S. and Brut, A. and Caccia, J. L. and Campistron, B. and Cautenet, S. and Colette, A. and Coll, I. and Corsmeier, U. and Cros, B. and Dabas, A. and Delbarre, H. and Dufour, A. and Durand, P. and Guenard, V. and Hasel, M. and Kalthoff, N. and Kottmeier, C. and Lasry, F. and Lemonsu, A. and Lohou, F. and Masson, V. and Menut, L. and Moppert, C. and Peuch, V. H. and Puygrenier, V. and Reitebuch, O. and Vautard, R., Regional transport and dilution during high-pollution episodes in southern France: Summary of findings from the Field Experiment to Constraint Models of Atmospheric Pollution and Emissions Transport (ESCOMPTE), Journal of Geophysical Research-Atmospheres, 112, D13, 2007

42. Hodzic, A. and Madronich, S. and Bohn, B. and Massie, S. and Menut, L. and Wiedinmyer, C., Wildfire particulate matter in Europe during summer 2003: meso-scale modeling of smoke emissions, transport and radiative effects, Atmospheric Chemistry and Physics, 7, 15, 4043-4064, 2007

41. Lasry, Fanny and Coll, Isabelle and Fayet, Sylvain and Havre, Maxime and Vautard, Robert, Short-term measures for the control of ozone peaks: expertise from CTM simulations, Journal of Atmospheric Chemistry, 57, 2, 107-134, 2007

40. Meleux, Frederik and Solmon, Fabien and Giorgi, Filippo, Increase in summer European ozone amounts due to climate change, Atmospheric Environment, 41, 35, 7577-7587, 2007

39. Monteiro, A. and Miranda, A. I. and Borrego, C. and Vautard, R., Air quality assessment for Portugal, Science of the Total Environment, 373, 1, 22-31, 2007

38. Monteiro, A. and Miranda, A. I. and Borrego, C. and Vautard, R. and Ferreira, J. and Perez, A. T., Long-term assessment of particulate matter using CHIMERE model, Atmospheric Environment, 41, 36, 7726-7738, 2007

37. Pirovano, G. and Coll, I. and Bedogni, M. and Alessandrini, S. and Costa, M. P. and Gabusi, V. and Lasry, F. and Menut, L. and Vautard, R., On the influence of meteorological input on photochemical modelling of a severe episode over a coastal area, Atmospheric Environment, 41, 30, 6445-6464, 2007

36. Pison, Isabelle and Menut, Laurent and Bergametti, Gilles, Inverse modeling of surface NOx anthropogenic emission fluxes in the Paris area during the Air Pollution Over Paris Region (ESQUIF) campaign, Journal of Geophysical Research-Atmospheres, 112, D24, 2007
35. Salameh, T. and Drobinski, P. and Menut, L. and Bessagnet, B. and Flamant, C. and Hodzic, A. and Vautard, R., Aerosol distribution over the western Mediterranean basin during a Tramontane/Mistral event, Annales Geophysicae, 25, 11, 2271-2291, 2007
34. van Loon, M. and Vautard, R. and Schaap, M. and Bergstrom, R. and Bessagnet, B. and Brandt, J. and Builtjes, P. J. H. and Christensen, J. H. and Cuvelier, C. and Graff, A. and Jonson, J. E. and Krol, M. and Langner, J. and Roberts, P. and Rouil, L. and Stern, R. and Tarrason, L. and Thunis, P. and Vignati, E. and White, L. and Wind, P., Evaluation of long-term ozone simulations from seven regional air quality models and their ensemble, Atmospheric Environment, 41, 10, 2083-2097, 2007
33. Vautard, Robert and Maidi, Mohamed and Menut, Laurent and Beekmann, Matthias and Colette, Augustin, Boundary layer photochemistry simulated with a two-stream convection scheme, Atmospheric Environment, 41, 37, 8275-8287, 2007
32. Hodzic, A. and Bessagnet, B. and Vautard, R., A model evaluation of coarse-mode nitrate heterogeneous formation on dust particles, Atmospheric Environment, 40, 22, 4158-4171, 2006
31. Hodzic, A. and Vautard, R. and Chazette, P. and Menut, L. and Bessagnet, B., Aerosol chemical and optical properties over the Paris area within ESQUIF project, Atmospheric Chemistry and Physics, 6, 3257-3280, 2006
30. Hodzic, A. and Vautard, R. and Chepfer, H. and Goloub, P. and Menut, L. and Chazette, P. and Deuze, J. L. and Apituley, A. and Couvert, P., Evolution of aerosol optical thickness over Europe during the August 2003 heat wave as seen from CHIMERE model simulations and POLDER data, Atmospheric Chemistry and Physics, 6, 1853-1864, 2006
29. Konovalov, I. B. and Beekmann, M. and Richter, A. and Burrows, J. P., Inverse modelling of the spatial distribution of NO(x) emissions on a continental scale using satellite data, Atmospheric Chemistry and Physics, 6, 1747-1770, 2006
28. Pison, I. and Menut, L. and Blond, N., Inverse modeling of emissions for local photooxidant pollution: Testing a new methodology with kriging constraints, Annales Geophysicae, 24, 6, 1523-1535, 2006
27. Szopa, S. and Hauglustaine, D. A. and Vautard, R. and Menut, L., Future global tropospheric ozone changes and impact on European air quality, Geophysical Research Letters, 33, 14, 2006
26. Vautard, Robert and Szopa, Sophie and Beekmann, Matthias and Menut, Laurent and Hauglustaine, Didier A. and Rouil, Laurence and Roemer, Michiel, Are decadal anthropogenic emission reductions in Europe consistent with surface ozone observations?, Geophysical Research Letters, 33, 13, 2006
25. Brulfert, G. and Chemel, C. and Chaxel, E. and Chollet, J. P., Modelling photochemistry in alpine valleys, Atmospheric Chemistry and Physics, 5, 2341-2355, 2005
24. Hodzic, A. and Vautard, R. and Bessagnet, B. and Lattuati, M. and Moreto, F., Long-term urban aerosol simulation versus routine particulate matter observations, Atmospheric Environment, 39, 32, 5851-5864, 2005
23. Konovalov, I. B. and Beekmann, M. and Vautard, R. and Burrows, J. P. and Richter, A. and Nuss, H. and Elansky, N., Comparison and evaluation of modelled and GOME measurement derived tropospheric NO(2) columns over Western and Eastern Europe, Atmospheric Chemistry and Physics, 5, 169-190, 2005
22. Monteiro, A. and Vautard, R. and Borrego, C. and Miranda, A. I., Long-term simulations of photo oxidant pollution over Portugal using the CHIMERE model, Atmospheric Environment, 39, 17, 3089-3101, 2005
21. Moukhtar, S. and Bessagnet, B. and Rouil, L. and Simon, V., Monoterpene emissions from Beech (Fagus sylvatica) in a French forest and impact on secondary pollutants formation at regional scale, Atmospheric Environment, 39, 19, 3535-3547, 2005
20. Vautard, R. and Bessagnet, B. and Chin, M. and Menut, L., On the contribution of natural Aeolian sources to particulate matter concentrations in Europe: Testing hypotheses with a modelling approach, Atmospheric Environment, 39, 18, 3291-3303, 2005
19. Vautard, R. and Honore, C. and Beekmann, M. and Rouil, L., Simulation of ozone during the August 2003 heat wave and emission control scenarios, Atmospheric Environment, 39, 16, 2957-2967, 2005
18. Zhang, B. N. and Vautard, R. and Oanh, N. T. K., Comparison of two photochemical modeling systems in a tropical urban area, Journal of the Air and Waste Management Association, 55, 7, 903-918, 2005
17. Blond, N. and Vautard, R., Three-dimensional ozone analyses and their use for short-term ozone forecasts, Journal of Geophysical Research-Atmospheres, 109, D17, 2004
16. Hodzic, A. and Chepfer, H. and Vautard, R. and Chazette, P. and Beekmann, M. and Bessagnet, B. and Chatenet, B. and Cuesta, J. and Drobinski, P. and Goloub, P. and Haeffelin, M. and Morille, Y., Comparison of aerosol chemistry transport model simulations with lidar and Sun photometer observations at a site near Paris, Journal of Geophysical Research-Atmospheres, 109, D23, 2004
15. Labouze, E. and Honore, U. and Moulay, L. and Couffignal, B. and Beekmann, M., Photochemical Ozone Creation Potentials - A new set of characterization factors for different gas species on the scale of Western Europe, International Journal of Life Cycle Assessment, 9, 3, 187-195, 2004
14. Pison, I. and Menut, L., Quantification of the impact of aircraft traffic emissions on tropospheric ozone over Paris area, Atmospheric Environment, 38, 7, 971-983, 2004
13. Wackernagel, H. and Lajaunie, C. and Blond, N. and Roth, C. and Vautard, R., Geostatistical risk mapping with chemical transport model output and ozone station data, Ecological Modelling, 179, 2, 177-185, 2004
12. Beegum, S. Naseema and Gherboudj, Imen and Chaouch, Naira and Couvidat, Florian and Menut, Laurent and Ghedira, Hosni, Simulating aerosols over Arabian Peninsula with CHIMERE: Sensitivity to soil, surface parameters and anthropogenic emission inventories, Atmospheric Environment, 128, 185-197, Beekmann, M. and Derognat, C., Monte Carlo uncertainty analysis of a regional-scale transport chemistry model constrained by measurements from the Atmospheric Pollution Over the Paris Area (ESQUIF) campaign, Journal of Geophysical Research-Atmospheres, 108, D17, 2003

11. Blond, N. and Bel, L. and Vautard, R., Three-dimensional ozone data analysis with an air quality model over the Paris area, Journal of Geophysical Research-Atmospheres, 108, D23, 2003
10. Derognat, C. and Beekmann, M. and Baeumle, M. and Martin, D. and Schmidt, H., Effect of biogenic volatile organic compound emissions on tropospheric chemistry during the Atmospheric Pollution Over the Paris Area (ESQUIF) campaign in the Ile-de-France region, Journal of Geophysical Research-Atmospheres, 108, D17, 2003
9. Schmidt, H. and Martin, D., Adjoint sensitivity of episodic ozone in the Paris area to emissions on the continental scale, Journal of Geophysical Research-Atmospheres, 108, D17, 2003
8. Sillman, S. and Vautard, R. and Menut, L. and Kley, D., O-3-NOx-VOC sensitivity and NOx-VOC indicators in Paris: Results from models and atmospheric pollution over the Paris area (ESQUIF) measurements, Journal of Geophysical Research-Atmospheres, 108, D12, 2003
7. Vautard, R. and Martin, D. and Beekmann, M. and Drobinski, P. and Friedrich, R. and Jaubertie, A. and Kley, D. and Lattuati, M. and Moral, P. and Neininger, B. and Theloke, J., Paris emission inventory diagnostics from ESQUIF airborne measurements and a chemistry transport model, Journal of Geophysical Research-Atmospheres, 108, D17, 2003
6. Vautard, R. and Menut, L. and Beekmann, M. and Chazette, P. and Flamant, P. H. and Gombert, D. and Guedalia, D. and Kley, D. and Lefebvre, M. P. and Martin, D. and Megie, G. and Perros, P. and Toupance, G., A synthesis of the Air Pollution over the Paris Region (ESQUIF) field campaign, Journal of Geophysical Research-Atmospheres, 108, D17, 2003
5. Schmidt, H. and Derognat, C. and Vautard, R. and Beekmann, M., A comparison of simulated and observed ozone mixing ratios for the summer of 1998 in Western Europe, Atmospheric Environment, 35, 36, 6277-6297, 2001
4. Vautard, R. and Beekmann, M. and Roux, J. and Gombert, D., Validation of a hybrid forecasting system for the ozone concentrations over the Paris area, Atmospheric Environment, 35, 14, 2449-2461, 2001
3. Honore, C. and Vautard, R. and Beekmann, M., Low and high NOx chemical regimes in an urban environment, Environmental Modelling and Software, 15, 6-7, 559-564, 2000
2. Honore, C. and Vautard, R. and Beekmann, M., Photochemical regimes in urban atmospheres: The influence of dispersion, Geophysical Research Letters, 27, 13, 1895-1898, 2000
1. Vautard, R. and Beekmann, M. and Menut, L., Applications of adjoint modelling in atmospheric chemistry: sensitivity and inverse modelling, Environmental Modelling and Software, 15, 6-7, 703-709, 2000

B Changes in WRF for the coupling

B.1 Introduction

The WRF/CHIMERE coupled model is distributed with the WRF version 3.6.1, including all necessary changes. As the WRF model evolves every year, we propose in this section the detailed changes, in order to upgrade a new version in the future if necessary. The modified WRF files for the coupling with CHIMERE via OASIS are:

1. The registry files:
   - Registry/registry.dimspec p.200
   - Registry/Registry.EM_COMMON p.200
2. The solver part:
   - dyn_em/module_first_rk_step_part1.F p.203
   - dyn_em/solve_em.F p.204
3. Coupling modules:
   - frame/module_cpl.F p.205
   - frame/module_cpl_oasis3.F p.214
4. Drivers:
   - frame/module_driver_constants.F p.226
   - frame/module_integrate.F p.226
   - main/module_wrf_top.F p.226
5. Physics packages:
   - phys/module_microphysics_driver.F p.227
   - phys/module_radiation_driver.F p.229
   - phys/module_surface_driver.F p.230
   - phys/module_mp_thompson.F p.230
   - phys/module_ra_rrtmg_sw.F p.237
   - phys/module_ra_rrtmg_lw.F p.238

B.2 Registry files

WRF Registry is an "Active data-dictionary" for managing WRF data structures. It allows the WRF software to automatically generate sections of WRF code from a database. About 2000 - 3000 Registry entries are used to auto-generate about 160,000 lines of WRF code. These include allocate statements for state data and local (i1) data, interprocessor communications, such as halo and periodic boundary updates, code for defining and managing run-time configuration information, code for forcing, feedback, shifting, and interpolation of nest data.
The following Registry types have been added or modified for WRF/CHIMERE coupling:

1. Dimspec: Describes dimensions that are used to define arrays in the model;
2. State: describes state variables and arrays in the domain structure;
3. rconfig: describes a configuration (e.g. namelist) variable or array;

4. Package: describes attributes of a package (e.g. microphysics scheme).

The dimension entries are specified in the registry.dimspec, while the state variables, namelist parameters, and physical package parameters are defined in Registry.EM_COMMON.

B.2.1 Registry/registry.dimspec

The following dimensions have to be defined for WRF/CHIMERE coupling:

```
# WRF-CHIMERE
ifdef EM_CORE=1

# Number of bins
dimspec nb 4 namelist=n_bin  z number_bin

endif
```

This entry defines a new array dimension, which is the number of aersol bins, with internal name nb, dimension order 4, the size parameter specified in the namelist as n_bin, coordinate axis z, and the dimension name in the output NetCDF file number_bin.

B.2.2 Registry/Registry.EM_COMMON

B.2.2.1 State entries

#	Type	Sym	Dims	Use	Tlev	Stag	IO	Dname	Descrip
state	real	u	ikjb	dyn_em	2	X	i01rhusdf	"U"	"X WIND COMPONENT"

Table B.1: *An example of a state entry in Registry.EM file.*

A state entry in the Registry.EM file looks as shown in Table B.1 . The state entry elements are:

1. The "state" keyword to open the state entry;
2. Type: The type of the state variable or array (real, double, integer, logical, character, or derived);
3. Sym: The symbolic name of the variable or array;
4. Dims: A string denoting the dimensionality of the array or a hyphen (-);
5. Use: A string denoting association with a solver or 4D scalar array, or a hyphen (-);
6. NumTLev: An integer indicating the number of time levels (for arrays) or hyphen (for variable);
7. Stagger: String indicating staggered dimensions of variable (X, Y, Z, or hyphen);
8. IO: String that specifies if the variable is to be subject to initial, restart, history, or boundary I/O. The string may consist of the following symbols, in any order or combination:
   - 'h' (subject to history I/O);
   - 'i' (initial dataset);
   - 'r' (restart dataset);
   - 'b' (lateral boundary dataset).
9. DName: Metadata name for the variable;
10. Units: Metadata units of the variable;
11. Descrip: Metadata description of the variable.

In the example above, the single entry results in over 100 lines of code automatically added to more than 40 different locations in the WRF model, the real and ideal initialization programs, and in the WRF-Var package. These include: (i) nesting code to interpolate, force, feedback, and smooth u; (ii) addition of u to the input, restart, history, and LBC I/O streams.

B.2.2.2 Rconfig entries.

#	Type	Sym	How set	Nentries	Default
rconfig	integer	spec_bdy_width	namelist,bdy_control	1	1

Table B.2: *An example of rconfig entry in Registry.EM.*

Rconfig entries define namelist parameters. An rconfig entry looks as shown in Table B.2. The rconfig entry elements are:

1. The "rconfig" keyword to open the rconfig entry;
2. Type: the type of the namelist variable (integer, real, logical, string);
3. Sym: the name of the namelist variable or array;
4. How set: indicates how the variable is set: e.g. namelist or derived, and if namelist, which block of the namelist it is set in;
5. Nentries: specifies the dimensionality of the namelist variable or array. If 1 (one) it is a variable and applies to all domains; otherwise specify max_domains (which is an integer parameter defined in module_driver_constants.F);
6. Default: the default value of the variable to be used if none is specified in the namelist; hyphen (-) for no default.

B.2.2.3 Package entries.

#	Package name	Associated rconfig choice	Package state vars	Associated variables
package	wsm3scheme	mp_physics==3	-	moist:qv,qc,qr;state:re_ice,...

Table B.3: *An example of package entry in Registry.EM.*

Package entries describe attributes of a package (e.g. microphysics). An package entry looks as shown in Table B.3. The package entry elements are:

1. The "package" keyword to open the package entry;
2. Package name: the name of the package: e.g., wsm3scheme;
3. Associated rconfig choice: the name of a rconfig variable and the value of that variable that choses this package. In the example above the value mp_physics=3 in the namelist will choose the "WSM3" microphysics scheme;
4. Package state vars: unused, hyphen (-) is specied;
5. Associated variables: the names of 4D scalar arrays (moist, chem, scalar) and the fields within those arrays this package uses, and the state variables (state: u_gc, ...).

B.2.2.4 Changes in Registry.EM related to WRF/CHIMERE coupling:

The history output must be added for a number of variables, by replacing 'r' $\rightarrow$ 'rh' in their state entries. For instance:

```
# TKE
state real tke ikj dyn_em 2 - rh "tke" "TURBULENCE KINETIC ENERGY" "m2 s-2"
```

The state entries for the following variables should be modified:

- tke: Turbulence kinetic energy
- alt: Inverse density
- re_cloud: Effective radius cloud water

- re_ice: Effective radius cloud ice
- EXCH_H: Scalar exchange coefficients
- EXCH_M: Exchange coefficients
- h_diabatic: Microphysics latent heating
- qv_diabatic: Microphysics qv tendency
- qc_diabatic: Microphysics qc tendency
- GD_CLOUD: Cloud water/ice mixing raio in GD cloud
- GD_CLOUD2: Test for GD cloud
- QC_CU: Cloud water mixing ratio from a CU scheme
- QI_CU: Cloud ice mixung ratio from a CU scheme
- RTHFTEN: Total advective potential temperature tendency
- RQVFTEN: Total advective moisture tendency
- RTHRATENLW: Uncoupled theta tendency due to long wave radiation
- RTHRATSW: Uncoupled theta tendency due to short wave radiation
- xkhv Vertical eddy diffusivity of heat

The following new state variables have been added for the aerosol/radiation coupling:

```
#Aerosol optical properties for WRF-CHIMERE
state real aod_300 ikj misc 1 - - "AOD_300" "Aerosol optical depth from CHIMERE 300 nm" "?"
state real aod_400 ikj misc 1 - h "AOD_400" "Aerosol optical depth from CHIMERE 400 nm" "?"
state real aod_999 ikj misc 1 - - "AOD_999" "Aerosol optical depth from CHIMERE 999 nm" "?"
state real ssa_400 ikj misc 1 - h "SSA_400" "Aerosol single-scattering albedo from CHIMERE 400 nm" "?"
state real ssa_600 ikj misc 1 - - "SSA_600" "Aerosol single-scattering albedo from CHIMERE 600 nm" "?"
state real g_400 ikj misc 1 - h "G_400" "Aerosol assymetry parameter from CHIMERE 400 nm" "?"
state real g_600 ikj misc 1 - - "G_600" "Aerosol assymetry parameter from CHIMERE 600 nm" "?"
state real aod_lw1 ikj misc 1 - - "AOD_LW1" "Aerosol optical depth from CHIMERE LW1" "?"
state real aod_lw2 ikj misc 1 - - "AOD_LW2" "Aerosol optical depth from CHIMERE LW2" "?"
state real aod_lw3 ikj misc 1 - - "AOD_LW3" "Aerosol optical depth from CHIMERE LW3" "?"
state real aod_lw4 ikj misc 1 - - "AOD_LW4" "Aerosol optical depth from CHIMERE LW4" "?"
state real aod_lw5 ikj misc 1 - - "AOD_LW5" "Aerosol optical depth from CHIMERE LW5" "?"
state real aod_lw6 ikj misc 1 - - "AOD_LW6" "Aerosol optical depth from CHIMERE LW6" "?"
state real aod_lw7 ikj misc 1 - - "AOD_LW7" "Aerosol optical depth from CHIMERE LW7" "?"
state real aod_lw8 ikj misc 1 - - "AOD_LW8" "Aerosol optical depth from CHIMERE LW8" "?"
state real aod_lw9 ikj misc 1 - - "AOD_LW9" "Aerosol optical depth from CHIMERE LW9" "?"
state real aod_lw10 ikj misc 1 - - "AOD_LW10" "Aerosol optical depth from CHIMERE LW10" "?"
state real aod_lw11 ikj misc 1 - - "AOD_LW11" "Aerosol optical depth from CHIMERE LW11" "?"
state real aod_lw12 ikj misc 1 - - "AOD_LW12" "Aerosol optical depth from CHIMERE LW12" "?"
state real aod_lw13 ikj misc 1 - - "AOD_LW13" "Aerosol optical depth from CHIMERE LW13" "?"
state real aod_lw14 ikj misc 1 - - "AOD_LW14" "Aerosol optical depth from CHIMERE LW14" "?"
state real aod_lw15 ikj misc 1 - - "AOD_LW15" "Aerosol optical depth from CHIMERE LW15" "?"
state real aod_lw16 ikj misc 1 - - "AOD_LW16" "Aerosol optical depth from CHIMERE LW16" "?"
```

The following new state variables have been added for the indirect aerosol effects:

```
# Variables for WRF-CHIMERE (indirect effects)
state real dropsource ikj misc 1 - rh "dropsource" "Cloud droplet number source" "#/kg/s"
state real chm_size_dist ikj{nb} misc 1 - r "CHM_SIZE_DIST" "CHIMERE size ditributition" "#/kg"
state real chm_hygro_size_dist ikj{nb} misc 1 - r "CHM_HYGRO_SIZE_DIST" "CHIMERE
hygroscopic aerosol size distribution" "#/kg"
state real chm_ice_nucl ikj misc 1 - rh "CHM_ICE_NUCL" "CHIMERE ice nuclei" "#/kg"
state real hygro_aer ikj misc 1 - rh "HYGRO_AER" "Aerosol hygroscopicity"

state real num01 ikj misc 1 - h "NUM01" "num01"
state real num02 ikj misc 1 - h "NUM02" "num02"
state real num10 ikj misc 1 - h "NUM10" "num10"

state real ccn1 ikj misc 1 - rh "ccn1" "CCN concentration at S=0.02%" "#/cm3"
state real ccn2 ikj misc 1 - rh "ccn2" "CCN concentration at S=0.05%" "#/cm3"
state real ccn3 ikj misc 1 - rh "ccn3" "CCN concentration at S=0.1%" "#/cm3"
state real ccn4 ikj misc 1 - rh "ccn4" "CCN concentration at S=0.2%" "#/cm3"
state real ccn5 ikj misc 1 - rh "ccn5" "CCN concentration at S=0.5%" "#/cm3"
state real ccn6 ikj misc 1 - rh "ccn6" "CCN concentration at S=1.0%" "#/cm3"

state real rainrate ikj misc 1 - h "RAINRATE" "TOTAL RAIN PRODUCTION RATE" "s-1"
state real evaprate ikj misc 1 - h "EVAPRATE" "RAIN EVAPORATION RATE" "s-1"
```

The following variables have been excluded from the history output (replaced 'rhdu' → 'rdu' or 'rhd' → 'rd' in their state entries) :

- ACSWUPT: Accumulated upwelling shortwave flux at top

- ACSWUPTC: Accumulated upwelling clear sky shortwave flux at top
- ACSWDNT: Accumulated downwelling shortwave flux at top
- ACSWDNTC: Accumulated downwelling clear sky shortwave flux at top
- ACSWUPB: Accumulated upwelling shortwave flux at bottom
- ACSWUPBC: Accumulated upwelling clear sky shortwave flux at bottom
- ACSWDNB: Accumulated downwelling shortwave flux at bottom
- ACSWDNBC: Accumulated downwelling clear sky shortwave flux at bottom
- ACLWUPT: Accumulated upwelling longwave flux at top
- ACLWUPTC: Accumulated upwelling clear sky longwave flux at top
- ACLWDNT: Accumulated downwelling longwave flux at top
- ACLWDNTC: Accumulated downwelling clear sky longwave flux at top
- ACLWUPB: Accumulated upwelling longwave flux at bottom
- ACLWUPBC: Accumulated upwelling clear sky longwave flux at bottom
- ACLWDNB: Accumulated downwelling longwave flux at bottom
- ACLWDNBC: Accumulated downwelling clear sky longwave flux at bottom
- I_ACSWUPT: Bucket for upwelling shortwave flux at top
- I_ACSWUPTC: Bucket for upwelling clear sky shortwave flux at top
- I_ACSWDNT: Bucket for downwelling shortwave flux at top
- I_ACSWDNTC: Bucket for downwelling clear sky shortwave flux at top
- I_ACSWUPB: Bucket for upwelling shortwave flux at bottom
- I_ACSWUPBC: Bucket for upwelling clear sky shortwave flux at bottom
- I_ACSWDNB: Bucket for downwelling shortwave flux at bottom
- I_ACSWDNBC: Bucket for downwelling clear sky shortwave flux at bottom
- I_ACLWUPT: Bucket for upwelling longwave flux at top
- I_ACLWUPTC: Bucket for upwelling clear sky longwave flux at top
- I_ACLWDNT: Bucket for downwelling longwave flux at top
- I_ACLWDNTC: Bucket for downwelling clear sky longwave flux at top
- I_ACLWUPB: Bucket for upwelling longwave flux at bottom
- I_ACLWUPBC: Bucket for upwelling clear sky longwave flux at bottom
- I_ACLWDNB: Bucket for downwelling longwave flux at bottom
- I_ACLWDNBC: Bucket for downwelling clear sky longwave flux at bottom

The following namelist variables have to be added (rconfig entries):

```
rconfig real chm_sections namelist,domains max_eta -1.
.......
#Option for aerosol direct effects in WRF-CHIMERE
rconfig integer direct_feed_chimere namelist,physics 1 0 - "direct_feed_chimere" "
if aersol direct feedback with CHIMERE" ""
#Option for aerosol indirect effects in WRF-CHIMERE
rconfig integer indirect_feed_chimere namelist,physics 1 0 - "indirect_feed_chimere" "
if aersol direct feedback with CHIMERE" ""
rconfig integer n_bin namelist,physics 1 0 - "n_bin" "number of bin of CHIMERE size distribution"
```

The following physical scheme options have to be defined (package entries):

```
# package for aerosol-cloud interaction with WRF-CHIMERE
package noprognchm indirect_feed_chimere==0 - -
package prognchm indirect_feed_chimere==1 - state:dropsource,chm_size_dist,
  chm_hygro_size_dist,hygro_aer,num01,
  num02,num10,chm_ice_nucl,ccn1,ccn2,ccn3,ccn4,ccn5,ccn6
```

B.3 Solver part

B.3.1 module_first_rk_step_part1.F

The first_rk_step_part1 routine calls the radiation_driver with its new arguments specific to the WRF/CHIMERE coupling. Right after the #if (WRF_CHEM == 1) entry add the following code to specify the aerosol/radiation coupling parameters and fields:

```
#ifdef cpl_wrf_chimere
 &   ,direct_feed_chimere=config_flags%direct_feed_chimere              &
 &   ,aod_300=grid%aod_300,aod_400=grid%aod_400,aod_999=grid%aod_999    &
 &   ,ssa_400=grid%ssa_400,ssa_600=grid%ssa_600                         &
 &   ,g_400=grid%g_400,g_600=grid%g_600                                 &
 &   ,aod_lw1=grid%aod_lw1,   aod_lw2=grid%aod_lw2,   aod_lw3=grid%aod_lw3   &
     ,aod_lw4=grid%aod_lw4,   aod_lw5=grid%aod_lw5,   aod_lw6=grid%aod_lw6   &
     ,aod_lw7=grid%aod_lw7,   aod_lw8=grid%aod_lw8,   aod_lw9=grid%aod_lw9   &
     ,aod_lw10=grid%aod_lw10, aod_lw11=grid%aod_lw11, aod_lw12=grid%aod_lw12 &
     ,aod_lw13=grid%aod_lw13, aod_lw14=grid%aod_lw14, aod_lw15=grid%aod_lw15 &
     ,aod_lw16=grid%aod_lw16                                            &
#endif
```

B.3.2 solve_em.F

The following changes are needed in the solve_em subroutine:

1. Two new subroutines and two flags are additionally used from the module_cpl module: cpl_rcv2, cpl_snd, wrf_chimere, wrf_nemo, e.g.:

```
USE module_cpl, ONLY : coupler_on, cpl_settime, cpl_rcv2,  &
                       cpl_store_input, wrf_nemo, wrf_chimere, &
                       cpl_snd
```

2. The NEMO-related cpl_store_input subroutine is called only when the wrf_nemo flag is True:

```
IF( coupler_on .and. wrf_nemo ) CALL cpl_store_input( grid, config_flags )
```

3. Right before calling the first_rk_step_part1 subroutine, the coupling fields are received from CHIMERE:

```
!! GET CHIMERE DATA IF DEFINED
IF (coupler_on.AND.wrf_chimere) THEN
   CALL cpl_snd( grid )
  CALL  cpl_rcv2(grid,grid%id,config_flags      &
                &   ,ids,ide,jds,jde,kds,kde    &
                &   ,ims,ime,jms,jme,kms,kme    &
                &   ,ips,ipe,jps,jpe,kps,kpe )
ENDIF
```

4. New parameters and fields are passed as arguments to the microphysics_driver to take into account the indirect effects. Right after the ccn_conc argument, add the following code:

```
#ifdef cpl_wrf_chimere
  &   ,indirect_feed_chimere=config_flags%indirect_feed_chimere &
  &   ,n_bin=config_flags%n_bin                                 &
  &   ,max_sect=max_eta                                         &
  &   ,sect=model_config_rec%chm_sections                       &
  &   ,dropsource=grid%dropsource                               &
  &   ,num_aer=grid%chm_size_dist                               &
  &   ,num_hygro_aer=grid%chm_hygro_size_dist                   &
  &   ,hygro_aer=grid%hygro_aer                                 &
  &   ,in_aer=grid%chm_ice_nucl                                 &
  &   ,num01=grid%num01                                         &
  &   ,num02=grid%num02                                         &
  &   ,num10=grid%num10                                         &
  &   ,ccn1=grid%ccn1                                           &
  &   ,ccn2=grid%ccn2                                           &
  &   ,ccn3=grid%ccn3                                           &
  &   ,ccn4=grid%ccn4                                           &
  &   ,ccn5=grid%ccn5                                           &
  &   ,ccn6=grid%ccn6                                           &
  &   ,rainrate=grid%rainrate                                   &
  &   ,evaprate=grid%evaprate                                   &
#endif
```

5. The following function is called only in case of WRF/NEMO coupling, so additional condition ".and. wrf_nemo" is needed in the "IF" statement:

```
IF (coupler_on .and. wrf_nemo) &
grid%just_read_auxinput4 = Is_alarm_tstep(grid%domain_clock, grid%alarms(AUXINPUT4_ALARM))
```

B.4 Coupling modules

B.4.1 module_cpl.F

The frame/module_cpl.F module was originally developed for the coupling between WRF and NEMO models. The following changes need to be made for the coupling with CHIMERE.

1. Declare the new subroutine cpl_rcv2. After PUBLIC cpl_rcv, insert PUBLIC cpl_rcv2 to have:

```
PUBLIC cpl_rcv
PUBLIC cpl_rcv2
```

2. Initialize the flags wrf_nemo and wrf_chimere depending on the conditional compilation flag cpl_wrf_chimere. After the preprocessor condition #ifdef key_cpp_oasis3 add another one that defines the wrf_chimere and wrf_nemo flags:

```
#ifdef cpl_wrf_nemo
   LOGICAL     , PARAMETER, PUBLIC :: wrf_nemo = .TRUE.
#else
   LOGICAL     , PARAMETER, PUBLIC :: wrf_nemo = .FALSE.
#endif
#ifdef cpl_wrf_chimere
   LOGICAL     , PARAMETER, PUBLIC :: wrf_chimere = .TRUE.
#else
   LOGICAL     , PARAMETER, PUBLIC :: wrf_chimere = .FALSE.
#endif
```

3. Declare the structure containing parameters defined in the namcouple file. Right before the "CONTAINS" statement:

```
! Structure containing all necesary namcouple info.
TYPE(NAMCOUPLE_TYPE)  :: namcouple_info
CHARACTER(len=256)    :: namcouple_file   ! Path to the namcouple file
INTEGER               :: crank, ierr, ifield
```

4. Initialization of the coupling fields related to NEMO in the cpl_init subroutine must be done only if wrf_nemo is True. Add the "IF (wrf_nemo) / END IF" pair around the following code to obtain:

```
IF (wrf_nemo) THEN
   ! we could imagine to define rcvname and sndname through the namelist...
   ! define all possible coupling names with _d<domain> of WRF and the external model(s)
   DO jext = 1, max_extdomains
   ..................

         sndname(jwrf,jext,6) = clprefix//'TAUMOD'
         ! send the wind tress module at atmosphere-ocean interface
      END DO
   END DO
END IF
```

5. At the end of the cpl_init subroutine, right after the call to cpl_oasis_init, read the namcouple parameters and initialize the fields to exchange with CHIMERE:

```
IF (wrf_chimere) THEN
   CALL mpi_comm_rank(kl_comm,crank,ierr)

   IF (crank == 0) THEN
      ! Read namcouple file
      CALL cpl_oasis_read_namcouple(namcouple_info)
   END IF

   ! Allocate and broadcast parameters to other processors.
   CALL mpi_bcast(namcouple_info%nfields,1,MPI_INTEGER,0,kl_comm,ierr)

   IF (.NOT. ALLOCATED(namcouple_info%source_grid)) &
   & ALLOCATE(namcouple_info%source_grid(namcouple_info%nfields))
   CALL mpi_bcast(namcouple_info%source_grid,namcouple_info%nfields*4, &
   &              MPI_CHARACTER,0,kl_comm,ierr)

   IF (.NOT. ALLOCATED(namcouple_info%target_grid)) &
   & ALLOCATE(namcouple_info%target_grid(namcouple_info%nfields))
```

```
   CALL mpi_bcast(namcouple_info%target_grid,namcouple_info%nfields*4, &
   &              MPI_CHARACTER,0,kl_comm,ierr)

   IF (.NOT. ALLOCATED(namcouple_info%send_variables)) &
   & ALLOCATE(namcouple_info%send_variables(namcouple_info%nfields))
   CALL mpi_bcast(namcouple_info%send_variables,namcouple_info%nfields*128, &
   &              MPI_CHARACTER,0,kl_comm,ierr)

   IF (.NOT. ALLOCATED(namcouple_info%recv_variables)) &
   & ALLOCATE(namcouple_info%recv_variables(namcouple_info%nfields))
   CALL mpi_bcast(namcouple_info%recv_variables,namcouple_info%nfields*128, &
   &             MPI_CHARACTER,0,kl_comm,ierr)

   IF (.NOT. ALLOCATED(namcouple_info%coupling_freq)) &
   & ALLOCATE(namcouple_info%coupling_freq(namcouple_info%nfields))
   CALL mpi_bcast(namcouple_info%coupling_freq,namcouple_info%nfields, &
   &              MPI_INTEGER,0,kl_comm,ierr)

   ! Define coupling names with _d<domain> of WRF and the external model(s)
   DO ifield=1,namcouple_info%nfields
      ! Variables that are sent by WRF.
      IF (namcouple_info%source_grid(ifield)(1:3) == 'wrf') THEN
         sndname(1,1,ifield) = TRIM(namcouple_info%send_variables(ifield))
      END IF

      ! Variables that are received by WRF.
      IF (namcouple_info%target_grid(ifield)(1:3) == 'wrf') THEN
         rcvname(1,1,ifield) = TRIM(namcouple_info%recv_variables(ifield))
      END IF
   END DO
END IF
```

6. In the subroutine cpl_set_dm_communicator, add an additional argument namcouple_info to the call of the cpl_oasis_define routine that needs the coupling parameters read above.

```
CALL cpl_oasis_define( sndname, rcvname, namcouple_info ) ! define io_quilting to OASIS
```

7. Subroutine cpl_defdomain gets an additional argument nbin.

```
SUBROUTINE cpl_defdomain( grid, nbin)
.............
INTEGER, INTENT(IN), OPTIONAL :: nbin
```

8. Instead of the last line of cpl_defdomain put the following #ifdef block, to pass the namcouple_info and nbin arguments to cpl_oasis_define in the case of coupling with CHIMERE:

```
#ifdef cpl_wrf_chimere
 IF ( coupler_name == 'oasis' ) CALL cpl_oasis_define( sndname, rcvname, namcouple_info, grid, nbin)
#else
 IF ( coupler_name == 'oasis' ) CALL cpl_oasis_define( sndname, rcvname, namcouple_info, grid )
#endif
```

9. The function cpl_get_fldid takes the suffix of a coupling variable name and returns the field ID. It needs to be adapted to the WRF/CHIMERE coupling name convention. The loop DO jfld = 1, max_cplfld in function cpl_get_fldid becomes:

```
DO jfld = 1, max_cplfld
   IF(wrf_nemo) THEN
      IF( clprefix//TRIM(cdsuffix) == TRIM(sndname(1,1,jfld)) )   cpl_get_fldid = jfld
      IF( clprefix//TRIM(cdsuffix) == TRIM(rcvname(1,1,jfld)) )   cpl_get_fldid = jfld
   ELSE IF(wrf_chimere) THEN
      IF( TRIM(cdsuffix) == TRIM(sndname(1,1,jfld)) )   cpl_get_fldid = jfld
      IF( TRIM(cdsuffix) == TRIM(rcvname(1,1,jfld)) )   cpl_get_fldid = jfld
   END IF
END DO
```

10. And now a crucial part: receiving fields from CHIMERE. You need to change this if you wished to add/remove coupling fields. The new subroutine cpl_rcv2 receives the required fields from CHIMERE, on the grid and partition defined for coupling:

```
SUBROUTINE cpl_rcv2(grid,id,config_flags           &
   &                    ,ids,ide,jds,jde,kds,kde   &
   &                    ,ims,ime,jms,jme,kms,kme   &
   &                    ,ips,ipe,jps,jpe,kps,kpe )
  !!---------------------------------------------------------------------
  !!             ***  SUBROUTINE cpl_rcv2  ***
  !!
  !! ** Purpose : receiving data from CHIMERE
  !!---------------------------------------------------------------------
  TYPE(domain), INTENT(IN),    POINTER    :: grid
  TYPE (grid_config_rec_type), INTENT(IN) :: config_flags
  INTEGER,                     INTENT(IN) :: id ! wrf domain index
  INTEGER,                     INTENT(IN) :: ids,ide,jds,jde,kds,kde
  INTEGER,                     INTENT(IN) :: ims,ime,jms,jme,kms,kme
  INTEGER,                     INTENT(IN) :: ips,ipe,jps,jpe,kps,kpe
  ! Local
  INTEGER :: jf
  INTEGER :: ine,jne,kne

  IF ( config_flags%direct_feed_chimere == 0 .AND. &
       config_flags%indirect_feed_chimere == 0) return

  ine = ipe
  if (ine.ge.ide) ine = ine-1
  jne = jpe
  if (jne.ge.jde) jne = jne-1
  kne = kpe-1

  DO jf = 1,namcouple_info%nfields
     ! Cycle if it is not time to receive variables.
     IF (mod(nsecrun,namcouple_info%coupling_freq(jf)) /= 0) cycle

     IF( config_flags%direct_feed_chimere == 1) THEN
        IF( namcouple_info%recv_variables(jf) == 'RCV_AOD_300nm' ) THEN
               CALL cpl_rcv( id,  namcouple_info%recv_variables(jf),  &
                   &              ids, ide, jds, jde, kds, kde,       &
                   &              ims, ime, jms, jme, kms, kme,       &
                   &              ips, ine, jps, jne, kps, kne,       &
                   &                1, grid%cplmask, grid%aod_300 )
        ELSEIF( namcouple_info%recv_variables(jf) == 'RCV_AOD_400nm' ) THEN
                CALL cpl_rcv( id,  namcouple_info%recv_variables(jf),  &
                    &              ids, ide, jds, jde, kds, kde,       &
                    &              ims, ime, jms, jme, kms, kme,       &
                    &              ips, ine, jps, jne, kps, kne,       &
                    &                1, grid%cplmask, grid%aod_400 )
        ELSEIF( namcouple_info%recv_variables(jf) == 'RCV_AOD_999nm' ) THEN
                CALL cpl_rcv( id, namcouple_info%recv_variables(jf),   &
                    &              ids, ide, jds, jde, kds, kde,       &
                    &              ims, ime, jms, jme, kms, kme,       &
                    &              ips, ine, jps, jne, kps, kne,       &
                    &                1, grid%cplmask, grid%aod_999)
        ELSEIF( namcouple_info%recv_variables(jf) == 'RCV_albedo_400nm' ) THEN
                CALL cpl_rcv( id, namcouple_info%recv_variables(jf),   &
                   &             ids, ide, jds, jde, kds, kde,        &
                   &             ims, ime, jms, jme, kms, kme,        &
                   &             ips, ine, jps, jne, kps, kne,        &
                   &               1, grid%cplmask, grid%ssa_400 )
        ELSEIF( namcouple_info%recv_variables(jf) == 'RCV_albedo_600nm' ) THEN
                CALL cpl_rcv( id, namcouple_info%recv_variables(jf),   &
                    &              ids, ide, jds, jde, kds, kde,       &
                    &              ims, ime, jms, jme, kms, kme,       &
                    &              ips, ine, jps, jne, kps, kne,       &
                    &                1, grid%cplmask, grid%ssa_600 )
        ELSEIF( namcouple_info%recv_variables(jf) == 'RCV_asymmetry_factor_400nm' ) THEN
                CALL cpl_rcv( id, namcouple_info%recv_variables(jf),   &
                    &              ids, ide, jds, jde, kds, kde,       &
                    &              ims, ime, jms, jme, kms, kme,       &
                    &              ips, ine, jps, jne, kps, kne,       &
                    &                1, grid%cplmask, grid%g_400 )
        ELSEIF( namcouple_info%recv_variables(jf) == 'RCV_asymmetry_factor_600nm' ) THEN
                CALL cpl_rcv( id, namcouple_info%recv_variables(jf),   &
                    &              ids, ide, jds, jde, kds, kde,       &
                    &              ims, ime, jms, jme, kms, kme,       &
                    &              ips, ine, jps, jne, kps, kne,       &
                    &                1, grid%cplmask, grid%g_600 )
        ELSEIF( namcouple_info%recv_variables(jf) == 'RCV_AOD_LW1' ) THEN
                CALL cpl_rcv( id, namcouple_info%recv_variables(jf),   &
                    &              ids, ide, jds, jde, kds, kde,       &
                    &              ims, ime, jms, jme, kms, kme,       &
                    &              ips, ine, jps, jne, kps, kne,       &
                    &                1, grid%cplmask, grid%aod_lw1 )
```

```
        ELSEIF( namcouple_info%recv_variables(jf) == 'RCV_AOD_LW2' ) THEN
                CALL cpl_rcv( id, namcouple_info%recv_variables(jf),   &
                    &               ids, ide, jds, jde, kds, kde,      &
                    &               ims, ime, jms, jme, kms, kme,      &
                    &               ips, ine, jps, jne, kps, kne,      &
                    &                 1, grid%cplmask, grid%aod_lw2 )
        ELSEIF( namcouple_info%recv_variables(jf) == 'RCV_AOD_LW3' ) THEN
                CALL cpl_rcv( id, namcouple_info%recv_variables(jf),   &
                    &               ids, ide, jds, jde, kds, kde,      &
                    &               ims, ime, jms, jme, kms, kme,      &
                    &               ips, ine, jps, jne, kps, kne,      &
                    &                 1, grid%cplmask, grid%aod_lw3 )
        ELSEIF( namcouple_info%recv_variables(jf) == 'RCV_AOD_LW4' ) THEN
                CALL cpl_rcv( id, namcouple_info%recv_variables(jf),   &
                    &               ids, ide, jds, jde, kds, kde,      &
                    &               ims, ime, jms, jme, kms, kme,      &
                    &               ips, ine, jps, jne, kps, kne,      &
                    &                 1, grid%cplmask, grid%aod_lw4 )
        ELSEIF( namcouple_info%recv_variables(jf) == 'RCV_AOD_LW5' ) THEN
                CALL cpl_rcv( id, namcouple_info%recv_variables(jf),   &
                    &               ids, ide, jds, jde, kds, kde,      &
                    &               ims, ime, jms, jme, kms, kme,      &
                    &               ips, ine, jps, jne, kps, kne,      &
                    &                 1, grid%cplmask, grid%aod_lw5 )
        ELSEIF( namcouple_info%recv_variables(jf) == 'RCV_AOD_LW6' ) THEN
                CALL cpl_rcv( id, namcouple_info%recv_variables(jf),   &
                    &               ids, ide, jds, jde, kds, kde,      &
                    &               ims, ime, jms, jme, kms, kme,      &
                    &               ips, ine, jps, jne, kps, kne,      &
                    &                 1, grid%cplmask, grid%aod_lw6 )
        ELSEIF( namcouple_info%recv_variables(jf) == 'RCV_AOD_LW7' ) THEN
                CALL cpl_rcv( id, namcouple_info%recv_variables(jf),   &
                    &               ids, ide, jds, jde, kds, kde,      &
                    &               ims, ime, jms, jme, kms, kme,      &
                    &               ips, ine, jps, jne, kps, kne,      &
                    &                 1, grid%cplmask, grid%aod_lw7 )
        ELSEIF( namcouple_info%recv_variables(jf) == 'RCV_AOD_LW8' ) THEN
                CALL cpl_rcv( id, namcouple_info%recv_variables(jf),   &
                    &               ids, ide, jds, jde, kds, kde,      &
                    &               ims, ime, jms, jme, kms, kme,      &
                    &               ips, ine, jps, jne, kps, kne,      &
                    &                 1, grid%cplmask, grid%aod_lw8 )
        ELSEIF( namcouple_info%recv_variables(jf) == 'RCV_AOD_LW9' ) THEN
                CALL cpl_rcv( id, namcouple_info%recv_variables(jf),   &
                    &               ids, ide, jds, jde, kds, kde,      &
                    &               ims, ime, jms, jme, kms, kme,      &
                    &               ips, ine, jps, jne, kps, kne,      &
                    &                 1, grid%cplmask, grid%aod_lw9 )
        ELSEIF( namcouple_info%recv_variables(jf) == 'RCV_AOD_LW10' ) THEN
                CALL cpl_rcv( id, namcouple_info%recv_variables(jf),   &
                    &               ids, ide, jds, jde, kds, kde,      &
                    &               ims, ime, jms, jme, kms, kme,      &
                    &               ips, ine, jps, jne, kps, kne,      &
                    &                 1, grid%cplmask, grid%aod_lw10 )
        ELSEIF( namcouple_info%recv_variables(jf) == 'RCV_AOD_LW11' )THEN
                CALL cpl_rcv( id, namcouple_info%recv_variables(jf),   &
                    &               ids, ide, jds, jde, kds, kde,      &
                    &               ims, ime, jms, jme, kms, kme,      &
                    &               ips, ine, jps, jne, kps, kne,      &
                    &                 1, grid%cplmask, grid%aod_lw11 )
        ELSEIF( namcouple_info%recv_variables(jf) == 'RCV_AOD_LW12' )THEN
                CALL cpl_rcv( id, namcouple_info%recv_variables(jf),   &
                    &               ids, ide, jds, jde, kds, kde,      &
                    &               ims, ime, jms, jme, kms, kme,      &
                    &               ips, ine, jps, jne, kps, kne,      &
                    &                 1, grid%cplmask, grid%aod_lw12 )
        ELSEIF( namcouple_info%recv_variables(jf) == 'RCV_AOD_LW13' )THEN
                CALL cpl_rcv( id, namcouple_info%recv_variables(jf),   &
                    &               ids, ide, jds, jde, kds, kde,      &
                    &               ims, ime, jms, jme, kms, kme,      &
                    &               ips, ine, jps, jne, kps, kne,      &
                    &                 1, grid%cplmask, grid%aod_lw13 )
        ELSEIF( namcouple_info%recv_variables(jf) == 'RCV_AOD_LW14' )THEN
                CALL cpl_rcv( id, namcouple_info%recv_variables(jf),   &
                    &               ids, ide, jds, jde, kds, kde,      &
                    &               ims, ime, jms, jme, kms, kme,      &
                    &               ips, ine, jps, jne, kps, kne,      &
                    &                 1, grid%cplmask, grid%aod_lw14 )
        ELSEIF( namcouple_info%recv_variables(jf) == 'RCV_AOD_LW15' )THEN
                CALL cpl_rcv( id, namcouple_info%recv_variables(jf),   &
                    &               ids, ide, jds, jde, kds, kde,      &
```

```
                    &              ims, ime, jms, jme, kms, kme,       &
                    &              ips, ine, jps, jne, kps, kne,       &
                    &                1, grid%cplmask, grid%aod_lw15 )
      ELSEIF( namcouple_info%recv_variables(jf) == 'RCV_AOD_LW16' ) THEN
              CALL cpl_rcv( id, namcouple_info%recv_variables(jf),   &
                    &              ids, ide, jds, jde, kds, kde,       &
                    &              ims, ime, jms, jme, kms, kme,       &
                    &              ips, ine, jps, jne, kps, kne,       &
                    &                1, grid%cplmask, grid%aod_lw16 )
      ENDIF
    ENDIF

    IF( config_flags%indirect_feed_chimere == 1) THEN
      IF    ( namcouple_info%recv_variables(jf) == 'RCV_NUM_AER' ) THEN
              CALL cpl_rcv_4D( id,namcouple_info%recv_variables(jf),&
                 &             config_flags%n_bin,                  &
                 &             ids, ide, jds, jde, kds, kde,        &
                 &             ims, ime, jms, jme, kms, kme,        &
                 &             ips, ine, jps, jne, kps, kne,        &
                 &               1, grid%chm_size_dist             )
      ELSEIF( namcouple_info%recv_variables(jf) == 'RCV_NUM_HYGRO_AER' ) THEN
              CALL cpl_rcv_4D( id,namcouple_info%recv_variables(jf),&
                 &             config_flags%n_bin,                  &
                 &             ids, ide, jds, jde, kds, kde,        &
                 &             ims, ime, jms, jme, kms, kme,        &
                 &             ips, ine, jps, jne, kps, kne,        &
                 &               1, grid%chm_hygro_size_dist          )
      ELSEIF( namcouple_info%recv_variables(jf) == 'RCV_ICE_NUCL' )  THEN
              CALL cpl_rcv( id,  namcouple_info%recv_variables(jf),  &
                 &              ids, ide, jds, jde, kds, kde,        &
                 &              ims, ime, jms, jme, kms, kme,        &
                 &              ips, ine, jps, jne, kps, kne,        &
                 &                1, grid%cplmask, grid%chm_ice_nucl  )
      ELSEIF( namcouple_info%recv_variables(jf) == 'RCV_HYGRO_AER' ) THEN
              CALL cpl_rcv( id,  namcouple_info%recv_variables(jf),  &
                 &              ids, ide, jds, jde, kds, kde,        &
                 &              ims, ime, jms, jme, kms, kme,        &
                 &              ips, ine, jps, jne, kps, kne,        &
                 &                1, grid%cplmask, grid%hygro_aer  )
      ENDIF
    ENDIF

  ENDDO

END SUBROUTINE cpl_rcv2
```

The subroutines cpl_snd2, cpl_sndfield, cpl_rcv, cpl_oasis_snd, and cpl_oasis_rcv are adapted to the CHIMERE coupling grid (1D).

11. The subroutine cpl_snd2 also includes variables specific to coupling with CHIMERE:

    (a) Add new USE statement:

    ```
    #ifdef cpl_wrf_chimere
          USE module_state_description , ONLY : p_qv,p_qc,p_qi,p_qr,p_qs
    #endif
    ```

    (b) The variable cplsnd becomes 1D:

    ```
    #ifdef cpl_wrf_chimere
          INTEGER :: Npts
          INTEGER :: i,j,k,l,jf
          INTEGER :: ins,ine,jns,jne,kns,kne
          REAL, DIMENSION( : ), ALLOCATABLE :: cplsnd
    #else
          REAL, DIMENSION( ips:ipe, jps:jpe ) :: cplsnd
    #endif
    ```

    (c) The field preparation and indexing has been rewritten for the coupling with CHIMERE. Right after the #if (EM_CORE == 1) directive, insert an additonal condition to separate the NEMO-related code:

    ```
    #ifdef cpl_wrf_nemo
    ```

    (d) Another crucial part (just in case you were slumbering): sending WRF fields to CHIMERE! You will need to add/remove entries in this code if you wanted to add or remove coupling variables. Right before the #endif corresponding to the #if (EM_CORE == 1) directive add the following code:

```
#endif

#ifdef cpl_wrf_chimere
   DO jf = 1, namcouple_info%nfields
      IF (mod(nsecrun,namcouple_info%coupling_freq(jf)) /= 0) cycle
      IF (namcouple_info%source_grid(jf)(1:3) /= "wrf") cycle
      ifldid      = cpl_get_fldid( namcouple_info%send_variables(jf) )
      lltosend(:) = cpl_tosend( grid%id, ifldid, max_edom )
      IF ( COUNT(lltosend) > 0 ) THEN

         ins = ips
         jns = jps
         kns = kps
         ine = ipe
         jne = jpe
         l = 1

         ! Allocate cplsnd and compute indexes.
         IF (namcouple_info%source_grid(jf) == 'wrf1') THEN
            if (ine.ge.ide) ine = ine-1
            if (jne.ge.jde) jne = jne-1
            kne = kpe-1
            Npts = (ine-ins+1)*(jne-jns+1)
         ELSE IF (namcouple_info%source_grid(jf) == 'wrf2') THEN
            if (ine.ge.ide) ine = ine-1
            if (jne.ge.jde) jne = jne-1
            kne = kpe-1
            Npts = (ine-ins+1)*(jne-jns+1)*(kne-kns+1)
         ELSE IF (namcouple_info%source_grid(jf) == 'wrf3') THEN
            if (ine.ge.ide) ine = ine-1
            if (jne.ge.jde) jne = jne-1
            kne = kpe
            Npts = (ine-ins+1)*(jne-jns+1)*(kne-kns+1)
         ELSE IF (namcouple_info%source_grid(jf) == 'wrf4') THEN
            if (jne.ge.jde) jne = jne-1
            kne = kpe-1
            Npts = (ine-ins+1)*(jne-jns+1)*(kne-kns+1)
         ELSE IF (namcouple_info%source_grid(jf) == 'wrf5') THEN
            if (ine.ge.ide) ine = ine-1
            kne = kpe-1
            Npts = (ine-ins+1)*(jne-jns+1)*(kne-kns+1)
         END IF
         ALLOCATE(cplsnd(Npts))

         ! Reshape of sending variable.
         IF (namcouple_info%source_grid(jf) == 'wrf1') THEN
            do j = jns,jne
               do i = ins,ine
                  ! Surface Pressure
                  IF( namcouple_info%send_variables(jf) == 'SND_PSFC' ) THEN
                     cplsnd(l) = grid%PSFC(i,j)

                     ! Surface SW Radiation
                  ELSE IF( namcouple_info%send_variables(jf) == 'SND_SWDOWN' ) THEN
                     cplsnd(l) = grid%SWDOWN(i,j)

                     ! Surface LW Radiation
                  ELSE IF( namcouple_info%send_variables(jf) == 'SND_GLW' ) THEN
                     cplsnd(l) = grid%GLW(i,j)

                     ! 2m Water Vapor
                  ELSE IF( namcouple_info%send_variables(jf) == 'SND_Q2' ) THEN
                     cplsnd(l) = grid%Q2(i,j)

                     ! Convective Rain
                  ELSE IF( namcouple_info%send_variables(jf) == 'SND_RAINC' ) THEN
                     cplsnd(l) = grid%RAINC(i,j)

                     ! Grid Resolved Rain
                  ELSE IF( namcouple_info%send_variables(jf) == 'SND_RAINNC' ) THEN
                     cplsnd(l) = grid%RAINNC(i,j)

                     ! Surface Sensible Heat Flux
                  ELSE IF( namcouple_info%send_variables(jf) == 'SND_HFX' ) THEN
                     cplsnd(l) = grid%HFX(i,j)

                     ! Surface Latent Heat Flux
                  ELSE IF( namcouple_info%send_variables(jf) == 'SND_LH' ) THEN
                     cplsnd(l) = grid%LH(i,j)

                     ! Friction Velocity
```

```
                ELSE IF( namcouple_info%send_variables(jf) == 'SND_UST' ) THEN
                   cplsnd(l) = grid%UST(i,j)

                   ! 2m Temperature
                ELSE IF( namcouple_info%send_variables(jf) == 'SND_T2' ) THEN
                   cplsnd(l) = grid%T2(i,j)

                   ! PBLH
                ELSE IF( namcouple_info%send_variables(jf) == 'SND_PBLH' ) THEN
                   cplsnd(l) = grid%PBLH(i,j)

                   ! Equi. Acc. Snow Depth
                ELSE IF( namcouple_info%send_variables(jf) == 'SND_SNOW' ) THEN
                   cplsnd(l) = grid%SNOW(i,j)

                   ! Snow Depth
                ELSE IF( namcouple_info%send_variables(jf) == 'SND_SNOWH' ) THEN
                   cplsnd(l) = grid%SNOWH(i,j)

                   ! U10
                ELSE IF( namcouple_info%send_variables(jf) == 'SND_U10' ) THEN
                   cplsnd(l) = grid%U10(i,j)

                   ! V10
                ELSE IF( namcouple_info%send_variables(jf) == 'SND_V10' ) THEN
                   cplsnd(l) = grid%V10(i,j)

                   ! Soil Moisture
                ELSE IF( namcouple_info%send_variables(jf) == 'SND_SMOIS' ) THEN
                   cplsnd(l) = grid%SMOIS(i,1,j)

                   ! Sea Ice
                ELSE IF( namcouple_info%send_variables(jf) == 'SND_SEAICE' ) THEN
                   cplsnd(l) = grid%XICE(i,j)
                END IF
                l = l+1
             enddo
          enddo
       ELSE
          do k = kns,kne
             do j = jns,jne
                do i = ins,ine
                   ! Base State Pressure
                   IF( namcouple_info%send_variables(jf) == 'SND_PB' ) THEN
                      cplsnd(l) = grid%PB(i,k,j)

                      ! Perturbation to the Base State Pressure
                   ELSE IF( namcouple_info%send_variables(jf) == 'SND_P' ) THEN
                      cplsnd(l) = grid%P(i,k,j)

                      ! Hydrostatic Pressure
                   ELSE IF( namcouple_info%send_variables(jf) == 'SND_PHYD' ) THEN
                         cplsnd(l) = grid%P_HYD(i,k,j)

                      ! Perturbation to the Base Potential Temperature
                   ELSE IF( namcouple_info%send_variables(jf) == 'SND_T' ) THEN
                      cplsnd(l) = grid%T_1(i,k,j)

                      ! Cloud Water mixing ratio
                   ELSE IF( namcouple_info%send_variables(jf) == 'SND_QCLOUD' ) THEN
                      cplsnd(l) = grid%MOIST(i,k,j,p_qc)+grid%QC_CU(i,k,j)


                      ! Rain Water mixing ratio
                   ELSE IF( namcouple_info%send_variables(jf) == 'SND_QRAIN' ) THEN
                      cplsnd(l) = grid%MOIST(i,k,j,p_qr)

                      ! Snow mixing ratio
                   ELSE IF( namcouple_info%send_variables(jf) == 'SND_QSNOW' ) THEN
                      cplsnd(l) = grid%MOIST(i,k,j,p_qs)

                      ! Water Vapor mixing ratio
                   ELSE IF( namcouple_info%send_variables(jf) == 'SND_QVAPOR' ) THEN
                      cplsnd(l) = grid%MOIST(i,k,j,p_qv)

                      ! Ice Water mixing ratio
                   ELSE IF( namcouple_info%send_variables(jf) == 'SND_QICE' ) THEN
                     cplsnd(l) = grid%MOIST(i,k,j,p_qi)+grid%QI_CU(i,k,j)

                      ! Perturbation geopotential
                   ELSE IF( namcouple_info%send_variables(jf) == 'SND_PH' ) THEN
```

```
                        cplsnd(l) = grid%PH_2(i,k,j)

                        ! Base-state geopotential
                     ELSE IF( namcouple_info%send_variables(jf) == 'SND_PHB' ) THEN
                        cplsnd(l) = grid%PHB(i,k,j)

                        ! U (stag lon)
                     ELSE IF( namcouple_info%send_variables(jf) == 'SND_U' ) THEN
                        cplsnd(l) = grid%U_2(i,k,j)

                        ! V (stag lat)
                     ELSE IF( namcouple_info%send_variables(jf) == 'SND_V' ) THEN
                        cplsnd(l) = grid%V_2(i,k,j)

                     END IF
                     l = l+1
                  enddo
               enddo
            enddo
         END IF

         CALL cpl_sndfield( grid%id, lltosend, ifldid, cplsnd )
      END IF
      IF (ALLOCATED(cplsnd)) DEALLOCATE (cplsnd)
   ENDDO
#endif
```

12. The cpl_sndfield subroutine has added support for 1D exchange data / point partition used for WRF/CHIMERE coupling. Declaration of the pdata array becomes:

```
#ifdef cpl_wrf_chimere
   REAL, DIMENSION(:),   INTENT(IN) :: pdata     ! data to be sent
#else
   REAL, DIMENSION(:,:), INTENT(IN) :: pdata     ! data to be sent
#endif
```

13. The cpl_rcv is also adapted for 3D field exchanges through a 1D indexing space (point partition). Declaration of the pdatacpl array now becomes:

```
#ifdef cpl_wrf_chimere
  REAL, DIMENSION( ims:ime, kms:kme,  jms:jme ),  INTENT(  OUT) :: pdatacpl    ! coupling data
#else
  REAL, DIMENSION( ims:ime, jms:jme ), INTENT(  OUT) :: pdatacpl    ! coupling data
#endif
```

And declaration of the zdata array:

```
#ifdef cpl_wrf_chimere
   REAL, DIMENSION( (ipe-ips+1)*(jpe-jps+1)*(kpe-kps+1) ) :: zdata ! data received from the coupler
   INTEGER :: i,j,k,l
#endif
#ifdef cpl_wrf_nemo
   REAL, DIMENSION( ips:ipe, jps:jpe ) :: zdata ! data received from the coupler
#endif
```

The array indexing changes accordingly:

(a) At the end of the subroutine, add the conditional compilation directives for NEMO around the block starting with the line

```
ifldid = cpl_get_fldid( cdsuffix )
```

so that the code becomes:

```
!! Get data from NEMO
#ifdef cpl_wrf_nemo
   ifldid = cpl_get_fldid( cdsuffix )
..........
        END IF
    END DO
#endif
```

(b) Then add the following code using the new array indexing for CHIMERE:

```
!! Get data from CHIMERE
#ifdef cpl_wrf_chimere
  ifldid = cpl_get_fldid( cdsuffix )

   DO jext = 1, max_edom
      IF( cpl_toreceive( kdomwrf, jext, ifldid ) ) THEN
        IF( coupler_name == 'oasis' ) THEN
          CALL cpl_oasis_rcv( kdomwrf, jext, ifldid, nsecrun, zdata )
          l = 1
          DO k = kps,kpe
             DO j = jps,jpe
                DO i = ips,ipe
                    pdatacpl(i,k,j) = zdata(l)
                    l = l+1
                ENDDO
             ENDDO
          ENDDO
        END IF
      END IF
   END DO
#endif
```

14. Finally a new subroutine cpl_rcv_4d has been added to receive the 4D variables from CHIMERE.

```
!!!!!!!!!!!!!!!!!!!!!
#ifdef cpl_wrf_chimere
   SUBROUTINE cpl_rcv_4D( kdomwrf, cdsuffix, dim4,      &
      &                   ids, ide, jds, jde, kds, kde, &
      &                   ims, ime, jms, jme, kms, kme, &
      &                   ips, ipe, jps, jpe, kps, kpe, &
      &                   max_edom, pdatacpl)
      !!---------------------------------------------------------------------
      !!             ***  SUBROUTINE cpl_rcv_4D  ***
      !!
      !! ** Purpose :   receive coupling 4D data from CHIMERE
      !!---------------------------------------------------------------------
      INTEGER, INTENT(IN   )      :: kdomwrf     ! wrf domain index
      CHARACTER(*), INTENT(IN   ) :: cdsuffix    ! field name suffix
      INTEGER, INTENT(IN   )      :: dim4        ! dimension of the fourth dimension
      INTEGER, INTENT(IN   )      :: ids,ide,jds,jde,kds,kde
      INTEGER, INTENT(IN   )      :: ims,ime,jms,jme,kms,kme
      INTEGER, INTENT(IN   )      :: ips,ipe,jps,jpe,kps,kpe
      INTEGER, INTENT(IN   )      :: max_edom    ! max number of external model domains
      ! coupling data
      REAL, DIMENSION( ims:ime, kms:kme, jms:jme, dim4 ), INTENT(  OUT) :: pdatacpl
      !
      INTEGER :: jext                                ! external domain index
      INTEGER :: ifldid                              ! field index
      ! data received from the coupler
      REAL, DIMENSION( (ipe-ips+1)*(jpe-jps+1)*(kpe-kps+1)*dim4 )   :: zdata
      INTEGER :: i,j,k,l,m

      !!---------------------------------------------------------------------

      ifldid = cpl_get_fldid( cdsuffix )

      DO jext = 1, max_edom
         IF( cpl_toreceive( kdomwrf, jext, ifldid ) ) THEN
           IF( coupler_name == 'oasis' ) THEN
              CALL cpl_oasis_rcv( kdomwrf, jext, ifldid, nsecrun, zdata )
              l = 1
              DO m = 1,dim4
                DO k = kps,kpe
                  DO j = jps,jpe
                     DO i = ips,ipe
                         pdatacpl(i,k,j,m) = zdata(l)
                         l = l+1
                     ENDDO
                  ENDDO
                ENDDO
              ENDDO
           ENDIF
         ENDIF
      ENDDO

   END SUBROUTINE cpl_rcv_4D
#endif
!!!!!!!!!!!!!!!!!!!!!
```

B.4.2 module_cpl_oasis3.F

This module contains data structures and subroutines called by the higher-level subroutines of the module_cpl module. Most of them call OASIS subroutines to exchange data via the defined partition. The subroutines cpl_oasis_define, cpl_oasis_snd, cpl_oasis_rcv were adapted to take into account the "Points" partition and the 3D data fields used for data transfer between WRF and CHIMERE. The new cpl_oasis_read_namcouple subroutine reads parameters from the auto-generated namcouple file. Also, a few empty (dummy) subroutines have been added for compilation when the cpl_wrf_chimere flag is desactivated.
The main module part contains the following changes.

1. The dbl2d array to store received field becomes 1D:

```
#ifdef cpl_wrf_chimere
     #if ( RWORDSIZE == 8 )
          REAL        , POINTER, DIMENSION(:) ::   dbl2d   ! 1d array to store received field
     #else
          REAL(kind=8), POINTER, DIMENSION(:) ::   dbl2d   ! 1d array to store received field
     #endif
#else
     #if ( RWORDSIZE == 8 )
          REAL        , POINTER, DIMENSION(:,:) ::   dbl2d   ! 2d array to store received field
     #else
          REAL(kind=8), POINTER, DIMENSION(:,:) ::   dbl2d   ! 2d array to store received field
     #endif
#endif
```

2. The structure to store the namcouple file data, as well as the maximum number of lines read:

```
  INTEGER,parameter  :: Length_file_max=1000  ! Maximum number of lines read in namcouple file.

   ! namcouple_type is a structure that can contains all necesary namcouple info.
   ! It is filled by a call to the "read_namcouple" subroutine.
   type, PUBLIC :: namcouple_type
      ! First sectoin of namcouple file
      INTEGER                                  :: nfields
      INTEGER                                  :: nbmodel
      INTEGER                                  :: runtime
      INTEGER                                  :: nlogprt
      CHARACTER(len=6),dimension(:),allocatable :: model_name

      ! Second section of namcouple file
      ! First line
      CHARACTER(len=128),dimension(:),allocatable :: send_variables
      CHARACTER(len=128),dimension(:),allocatable :: recv_variables
      INTEGER,dimension(:),allocatable            :: index_table
      INTEGER,dimension(:),allocatable            :: coupling_freq
      INTEGER,dimension(:),allocatable            :: n_transformation
      CHARACTER(len=32),dimension(:),allocatable  :: restart
      CHARACTER(len=8),dimension(:),allocatable   :: field_status

      ! Second line
      INTEGER,dimension(:),allocatable            :: n_point_source_first_dim
      INTEGER,dimension(:),allocatable            :: n_point_source_second_dim
      INTEGER,dimension(:),allocatable            :: n_point_target_first_dim
      INTEGER,dimension(:),allocatable            :: n_point_target_second_dim
      CHARACTER(len=4),dimension(:),allocatable   :: source_grid
      CHARACTER(len=4),dimension(:),allocatable   :: target_grid
      INTEGER,dimension(:),allocatable            :: lag
      INTEGER,dimension(:),allocatable            :: seq

      ! Transformations
      CHARACTER(len=256),dimension(:),allocatable :: transformations_names

      ! Third line
      CHARACTER(len=1),dimension(:),allocatable   :: source_grid_first_dim_characteristic
      INTEGER,dimension(:),allocatable            :: source_grid_first_dim
      CHARACTER(len=1),dimension(:),allocatable   :: target_grid_first_dim_characteristic
      INTEGER,dimension(:),allocatable            :: target_grid_first_dim
   end type namcouple_type
```

3. The new routines need to be declared:

```
   PUBLIC cpl_oasis_read_namcouple
   PUBLIC cpl_oasis_check_file_length
   PUBLIC cpl_oasis_remove_comments
```

```
   PUBLIC cpl_oasis_read_integer_namcouple
   PUBLIC cpl_oasis_search_in_file
   PUBLIC cpl_oasis_get_number_of_elements
```

4. The subroutine cpl_oasis_define receives an additional argument namcouple_info:

```
SUBROUTINE cpl_oasis_define( cdsndname, cdrcvname, namcouple_info, pgrid, nbin)
.........
   TYPE(NAMCOUPLE_TYPE),INTENT(IN),optional :: namcouple_info ! Information read in the namcouple file
   INTEGER,INTENT(IN),optional              :: nbin ! Number of bin of CHIMERE aerosol size distribution
```

5. Partition variables are defined differently for CHIMERE. Right before declaration of variables id_part, iparal, and ishape add the directives to separate NEMO and CHIMERE variables, as well as CHIMERE-specific declarations. You code will look like:

```
#ifdef cpl_wrf_nemo
      INTEGER :: id_part                ! partition id in oasis
      INTEGER :: iparal(5)              ! OASIS box partition
      INTEGER :: ishape(2,2)            ! shape of arrays passed to PSMILe
#endif
#ifdef cpl_wrf_chimere
      INTEGER, DIMENSION(6)             :: id_part ! 6 partitions are required for the WRF/CHIMERE coupling.
      INTEGER, DIMENSION(:), ALLOCATABLE :: iparal  ! OASIS points partition
      INTEGER, DIMENSION(6,2)           :: ishape  ! shape of arrays passed to PSMILe
      INTEGER :: Npts,ipartition
      INTEGER :: i,j,k,l
      INTEGER :: ins,ine,jns,jne,kns,kne
#endif
```

6. Likewise, separate the partition definition implementations. Wrap the NEMO-related code with the #ifdef cpl_wrf_nemo and #endif directives. Then add the new code for CHIMERE that defines partitions for WRF grids wrf1 to wrf6. The code will look like:

   The code for the NEMO case and already implemented in WRF is:

```
#ifdef cpl_wrf_nemo
      CALL wrf_message('cpl_oasis_define : initialization in coupled ocean/atmosphere case')
      CALL wrf_debug(nlevdbg, '~~~~~~~~~~~~~~~~~~~~~~~~')
.....................................................................
#endif
```

   The new code developed for CHIMERE in WRF is:

```
#ifdef cpl_wrf_chimere
   CALL wrf_message('cpl_oasis_define : initialization in coupled wrf/chimere case')
   CALL wrf_debug(nlevdbg, '~~~~~~~~~~~~~~~~~~~~~~~~')

   llcompute_core = PRESENT(pgrid)

   ! ------------------------------------------------------------------
   ! ... Define communicator used between computing cores.
   CALL oasis_set_couplcomm( ndm_comm, ierror )               ! provide this communicator to OASIS3-MCT
   IF ( ierror /= OASIS_Ok ) CALL cpl_oasis_abort( 'cpl_oasis_define', 'Failure in oasis_set_couplcomm')

   ! ------------------------------------------------------------------
   ! ... Define the partitions.
   ! ------------------------------------------------------------------
   IF( llcompute_core ) THEN

      ! ... get mpi domain position
      CALL get_ijk_from_grid( pgrid, ids, ide, jds, jde, kds, kde, &
         &                           ims, ime, jms, jme, kms, kme, &
         &                           ips, ipe, jps, jpe, kps, kpe  )

      ! DEFINITION OF UNSTAGGERED GRIDS PARTITIONS AND SHAPES
      ins = ips
      jns = jps
      kns = kps

      ine = ipe
      if (ine.ge.ide) ine = ine-1
      jne = jpe
      if (jne.ge.jde) jne = jne-1
      kne = kpe-1
```

```
! 2D VARIABLES
Npts = (ine-ins+1)*(jne-jns+1)
allocate(iparal(Npts+2))
iparal(1) = 4
iparal(2) = Npts
ipartition = 3
do j = jns,jne
   do i = ins,ine
      iparal(ipartition) = i+(j-1)*(ide-1)
      ipartition = ipartition+1
   enddo
enddo

ishape(1,1) = 1
ishape(1,2) = Npts

WRITE(cltxt,*) 'Define the partition for 2D Variables'       ;   CALL wrf_debug(nlevdbg, cltxt)
WRITE(cltxt,*) ' multiexchg: ishape(1,1) =', ishape(1,1)      ;   CALL wrf_debug(nlevdbg, cltxt)
WRITE(cltxt,*) ' multiexchg: ishape(1,2) =', ishape(1,2)      ;   CALL wrf_debug(nlevdbg, cltxt)

CALL oasis_def_partition( id_part(1), iparal, ierror )
IF( ierror /= OASIS_Ok ) CALL cpl_oasis_abort( 'cpl_oasis_define', &
&'Failure in oasis_def_partition for 2D Variables')
deallocate(iparal)

! FOR 3D VARIABLES
Npts = (ine-ins+1)*(jne-jns+1)*(kne-kns+1)
allocate(iparal(Npts+2))
iparal(1) = 4
iparal(2) = Npts
ipartition = 3
do k = kns,kne
   do j = jns,jne
      do i = ins,ine
         iparal(ipartition) = i+(j-1)*(ide-1)+(k-1)*(ide-1)*(jde-1)
         ipartition = ipartition+1
      enddo
   enddo
enddo

ishape(2,1) = 1
ishape(2,2) = Npts

WRITE(cltxt,*) 'Define the partition for 3D Variables'       ;   CALL wrf_debug(nlevdbg, cltxt)
WRITE(cltxt,*) ' multiexchg: ishape(2,1) =', ishape(2,1)      ;   CALL wrf_debug(nlevdbg, cltxt)
WRITE(cltxt,*) ' multiexchg: ishape(2,2) =', ishape(2,2)      ;   CALL wrf_debug(nlevdbg, cltxt)

CALL oasis_def_partition( id_part(2), iparal, ierror )
IF( ierror /= OASIS_Ok ) CALL cpl_oasis_abort('cpl_oasis_define',&
&'Failure in oasis_def_partition for 3D Variables')
deallocate(iparal)


! DEFINITION OF STAGGERED GRIDS PARTITIONS AND SHAPES

! W (vertical stag)
ine = ipe
if (ine.ge.ide) ine = ine-1
jne = jpe
if (jne.ge.jde) jne = jne-1
kne = kpe

Npts = (ine-ins+1)*(jne-jns+1)*(kne-kns+1)
allocate(iparal(Npts+2))
iparal(1) = 4
iparal(2) = Npts
ipartition = 3
do k = kns,kne
   do j = jns,jne
      do i = ins,ine
         iparal(ipartition) = i+(j-1)*(ide-1)+(k-1)*(ide-1)*(jde-1)
         ipartition = ipartition+1
      enddo
   enddo
enddo

ishape(3,1) = 1
ishape(3,2) = Npts

WRITE(cltxt,*) 'Define the partition for 3D Variables'       ;   CALL wrf_debug(nlevdbg, cltxt)
```

```
      WRITE(cltxt,*) ' multiexchg: ishape(3,1) =', ishape(3,1)      ;   CALL wrf_debug(nlevdbg, cltxt)
      WRITE(cltxt,*) ' multiexchg: ishape(3,2) =', ishape(3,2)      ;   CALL wrf_debug(nlevdbg, cltxt)

      CALL oasis_def_partition( id_part(3), iparal, ierror )
      IF( ierror /= OASIS_Ok ) CALL cpl_oasis_abort('cpl_oasis_define',&
      &'Failure in oasis_def_partition for 3D Variables (staggered z)')
      deallocate(iparal)

      ! U (lon stag)
      ine = ipe
      jne = jpe
      if (jne.ge.jde) jne = jne-1
      kne = kpe-1

      Npts = (ine-ins+1)*(jne-jns+1)*(kne-kns+1)
      allocate(iparal(Npts+2))
      iparal(1) = 4
      iparal(2) = Npts
      ipartition = 3
      do k = kns,kne
         do j = jns,jne
            do i = ins,ine
               iparal(ipartition) = i+(j-1)*ide+(k-1)*ide*(jde-1)
               ipartition = ipartition+1
            enddo
         enddo
      enddo

      ishape(4,1) = 1
      ishape(4,2) = Npts

      WRITE(cltxt,*) 'Define the partition for 3D Variables'       ;   CALL wrf_debug(nlevdbg, cltxt)
      WRITE(cltxt,*) ' multiexchg: ishape(4,1) =', ishape(4,1)      ;   CALL wrf_debug(nlevdbg, cltxt)
      WRITE(cltxt,*) ' multiexchg: ishape(4,2) =', ishape(4,2)      ;   CALL wrf_debug(nlevdbg, cltxt)

      CALL oasis_def_partition( id_part(4), iparal, ierror )
      IF( ierror /= OASIS_Ok ) CALL cpl_oasis_abort('cpl_oasis_define',&
      &'Failure in oasis_def_partition for 3D Variables (staggered x)')
      deallocate(iparal)

      ! V (lat stag)
      ine = ipe
      if (ine.ge.ide) ine = ine-1
      jne = jpe
      kne = kpe-1

      Npts = (ine-ins+1)*(jne-jns+1)*(kne-kns+1)
      allocate(iparal(Npts+2))
      iparal(1) = 4
      iparal(2) = Npts
      ipartition = 3
      do k = kns,kne
         do j = jns,jne
            do i = ins,ine
               iparal(ipartition) = i+(j-1)*(ide-1)+(k-1)*(ide-1)*jde
               ipartition = ipartition+1
            enddo
         enddo
      enddo

      ishape(5,1) = 1
      ishape(5,2) = Npts

      WRITE(cltxt,*) 'Define the partition for 3D Variables'       ;   CALL wrf_debug(nlevdbg, cltxt)
      WRITE(cltxt,*) ' multiexchg: ishape(5,1) =', ishape(5,1)      ;   CALL wrf_debug(nlevdbg, cltxt)
      WRITE(cltxt,*) ' multiexchg: ishape(5,2) =', ishape(5,2)      ;   CALL wrf_debug(nlevdbg, cltxt)

      CALL oasis_def_partition( id_part(5), iparal, ierror )
      IF( ierror /= OASIS_Ok ) CALL cpl_oasis_abort('cpl_oasis_define',&
      &'Failure in oasis_def_partition for 3D Variables (staggered y)')
      deallocate(iparal)

      ! FOR 4D VARIABLES: UNSTAGGERED
      ine = ipe
      if (ine.ge.ide) ine = ine-1
      jne = jpe
      if (jne.ge.jde) jne = jne-1
      kne = kpe-1

      Npts = (ine-ins+1)*(jne-jns+1)*(kne-kns+1)*nbin
      allocate(iparal(Npts+2))
```

```
      iparal(1) = 4
      iparal(2) = Npts
      ipartition = 3
      do l = 1,nbin
        do k = kns,kne
          do j = jns,jne
            do i = ins,ine
                iparal(ipartition) = i+(j-1)*(ide-1)+(k-1)*(ide-1)*(jde-1)+(l-1)*(ide-1)*(jde-1)*(kpe-1)
                ipartition = ipartition+1
            enddo
          enddo
        enddo
      enddo

      ishape(6,1) = 1
      ishape(6,2) = Npts

      WRITE(cltxt,*) 'Define the partition for 4D Variables'       ;   CALL wrf_debug(nlevdbg, cltxt)
      WRITE(cltxt,*) ' multiexchg: ishape(6,1) =', ishape(6,1)     ;   CALL wrf_debug(nlevdbg, cltxt)
      WRITE(cltxt,*) ' multiexchg: ishape(6,2) =', ishape(6,2)     ;   CALL wrf_debug(nlevdbg, cltxt)

      CALL oasis_def_partition( id_part(6), iparal, ierror )
      IF( ierror /= OASIS_Ok ) CALL cpl_oasis_abort('cpl_oasis_define',&
      &'Failure in oasis_def_partition for 4D Variables')
      deallocate(iparal)
   ELSE
      CALL wrf_debug(nlevdbg, 'no partition for IO cores')

      allocate(iparal(3))
      iparal(:) = 0   ! Dummy partition for IO cores

      CALL oasis_def_partition( id_part(1), iparal, ierror )
      IF( ierror /= OASIS_Ok ) CALL cpl_oasis_abort('cpl_oasis_define',&
      &'Failure in oasis_def_partition for 2D Variables')

      CALL oasis_def_partition( id_part(2), iparal, ierror )
      IF( ierror /= OASIS_Ok ) CALL cpl_oasis_abort('cpl_oasis_define',&
      &'Failure in oasis_def_partition for 3D Variables')

      CALL oasis_def_partition( id_part(3), iparal, ierror )
      IF( ierror /= OASIS_Ok ) CALL cpl_oasis_abort('cpl_oasis_define',&
      &'Failure in oasis_def_partition for 3D Variables (staggered vertical)')

      CALL oasis_def_partition( id_part(4), iparal, ierror )
      IF( ierror /= OASIS_Ok ) CALL cpl_oasis_abort('cpl_oasis_define',&
      &'Failure in oasis_def_partition for 3D Variables (staggered lon)')

      CALL oasis_def_partition( id_part(5), iparal, ierror )
      IF( ierror /= OASIS_Ok ) CALL cpl_oasis_abort('cpl_oasis_define',&
      &'Failure in oasis_def_partition for 3D Variables (staggered lat)')

      CALL oasis_def_partition( id_part(6), iparal, ierror )
      IF( ierror /= OASIS_Ok ) CALL cpl_oasis_abort('cpl_oasis_define',&
      &'Failure in oasis_def_partition for 4D Variables')
      deallocate(iparal)
   ENDIF

   ! ----------------------------------------------------------------
   ! ... Define the variables that can be send/received by WRF
   ! ----------------------------------------------------------------
   ! coupling field related to this nest
   IF( llcompute_core ) THEN ; idwrf1 = pgrid%id ; idwrf2 = pgrid%id
    ! define all (dummy) coupling fields
   ELSE                      ; idwrf1 = 1        ; idwrf2 = max_domains
   ENDIF

   ! ----------------------------------------------------------------
   ! ... Define sent variables.
   ! ----------------------------------------------------------------
   DO jf = 1, namcouple_info%nfields
      ssnd(1,1,jf)%clname = TRIM(cdsndname(1,1,jf))
      IF( namcouple_info%source_grid(jf) == 'wrf1' ) THEN
         CALL oasis_def_var(ssnd(1,1,jf)%nid, ssnd(1,1,jf)%clname, id_part(1) , (/1,1/), &
         &OASIS_Out, ishape(1,:), OASIS_Real, ierror)
      ENDIF
      IF( namcouple_info%source_grid(jf) == 'wrf2' ) THEN
         CALL oasis_def_var(ssnd(1,1,jf)%nid, ssnd(1,1,jf)%clname, id_part(2) , (/1,1/), &
         &OASIS_Out, ishape(2,:), OASIS_Real, ierror)
      ENDIF
      IF( namcouple_info%source_grid(jf) == 'wrf3' ) THEN
         CALL oasis_def_var(ssnd(1,1,jf)%nid, ssnd(1,1,jf)%clname, id_part(3) , (/1,1/), &
```

```
         &OASIS_Out, ishape(3,:), OASIS_Real, ierror)
      ENDIF
      IF( namcouple_info%source_grid(jf) == 'wrf4' ) THEN
         CALL oasis_def_var(ssnd(1,1,jf)%nid, ssnd(1,1,jf)%clname, id_part(4) , (/1,1/), &
         &OASIS_Out, ishape(4,:), OASIS_Real, ierror)
      ENDIF
      IF( namcouple_info%source_grid(jf) == 'wrf5' ) THEN
         CALL oasis_def_var(ssnd(1,1,jf)%nid, ssnd(1,1,jf)%clname, id_part(5) , (/1,1/), &
         &OASIS_Out, ishape(5,:), OASIS_Real, ierror)
      ENDIF
      IF( namcouple_info%source_grid(jf) == 'wrf6' ) THEN
         CALL oasis_def_var(ssnd(1,1,jf)%nid, ssnd(1,1,jf)%clname, id_part(6), (/1,1/), &
         &OASIS_Out, ishape(6,:), OASIS_Real, ierror)
      ENDIF
      IF( ierror /= OASIS_Ok ) THEN
         WRITE(cltxt,*) 'wrf domain 1 external model domain 1', &
              ' field ',jf,' (',TRIM(ssnd(1,1,jf)%clname),'): oasis_def_var failed'
         CALL wrf_message( cltxt )
         CALL cpl_oasis_abort( 'cpl_oasis_define', 'Failure in oasis_def_var')
      ENDIF
      WRITE(cltxt,*) 'cpl_oasis_define ok for :', TRIM(ssnd(1,1,jf)%clname), ssnd(1,1,jf)%nid
      CALL wrf_debug(nlevdbg2, cltxt)
      IF( ssnd(1,1,jf)%nid /= -1 ) THEN
         WRITE(cltxt,*) ' var snd: ', ssnd(1,1,jf)%nid, ' ',TRIM(ssnd(1,1,jf)%clname)
         CALL wrf_debug(nlevdbg, cltxt)
      ENDIF

   ! ------------------------------------------------------------------
   ! ... Define received variables.
   ! ------------------------------------------------------------------
      srcv(1,1,jf)%clname = TRIM(cdrcvname(1,1,jf))
      IF( namcouple_info%target_grid(jf) == 'wrf1' ) THEN
         CALL oasis_def_var(srcv(1,1,jf)%nid, srcv(1,1,jf)%clname, id_part(1) , (/1,1/), &
         &OASIS_In, ishape(1,:), OASIS_Real, ierror)
      ENDIF
      IF( namcouple_info%target_grid(jf) == 'wrf2' ) THEN
         CALL oasis_def_var(srcv(1,1,jf)%nid, srcv(1,1,jf)%clname, id_part(2) , (/1,1/), &
         &OASIS_In, ishape(2,:), OASIS_Real, ierror)
      ENDIF
      IF( namcouple_info%target_grid(jf) == 'wrf3' ) THEN
         CALL oasis_def_var(srcv(1,1,jf)%nid, srcv(1,1,jf)%clname, id_part(3) , (/1,1/), &
         &OASIS_In, ishape(3,:), OASIS_Real, ierror)
      ENDIF
      IF( namcouple_info%target_grid(jf) == 'wrf4' ) THEN
         CALL oasis_def_var(srcv(1,1,jf)%nid, srcv(1,1,jf)%clname, id_part(4) , (/1,1/), &
         &OASIS_In, ishape(4,:), OASIS_Real, ierror)
      ENDIF
      IF( namcouple_info%target_grid(jf) == 'wrf5' ) THEN
         CALL oasis_def_var(srcv(1,1,jf)%nid, srcv(1,1,jf)%clname, id_part(5) , (/1,1/), &
         &OASIS_In, ishape(5,:), OASIS_Real, ierror)
      ENDIF
      IF( namcouple_info%target_grid(jf) == 'wrf6' ) THEN
         CALL oasis_def_var(srcv(1,1,jf)%nid, srcv(1,1,jf)%clname, id_part(6) , (/1,1/), &
         &OASIS_In, ishape(6,:), OASIS_Real, ierror)
      ENDIF
      IF( ierror /= OASIS_Ok ) THEN
         WRITE(cltxt,*) 'wrf domain 1 external model domain 1', &
              ' field ',jf,' (',TRIM(srcv(1,1,jf)%clname),'): oasis_def_var failed'
         CALL wrf_message( cltxt )
         CALL cpl_oasis_abort( 'cpl_oasis_define', 'Failure in oasis_def_var')
      ENDIF
      WRITE(cltxt,*) 'cpl_oasis_define ok for :', TRIM(srcv(1,1,jf)%clname), srcv(1,1,jf)%nid
      CALL wrf_debug(nlevdbg2, cltxt)
      IF( srcv(1,1,jf)%nid /= -1 ) THEN
         WRITE(cltxt,*) ' var rcv: ', srcv(1,1,jf)%nid, ' ',TRIM(srcv(1,1,jf)%clname)
         CALL wrf_debug(nlevdbg, cltxt)
      ENDIF
      IF( srcv(1,1,jf)%nid /= -1 .AND. llcompute_core ) THEN !allocate received array
       IF( namcouple_info%target_grid(jf) == 'wrf1' ) ALLOCATE( srcv(1,1,jf)%dbl2d(ishape(1,2)), stat =ierror )
       IF( namcouple_info%target_grid(jf) == 'wrf2' ) ALLOCATE( srcv(1,1,jf)%dbl2d(ishape(2,2)), stat =ierror )
       IF( namcouple_info%target_grid(jf) == 'wrf3' ) ALLOCATE( srcv(1,1,jf)%dbl2d(ishape(3,2)), stat =ierror )
       IF( namcouple_info%target_grid(jf) == 'wrf4' ) ALLOCATE( srcv(1,1,jf)%dbl2d(ishape(4,2)), stat =ierror )
       IF( namcouple_info%target_grid(jf) == 'wrf5' ) ALLOCATE( srcv(1,1,jf)%dbl2d(ishape(5,2)), stat =ierror )
       IF( namcouple_info%target_grid(jf) == 'wrf6' ) ALLOCATE( srcv(1,1,jf)%dbl2d(ishape(6,2)), stat =ierror )
       IF(ierror > 0)   CALL cpl_oasis_abort('cpl_oasis_define','Failure in allocating srcv')
      END IF
   END DO !jf
#endif
```

7. In the cpl_oasis_snd subroutine the array pdata becomes 1D for CHIMERE:

```
#ifdef cpl_wrf_chimere
      REAL, DIMENSION(:),   INTENT(IN) :: pdata     ! data to be sent
#else
      REAL, DIMENSION(:,:), INTENT(IN) :: pdata     ! data to be sent
#endif
..............

#if ( RWORDSIZE == 8 )
      CALL oasis_put(ssnd(kdomwrf,kdomext,kfldid)%nid, ksec, pdata, info)
#else
      CALL oasis_put(ssnd(kdomwrf,kdomext,kfldid)%nid, ksec, DBLE(pdata), info)
#endif
```

8. The same for the array pcplrcv in the cpl_oasis_rcv subroutine:

```
#ifdef cpl_wrf_chimere
      REAL, DIMENSION(:),   INTENT(  OUT) :: pcplrcv   ! output data
#else
      REAL, DIMENSION(:,:), INTENT(  OUT) :: pcplrcv   ! output data
#endif
.......................

#if ( RWORDSIZE == 8 )
      pcplrcv =      srcv(kdomwrf,kdomext,kfldid)%dbl2d
#else
      pcplrcv = REAL(srcv(kdomwrf,kdomext,kfldid)%dbl2d, kind=4)
#endif
```

9. The new routines to read the namcouple file and the coupling parameters and fields defined therein have been added at the end of module_cpl_oasis3.F:

```
SUBROUTINE cpl_oasis_read_namcouple(namcouple_info)
  !!---------------------------------------------------------------------
  !!              ***  ROUTINE cpl_oasis_read_namcouple  ***
  !!
  !! ** Purpose : read namcouple file and fill the namcouple_info structure.
  !!---------------------------------------------------------------------
  TYPE(NAMCOUPLE_TYPE),intent(inout)          :: namcouple_info ! Information read in the namcouple file
  CHARACTER(len=256)                          :: namcouple_file
  CHARACTER(len=256)                          :: line
  CHARACTER(len=256)                          :: namcouple_tmp
  CHARACTER(len=256)                          :: lag_seq1, lag_seq2
  INTEGER                                     :: ioval_namcouple
  INTEGER                                     :: i, j, int_tmp, n_element, int1, int2
  CHARACTER(len=8), dimension(:), allocatable :: trans_name_loc
  !!---------------------------------------------------------------------

  namcouple_file = 'namcouple'

  ! Check file length is inferior to "Length_file_max"
  CALL cpl_oasis_check_file_length(namcouple_file)

  ! Remove all comment line in a temporary file.
  CALL cpl_oasis_remove_comments(namcouple_file)
  namcouple_tmp=TRIM(namcouple_file)//'_tmp'

  ! Get the number of fields
  CALL cpl_oasis_read_integer_namcouple(namcouple_tmp,'$NFIELDS',namcouple_info%nfields)

  ! Get the number of model
  CALL cpl_oasis_read_integer_namcouple(namcouple_tmp,'$NBMODEL',namcouple_info%nbmodel,line)

  ! Get the model names
  allocate(namcouple_info%model_name(namcouple_info%nbmodel))
  READ(line,*) int_tmp,namcouple_info%model_name(:)

  ! Get the runtime
  CALL cpl_oasis_read_integer_namcouple(namcouple_tmp,'$RUNTIME',namcouple_info%runtime)

  ! Allocations
  IF (namcouple_info%nfields > 0) THEN
     allocate(namcouple_info%send_variables(namcouple_info%nfields))
     allocate(namcouple_info%recv_variables(namcouple_info%nfields))
     allocate(namcouple_info%index_table(namcouple_info%nfields))
     allocate(namcouple_info%coupling_freq(namcouple_info%nfields))
     allocate(namcouple_info%n_transformation(namcouple_info%nfields))
```

```
    allocate(namcouple_info%restart(namcouple_info%nfields))
    allocate(namcouple_info%field_status(namcouple_info%nfields))
    allocate(namcouple_info%source_grid(namcouple_info%nfields))
    allocate(namcouple_info%target_grid(namcouple_info%nfields))
    allocate(namcouple_info%source_grid_first_dim_characteristic(namcouple_info%nfields))
    allocate(namcouple_info%source_grid_first_dim(namcouple_info%nfields))
    allocate(namcouple_info%n_point_source_first_dim(namcouple_info%nfields))
    allocate(namcouple_info%n_point_source_second_dim(namcouple_info%nfields))
    allocate(namcouple_info%n_point_target_first_dim(namcouple_info%nfields))
    allocate(namcouple_info%n_point_target_second_dim(namcouple_info%nfields))
    allocate(namcouple_info%target_grid_first_dim_characteristic(namcouple_info%nfields))
    allocate(namcouple_info%target_grid_first_dim(namcouple_info%nfields))
    allocate(namcouple_info%lag(namcouple_info%nfields))
    allocate(namcouple_info%seq(namcouple_info%nfields))
    allocate(namcouple_info%transformations_names(namcouple_info%nfields))
END IF

! Get the nlogprt
CALL cpl_oasis_read_integer_namcouple(namcouple_tmp,'$NLOGPRT',namcouple_info%nlogprt)

! Open namcouple
OPEN(unit=103,file=TRIM(namcouple_tmp),status='OLD')

! Set the reading pointer at the begining of the '$STRINGS' section
CALL cpl_oasis_search_in_file(103,'$STRINGS')

DO i=1,namcouple_info%nfields
   ! Read the first line
   READ(103,'(A)') line
   READ(line,*) namcouple_info%send_variables(i), namcouple_info%recv_variables(i), &
        namcouple_info%index_table(i), namcouple_info%coupling_freq(i),             &
        namcouple_info%n_transformation(i), namcouple_info%restart(i),              &
        namcouple_info%field_status(i)

   IF (namcouple_info%coupling_freq(i) <= 0) THEN
      CALL wrf_message2('     ==== ABORTING ====')
      CALL wrf_message2('Error with the namcouple frequency')
      CALL wrf_abort
   END IF

   ! If the 6th char that is read is 'OUTPUT', THEN it means there is no restart name file.
   IF (TRIM(namcouple_info%restart(i)) == 'OUTPUT') THEN
      namcouple_info%restart(i) = ''
      namcouple_info%field_status(i) = 'OUTPUT'
   END IF

   ! With OASIS3-MCT IGNORED is equivalent to EXPORTED
   IF (TRIM(namcouple_info%field_status(i)) == 'IGNORED') THEN
      namcouple_info%field_status(i) = 'EXPORTED'
   END IF

   ! With OASIS3-MCT IGNOUT is equivalent to EXPOUT
   IF (TRIM(namcouple_info%field_status(i)) == 'IGNOUT') THEN
      namcouple_info%field_status(i) = 'EXPOUT'
   END IF

   IF (TRIM(namcouple_info%field_status(i)) == 'EXPORTED' .OR. &
        TRIM(namcouple_info%field_status(i)) == 'EXPOUT') THEN

      ! Read second line
      READ(103,'(A)') line
      CALL cpl_oasis_get_number_of_elements(line,n_element)

      lag_seq1 = ''
      lag_seq2 = ''
      IF (n_element == 2) THEN
         READ(line,*) namcouple_info%source_grid(i), namcouple_info%target_grid(i)
      ELSE IF(n_element == 3) THEN
         READ(line,*) namcouple_info%source_grid(i), namcouple_info%target_grid(i), &
              lag_seq1
      ELSE IF(n_element == 4) THEN
         READ(line,*) namcouple_info%source_grid(i), namcouple_info%target_grid(i), &
              lag_seq1, lag_seq2
      ELSE IF(n_element == 6) THEN
         READ(line,*) namcouple_info%n_point_source_first_dim(i), &
              namcouple_info%n_point_source_second_dim(i),        &
              namcouple_info%n_point_target_first_dim(i),         &
              namcouple_info%n_point_target_second_dim(i),        &
              namcouple_info%source_grid(i), namcouple_info%target_grid(i)
      ELSE IF(n_element == 7) THEN
         READ(line,*) namcouple_info%n_point_source_first_dim(i),         &
```

```
               namcouple_info%n_point_source_second_dim(i),                &
               namcouple_info%n_point_target_first_dim(i),                 &
               namcouple_info%n_point_target_second_dim(i),                &
               namcouple_info%source_grid(i), namcouple_info%target_grid(i), &
               lag_seq1
         ELSE IF(n_element == 8) THEN
            READ(line,*) namcouple_info%n_point_source_first_dim(i),         &
               namcouple_info%n_point_source_second_dim(i),                &
               namcouple_info%n_point_target_first_dim(i),                 &
               namcouple_info%n_point_target_second_dim(i),                &
               namcouple_info%source_grid(i), namcouple_info%target_grid(i), &
               lag_seq1, lag_seq2
         ELSE
            CALL wrf_message2('     ==== ABORTING ====')
            CALL wrf_message2('The number of element on third line is not correct.')
            CALL wrf_abort
         END IF

         ! LAG or SEQ
         IF (TRIM(lag_seq1) /= '') THEN
            IF (TRIM(lag_seq1(1:4)) == 'SEQ=') THEN
               READ(lag_seq1(5:len_TRIM(lag_seq1)),*) namcouple_info%seq(i)
            ELSE IF (TRIM(lag_seq1(1:4)) == 'LAG=') THEN
               READ(lag_seq1(5:len_TRIM(lag_seq1)),*) namcouple_info%lag(i)
            ELSE
               CALL wrf_message2('     ==== ABORTING ====')
               CALL wrf_message2('Error in LAG or SEQ values')
               CALL wrf_abort
            END IF
         END IF

         IF (TRIM(lag_seq2) /= '') THEN
            IF (TRIM(lag_seq2(1:4)) == 'SEQ=') THEN
               READ(lag_seq2(5:len_TRIM(lag_seq2)),*) namcouple_info%seq(i)
            ELSE IF (TRIM(lag_seq2(1:4)) == 'LAG=') THEN
               READ(lag_seq2(5:len_TRIM(lag_seq2)),*) namcouple_info%lag(i)
            ELSE
               CALL wrf_message2('     ==== ABORTING ====')
               CALL wrf_message2('Error in LAG or SEQ values')
               CALL wrf_abort
            END IF
         END IF

         ! Read third line
         READ(103,'(A)') line
         READ(line,*) namcouple_info%source_grid_first_dim_characteristic(i), &
              namcouple_info%source_grid_first_dim(i),                        &
              namcouple_info%target_grid_first_dim_characteristic(i),         &
              namcouple_info%target_grid_first_dim(i)

      ELSE IF (TRIM(namcouple_info%field_status(i)) == 'OUTPUT') THEN
         IF (namcouple_info%send_variables(i) /= namcouple_info%recv_variables(i)) THEN
            CALL wrf_message2('     ==== ABORTING ====')
            CALL wrf_message2('The source symbolic name must be repeated twice in case of OUTPUT status.')
            CALL wrf_abort
         END IF

         ! Read second line
         READ(103,'(A)') line
         READ(line, *) namcouple_info%source_grid(i), namcouple_info%target_grid(i)

      ELSE IF (TRIM(namcouple_info%field_status(i)) == 'INPUT') THEN
         IF (namcouple_info%send_variables(i) /= namcouple_info%recv_variables(i)) THEN
            CALL wrf_message2('     ==== ABORTING ====')
            CALL wrf_message2('The source symbolic name must be repeated twice in case of INPUT status.')
            CALL wrf_abort
         END IF

      ELSE IF (TRIM(namcouple_info%field_status(i)) /= 'AUXILARY') THEN
            CALL wrf_message2('     ==== ABORTING ====')
            CALL wrf_message2('The field status: "'//TRIM(namcouple_info%field_status(i))//'" is not recognized.')
            CALL wrf_abort
      END IF

      ! Transformations
      IF (namcouple_info%n_transformation(i) > 0) THEN
         allocate(trans_name_loc(namcouple_info%n_transformation(i)))
         READ(103,'(A)') namcouple_info%transformations_names(i)
         namcouple_info%transformations_names(i)=TRIM(namcouple_info%transformations_names(i))
         READ(namcouple_info%transformations_names(i),*) trans_name_loc(:)
```

```
        ! Transformation info will not be used here.
        ! However, the right number of lines must be skipped
        DO j=1,namcouple_info%n_transformation(i)
           IF (trans_name_loc(j) /= 'LOCTRANS' .and. &
                TRIM(namcouple_info%field_status(i)) == 'OUTPUT') THEN
              CALL wrf_message2('     ==== ABORTING ====')
              CALL wrf_message2('LOCTRANS is the only transformation supported for OUTPUT fields.')
              CALL wrf_abort
           END IF

           IF (trans_name_loc(j) == 'LOCTRANS' .or. trans_name_loc(j) == 'CHECKIN' .or. &
                trans_name_loc(j) == 'MAPPING' .or. trans_name_loc(j) == 'SCRIPR' .or.  &
                trans_name_loc(j) == 'CONSERV' .or. trans_name_loc(j) == 'CHECKOUT') THEN
              READ(103,'(A)') line
           ELSE IF (trans_name_loc(j) == 'BLASOLD' .or. trans_name_loc(j) == 'BLASNEW') THEN
              READ(103,'(A)') line
              READ(line,*) int1,int2
              IF (int2 == 1) READ(103,'(A)') line
           ELSE
              CALL wrf_message2('     ==== ABORTING ====')
              CALL wrf_message2('Transformation '//trans_name_loc(j)//' is unknown.')
              CALL wrf_abort
           END IF
        END DO
        deallocate(trans_name_loc)
     END IF
  END DO

  READ(103,'(A)') line
  READ(line,*) line

  IF (TRIM(line) /= "$END") THEN
     CALL wrf_message2('     ==== ABORTING ====')
     CALL wrf_message2('Error in namcouple, there are extra field descriptions that were not read:')
     CALL wrf_message2(TRIM(line))
     CALL wrf_abort
  END IF

  ! Close namcouple
  close(103)
END SUBROUTINE cpl_oasis_read_namcouple


SUBROUTINE cpl_oasis_check_file_length(input_file)
  !!---------------------------------------------------------------------
  !!              ***  ROUTINE cpl_oasis_check_file_length  ***
  !!
  !! ** Purpose : run through the file and check if the end of the file is reached.
  !!---------------------------------------------------------------------
  !
  CHARACTER(len=*),intent(in)   :: input_file
  INTEGER                       :: i
  CHARACTER(len=256)            :: line
  !!---------------------------------------------------------------------

  ! Open input file
  OPEN(unit=101,file=TRIM(input_file),status='OLD')

  DO i=1,Length_file_max
     READ(101,*,end=1001) line
  END DO

  CALL wrf_message2('     ==== ABORTING ====')
  CALL wrf_message2('The Length_file_max variable must be higher than the number of lines in the namcouple file.')
  CALL wrf_message2('Please increase Length_file_max in: ', __FILE__, ' Exiting')
  CALL wrf_abort

1001 continue
  CLOSE(101)
END SUBROUTINE cpl_oasis_check_file_length


SUBROUTINE cpl_oasis_remove_comments(input_file)
  !!---------------------------------------------------------------------
  !!              ***  ROUTINE cpl_oasis_remove_comments  ***
  !!
  !! ** Purpose : read input file and create an output file without comments line.
  !!---------------------------------------------------------------------
  !
  CHARACTER(len=*),intent(in)   :: input_file
  INTEGER                       :: i
```

```
  CHARACTER(len=256)            :: line
  CHARACTER                     :: first_char
  !!-------------------------------------------------------------------

  ! Open input file
  OPEN(unit=101,file=TRIM(input_file),status='OLD')

  ! Open output file
  OPEN(unit=102,file=TRIM(input_file)//'_tmp',status='REPLACE')

  DO i=1,Length_file_max
     READ(101,'(A)',end=1002) line
     READ(line,*) first_char
     IF (first_char /= '#') THEN
        write(102,*) TRIM(line)
     END IF
  END DO

1002 continue
  ! Close files
  close(101)
  close(102)

END SUBROUTINE cpl_oasis_remove_comments


SUBROUTINE cpl_oasis_read_integer_namcouple(input_file,in_string,field_out,out_line)
  !!-------------------------------------------------------------------
  !!              ***  ROUTINE cpl_oasis_read_integer_namcouple  ***
  !!
  !! ** Purpose : return the first element, as an integer, on the line following the in_string line.
  !!              Optionally it can return the entire line.
  !!-------------------------------------------------------------------
  !
  CHARACTER(len=*),intent(in)            :: input_file
  CHARACTER(len=*),intent(in)            :: in_string
  INTEGER,intent(out)                    :: field_out
  CHARACTER(len=*),intent(out),optional  :: out_line
  CHARACTER(len=256)                     :: line
  INTEGER                                :: i
  LOGICAL                                :: found
  !!-------------------------------------------------------------------

  OPEN(unit=101,file=TRIM(input_file),status='OLD')

  found=.false.
  DO i=1,Length_file_max
     IF (.not.found) THEN
        READ(101,*) line
        IF(TRIM(line) == TRIM(in_string)) THEN
           READ(101,'(A)') line
           READ(line,*) field_out
           IF (present(out_line)) out_line = TRIM(line)
           found=.true.
        END IF
     END IF
  END DO

  close(101)

  IF (.not. found) THEN
     CALL wrf_message2('     ==== ABORTING ====')
     CALL wrf_message2(TRIM(in_string)//' was not found in namcouple file. Exiting')
     CALL wrf_abort
  END IF
END SUBROUTINE cpl_oasis_read_integer_namcouple


SUBROUTINE cpl_oasis_search_in_file(ioval,in_string)
  !!-------------------------------------------------------------------
  !!              ***  ROUTINE cpl_oasis_search_in_file  ***
  !!
  !! ** Purpose : move the reading pointer on in_string line of a given file.
  !!-------------------------------------------------------------------
  !
  INTEGER,intent(in)                     :: ioval
  CHARACTER(len=*),intent(in)            :: in_string
  LOGICAL                                :: found
  CHARACTER(len=256)                     :: line
  INTEGER                                :: i
  !!-------------------------------------------------------------------
```

```
  found=.false.
  DO i=1,Length_file_max
     IF (.not.found) THEN
        READ(ioval,*) line
        IF(TRIM(line) == TRIM(in_string)) found=.true.
     END IF
  END DO

  IF (.not. found) THEN
     CALL wrf_message2('     ==== ABORTING ====')
     CALL wrf_message2(TRIM(in_string)//' was not found in namcouple file. Exiting')
     CALL wrf_abort
  END IF

END SUBROUTINE cpl_oasis_search_in_file


SUBROUTINE cpl_oasis_get_number_of_elements(line,num)
  !!----------------------------------------------------------------------
  !!              ***  ROUTINE cpl_oasis_get_number_of_elements  ***
  !!
  !! ** Purpose : get the number of element present in a given string.
  !!----------------------------------------------------------------------
  !
  CHARACTER(len=*),intent(in) :: line
  INTEGER, intent(out)        :: num
  CHARACTER(len=256)          :: buf
  INTEGER                     :: i
  !!----------------------------------------------------------------------

  num = 0
  buf = TRIM(line)
  DO while ( len_TRIM(buf) > 0 )
     i = index(TRIM(buf), ' ', back=.true.)
     buf = TRIM(buf(1:i))
     num = num + 1
  END DO

END SUBROUTINE cpl_oasis_get_number_of_elements
```

10. The namcouple_type structure has also to be added in the "dummy" section of the module (see item 2 for this module):

```
 ! namcouple_type is a structure that can contains all necesary namcouple info.
   ! It is filled by a call to the "read_namcouple" subroutine.
   type, PUBLIC :: namcouple_type
      ! First sectoin of namcouple file
      INTEGER                                  :: nfields
      INTEGER                                  :: nbmodel
      INTEGER                                  :: runtime
      INTEGER                                  :: nlogprt
      CHARACTER(len=6),dimension(:),allocatable :: model_name

      ! Second section of namcouple file
      ! First line
      CHARACTER(len=128),dimension(:),allocatable :: send_variables
      CHARACTER(len=128),dimension(:),allocatable :: recv_variables
      INTEGER,dimension(:),allocatable            :: index_table
      INTEGER,dimension(:),allocatable            :: coupling_freq
      INTEGER,dimension(:),allocatable            :: n_transformation
      CHARACTER(len=32),dimension(:),allocatable  :: restart
      CHARACTER(len=8),dimension(:),allocatable   :: field_status

      ! Second line
      INTEGER,dimension(:),allocatable            :: n_point_source_first_dim
      INTEGER,dimension(:),allocatable            :: n_point_source_second_dim
      INTEGER,dimension(:),allocatable            :: n_point_target_first_dim
      INTEGER,dimension(:),allocatable            :: n_point_target_second_dim
      CHARACTER(len=4),dimension(:),allocatable   :: source_grid
      CHARACTER(len=4),dimension(:),allocatable   :: target_grid
      INTEGER,dimension(:),allocatable            :: lag
      INTEGER,dimension(:),allocatable            :: seq

      ! Transformations
      CHARACTER(len=256),dimension(:),allocatable :: transformations_names

      ! Third line
      CHARACTER(len=1),dimension(:),allocatable   :: source_grid_first_dim_characteristic
      INTEGER,dimension(:),allocatable            :: source_grid_first_dim
```

```
      CHARACTER(len=1),dimension(:),allocatable   :: target_grid_first_dim_characteristic
      INTEGER,dimension(:),allocatable            :: target_grid_first_dim
   end type namcouple_type
```

11. The cpl_oasis_define subroutine in the "dummy" part also has a new argument namcouple_info:

```
SUBROUTINE cpl_oasis_define( cdsndname, cdrcvname, namcouple_info, pgrid )
............
TYPE(NAMCOUPLE_TYPE),INTENT(IN),optional :: namcouple_info ! Information read in the namcouple file
```

12. Finally the namcouple-reading subroutines need also be declared as empty in the "dummy" part:

```
SUBROUTINE cpl_oasis_read_namcouple(namcouple_info)
     TYPE(NAMCOUPLE_TYPE),intent(out) :: namcouple_info ! Information read in the namcouple file
   END SUBROUTINE cpl_oasis_read_namcouple

   SUBROUTINE cpl_oasis_check_file_length(input_file)
     CHARACTER(len=*),intent(in)    :: input_file
   END SUBROUTINE cpl_oasis_check_file_length

   SUBROUTINE cpl_oasis_remove_comments(input_file)
     CHARACTER(len=*),intent(in)    :: input_file
   END SUBROUTINE cpl_oasis_remove_comments

   SUBROUTINE cpl_oasis_read_integer_namcouple(input_file,in_string,field_out,out_line)
     CHARACTER(len=*),intent(in)             :: input_file
     CHARACTER(len=*),intent(in)             :: in_string
     INTEGER,intent(out)                     :: field_out
     CHARACTER(len=*),intent(out),optional   :: out_line
   END SUBROUTINE cpl_oasis_read_integer_namcouple

   SUBROUTINE cpl_oasis_search_in_file(ioval,in_string)
     INTEGER,intent(in)                      :: ioval
     CHARACTER(len=*),intent(in)             :: in_string
   END SUBROUTINE cpl_oasis_search_in_file

   SUBROUTINE cpl_oasis_get_number_of_elements(line,num)
     CHARACTER(len=*),intent(in) :: line
     INTEGER, intent(out)        :: num
   END SUBROUTINE cpl_oasis_get_number_of_elements
```

B.5 Driver part

B.5.1 module_driver_constants.F

You need to increase the maximum number of fields that can be sent or received in coupled mode:

```
 INTEGER , PARAMETER :: max_cplfld = 100
```

B.5.2 module_integrate.F

The NEMO version of cpl_snd must be called only when NEMO coupling is activated. Add the directive around this subroutine call:

```
#ifdef cpl_wrf_nemo
   IF (coupler_on) CALL cpl_snd( grid_ptr )
#endif
```

B.5.3 module_wrf_top.F

In the wrf_init subroutine, cpl_defdomain is called with an additional argument, which is the number of aerosol bins defined for CHIMERE. The IF (coupler_on) statement becomes:

```
  IF (coupler_on) THEN
#ifdef cpl_wrf_chimere
     CALL cpl_defdomain( head_grid, config_flags%n_bin )
#else
```

```
     CALL cpl_defdomain( head_grid )
#endif
   ENDIF
```

B.6 Physics packages

B.6.1 module_microphysics_driver.F

The subroutine microphysics_driver receives additional arguments, which are the variables used of indirect effects coupling (those are currently the last subroutine arguments):

```
#ifdef cpl_wrf_chimere
   ,indirect_feed_chimere                &
   ,n_bin                                &
   ,max_sect                             &
   ,sect                                 &
   ,dropsource                           &
   ,num_aer                              &
   ,num_hygro_aer                        &
   ,hygro_aer                            &
   ,in_aer                               &
   ,num01,num02,num10                    &
   ,ccn1,ccn2,ccn3,ccn4,ccn5,ccn6        &
   ,rainrate,evaprate                    &
#endif
```

In the Framework section, the cal_cldfra1 subroutine and the coupling flags need to be visible:

```
#ifdef cpl_wrf_chimere
use module_radiation_driver, ONLY : cal_cldfra1
use module_cpl, ONLY : coupler_on, wrf_chimere
#endif
```

The new subroutine arguments are declared as follows:

```
 #ifdef cpl_wrf_chimere
   INTEGER, INTENT(IN) :: indirect_feed_chimere,n_bin,max_sect
   REAL,DIMENSION(max_sect),INTENT(IN) :: sect
   REAL,DIMENSION(ims:ime,kms:kme,jms:jme,n_bin),INTENT(IN) :: num_aer,num_hygro_aer
   REAL,DIMENSION(ims:ime,kms:kme,jms:jme),INTENT(IN)  :: hygro_aer
   REAL,DIMENSION(ims:ime,kms:kme,jms:jme),INTENT(IN)  :: in_aer
   REAL,DIMENSION(ims:ime,kms:kme,jms:jme),INTENT(OUT) :: dropsource
   REAL,DIMENSION(ims:ime,kms:kme,jms:jme),INTENT(OUT) :: num01,num02,num10
   REAL,DIMENSION(ims:ime,kms:kme,jms:jme),INTENT(OUT) :: ccn1,ccn2,ccn3,ccn4,ccn5,ccn6
   REAL,DIMENSION(ims:ime,kms:kme,jms:jme),INTENT(OUT) :: rainrate,evaprate

   !! Local
   REAL :: kzzz
   REAL,DIMENSION(ims:ime,kms:kme,jms:jme) :: temp,wtke
   REAL,DIMENSION(ims:ime,kms:kme,jms:jme) :: cldfra_chm,lcldfra_chm
#endif
```

In the subroutine body, in the case of the Thompson aerosol-aware scheme choice, a number of microphysics computations are performed: turbulent vertical velocity, cloud fraction , and the CCN spectrum are diagnozed. Right after the wrf_chem directive, add the following code:

```
#ifdef cpl_wrf_chimere
         IF(coupler_on.AND.wrf_chimere.AND.indirect_feed_chimere==1) THEN

           ! Diagnosing turbolent vertical velocity
           DO i=its,ite
             DO j=jts,jte
               DO k=kts,kte
                 num01(i,k,j)=num_aer(i,k,j,1)
                 num02(i,k,j)=num_aer(i,k,j,2)
                 num10(i,k,j)=num_aer(i,k,j,10)
                 IF(k==kts) then
                    wtke(i,k,j) = 0.2
                 ELSE
                    kzzz = exch_h(i,k,j)
                    kzzz = min(kzzz,100.)
                    kzzz = max(kzzz,0.01)
                    wtke(i,k,j) = sqrt(2.*4.*atan(1.))*exch_h(i,k,j)/dz8w(i,k,j)
                    wtke(i,k,j) = max(wtke(i,k,j),0.2)
```

```
                ENDIF
              ENDDO
            ENDDO
          ENDDO

          ! Diagnosing cloud fraction
          DO i=its,ite
            DO j=jts,jte
              DO k=kts,kte
                temp(i,k,j) = th(i,k,j)*pi_phy(i,k,j)
              ENDDO
            ENDDO
          ENDDO
          CALL cal_cldfra1(cldfra_chm,qv_curr,qc_curr,     &
                  qi_curr,qs_curr,                         &
                  F_QV,F_QC,F_QI,F_QS,temp,p,              &
                  F_ICE_PHY,F_RAIN_PHY,                    &
                  ids,ide, jds,jde, kds,kde,               &
                  ims,ime, jms,jme, kms,kme,               &
                  its,ite, jts,jte, kts,kte                )
          DO i=its,ite
           DO j=jts,jte
             DO k=kts,kte
               IF(cldfra_chm(i,k,j).gt.0.0) THEN
                 lcldfra_chm(i,k,j) = cldfra_chm(i,k,j) * &
                          qc_curr(i,k,j)/(qc_curr(i,k,j)+qi_curr(i,k,j)+qs_curr(i,k,j))
               ELSE
                 lcldfra_chm(i,k,j) = 0.0
               ENDIF
             ENDDO
           ENDDO
          ENDDO
          ! Diagnosing CCN spectrum
          call ccn_calc(n_bin,sect,num_aer,hygro_aer,      &
                        ccn1,ccn2,ccn3,ccn4,ccn5,ccn6,     &
                        ids,ide,jds,jde,kds,kde,           &
                        ims,ime,jms,jme,kms,kme,           &
                        its,ite,jts,jte,kts,kte            )
        ENDIF
#endif
```

The new arguments defined above are passed to the mp_gt_driver subroutine (right after the wrf_chem directive):

```
#ifdef cpl_wrf_chimere
    INDIRECT_FEED_CHIMERE=indirect_feed_chimere, &
    N_BIN=n_bin,                          &
    SECT=sect(1:n_bin+1),                 &
    NUM_aer=num_aer,                      &
    NUM_hygro_aer=num_hygro_aer,          &
    HYGRO_AER=hygro_aer,                  &
    ICE_NUCL=in_aer,                      &
    WTKE=wtke,                            &
    LCLDFRA=lcldfra_chm,                  &
    DROPSOURCE=dropsource,                &
    RAINRATE=rainrate,                    &
    EVAPRATE=evaprate,                    &
#endif
```

The arguments are also passed to mp_gt_driver in the case of Thompson scheme.
Note: these additional arguments are passed to both Thomson Aerosol-Aware and Thompson schemes (CASE (THOMPSONAERO) and CASE (THOMPSON) conditions), but the microphysics diagnostics are only performed for the Thomson Aerosol-Aware scheme.
The new subroutine ccn_calc that diagnozes the CCN spectrum is defined at the end of the module:

```
#ifdef cpl_wrf_chimere
   SUBROUTINE ccn_calc(nbin,sect,num_aer,hygro_aer,      &
                       ccn1,ccn2,ccn3,ccn4,ccn5,ccn6,    &
                       ids,ide, jds,jde, kds,kde,        &
                       ims,ime, jms,jme, kms,kme,        &
                       its,ite, jts,jte, kts,kte         )

    ! I/O variable
    INTEGER,INTENT(IN)                 :: nbin
    REAL,DIMENSION(nbin+1),INTENT(IN) :: sect
    REAL,DIMENSION(ims:ime,kms:kme,jms:jme,nbin),INTENT(IN) :: num_aer
    REAL,DIMENSION(ims:ime,kms:kme,jms:jme),INTENT(IN) :: hygro_aer
    REAL,DIMENSION(ims:ime,kms:kme,jms:jme),INTENT(OUT):: ccn1,ccn2,ccn3, &
                                                          ccn4,ccn5,ccn6
    INTEGER, INTENT(IN)   :: ids,ide, jds,jde, kds,kde, &
```

```
                            ims,ime, jms,jme, kms,kme, &
                            its,ite, jts,jte, kts,kte

    ! Local variable
    INTEGER,PARAMETER :: nsat=6
    REAL,PARAMETER :: ss(nsat)= &  ! supersaturation (%) of the specturm
       (/0.02,0.05,0.1,0.2,0.5,1.0/)
    INTEGER :: i_start,j_start,i_end,j_end
    INTEGER :: i,j,k,l,m
    REAL    :: ak,dc,frac
    REAL,DIMENSION(nsat) :: logss


    ! Surface tension term
    ak = log(1.2*1.e-9)

    ! Supersaturation spectrum
    logss = log(ss*0.01)

    ! Inizialize CCN
    ccn1 = 0.0
    ccn2 = 0.0
    ccn3 = 0.0
    ccn4 = 0.0
    ccn5 = 0.0
    ccn6 = 0.0

    ! CCN diagnostic
    do i=its,ite
      do j=jts,jte
        do k=kts,kte
          do l=1,nsat

            ! Critical activation diameter for each supersaturation level
            dc = ak-log(3.)-2./3.*(logss(l)-log(2.)+0.5*log(hygro_aer(i,k,j)))
            dc = 2.*exp(dc)*1.e6

            do m=1,nbin
              ! Fraction of each section acting as CCN
              frac = 0.0
              if(dc.le.sect(m)) then
                frac = 1.
              elseif(dc.gt.sect(m+1)) then
                frac = 0.
              elseif(dc.gt.sect(m).AND.dc.le.sect(m+1)) then
                frac = log(sect(m+1)/dc)/log(sect(m+1)/sect(m))
              endif

              ! CCN concentration (#/cm3)
              if(frac.gt.0) then
                if(l==1) then
                  ccn1(i,k,j) = ccn1(i,k,j)+frac*num_aer(i,k,j,m)*1.e-6
                elseif(l==2) then
                  ccn2(i,k,j) = ccn2(i,k,j)+frac*num_aer(i,k,j,m)*1.e-6
                elseif(l==3) then
                  ccn3(i,k,j) = ccn3(i,k,j)+frac*num_aer(i,k,j,m)*1.e-6
                elseif(l==4) then
                  ccn4(i,k,j) = ccn4(i,k,j)+frac*num_aer(i,k,j,m)*1.e-6
                elseif(l==5) then
                  ccn5(i,k,j) = ccn5(i,k,j)+frac*num_aer(i,k,j,m)*1.e-6
                elseif(l==6) then
                  ccn6(i,k,j) = ccn6(i,k,j)+frac*num_aer(i,k,j,m)*1.e-6
                endif
              endif
            enddo !m

          enddo !l
        enddo !k
      enddo !j
    enddo !i
   END SUBROUTINE ccn_calc
#endif
```

B.6.2 module_radiation_driver.F

1. The radiation_driver subroutine takes new (optional) arguments related to direct-effects coupling. They are specified right before the progn argument:

```
 ,direct_feed_chimere      &
              ,aod_300,aod_400,aod_999  &
              ,ssa_400,ssa_600          &
              ,g_400,g_600              &
              ,aod_lw1,aod_lw2,aod_lw3,aod_lw4,aod_lw5   &
              ,aod_lw6,aod_lw7,aod_lw8,aod_lw9,aod_lw10  &
              ,aod_lw11,aod_lw12,aod_lw13,aod_lw14       &
              ,aod_lw15,aod_lw16                         &
```

2. These optional arguments can be declared after the similar optional input arguments tauaerlwxx related to WRF/CHEM:

```
 INTEGER,OPTIONAL,INTENT(IN   ) ::   direct_feed_chimere
   REAL, DIMENSION( ims:ime, kms:kme, jms:jme ), OPTIONAL,     &
         INTENT(INOUT   ) ::          aod_300,aod_400,aod_999, &
                                      ssa_400,ssa_600,         &
                                      g_400,g_600,             &
                                      aod_lw1,aod_lw2,aod_lw3,aod_lw4,    &
                                      aod_lw5,aod_lw6,aod_lw7,aod_lw8,    &
                                      aod_lw9,aod_lw10,aod_lw11,aod_lw12, &
                                      aod_lw13,aod_lw14,aod_lw15,aod_lw16
```

3. These variables are then passed over to the RRTMG_LWRAD subroutine, which runs the actual LW radiation model. In the CASE (RRTMG_LWSCHEME) section, right after the #if (WRF_CHEM == 1) section, put the following code:

```
#ifdef cpl_wrf_chimere
                 direct_feed_chimere=direct_feed_chimere,            &
                 aod_lw1=aod_lw1,aod_lw2=aod_lw2,aod_lw3=aod_lw3,aod_lw4=aod_lw4,        &
                 aod_lw5=aod_lw5,aod_lw6=aod_lw6,aod_lw7=aod_lw7,aod_lw8=aod_lw8,        &
                 aod_lw9=aod_lw9,aod_lw10=aod_lw10,aod_lw11=aod_lw11,aod_lw12=aod_lw12,  &
                 aod_lw13=aod_lw13,aod_lw14=aod_lw14,aod_lw15=aod_lw15,aod_lw16=aod_lw16,&
#endif
```

4. Similarly the shortwave radiation model RRTMG_SWRAD also takes the corresponding variables as new input arguments in case of coupling with CHIMERE. In the CASE (RRTMG_SWSCHEME) section, right after the #if (WRF_CHEM == 1) section, insert the following:

```
#ifdef cpl_wrf_chimere
                   direct_feed_chimere=direct_feed_chimere,          &
                   aod_300=aod_300,aod_400=aod_400,aod_999=aod_999,  &
                   ssa_400=ssa_400,ssa_600=ssa_600,                  &
                   g_400=g_400,g_600=g_600,                          &
#endif
```

B.6.3 module_surface_driver.F

1. An additional flag wrf_nemo is imported from module_cpl:

```
 USE module_cpl, ONLY : coupler_on, cpl_rcv, wrf_nemo
```

2. The NEMO-related variables are received only when the wrf_nemo flag is True:

```
 IF ( coupler_on .and. wrf_nemo .and. present(cplmask) .and. present(sst_input) ) THEN
     CALL cpl_rcv( id, 'SST',          &
        &              ids, ide, jds, jde, kds, kde, &
        &              ims, ime, jms, jme, kms, kme, &
        &              ips, ipe, jps, jpe, kps, kpe, &
        &              max_edom, cplmask, SST, SST_INPUT )
        ..............................
```

B.6.4 module_mp_thompson.F

The coupling with CHIMERE was initially developed for WRF 3.7.1 version. However, there were important updates in the Thompson microphysics scheme in WRF 3.8.1. They have been applied in the current version of the coupled model.

This section describes changes in the module_mp_thompson module related to the coupling with CHIMERE, with respect to the 3.8.1 version.
The new arguments of the mp_gt_driver subroutine need to be declared in the arguments list, right after the WRF_CHEM directive:

```
#ifdef cpl_wrf_chimere
       indirect_feed_chimere,n_bin,sect,         &
       num_aer,num_hygro_aer,hygro_aer,ice_nucl, &
       wtke,lcldfra,                             &
       dropsource,rainrate,evaprate,             &
#endif
```

... and n the variables declaration section:

```
#ifdef cpl_wrf_chimere
 INTEGER,INTENT(IN) :: indirect_feed_chimere,n_bin
 REAL,INTENT(IN),DIMENSION(n_bin+1) :: sect
 REAL,DIMENSION(ims:ime,kms:kme,jms:jme),OPTIONAL,INTENT(IN) :: hygro_aer,ice_nucl,wtke,lcldfra
 REAL,DIMENSION(ims:ime,kms:kme,jms:jme,n_bin),OPTIONAL,INTENT(IN) :: num_aer,num_hygro_aer
 REAL,DIMENSION(ims:ime,kms:kme,jms:jme),OPTIONAL,INTENT(OUT) :: dropsource,rainrate,evaprate
#endif
```

The mp_gt_driver subroutine converts the 3D WRF variables to 1D to use inside microphysics schemes. The new variables introduced for the indirect effects coupling thus need to have their 1D equivalents:

```
#ifdef cpl_wrf_chimere
      INTEGER :: nb
      REAL,DIMENSION(kts:kte) :: hygro_aer1d,ice_nucl1d,wtke1d,dropsource1d
      REAL,DIMENSION(kts:kte,n_bin) :: num_aer1d,num_hygro_aer1d
      REAL,DIMENSION(kts:kte) :: rainrate1d,evaprate1d
       REAL,DIMENSION(kts:kte) :: lcldf1d
#endif
```

These variables are computed inside the if (is_aerosol_aware) block, e.g. after the assignment of the nwfa1 variable:

```
#ifdef cpl_wrf_chimere
            if(indirect_feed_chimere==1) then
              do k = kts, kte
                do nb = 1, n_bin
                  num_aer1d(k,nb) = max(num_aer(i,k,j,nb),1.e-20)
                  num_hygro_aer1d(k,nb) = max(num_hygro_aer(i,k,j,nb),1.e-20)
                enddo
                hygro_aer1d(k) = max(hygro_aer(i,k,j),1.e-20)
                ice_nucl1d(k) = max(ice_nucl(i,k,j),1.e-20)
                wtke1d(k) = wtke(i,k,j)
                lcldf1d(k) = lcldfra(i,k,j)
                dropsource1d(k) = 0.0
              enddo
            endif
#endif
```

The microphysics scheme mp_thompson is called with these 1D arrays as new arguments, right before the kts argument:

```
#ifdef cpl_wrf_chimere
                     indirect_feed_chimere,n_bin,sect,num_aer1d,       &
                     num_hygro_aer1d,hygro_aer1d,ice_nucl1d,           &
                     wtke1d,lcldf1d,dropsource1d,rainrate1d,evaprate1d,&
#endif
```

The 3D variables are then restored from their 1D versions updated in the microphysics scheme. Just after the if (is_aerosol_aware) condition that follows the "Reset lowest model level to initial state aerosols..." comments, add the following calculations:

```
#ifdef cpl_wrf_chimere
         do k = kts, kte
            rainrate(i,k,j) = rainrate1d(k)
            evaprate(i,k,j) = evaprate1d(k)
            if(indirect_feed_chimere==1) dropsource(i,k,j) = dropsource1d(k)
         enddo
#endif
```

The mp_thompson subroutine takes the additional coupling-related arguments (1D arrays except those depending on the number of CHIMERE bins, which are 2D), listed right before the kts variable:

```
#ifdef cpl_wrf_chimere
                         indirect_feed_chimere,n_bin,sect,num_aer1d,    &
                         num_hygro_aer1d,                               &
                         hygro_aer1d,ice_nucl1d,wtke1d,lcldf1d,         &
                         dropsource1d,rainrate1d,evaprate1d,            &
#endif
```

The input arguments are declared after the WRF_CHEM "IF" directive:

```
#ifdef cpl_wrf_chimere
      INTEGER,INTENT(IN) :: indirect_feed_chimere,n_bin
      REAL,INTENT(IN),DIMENSION(n_bin+1) :: sect
      REAL,DIMENSION(kts:kte),INTENT(IN) :: hygro_aer1d,ice_nucl1d,wtke1d,lcldf1d
      REAL,DIMENSION(kts:kte,n_bin),INTENT(IN) :: num_aer1d,num_hygro_aer1d
      REAL,DIMENSION(kts:kte),INTENT(OUT) :: dropsource1d,rainrate1d,evaprate1d
#endif
```

And the local variables can be declared further below, e.g., after the diffu, visco, variables:

```
#ifdef cpl_wrf_chimere
      REAL, DIMENSION(kts:kte,n_bin):: num_aer,num_hygro_aer
      REAL, DIMENSION(kts:kte) :: ice_nucl
      REAL, DIMENSION(n_bin)   :: act_frac
#endif
```

They are initialized after those enclosed by the WRF_CHEM directive:

```
#ifdef cpl_wrf_chimere
      do k = kts,kte
         rainrate1d(k) = 0.
         evaprate1d(k) = 0.
      enddo
#endif
```

Local arrays are defined from the input arrays in the do k = kts, kte loop ("Put column of data into local arrays"), e.g., after the nifa(k) assignment:

```
#ifdef cpl_wrf_chimere
         if(indirect_feed_chimere==1) then
            num_aer(k,:) = num_aer1d(k,:)*rho(k)
            num_hygro_aer(k,:) = num_hygro_aer1d(k,:)*rho(k)
            ice_nucl(k) = ice_nucl1d(k)*rho(k)
         endif
#endif
```

The ice nucleation was updated ("BEGIN NEW ICE NUCLEATION" part):

```
          if (dustyIce .AND. is_aerosol_aware) then
#ifdef cpl_wrf_chimere
           if(indirect_feed_chimere==1) then
              xni = iceDeMott(tempc,qvs(k),qvs(k),qvsi(k),rho(k),ice_nucl(k))
           else
              xni = iceDeMott(tempc,qvs(k),qvs(k),qvsi(k),rho(k),nifa(k))
           endif
#else
           xni = iceDeMott(tempc,qvs(k),qvs(k),qvsi(k),rho(k),nifa(k))
#endif
```

And for deposition nucleation of dust/mineral:

```
          if ( (ssati(k).ge. 0.25) .or. (ssatw(k).gt. eps &
                                .and. temp(k).lt.253.15) ) then
           if (dustyIce .AND. is_aerosol_aware) then
#ifdef cpl_wrf_chimere
            if(indirect_feed_chimere==1) then
              xnc = iceDeMott(tempc,qv(k),qvs(k),qvsi(k),rho(k),ice_nucl(k))
            else
              xnc = iceDeMott(tempc,qv(k),qvs(k),qvsi(k),rho(k),nifa(k))
            endif
#else
            xnc = iceDeMott(tempc,qv(k),qvs(k),qvsi(k),rho(k),nifa(k))
#endif
```

As well as for freezing of aqueous aerosols:

```
          if (is_aerosol_aware .AND. homogIce .AND. (xni.le.500.E3)     &
     &                .AND.(temp(k).lt.238).AND.(ssati(k).ge.0.4) ) then
#ifdef cpl_wrf_chimere
            if(indirect_feed_chimere==1) then
              xnc = iceKoop_chm(temp(k),qv(k),qvs(k),num_hygro_aer(k,:),dtsave,sect,n_bin)
            else
              xnc = iceKoop(temp(k),qv(k),qvs(k),nwfa(k), dtsave)
            endif
#else
            xnc = iceKoop(temp(k),qv(k),qvs(k),nwfa(k), dtsave)
#endif
```

In the "Cloud water condensation and evaporation" part, the drop_activation subroutine is called in case of coupling with indirect effects :

```
            if (is_aerosol_aware) then
               xnc = MAX(2., activ_ncloud(temp(k), w1d(k), nwfa(k)))
#ifdef cpl_wrf_chimere
               if(indirect_feed_chimere==1) then
                  call drop_activation(n_bin,sect,num_aer(k,:),hygro_aer1d(k), &
                                       w1d(k),wtke1d(k),lcldf1d(k),            &
                                       temp(k),pres(k),act_frac)
                  xnc=max(2.,sum(act_frac*num_aer(k,:)))
               else
                  xnc = MAX(2., activ_ncloud(temp(k), w1d(k), nwfa(k)))
               endif
#else
               xnc = MAX(2., activ_ncloud(temp(k), w1d(k), nwfa(k)))
#endif
            else
               xnc = Nt_c
            endif
```

And the arrays dropsource1d, evaprate1d, and rainrate1d are assigned:

```
#ifdef cpl_wrf_chimere
          if(indirect_feed_chimere==1) dropsource1D(k) = pnc_wcd(k)
#endif
.................
#ifdef cpl_wrf_chimere
      do k = kts,kte
         evaprate1d(k) = prv_rev(k) - (min(zeroD0,prs_sde(k)) + &
                                       min(zeroD0,prg_gde(k)))
         rainrate1d(k) = prr_wau(k) + prr_rcw(k) + prs_scw(k) + &
                                      prg_scw(k) + prs_iau(k) + &
                                      prg_gcw(k) + prs_sci(k) + &
                                      pri_rci(k)
      enddo
#endif
```

The freezing rate from the [Koop et al., 2001] parameterization has a special treatment for CHIMERE:

```
#ifdef cpl_wrf_chimere
!!! Special case for WRF-CHIMERE

      real function iceKoop_chm(temp, qv, qvs, naero, dt, sect, nbin)
      implicit none

      INTEGER, INTENT(IN) :: nbin
      REAL, INTENT(IN):: temp, qv, qvs, DT
      REAL, INTENT(IN):: naero(nbin), sect(nbin+1)
      INTEGER :: i
      REAL:: mu_diff, a_w_i, delta_aw, log_J_rate, J_rate, prob_h, satw
      REAL:: volume,rad,xni

      xni = 0.0
      satw = qv/qvs
      mu_diff    = 210368.0 + (131.438*temp) - (3.32373E6/temp) &
     &           - (41729.1*alog(temp))
      a_w_i      = exp(mu_diff/(R_uni*temp))
      delta_aw   = satw - a_w_i
      log_J_rate = -906.7 + (8502.0*delta_aw) &
     &           - (26924.0*delta_aw*delta_aw) &
     &           + (29180.0*delta_aw*delta_aw*delta_aw)
      log_J_rate = MIN(20.0, log_J_rate)
      ! FROM 3.8.1
      !J_rate     = 0.01*(10.**log_J_rate) ! cm-3 s-1
      J_rate     = 10.**log_J_rate  !cm-3 s-1
```

```
      xni = 0.0
      do i=1,nbin
        if(sect(i).ge.0.1) then
         rad = (sect(i)+sect(i+1))/4.
         volume = 4./3.*PI*(rad*1.e-4)**3  ! cm3
         prob_h     = MIN(1.-exp(-J_rate*volume*DT), 1.)
         if (prob_h .gt. 0.) then
           xni     = xni+prob_h*naero(i)
         endif
        endif
      enddo

      xni = MIN(xni,1000.E3)
      iceKoop_chm = MAX(0.0, xni)

      end FUNCTION iceKoop_chm
#endif
```

The drop_activation function and the error function from the Numerical Recipes used therein are implemented at the end of the module:

```
#ifdef cpl_wrf_chimere
   SUBROUTINE drop_activation(n_bin,sect,num_aer,hygros_aer, &
                              wmean,sigm,lcldf,temp,pres,frac)

   USE module_model_constants, only: g,rhowater,xlv,cp,r_d,r_v

   IMPLICIT NONE

   INTEGER,INTENT(IN) :: n_bin
   REAL,INTENT(IN)    :: wmean,sigm,lcldf,hygros_aer,temp,pres
   REAL,DIMENSION(n_bin+1),INTENT(IN):: sect
   REAL,DIMENSION(n_bin),INTENT(IN)  :: num_aer
   REAL,DIMENSION(n_bin),INTENT(OUT) :: frac

   ! Local variables
   real,parameter :: eps = 0.622      ! mw/mair
   real,parameter :: surften = 0.076  ! Surface tension of water w/respect to air (N/m)
   real,parameter :: p0 = 1013.25e2   ! Reference pressure (Pa)
   real,parameter :: pi = 4.*atan(1.)
   real,parameter :: log3 = log(3.)
   real,parameter :: log2 = log(2.)
   real,parameter :: wmaxf = 50.
   integer,parameter :: nwmax = 200
   real :: alfa,gamma1,ak,gc,esw
   real :: press,diff,conduct
   real :: ntot,zeta,eta
   real :: sse   ! Effective critical supersaturation
   real :: ssmax ! Maximum supersaturation
   real :: w,wmax,wold,dw,wmin
   real :: gauss_old,gauss_new,sumgauss
   real :: frac_mean,integ
   real :: dc ! Critical activation diameter
   real,dimension(n_bin) :: rm ! Central section radius
   real,dimension(n_bin) :: sm ! Effective supersaturation
   real,dimension(n_bin) :: frac_new,frac_old,sumfrac

   ! Auxiliar variable
   integer :: i,j
   real :: tmp1,tmp2
   real :: logak,loghy,normc

   frac = 0.0
   ntot = sum(num_aer)
   if(ntot.le.1.e-20) return

   !!!! DEFINE CONSTANTS

   ! log of hygroscopicity
   loghy = log(hygros_aer)

   ! Gaussian normalization constant
   normc = sqrt(2.*pi)*sigm

   ! Surface tension term
   ak=2.*surften/(r_v*temp*rhowater)
   logak = log(ak)

   ! Vapor diffusivity
   diff=0.211e-4*(p0/pres)*(temp/t_0)**1.94
```

```
! Thermal conductivity
conduct=(5.69+0.017*(temp-t_0))*4.186d2*1e-5 ! convert to J/m/s/deg

! Saturation vapour pressure
esw = 611.*exp(17.27*(temp-t_0)/(temp-35.86))

! Growth coefficient
tmp1 = (rhowater*R_v*temp)/(esw*diff) +   &
       (xlv*rhowater)/(conduct*temp)*(xlv/(R_v*temp)-1.0)
gc = 1.0/tmp1

! Parameterization parameters
alfa = g*(xlv/(cp*R_v*temp**2d0)-1.0/(R_d*temp))
gamma1 = press/esw/eps+xlv**2d0/(cp*R_v*temp**2d0)

! Central section radius
do i=1,n_bin
  rm(i) = (sect(i)+sect(i+1))/2.0
enddo
rm = rm/2.*1.e-6

! Particle critical supersaturation at rm
sm = 2.0/sqrt(hygros_aer)*(ak/(3.0*rm))**1.5

! Effective critical supersatiration
sse = sum(num_aer/(sm**(2./3.)))
sse = (ntot/sse)**1.5

! Set the parameters of the updraft spectrum
wmax = min(wmaxf,wmean+3.*sigm)
wmin = max(0.0,wmean-3.*sigm)
dw = max(0.01,wmax/float(nwmax-1))
w = wmin+dw
wold = wmin

!!! INITIALIZE THE INTEGRAL
sumfrac = 0.0
frac_old = 0.0
if(wmin.gt.0.0) then

 ! Non-dimensional parameters
 tmp1 = alfa*wmin/gc
 zeta = 2.0/3.0*ak*sqrt(tmp1)
 if(ntot.gt.1.e-10) then
   eta = 1./(2.*pi*rhowater*gamma1*ntot)*tmp1**1.5
   eta = max(eta,1.e-20)
 else
   eta = 1.e10
 endif

 ! Supersaturation maximum
 tmp1 = 0.5*(zeta/eta)**1.5 + (sse*sse/(eta+3.0*zeta))**0.75
 ssmax = sse/sqrt(tmp1)

 ! Critical activation diameter
 dc = logak-log3-2./3.*(log(ssmax)-log2+0.5*loghy)
 dc = 2.*exp(dc)*1.e6

 ! Activated fraction
 do i=1,n_bin
    if(dc.le.sect(i)) then
      frac_old(i) = 1.
    elseif(dc.gt.sect(i+1)) then
      frac_old(i) = 0.
    elseif(dc.gt.sect(i).AND.dc.le.sect(i+1)) then
      frac_old(i) = log(sect(i+1)/dc)/log(sect(i+1)/sect(i))
    endif
 enddo
endif

gauss_old = (wold-wmean)/sigm
gauss_old = exp(-0.5d0*gauss_old*gauss_old)

!!!! MAIN LOOP FOR INTEGRAL
sumfrac = 0.0
!sumgauss = 0.0

do i=1,nwmax

 ! Non-dimensional parameters
 tmp1 = alfa*w/gc
```

```
    zeta = 2.0/3.0*ak*sqrt(tmp1)
    if(ntot.gt.1.e-10) then
     eta = 1./(2.*pi*rhowater*gamma1*ntot)*tmp1**1.5
     eta = max(eta,1.e-20)
    else
     eta = 1.e10
    endif

    ! Supersaturation maximum
    tmp1 = 0.5*(zeta/eta)**1.5 + (sse*sse/(eta+3d0*zeta))**0.75d0
    ssmax = sse/sqrt(tmp1)

    ! Critical activation diameter
    dc = logak-log3-2./3.*(log(ssmax)-log2+0.5*loghy)
    dc = 2.*exp(dc)*1.e6

    ! Activated fraction
    do j=1,n_bin
       if(dc.le.sect(j)) then
         frac_new(j) = 1.
       elseif(dc.gt.sect(j+1)) then
         frac_new(j) = 0.
       elseif(dc.gt.sect(j).AND.dc.le.sect(j+1)) then
         frac_new(j) = log(sect(j+1)/dc)/log(sect(j+1)/sect(j))
       endif
    enddo

    ! Sum up the integral
    ! Gauss function
    gauss_new = (w-wmean)/sigm
    gauss_new = exp(-0.5d0*gauss_new*gauss_new)
    do j=1,n_bin
      sumfrac(j) = sumfrac(j)+0.5*(frac_new(j)*gauss_new+frac_old(j)*gauss_old)*dw
    enddo
    !sumgauss = sumgauss+0.5*(gauss_new+gauss_old)*dw

    ! Initialize the next iteraction
    wold = w
    frac_old = frac_new
    gauss_old = gauss_new

    if(i.gt.1.and.w.gt.wmax) go to 1001
    w = w+dw
   enddo

1001 CONTINUE

   tmp1 = (w-wmean)/sqrt(2.)/sigm
   tmp2 = (wmaxf-wmean)/sqrt(2.)/sigm
   integ = 0.5*normc*(erfc_num_recipes(tmp1)-erfc_num_recipes(tmp2))
   !sumgauss = sumgauss+integ
   do i=1,n_bin
     sumfrac(i) = sumfrac(i)+frac_new(i)*integ
     !frac(i) = sumfrac(i)/sumgauss
     frac(i) = sumfrac(i)/normc
     frac(i) = lcldf*frac(i)
     frac(i) = min(frac(i),0.99)
     frac(i) = max(frac(i),0.0)
   enddo

   return

   END SUBROUTINE drop_activation

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

   real function erfc_num_recipes( x )
!
!   from press et al, numerical recipes, 1990, page 164
!
     implicit none
     real x
     double precision erfc_dbl, dum, t, zz

     zz = abs(x)
     t = 1.0/(1.0 + 0.5*zz)

     dum =  ( -zz*zz - 1.26551223 + t*(1.00002368 + t*(0.37409196 + &
       t*(0.09678418 + t*(-0.18628806 + t*(0.27886807 +   &
                                       t*(-1.13520398 +   &
       t*(1.48851587 + t*(-0.82215223 + t*0.17087277 ))))))))))
```

```
     erfc_dbl = t * exp(dum)
     if (x .lt. 0.0) erfc_dbl = 2.0d0 - erfc_dbl

     erfc_num_recipes = erfc_dbl

     return
   end function erfc_num_recipes
#endif
```

B.6.5 module_ra_rrtmg_sw.F

The coupling flags and the cpl_rcv subroutine need to be imported:

```
#ifdef cpl_wrf_chimere
use module_cpl, ONLY : coupler_on, wrf_chimere, cpl_rcv
#endif
```

The RRTMG_SWRAD subroutine takes additional arguments: the direct effects coupling flag, short-wave optical depths, single-scattering albedos, and assymetry factors at the predefined wavelengths, right after the progn argument:

```
direct_feed_chimere,                        &
aod_300,aod_400,aod_999,                    &
ssa_400,ssa_600,                            &
g_400,g_600,
```

These variables are optional; they can be declared e.g. right after the asyaer3d_sw, etc. optional input arguments related to WRF/CHEM:

```
!If CHIMERE
  integer, optional, intent(in) ::   direct_feed_chimere
  real, dimension(ims:ime,kms:kme,jms:jme), optional, intent(in) ::   &
                                     aod_300,aod_400,aod_999,  &
                                     ssa_400,ssa_600,          &
                                     g_400,g_600
```

We then need to check if these optional variables have been provided in case of the direct effects coupling, in a similar way as done for WRF/CHEM. Just after the #if (WRF_CHEM == 1) block, add a similar one for CHIMERE:

```
#ifdef cpl_wrf_chimere
        IF (coupler_on.AND.wrf_chimere.AND.direct_feed_chimere==1) THEN ! If CHIMERE aerosol feedback
        IF ( .NOT. &
        ( PRESENT(aod_300) .AND. &
          PRESENT(aod_400) .AND. &
          PRESENT(aod_999) .AND. &
          PRESENT(ssa_400) .AND. &
          PRESENT(ssa_600) .AND. &
          PRESENT(g_400)   .AND. &
          PRESENT(g_600) ) ) THEN
        CALL wrf_error_fatal  &
        ('RRTMG-SW Warning: missing fields required for aerosol radiation' )
        ENDIF
        ENDIF
#endif
```

The computations follow the same method as for WRF/CHEM and can be implemented right afther the WRF/CHEM block that starts with:

```
#if ( WRF_CHEM == 1 )
   IF ( AER_RA_FEEDBACK == 1) then
```

Just after the closing #endif put the following code:

```
#ifdef cpl_wrf_chimere
    IF (coupler_on.AND.wrf_chimere.AND.direct_feed_chimere==1) THEN ! If CHIMERE aerosol feedback

        !!! FOLLOWING THE SAME METHOD USED FOR WRF/CHEM
        do nb = 1, nbndsw
         wavemid(nb)=0.5*(wavemin(nb)+wavemax(nb))  ! um
        do k = kts,kte      !wig

! convert optical properties at 300,400,600, and 999 to conform to the
! band wavelengths
! tauaer - use angstrom exponent
```

```
        if(aod_300(i,k,j).gt.thresh .and. aod_999(i,k,j).gt.thresh) then
           ang=alog(aod_300(i,k,j)/aod_999(i,k,j))/alog(999./300.)
           tauaer(ncol,k,nb)=aod_400(i,k,j)*(0.4/wavemid(nb))**ang
           !tauaer(ncol,k,nb)=tauaer600(i,k,j)*(0.6/wavemid(nb))**ang
           !if (i==30.and.j==49.and.k==2.and.nb==12) then
           ! write(0,*) 'TAU from 600 vs 400 in RRTMG',tauaer600(i,k,j),tauaer400(i,k,j)
           ! print*, 'TAU from 600 vs 400 in RRTMG',tauaer600(i,k,j),tauaer400(i,k,j)
           ! write(0,*) tauaer600(i,k,j)*(0.6/wavemid(nb))**ang,tauaer400(i,k,j)*(0.4/wavemid(nb))**ang
           ! print*, tauaer600(i,k,j)*(0.6/wavemid(nb))**ang,tauaer400(i,k,j)*(0.4/wavemid(nb))**ang
           !endif
! ssa - linear interpolation; extrapolation
           slope=(ssa_600(i,k,j)-ssa_400(i,k,j))/.2
           ssaaer(ncol,k,nb) = slope*(wavemid(nb)-.6)+ssa_600(i,k,j)
           if(ssaaer(ncol,k,nb).lt.0.4) ssaaer(ncol,k,nb)=0.4
           if(ssaaer(ncol,k,nb).ge.1.0) ssaaer(ncol,k,nb)=1.0
! g - linear interpolation;extrapolation
           slope=(g_600(i,k,j)-g_400(i,k,j))/.2
           asmaer(ncol,k,nb) = slope*(wavemid(nb)-.6)+g_600(i,k,j) ! notice reversed varaibles
           if(asmaer(ncol,k,nb).lt.0.5) asmaer(ncol,k,nb)=0.5
           if(asmaer(ncol,k,nb).ge.1.0) asmaer(ncol,k,nb)=1.0
         endif
      end do ! k
      end do ! nb

      !wig beg
      do nb = 1, nbndsw
         slope = 0.  !use slope as a sum holder
         do k = kts,kte
            slope = slope + tauaer(ncol,k,nb)
         end do
         if( slope < 0. ) then
            write(msg,'("ERROR: Negative total optical depth of ",f8.2," at point i,j,nb=",3i5)')&
            &                                                                  slope,i,j,nb
            call wrf_error_fatal(msg)
         else if( slope > 6. ) then
            call wrf_message("--------------------------")
            write(msg,'("WARNING: Large total sw optical depth of ",f8.2," at point i,j,nb=",3i5)') &
            &                                                                    slope,i,j,nb
            call wrf_message(msg)
            !call wrf_message("Diagnostics 1: k, tauaer300, tauaer400, tauaer600, tauaer999, tauaer")
            do k=kts,kte
               !write(msg,'(i4,5f8.2)') k, tauaer300(i,k,j), tauaer400(i,k,j), &
               !     tauaer600(i,k,j), tauaer999(i,k,j),tauaer(ncol,k,nb)
               !call wrf_message(msg)
               !czhao set an up-limit here to avoid segmentation fault
               !from extreme AOD
               tauaer(ncol,k,nb)=tauaer(ncol,k,nb)*6.0/slope
            end do

            !call wrf_message("Diagnostics 2: k, gaer300, gaer400, gaer600, gaer999")
            !do k=kts,kte
            !   write(msg,'(i4,4f8.2)') k, gaer300(i,k,j), gaer400(i,k,j), &
            !        gaer600(i,k,j), gaer999(i,k,j)
            !   call wrf_message(msg)
            !end do

            !call wrf_message("Diagnostics 3: k, waer300, waer400, waer600, waer999")
            !do k=kts,kte
            !   write(msg,'(i4,4f8.2)') k, waer300(i,k,j), waer400(i,k,j), &
            !        waer600(i,k,j), waer999(i,k,j)
            !   call wrf_message(msg)
            !end do

            !call wrf_message("Diagnostics 4: k, ssaal, asyal, taual")
            !do k=kts-1,kte
            !   write(msg,'(i4,3f8.2)') k, ssaaer(i,k,nb), asmaer(i,k,nb), tauaer(i,k,nb)
            !   call wrf_message(msg)
            !end do
            call wrf_message("--------------------------")
         endif
      enddo  ! nb
    ENDIF !CHIMERE aerosol feedback
#endif
```

B.6.6 module_ra_rrtmg_lw.F

The RRTMG_LWRAD subroutine takes new additional arguments: the direct effects coupling flag and longwave optical depths used for direct effect calculations, right after the progn argument:

```
direct_feed_chimere,                          &
aod_lw1,aod_lw2,aod_lw3,aod_lw4,aod_lw5,      &
aod_lw6,aod_lw7,aod_lw8,aod_lw9,aod_lw10,     &
aod_lw11,aod_lw12,aod_lw13,aod_lw14,          &
aod_lw15,aod_lw16,
```

These variables are optional and can be declared e.g. right before the aer_ra_feedback variable:

```
!If CHIMERE (Optional)
  INTEGER, optional, intent(in) ::   direct_feed_chimere
  REAL, dimension(ims:ime,kms:kme,jms:jme), optional, intent(in) ::   &
                                        aod_lw1,aod_lw2,aod_lw3,aod_lw4,     &
                                        aod_lw5,aod_lw6,aod_lw7,aod_lw8,     &
                                        aod_lw9,aod_lw10,aod_lw11,aod_lw12, &
                                        aod_lw13,aod_lw14,aod_lw15,aod_lw16
```

New variables Angstrom exponent and the wavelenghtes used for the LW radiation scheme also need to be defined, e.g. right before the #if (WRF_CHEM == 1) directive:

```
!! IF CHIMERE
   REAL :: ang    !Angstrom exponent
   REAL, DIMENSION(nbndlw) :: lambda !Length wawes of RRTMGLW
   DATA lambda /55.6,23.5,17.7,15.0,13.2,11.1,9.7,8.8,     &
                7.8, 7.0, 6.1, 5.2, 4.6, 4.3,4.0,3.4/
```

We then need to check if the optional AOD variables have been provided in case of the direct effects coupling, in a similar way as done for WRF/CHEM. Just after the #if (WRF_CHEM == 1) block, add a similar one for CHIMERE:

```
#ifdef cpl_wrf_chimere
   IF (direct_feed_chimere==1) THEN
   IF ( .NOT. &
      ( PRESENT(aod_lw1)  .AND. &
        PRESENT(aod_lw2)  .AND. &
        PRESENT(aod_lw3)  .AND. &
        PRESENT(aod_lw4)  .AND. &
        PRESENT(aod_lw5)  .AND. &
        PRESENT(aod_lw6)  .AND. &
        PRESENT(aod_lw7)  .AND. &
        PRESENT(aod_lw8)  .AND. &
        PRESENT(aod_lw9)  .AND. &
        PRESENT(aod_lw10)  .AND. &
        PRESENT(aod_lw11)  .AND. &
        PRESENT(aod_lw12)  .AND. &
        PRESENT(aod_lw13)  .AND. &
        PRESENT(aod_lw14)  .AND. &
        PRESENT(aod_lw15)  .AND. &
        PRESENT(aod_lw16) ) ) THEN
      CALL wrf_error_fatal  &
      ('RRTMG-LW Warning: missing fields required for aerosol radiation' )
   ENDIF
   ENDIF
#endif
```

The WRF aerosol optical depths tauaer are assigned at the end of the section "Aerosol effects. Added aerosol feedbacks from Chem , 03/2010 -czhao", right before calling the RRTMG longwave radiation model rrtmg_lw:

```
#ifdef cpl_wrf_chimere
IF (direct_feed_chimere==1) THEN
  do k = kts,kte
     if(aod_lw1(i,k,j).gt.thresh .and. aod_lw16(i,k,j).gt.thresh) then
        tauaer(ncol,k,1)  = aod_lw1(i,k,j)
        tauaer(ncol,k,2)  = aod_lw2(i,k,j)
        tauaer(ncol,k,3)  = aod_lw3(i,k,j)
        tauaer(ncol,k,4)  = aod_lw4(i,k,j)
        tauaer(ncol,k,5)  = aod_lw5(i,k,j)
        tauaer(ncol,k,6)  = aod_lw6(i,k,j)
        tauaer(ncol,k,7)  = aod_lw7(i,k,j)
        tauaer(ncol,k,8)  = aod_lw8(i,k,j)
        tauaer(ncol,k,9)  = aod_lw9(i,k,j)
        tauaer(ncol,k,10) = aod_lw10(i,k,j)
        tauaer(ncol,k,11) = aod_lw11(i,k,j)
        tauaer(ncol,k,12) = aod_lw12(i,k,j)
        tauaer(ncol,k,13) = aod_lw13(i,k,j)
        tauaer(ncol,k,14) = aod_lw14(i,k,j)
        tauaer(ncol,k,15) = aod_lw15(i,k,j)
        tauaer(ncol,k,16) = aod_lw16(i,k,j)
     endif
```

```
  enddo

 do nb = 1, nbndlw
     slope = 0.  !use slope as a sum holder
     do k = kts,kte
        slope = slope + tauaer(ncol,k,nb)
     end do
     if( slope < 0. ) then
        write(msg,'("ERROR: Negative total lw optical depth of ",f8.2," at point i,j,nb=",3i5)') &
        &                                                                    slope,i,j,nb
        call wrf_error_fatal(msg)
     else if( slope > 5. ) then
        call wrf_message("--------------------------")
        write(msg,'("WARNING: Large total lw optical depth of ",f8.2," at point i,j,nb=",3i5)') &
        &                                                                    slope,i,j,nb
        call wrf_message(msg)

        call wrf_message("Diagnostics 1: k, aod_lw1, aod_lw16")
        do k=kts,kte
           write(msg,'(i4,2f8.2)') k, aod_lw1(i,k,j), aod_lw16(i,k,j)
           call wrf_message(msg)
        end do
        call wrf_message("--------------------------")
     endif
  enddo  ! nb
ENDIF
#endif
```

C Example of an OASIS namcouple file for the WRF/CHIMERE coupling

```
# This is a typical input file for OASIS3-MCT.
# Keywords used in previous versions of OASIS3
# but now obsolete are marked "Not used"
# Don't hesitate to ask precisions or make suggestions (oasishelp@cerfacs.fr).
#
# Any line beginning with # is ignored. Blank lines are not allowed.
#
#############################################################################
 $NFIELDS
# The number of fields described in the second part of the namcouple.
#
51
 $END
#############################################################################
 $NBMODEL
# The number of models in this experiment + their names (6 characters)
#
2 chimer wrfexe
 $END
#############################################################################
 $RUNTIME
# The total simulated time for this run in seconds
#
86400
 $END
#############################################################################
 $NLOGPRT
# Amount of information written to OASIS3-MCT log files (see User Guide)
0  0
 $END
#############################################################################
 $STRINGS
#
# The above variables are the general parameters for the experiment.
# Everything below has to do with the fields being exchanged.
#
#####################################################
#
# Field 1: model1 to model2
#
#   First line:
# 1) and 2) Symbolic names for the field before and after interpolation
#           (8 characters maximum)
# 3) Index of field in cf_name_table.txt
# 4) Exchange frequency for the field in seconds (here 1 day)
# 5) Number of analysis to be performed
# 6) Restart input NetCDF file names
# 7) Field status: EXPORTED, EXPOUT, INPUT, OUTPUT
#    (For OASIS3-MCT, IGNORED/IGNOUT are converted to EXPORTED/EXPOUT)
#
#   Second line:
# 1)-2) and 3)-4) Source and target grids first and 2nd dimensions (optional)
# 5) and 6) Source and target locator prefix (4 characters)
# 7) LAG index if needed
# 8) SEQ index if needed
#
#   Third line:
# Overlap (P or R) and nbr of overlap grid points for source and target grids.
#
# List of analyses
#
SND_PSFC RCV_PSFC 1 900 1 not_used EXPORTED
wrf1 chm1
R 0 R 0
LOCTRANS
INSTANT
```

```
#
SND_SWDOWN RCV_SWDOWN 1 900 1 not_used EXPORTED
wrf1 chm1
R 0 R 0
LOCTRANS
INSTANT
#
SND_GLW RCV_GLW 1 900 1 not_used EXPORTED
wrf1 chm1
R 0 R 0
LOCTRANS
INSTANT
#
SND_Q2 RCV_Q2 1 900 1 not_used EXPORTED
wrf1 chm1
R 0 R 0
LOCTRANS
INSTANT
#
SND_RAINC RCV_RAINC 1 900 1 not_used EXPORTED
wrf1 chm1
R 0 R 0
LOCTRANS
INSTANT
#
SND_RAINNC RCV_RAINNC 1 900 1 not_used EXPORTED
wrf1 chm1
R 0 R 0
LOCTRANS
INSTANT
#
SND_HFX RCV_HFX 1 900 1 not_used EXPORTED
wrf1 chm1
R 0 R 0
LOCTRANS
INSTANT
#
SND_LH RCV_LH 1 900 1 not_used EXPORTED
wrf1 chm1
R 0 R 0
LOCTRANS
INSTANT
#
SND_UST RCV_UST 1 900 1 not_used EXPORTED
wrf1 chm1
R 0 R 0
LOCTRANS
INSTANT
#
SND_T2 RCV_T2 1 900 1 not_used EXPORTED
wrf1 chm1
R 0 R 0
LOCTRANS
INSTANT
#
SND_PBLH RCV_PBLH 1 900 1 not_used EXPORTED
wrf1 chm1
R 0 R 0
LOCTRANS
INSTANT
#
SND_SNOW RCV_SNOW 1 900 1 not_used EXPORTED
wrf1 chm1
R 0 R 0
LOCTRANS
INSTANT
#
SND_SNOWH RCV_SNOWH 1 900 1 not_used EXPORTED
wrf1 chm1
R 0 R 0
LOCTRANS
INSTANT
#
SND_U10 RCV_U10 1 900 1 not_used EXPORTED
wrf1 chm1
R 0 R 0
LOCTRANS
INSTANT
#
```

```
SND_V10 RCV_V10 1 900 1 not_used EXPORTED
wrf1 chm1
R 0 R 0
LOCTRANS
INSTANT
#
SND_SMOIS RCV_SMOIS 1 900 1 not_used EXPORTED
wrf1 chm1
R 0 R 0
LOCTRANS
INSTANT
#
SND_SEAICE RCV_SEAICE 1 900 1 not_used EXPORTED
wrf1 chm1
R 0 R 0
LOCTRANS
INSTANT
#
SND_PB RCV_PB 1 900 1 not_used EXPORTED
wrf2 chm2
R 0 R 0
LOCTRANS
INSTANT
#
SND_P RCV_P 1 900 1 not_used EXPORTED
wrf2 chm2
R 0 R 0
LOCTRANS
INSTANT
#
SND_T RCV_T 1 900 1 not_used EXPORTED
wrf2 chm2
R 0 R 0
LOCTRANS
INSTANT
#
SND_QCLOUD RCV_QCLOUD 1 900 1 not_used EXPORTED
wrf2 chm2
R 0 R 0
LOCTRANS
INSTANT
#
SND_QRAIN RCV_QRAIN 1 900 1 not_used EXPORTED
wrf2 chm2
R 0 R 0
LOCTRANS
INSTANT
#
SND_QVAPOR RCV_QVAPOR 1 900 1 not_used EXPORTED
wrf2 chm2
R 0 R 0
LOCTRANS
INSTANT
#
SND_QICE RCV_QICE 1 900 1 not_used EXPORTED
wrf2 chm2
R 0 R 0
LOCTRANS
INSTANT
#
SND_PH RCV_PH 1 900 1 not_used EXPORTED
wrf3 chm3
R 0 R 0
LOCTRANS
INSTANT
#
SND_PHB RCV_PHB 1 900 1 not_used EXPORTED
wrf3 chm3
R 0 R 0
LOCTRANS
INSTANT
#
SND_U RCV_U 1 900 1 not_used EXPORTED
wrf4 chm4
R 0 R 0
LOCTRANS
INSTANT
#
SND_V RCV_V 1 900 1 not_used EXPORTED
```

```
wrf5 chm5
R 0 R 0
LOCTRANS
INSTANT
#
SND_AOD_300nm RCV_AOD_300nm 1 1800 1 not_used EXPORTED
chm2 wrf2
R 0 R 0
LOCTRANS
INSTANT
#
SND_AOD_400nm RCV_AOD_400nm 1 1800 1 not_used EXPORTED
chm2 wrf2
R 0 R 0
LOCTRANS
INSTANT
#
SND_AOD_999nm RCV_AOD_999nm 1 1800 1 not_used EXPORTED
chm2 wrf2
R 0 R 0
LOCTRANS
INSTANT
#
SND_albedo_400nm RCV_albedo_400nm 1 1800 1 not_used EXPORTED
chm2 wrf2
R 0 R 0
LOCTRANS
INSTANT
#
SND_albedo_600nm RCV_albedo_600nm 1 1800 1 not_used EXPORTED
chm2 wrf2
R 0 R 0
LOCTRANS
INSTANT
#
SND_asymmetry_factor_400nm RCV_asymmetry_factor_400nm 1 1800 1 not_used EXPORTED
chm2 wrf2
R 0 R 0
LOCTRANS
INSTANT
#
SND_asymmetry_factor_600nm RCV_asymmetry_factor_600nm 1 1800 1 not_used EXPORTED
chm2 wrf2
R 0 R 0
LOCTRANS
INSTANT
#
SND_AOD_LW1 RCV_AOD_LW1 1 1800 1 not_used EXPORTED
chm2 wrf2
R 0 R 0
LOCTRANS
INSTANT
#
SND_AOD_LW2 RCV_AOD_LW2 1 1800 1 not_used EXPORTED
chm2 wrf2
R 0 R 0
LOCTRANS
INSTANT
#
SND_AOD_LW3 RCV_AOD_LW3 1 1800 1 not_used EXPORTED
chm2 wrf2
R 0 R 0
LOCTRANS
INSTANT
#
SND_AOD_LW4 RCV_AOD_LW4 1 1800 1 not_used EXPORTED
chm2 wrf2
R 0 R 0
LOCTRANS
INSTANT
#
SND_AOD_LW5 RCV_AOD_LW5 1 1800 1 not_used EXPORTED
chm2 wrf2
R 0 R 0
LOCTRANS
INSTANT
#
SND_AOD_LW6 RCV_AOD_LW6 1 1800 1 not_used EXPORTED
chm2 wrf2
```

```
R 0 R 0
LOCTRANS
INSTANT
#
SND_AOD_LW7 RCV_AOD_LW7 1 1800 1 not_used EXPORTED
chm2 wrf2
R 0 R 0
LOCTRANS
INSTANT
#
SND_AOD_LW8 RCV_AOD_LW8 1 1800 1 not_used EXPORTED
chm2 wrf2
R 0 R 0
LOCTRANS
INSTANT
#
SND_AOD_LW9 RCV_AOD_LW9 1 1800 1 not_used EXPORTED
chm2 wrf2
R 0 R 0
LOCTRANS
INSTANT
#
SND_AOD_LW10 RCV_AOD_LW10 1 1800 1 not_used EXPORTED
chm2 wrf2
R 0 R 0
LOCTRANS
INSTANT
#
SND_AOD_LW11 RCV_AOD_LW11 1 1800 1 not_used EXPORTED
chm2 wrf2
R 0 R 0
LOCTRANS
INSTANT
#
SND_AOD_LW12 RCV_AOD_LW12 1 1800 1 not_used EXPORTED
chm2 wrf2
R 0 R 0
LOCTRANS
INSTANT
#
SND_AOD_LW13 RCV_AOD_LW13 1 1800 1 not_used EXPORTED
chm2 wrf2
R 0 R 0
LOCTRANS
INSTANT
#
SND_AOD_LW14 RCV_AOD_LW14 1 1800 1 not_used EXPORTED
chm2 wrf2
R 0 R 0
LOCTRANS
INSTANT
#
SND_AOD_LW15 RCV_AOD_LW15 1 1800 1 not_used EXPORTED
chm2 wrf2
R 0 R 0
LOCTRANS
INSTANT
#
SND_AOD_LW16 RCV_AOD_LW16 1 1800 1 not_used EXPORTED
chm2 wrf2
R 0 R 0
LOCTRANS
INSTANT
#####################################################
#
 $END
```

D How to install MPI and NetCDF

D.1 Libraries to install

The CHIMERE model has been tested on GNU/Linux systems with Open MPI message passing libraries. The model requires several numerical tools, compilers and libraries. The system has to be installed with GNU **bash** Bourne shell, **awk** and **make**. It is also necessary to have python libraries. Note that ALL libraries must be compiled with the same compiler as the CHIMERE model. The model was developed and tested using the softwares versions as described in Table D.1 .

Software	URL	Tested versions
Fortran/C/C++ compilers		
GNU	https://gcc.gnu.org/	GCC 4.7.2
INTEL	https://software.intel.com/parallel-studio-xe	composer_xe_2013 parallel_studio_xe_2017
MPI		
Open MPI	https://www.open-mpi.org/software/ompi/	openmpi-2.0.1 openmpi-2.0.4
Librairies		
HDF5	https://support.hdfgroup.org/ftp/HDF5/releases/	HDF5 1.8.20
Unidata NetCDF-C	https://github.com/Unidata/netcdf-c/releases	netCDF-C 4.6.0
Unidata NetCDF-Fortran	https://github.com/Unidata/netcdf-fortran/releases	netCDF-Fortran 4.4.4
Blitz	https://sourceforge.net/projects/blitz	blitz-0.10
Grib API	https://confluence.ecmwf.int/display/GRIB/Home	grib-api 1.12.0 grib-api 1.26.1
Jasper	https://www.ece.uvic.ca/~frodo/jasper/#download	JasPer 1.900.1

Table D.1: *List of required external libraries.*

The model has been tested with gfortran and intel compilers. If these compilers are not installed by default on your computer, install one of them to be able to compile the NetCDF libraries and the CHIMERE model.
For any software the installation procedure is pretty similar :

- dowload the code on the website
- unpack it and go to the source directory
- configure the installation and then install

General commands look like that

```
tar xvfz my_sorftware.tar.z
cd my_softwar
./configure [options]
make
make install
```

In the following, we try to give some hints on the *[options]* needed to configure the installation of MPI, and NetCDF.

D.2 Installation of MPI

This section presents how to easily install MPI. We consider here that all computers are now under 64 bit systems.
For more informations please refer to `https://www.open-mpi.org/faq/?category=building#easy-build`.

Depending on your choice, GNU (gfortran, gcc) or INTEl (ifort, icc), you need to accordingly specify the various compilers. In the following, you need to replace :

path_to_your_compilers : location of your compiler installation

gcc_or_icc / g++_or_icpc / gfortran_or_ifort : gcc/g++/gfortran for GNU and icc/icpc/ifort for INTEL

path_to_your_MPI : final location of MPI directory with executables, libraries and include files

```
./configure CC=path_to_your_compilers/gcc_or_icc  \
            CXX=path_to_your_compilers/g++_or_icpc \
            FC=path_to_your_compilers/gfortran_or_ifort \
            CFLAGS=-m64 CXXFLAGS=-m64  FCFLAGS=-m64 \
           --prefix=path_to_your_MPI\
make all
make install
```

D.3 Installation of NetCDF-4 with HDF5

3 components need to be installed in the following order to enable the I/0 capabilities of the CHIMERE model :

1. HDF5 with parallel enabled
2. NetCDF-C linked againt the HDF5 library
3. NetCDF-Fortran linked against the NetCDF-C library

You need to choose a compiler (GNU or INTEL) and use it for all the components. You need to use accordingly the MPI library compiled before with this same compiler.

D.3.1 HDF5 installation with parallel enabled

In the following, you need to replace :

path_to_your_MPI : location of MPI directory with executables, libraries and include files defined previously

path_to_your_HDF5 : final location of HDF5 directory with executables, libraries and include files

```
./configure CC=path_to_your_MPI/bin/mpicc \
            FC=path_to_your_MPI/bin/mpif90 \
            CPPFLAGS=-Ipath_to_your_MPI/include \
            LDFLAGS=-Lpath_to_your_MPI/lib \
           --enable-parallel --enable-fortran \
           --prefix=path_to_your_HDF5\
make
make install
```

You may find more exhaustive information in the release_docs directory of the downloaded archive.

D.3.2 NetCDF-C installation

In the following, you need to replace :

path_to_your_MPI : location of MPI directory with executables, libraries and include files defined previously

path_to_your_HDF5 : location of HDF5 directory with executables, libraries and include files defined previously

path_to_your_NETCDF-C : final location of NetCDF-C directory with executables, libraries and include files

```
./configure CC=path_to_your_MPI/bin/mpicc \
             CPPFLAGS="-Ipath_to_your_MPI/include -Ipath_to_your_HDF5/include"\
             LDFLAGS="-Lpath_to_your_MPI/lib -Lpath_to_your_HDF5/lib" \
            --prefix=path_to_your_NETCDF-C\
make
make install
```

You may find more exhaustive information in the INSTALL.md file of the downloaded archive.

D.3.3 NetCDF-Fortran installation

In the following, you need to replace :

path_to_your_MPI : location of MPI directory with executables, libraries and include files defined previously

path_to_your_HDF5 : location of HDF5 directory with executables, libraries and include files defined previously

path_to_your_NETCDF-C : location of NetCDF-C directory with executables, libraries and include files defined previously

path_to_your_NETCDF-FORTRAN : final location of NetCDF-Fortran directory with executables, libraries and include files

```
./configure CC=path_to_your_MPI/bin/mpicc \
             FC=path_to_your_MPI/bin/mpif90 \
             CPPFLAGS="-Ipath_to_your_MPI/include -Ipath_to_your_HDF5/include  -Ipath_to_your_NETCDF-C/include"\
             LDFLAGS="-Lpath_to_your_MPI/lib -Lpath_to_your_HDF5/lib -Lpath_to_your_NETCDF-C/lib" \
            --prefix=path_to_your_NETCDF-FORTRAN\
make
make install
```

You may find more exhaustive information on the unidata website `https://www.unidata.ucar.edu/software/netcdf/docs/building_netcdf_fortran.html`.

E Structure of the CHIMERE netCDF files

This chapter describes the content of the several inout and output NetCDF files read or produced during a CHIMERE simulation. It includes:

- RAW anthropogenic and fire emission fluxes (before preprocessing)
- Anthropogenic, biogenic, dust and fire emissions after preprocessing and ready-to-use in the CHIMERE simulation
- Simulated species concentration and meteorological fields. The major CHIMERE output product named "out.[label].nc"
- Simulated deposition fluxes "dep.[label].nc"
- A copy of the last time-slot of the "out.[label].nc" in the "end.[label].nc" used in case of restart
- Meteorological fields read from the ECMWF and intepolated in the CHIMERE simulation mesh in the "exdo-mout.[label].nc"

E.1 The anthropogenic and fires emissions in the RAW format

E.1.1 Anthropogenic emissions

If you are using the program emiSURF you have input emissions with the following format, for example for NO and the month of April:

```
netcdf EMIS.MED60.04.NO.s {
dimensions:
        nlevel_emep = 7 ;
        nsectors = 1 ;
        Time = 24 ;
        type_day = 7 ;
        south_north = 109 ;
        west_east = 159 ;
variables:
        float EMEP_levels(nlevel_emep) ;
                EMEP_levels:units = "meters" ;
                EMEP_levels:long_name = "Cut_off_heights" ;
        float NO(nsectors, Time, type_day, nlevel_emep, south_north, west_east) ;
                NO:units = "molecule/cm2/s or g/km2/s" ;
                NO:long_name = "NO Emission" ;
        int SNAPsectors(nsectors) ;
                SNAPsectors:units = "none" ;
                SNAPsectors:long_name = "Id of output SNAP sectors" ;
        float lat(south_north, west_east) ;
                lat:units = "degrees_north" ;
                lat:long_name = "Latitude" ;
        float lon(south_north, west_east) ;
                lon:units = "degrees_east" ;
                lon:long_name = "Longitude" ;

// global attributes:
                :Title = "CHIMERE emissions" ;
                :Sub-title = "Hourly surface emission flux" ;
                :Chimere_type = "EMISSIONS" ;
                :Generating_process = "Generated by emiSURF2018" ;
                :Conventions = "None" ;
                :Domain = "MED60" ;
                :Year = "2010" ;
                :Version = "2018" ;
                :Landuse_for_Reggriding = "usgs" ;
                :Population_for_Residential_Sector = "no" ;
                :Roadmap_for_Road_traffic_Sector = "yes" ;
                :Sector_information = "Sectors by SNAP" ;
                :NCO = "4.4.8" ;
}
```

E.1.2 Fires emissions

The format for fires emissions is different since only the cells where fires are active are described.

```
netcdf EMISf_MED60_201405_NO {
dimensions:
        nFRPbins = 21 ;
        hours = 24 ;
        days = 31 ;
        sources = 286 ;
        south_north = 109 ;
        west_east = 159 ;
variables:
        float FRPbins(nFRPbins) ;
                FRPbins:units = "MW" ;
                FRPbins:long_name = "FRP bins limits" ;
        float NO(hours, days, sources) ;
                NO:units = "molecule/cm2/s" ;
                NO:long_name = "NO Emission" ;
        float area(south_north, west_east) ;
                area:units = "cm2" ;
                area:long_name = "Grid cell area" ;
        float avg_FRPbin(nFRPbins, days, sources) ;
                avg_FRPbin:units = "MW" ;
                avg_FRPbin:long_name = "Avg FRP in each FRP bin" ;
        float fracEmiss_FRPbin(nFRPbins, days, sources) ;
                fracEmiss_FRPbin:units = "-" ;
                fracEmiss_FRPbin:long_name = "Fraction of emissions in each FRP bin" ;
        float ilatsources(sources) ;
                ilatsources:units = "index" ;
                ilatsources:long_name = "Latitude_of_sources" ;
        float ilonsources(sources) ;
                ilonsources:units = "index" ;
                ilonsources:long_name = "Longitude_of_sources" ;
        float lat(south_north, west_east) ;
                lat:units = "degrees_north" ;
                lat:long_name = "Latitude" ;
        float lon(south_north, west_east) ;
                lon:units = "degrees_east" ;
                lon:long_name = "Longitude" ;
        float maxFRPsources(hours, days, sources) ;
                maxFRPsources:units = "MW" ;
                maxFRPsources:long_name = "Max Fire Radiative Power" ;

// global attributes:
                :Title = "APIFLAME" ;
                :Sub-title = "Hourly fire emissions for one species" ;
                :Chimere_type = "EMISSIONS" ;
                :Generating_process = "Generated by sectoremis" ;
                :Conventions = "None" ;
                :Domain = "MED60" ;
                :year = "2014" ;
                :month = "05" ;
                :NCO = "4.4.8" ;
}
```

E.2 The anthropogenic, biogenic, dust and fire emissions in the CHIMERE output format

These files are created during the simulation. They are on the CHIMERE grid and with an hourly time-step. They are present if the user wants to run a simulation with its own emissions or re-run the same simulation after some changes in these emissions.

Names of these files are defined in the chimere.par parameter file as:

```
   [$fnemisa]          Anthropic emiss. File (from Stage 2)  : ${datadir}/AEMISSIONS.${sim}.nc
   [$fnemisf]          Fires emiss. File (from Stage 2)      : ${datadir}/FEMISSIONS.${sim}.nc
   [$fnemisb]          Biogenic emission File                : ${datadir}/BEMISSIONS.${sim}.nc
   [$fnemissalt]       Sea salt emission File                : ${datadir}/SEMISSIONS.${sim}.nc
   [$fnemisd]          Dust emission File                    : ${datadir}/DEMISSIONS.${sim}.nc
```

E.2.1 AEMISSIONS: The anthropogenic emissions output file

In case of anthropogenic emissions generated with the emiSURF model for your CHIMERE simulation (*[emistyp]* flag set to "emisurf") the AEMISSIONS.nc file will be written by the CHIMERE preprocessing program prep_aemis. It has the following format.

```
netcdf AEMISSIONS.2014081800_2014082300_MED60o1_nc4 {
dimensions:
        Time = UNLIMITED ; // (122 currently)
        DateStrLen = 19 ;
        west_east = 159 ;
        south_north = 109 ;
        bottom_top = 20 ;
        SpStrLen = 23 ;
        Species = 26 ;
variables:
        char Times(Time, DateStrLen) ;
                Times:units = "Date string" ;
                Times:long_name = "Format YYYY-MM-DD_HH:MN:SS" ;
        char species(Species, SpStrLen) ;
                species:units = "-" ;
                species:long_name = "Species names" ;
        double lon(south_north, west_east) ;
                lon:units = "degrees_east" ;
                lon:long_name = "Longitude" ;
        double lat(south_north, west_east) ;
                lat:units = "degrees_north" ;
                lat:long_name = "Latitude" ;
        double APINEN(Time, bottom_top, south_north, west_east) ;
                APINEN:units = "molecule/cm2/s" ;
                APINEN:long_name = "APINEN Emission" ;
        double C2H4(Time, bottom_top, south_north, west_east) ;
                C2H4:units = "molecule/cm2/s" ;
                C2H4:long_name = "C2H4 Emission" ;
        double C2H6(Time, bottom_top, south_north, west_east) ;
                C2H6:units = "molecule/cm2/s" ;
                C2H6:long_name = "C2H6 Emission" ;
        double C3H6(Time, bottom_top, south_north, west_east) ;
                C3H6:units = "molecule/cm2/s" ;
                C3H6:long_name = "C3H6 Emission" ;
        double C5H8(Time, bottom_top, south_north, west_east) ;
                C5H8:units = "molecule/cm2/s" ;
                C5H8:long_name = "C5H8 Emission" ;
        double CH3CHO(Time, bottom_top, south_north, west_east) ;
                CH3CHO:units = "molecule/cm2/s" ;
                CH3CHO:long_name = "CH3CHO Emission" ;
        double CH3COE(Time, bottom_top, south_north, west_east) ;
                CH3COE:units = "molecule/cm2/s" ;
                CH3COE:long_name = "CH3COE Emission" ;
        double CH4(Time, bottom_top, south_north, west_east) ;
                CH4:units = "molecule/cm2/s" ;
                CH4:long_name = "CH4 Emission" ;
        double CO(Time, bottom_top, south_north, west_east) ;
                CO:units = "molecule/cm2/s" ;
                CO:long_name = "CO Emission" ;
        double HCHO(Time, bottom_top, south_north, west_east) ;
                HCHO:units = "molecule/cm2/s" ;
                HCHO:long_name = "HCHO Emission" ;
        double HONO(Time, bottom_top, south_north, west_east) ;
                HONO:units = "molecule/cm2/s" ;
                HONO:long_name = "HONO Emission" ;
        double NC4H10(Time, bottom_top, south_north, west_east) ;
                NC4H10:units = "molecule/cm2/s" ;
                NC4H10:long_name = "NC4H10 Emission" ;
        double NH3(Time, bottom_top, south_north, west_east) ;
                NH3:units = "molecule/cm2/s" ;
                NH3:long_name = "NH3 Emission" ;
        double NO(Time, bottom_top, south_north, west_east) ;
                NO:units = "molecule/cm2/s" ;
                NO:long_name = "NO Emission" ;
        double NO2(Time, bottom_top, south_north, west_east) ;
                NO2:units = "molecule/cm2/s" ;
                NO2:long_name = "NO2 Emission" ;
        double OXYL(Time, bottom_top, south_north, west_east) ;
                OXYL:units = "molecule/cm2/s" ;
                OXYL:long_name = "OXYL Emission" ;
        double SO2(Time, bottom_top, south_north, west_east) ;
                SO2:units = "molecule/cm2/s" ;
                SO2:long_name = "SO2 Emission" ;
        double H2SO4_fin(Time, bottom_top, south_north, west_east) ;
```

```
                H2SO4_fin:units = "molecule/cm2/s" ;
                H2SO4_fin:long_name = "H2SO4_fin Emission" ;
        double TOL(Time, bottom_top, south_north, west_east) ;
                TOL:units = "molecule/cm2/s" ;
                TOL:long_name = "TOL Emission" ;
        double TMB(Time, bottom_top, south_north, west_east) ;
                TMB:units = "molecule/cm2/s" ;
                TMB:long_name = "TMB Emission" ;
        double OCAR_fin(Time, bottom_top, south_north, west_east) ;
                OCAR_fin:units = "molecule/cm2/s" ;
                OCAR_fin:long_name = "OCAR_fin Emission" ;
        double BCAR_fin(Time, bottom_top, south_north, west_east) ;
                BCAR_fin:units = "molecule/cm2/s" ;
                BCAR_fin:long_name = "BCAR_fin Emission" ;
        double PPM_fin(Time, bottom_top, south_north, west_east) ;
                PPM_fin:units = "molecule/cm2/s" ;
                PPM_fin:long_name = "PPM_fin Emission" ;
        double OCAR_coa(Time, bottom_top, south_north, west_east) ;
                OCAR_coa:units = "molecule/cm2/s" ;
                OCAR_coa:long_name = "OCAR_coa Emission" ;
        double BCAR_coa(Time, bottom_top, south_north, west_east) ;
                BCAR_coa:units = "molecule/cm2/s" ;
                BCAR_coa:long_name = "BCAR_coa Emission" ;
        double PPM_coa(Time, bottom_top, south_north, west_east) ;
                PPM_coa:units = "molecule/cm2/s" ;
                PPM_coa:long_name = "PPM_coa Emission" ;

// global attributes:
                :_NCProperties = "version=1|netcdflibversion=4.6.0|hdf5libversion=1.8.20" ;
                :Title = "CHIMERE SUITE" ;
                :Sub-title = "Emissions file - Surface + Point sources" ;
                :Generating_process = "Generated by prepemis" ;
                :Conventions =  ;
                :Domain = "MED60" ;
```

However, if the user wants to use his/her own anthropogenic emission file without passing from the emiSURF program she must set the *[imakeaemis]* flag to 0 and make sure to provide the AEMISSIONS file in the expected format (see also Section 9).
In the specific case of subgrid scale modeling of the variability induced by heterogeneous anthropogenic emissions (i.e. *[sgmodel]* flag set to 1) the provided AEMISSIONS file needs to have an additional dimension Sectors as described in detail in Section 9.3.1.

E.2.2 BEMISSIONS: The biogenic emissions output file

```
netcdf BEMISSIONS.2014081800_2014082300_MED60o1_nc4 {
dimensions:
        Time = UNLIMITED ; // (121 currently)
        DateStrLen = 19 ;
        west_east = 159 ;
        south_north = 109 ;
        biospecies = 8 ;
variables:
        char Times(Time, DateStrLen) ;
                Times:units = "Date string" ;
                Times:long_name = "WRF-style Times" ;
        double lon(south_north, west_east) ;
                lon:units = "degrees_east" ;
                lon:long_name = "Longitude" ;
        double lat(south_north, west_east) ;
                lat:units = "degrees_north" ;
                lat:long_name = "Latitude" ;
        double C5H8(Time, south_north, west_east) ;
                C5H8:units = "g/m2/s" ;
                C5H8:long_name = "C5H8 emission flux" ;
        double APINEN(Time, south_north, west_east) ;
                APINEN:units = "g/m2/s" ;
                APINEN:long_name = "APINEN emission flux" ;
        double BPINEN(Time, south_north, west_east) ;
                BPINEN:units = "g/m2/s" ;
                BPINEN:long_name = "BPINEN emission flux" ;
        double LIMONE(Time, south_north, west_east) ;
                LIMONE:units = "g/m2/s" ;
                LIMONE:long_name = "LIMONE emission flux" ;
        double TERPEN(Time, south_north, west_east) ;
                TERPEN:units = "g/m2/s" ;
                TERPEN:long_name = "TERPEN emission flux" ;
```

```
        double OCIMEN(Time, south_north, west_east) ;
                OCIMEN:units = "g/m2/s" ;
                OCIMEN:long_name = "OCIMEN emission flux" ;
        double HUMULE(Time, south_north, west_east) ;
                HUMULE:units = "g/m2/s" ;
                HUMULE:long_name = "HUMULE emission flux" ;
        double NO(Time, south_north, west_east) ;
                NO:units = "g/m2/s" ;
                NO:long_name = "NO emission flux" ;

// global attributes:
                :_NCProperties = "version=1|netcdflibversion=4.6.0|hdf5libversion=1.8.20" ;
                :Title = "CHIMERE SUITE" ;
                :Sub-title = "Bioemissions file" ;
                :Generating_process = "Generated by diagbio" ;
                :Conventions =  ;
                :Domain = "MED60" ;
}
```

E.2.3 FEMISSIONS: The fires emissions output file

The format of FEMISSIONS is strictly the same as AEMISSIONS.

E.2.4 SEMISSIONS: The sea-salt emissions output file

```
netcdf SEMISSIONS.2014081800_2014082300_MED60o1_nc4 {
dimensions:
        Time = UNLIMITED ; // (121 currently)
        DateStrLen = 19 ;
        bins = 10 ;
        west_east = 159 ;
        south_north = 109 ;
        saltspecies = 4 ;
variables:
        char Times(Time, DateStrLen) ;
                Times:units = "Date string" ;
                Times:long_name = "WRF-style Times" ;
        double lon(south_north, west_east) ;
                lon:units = "degrees_east" ;
                lon:long_name = "Longitude" ;
        double lat(south_north, west_east) ;
                lat:units = "degrees_north" ;
                lat:long_name = "Latitude" ;
        double SALT(Time, bins, south_north, west_east) ;
                SALT:units = "g/m2/s" ;
                SALT:long_name = "SALT emission flux" ;
        double NA(Time, bins, south_north, west_east) ;
                NA:units = "g/m2/s" ;
                NA:long_name = "NA emission flux" ;
        double HCL(Time, bins, south_north, west_east) ;
                HCL:units = "g/m2/s" ;
                HCL:long_name = "HCL emission flux" ;
        double H2SO4(Time, bins, south_north, west_east) ;
                H2SO4:units = "g/m2/s" ;
                H2SO4:long_name = "H2SO4 emission flux" ;
        double DMS(Time, south_north, west_east) ;
                DMS:units = "g/m2/s" ;
                DMS:long_name = "DMS emission flux" ;

// global attributes:
                :_NCProperties = "version=1|netcdflibversion=4.6.0|hdf5libversion=1.8.20" ;
                :Title = "CHIMERE SUITE" ;
                :Sub-title = "Salt emissions file" ;
                :Generating_process = "Generated by diagsalt" ;
                :Conventions =  ;
                :Domain = "MED60" ;
}
```

E.2.5 DEMISSIONS: The mineral emissions output file

```
netcdf DEMISSIONS.2014081800_2014082300_MED60o1_nc4 {
dimensions:
        Time = UNLIMITED ; // (121 currently)
```

```
        DateStrLen = 19 ;
        west_east = 159 ;
        south_north = 109 ;
        bins = 10 ;
        ninterv = 11 ;
        nmode = 3 ;
variables:
        char Times(Time, DateStrLen) ;
                Times:units = "Date string" ;
                Times:long_name = "WRF-style Times" ;
        double lon(south_north, west_east) ;
                lon:units = "degrees_east" ;
                lon:long_name = "Longitude" ;
        double lat(south_north, west_east) ;
                lat:units = "degrees_north" ;
                lat:long_name = "Latitude" ;
        double DUST(Time, south_north, west_east) ;
                DUST:units = "g/m2/s" ;
                DUST:long_name = "DUST emission flux" ;
        double eDUST(Time, bins, south_north, west_east) ;
                eDUST:units = "g/m2/s" ;
                eDUST:long_name = "eDUST emission flux" ;
        double mmd_m_itv(ninterv) ;
                mmd_m_itv:units = "m" ;
                mmd_m_itv:long_name = "Size distribution intervals" ;
        double mmd_m(bins) ;
                mmd_m:units = "m" ;
                mmd_m:long_name = "Size distribution bins" ;

// global attributes:
                :_NCProperties = "version=1|netcdflibversion=4.6.0|hdf5libversion=1.8.20" ;
                :Title = "CHIMERE SUITE" ;
                :Sub-title = "Mineral dust emissions file" ;
                :Generating_process = "Generated by worker_dust" ;
                :Conventions =  ;
                :Domain = "MED60" ;
}
```

E.3 The main results file "out.[label].nc"

This section lists all variables in the main output file of a CHIMERE simulation. The Table E.1 details the meteorological variables contained in this file. For the particular case of subgrid-scale modeling due to heterogeneous anthropogenic emissions (*[sgmodel]* flag = 1) the structure of the "out.[label].nc" is presented in detail in Section 9.3.1.

Model variable	**Variable description**	**Units**
Times	Date string	NA
lon	Longitude	degrees-east
lat	Latitude	degrees-north
2-D Variables		
hght	Boundary Layer Height	m
usta	Friction Velocity u_*	m/s
uwta	Convection Velocity w_*	m/s
aerr	Aerodynamic Resistance	s/m
sshf	Surface Heat Flux	W/m^2
slhf	Surface Latent Heat Flux	W/m^2
u10m	10m Zonal Wind Speed	m/s
v10m	10m Meridional Wind Speed	m/s
w10m	10m Vertical Wind Speed	m/s
tem2	2m Temperature	K
soim	Soil Moisture	m^3/m^3
sreh	Surface relative humidity	%
pscf	Surface Pressure	Pa
swrd	Short Wave Radiation	W/m^2
atte	Radiation Attenuation Coefficient	[0-1]

Model variable	Variable description	Units
copc	Convective precipitation	kg/m^2/h
lspc	Large Scale precipitation	kg/m^2/h
toppc	Total precipitation	kg/m^2/h
3-D Variables		
hlay	Top of Layers' Height	m
thlay	Layers' Thickness	m
winz	Zonal Wind Speed	m/s
winm	Meridional Wind Speed	m/s
winw	Vertical Wind Speed	m/s
temp	Temperature	K
pres	Pression	Pa
airm	Air Density	kg/m^3
sphu	Specific Humidity	kg/kg
clwc	Liquid (+ice) Water Content	kg/kg
cliq	Cloud Liquid Water Mixing Ratio	kg/kg
cice	Ice Liquid Water Mixing Ratio	kg/kg
rain	Rain Water Mixing Ratio	kg/kg
kzzz	Vertical Diffusion Coefficient	m^2/s^2
Deep convection related fields		
dpeu	Entrainement in Updraft	kg/m^2/s
dped	Entrainement in Downdraft	kg/m^2/s
dpdu	Detrainement in Updraft	kg/m^2/s
dpdd	Detrainement in Downdraft	kg/m^2/s
flxu	Upward Mass Flux	kg/m^2/s
flxd	Downward Mass Flux	kg/m^2/s

Table E.1: *List of output meteorological variables*

```
netcdf out.2014081800_2014082300_MED60o1_nc4 {
dimensions:
        Time = UNLIMITED ; // (121 currently)
        DateStrLen = 19 ;
        west_east = 159 ;
        south_north = 109 ;
        bottom_top = 20 ;
        vcoord_dim = 20 ;
        wavelength_dim = 5 ;
        nbins = 10 ;
        ninterv = 11 ;
variables:
        double lon(south_north, west_east) ;
                lon:units = "degrees_east" ;
                lon:long_name = "Longitude" ;
        double lat(south_north, west_east) ;
                lat:units = "degrees_north" ;
                lat:long_name = "Latitude" ;
        double a_vcoord(vcoord_dim) ;
                a_vcoord:units = "no_units" ;
                a_vcoord:long_name = "A_sigma_coefficient" ;
        double b_vcoord(vcoord_dim) ;
                b_vcoord:units = "no_units" ;
                b_vcoord:long_name = "B_sigma_coefficient" ;
        double cut_off_diameters(ninterv) ;
                cut_off_diameters:units = "meters" ;
                cut_off_diameters:long_name = "cut_off_diameters" ;
        double mmd(nbins) ;
                mmd:units = "meters" ;
                mmd:long_name = "Mean_Mass_Median_Diameters" ;
        float wavelength(wavelength_dim) ;
                wavelength:units = "nm" ;
                wavelength:long_name = "Wavelength" ;
        char Times(Time, DateStrLen) ;
                Times:units = "Date string" ;
```

```
            Times:long_name = "WRF-style Times" ;
        float O3(Time, bottom_top, south_north, west_east) ;
            O3:units = "ppb vol" ;
            O3:long_name = "O3 Concentration" ;
        float NO2(Time, bottom_top, south_north, west_east) ;
            NO2:units = "ppb vol" ;
            NO2:long_name = "NO2 Concentration" ;
        float NO(Time, bottom_top, south_north, west_east) ;
            NO:units = "ppb vol" ;
            NO:long_name = "NO Concentration" ;
        float PAN(Time, bottom_top, south_north, west_east) ;
            PAN:units = "ppb vol" ;
            PAN:long_name = "PAN Concentration" ;
        float HNO3(Time, bottom_top, south_north, west_east) ;
            HNO3:units = "ppb vol" ;
            HNO3:long_name = "HNO3 Concentration" ;
        float H2O2(Time, bottom_top, south_north, west_east) ;
            H2O2:units = "ppb vol" ;
            H2O2:long_name = "H2O2 Concentration" ;
        float HONO(Time, bottom_top, south_north, west_east) ;
            HONO:units = "ppb vol" ;
            HONO:long_name = "HONO Concentration" ;
        float SO2(Time, bottom_top, south_north, west_east) ;
            SO2:units = "ppb vol" ;
            SO2:long_name = "SO2 Concentration" ;
        float CO(Time, bottom_top, south_north, west_east) ;
            CO:units = "ppb vol" ;
            CO:long_name = "CO Concentration" ;
        float OH(Time, bottom_top, south_north, west_east) ;
            OH:units = "ppb vol" ;
            OH:long_name = "OH Concentration" ;
        float CH4(Time, bottom_top, south_north, west_east) ;
            CH4:units = "ppb vol" ;
            CH4:long_name = "CH4 Concentration" ;
        float C2H6(Time, bottom_top, south_north, west_east) ;
            C2H6:units = "ppb vol" ;
            C2H6:long_name = "C2H6 Concentration" ;
        float NC4H10(Time, bottom_top, south_north, west_east) ;
            NC4H10:units = "ppb vol" ;
            NC4H10:long_name = "NC4H10 Concentration" ;
        float C2H4(Time, bottom_top, south_north, west_east) ;
            C2H4:units = "ppb vol" ;
            C2H4:long_name = "C2H4 Concentration" ;
        float C3H6(Time, bottom_top, south_north, west_east) ;
            C3H6:units = "ppb vol" ;
            C3H6:long_name = "C3H6 Concentration" ;
        float OXYL(Time, bottom_top, south_north, west_east) ;
            OXYL:units = "ppb vol" ;
            OXYL:long_name = "OXYL Concentration" ;
        float C5H8(Time, bottom_top, south_north, west_east) ;
            C5H8:units = "ppb vol" ;
            C5H8:long_name = "C5H8 Concentration" ;
        float HCHO(Time, bottom_top, south_north, west_east) ;
            HCHO:units = "ppb vol" ;
            HCHO:long_name = "HCHO Concentration" ;
        float CH3CHO(Time, bottom_top, south_north, west_east) ;
            CH3CHO:units = "ppb vol" ;
            CH3CHO:long_name = "CH3CHO Concentration" ;
        float GLYOX(Time, bottom_top, south_north, west_east) ;
            GLYOX:units = "ppb vol" ;
            GLYOX:long_name = "GLYOX Concentration" ;
        float MGLYOX(Time, bottom_top, south_north, west_east) ;
            MGLYOX:units = "ppb vol" ;
            MGLYOX:long_name = "MGLYOX Concentration" ;
        float CH3COE(Time, bottom_top, south_north, west_east) ;
            CH3COE:units = "ppb vol" ;
            CH3COE:long_name = "CH3COE Concentration" ;
        float NH3(Time, bottom_top, south_north, west_east) ;
            NH3:units = "ppb vol" ;
            NH3:long_name = "NH3 Concentration" ;
        float APINEN(Time, bottom_top, south_north, west_east) ;
            APINEN:units = "ppb vol" ;
            APINEN:long_name = "APINEN Concentration" ;
        float BPINEN(Time, bottom_top, south_north, west_east) ;
            BPINEN:units = "ppb vol" ;
            BPINEN:long_name = "BPINEN Concentration" ;
        float LIMONE(Time, bottom_top, south_north, west_east) ;
            LIMONE:units = "ppb vol" ;
            LIMONE:long_name = "LIMONE Concentration" ;
        float TERPEN(Time, bottom_top, south_north, west_east) ;
            TERPEN:units = "ppb vol" ;
```

```
        TERPEN:long_name = "TERPEN Concentration" ;
    float OCIMEN(Time, bottom_top, south_north, west_east) ;
        OCIMEN:units = "ppb vol" ;
        OCIMEN:long_name = "OCIMEN Concentration" ;
    float HUMULE(Time, bottom_top, south_north, west_east) ;
        HUMULE:units = "ppb vol" ;
        HUMULE:long_name = "HUMULE Concentration" ;
    float SALTAQ(Time, bottom_top, south_north, west_east) ;
        SALTAQ:units = "ppb vol" ;
        SALTAQ:long_name = "SALTAQ Concentration" ;
    float DUSTAQ(Time, bottom_top, south_north, west_east) ;
        DUSTAQ:units = "ppb vol" ;
        DUSTAQ:long_name = "DUSTAQ Concentration" ;
    float OCARAQ(Time, bottom_top, south_north, west_east) ;
        OCARAQ:units = "ppb vol" ;
        OCARAQ:long_name = "OCARAQ Concentration" ;
    float BCARAQ(Time, bottom_top, south_north, west_east) ;
        BCARAQ:units = "ppb vol" ;
        BCARAQ:long_name = "BCARAQ Concentration" ;
    float bBCAR(Time, nbins, bottom_top, south_north, west_east) ;
        bBCAR:units = "ug/m3" ;
        bBCAR:long_name = "BCAR concentration per size bin" ;
    float bDUST(Time, nbins, bottom_top, south_north, west_east) ;
        bDUST:units = "ug/m3" ;
        bDUST:long_name = "DUST concentration per size bin" ;
    float bOCAR(Time, nbins, bottom_top, south_north, west_east) ;
        bOCAR:units = "ug/m3" ;
        bOCAR:long_name = "OCAR concentration per size bin" ;
    float bPPM(Time, nbins, bottom_top, south_north, west_east) ;
        bPPM:units = "ug/m3" ;
        bPPM:long_name = "PPM concentration per size bin" ;
    float bSALT(Time, nbins, bottom_top, south_north, west_east) ;
        bSALT:units = "ug/m3" ;
        bSALT:long_name = "SALT concentration per size bin" ;
    float bAnA1D(Time, nbins, bottom_top, south_north, west_east) ;
        bAnA1D:units = "ug/m3" ;
        bAnA1D:long_name = "AnA1D concentration per size bin" ;
    float bAnBmP(Time, nbins, bottom_top, south_north, west_east) ;
        bAnBmP:units = "ug/m3" ;
        bAnBmP:long_name = "AnBmP concentration per size bin" ;
    float bBiA1D(Time, nbins, bottom_top, south_north, west_east) ;
        bBiA1D:units = "ug/m3" ;
        bBiA1D:long_name = "BiA1D concentration per size bin" ;
    float bBiBmP(Time, nbins, bottom_top, south_north, west_east) ;
        bBiBmP:units = "ug/m3" ;
        bBiBmP:long_name = "BiBmP concentration per size bin" ;
    float bISOPA1(Time, nbins, bottom_top, south_north, west_east) ;
        bISOPA1:units = "ug/m3" ;
        bISOPA1:long_name = "ISOPA1 concentration per size bin" ;
    float bH2SO4(Time, nbins, bottom_top, south_north, west_east) ;
        bH2SO4:units = "ug/m3" ;
        bH2SO4:long_name = "H2SO4 concentration per size bin" ;
    float bHNO3(Time, nbins, bottom_top, south_north, west_east) ;
        bHNO3:units = "ug/m3" ;
        bHNO3:long_name = "HNO3 concentration per size bin" ;
    float bNH3(Time, nbins, bottom_top, south_north, west_east) ;
        bNH3:units = "ug/m3" ;
        bNH3:long_name = "NH3 concentration per size bin" ;
    float bWATER(Time, nbins, bottom_top, south_north, west_east) ;
        bWATER:units = "ug/m3" ;
        bWATER:long_name = "WATER concentration per size bin" ;
    float DMS(Time, bottom_top, south_north, west_east) ;
        DMS:units = "ppb vol" ;
        DMS:long_name = "DMS Concentration" ;
    float DMSO(Time, bottom_top, south_north, west_east) ;
        DMSO:units = "ppb vol" ;
        DMSO:long_name = "DMSO Concentration" ;
    float MSA(Time, bottom_top, south_north, west_east) ;
        MSA:units = "ppb vol" ;
        MSA:long_name = "MSA Concentration" ;
    float TOTPAN(Time, bottom_top, south_north, west_east) ;
        TOTPAN:units = "ppb vol" ;
        TOTPAN:long_name = "TOTPAN Concentration" ;
    float NOX(Time, bottom_top, south_north, west_east) ;
        NOX:units = "ppb vol" ;
        NOX:long_name = "NOX Concentration" ;
    float OX(Time, bottom_top, south_north, west_east) ;
        OX:units = "ppb vol" ;
        OX:long_name = "OX Concentration" ;
    float HOX(Time, bottom_top, south_north, west_east) ;
        HOX:units = "ppb vol" ;
```

```
        HOX:long_name = "HOX Concentration" ;
    float NOY(Time, bottom_top, south_north, west_east) ;
        NOY:units = "ppb vol" ;
        NOY:long_name = "NOY Concentration" ;
    float ROOH(Time, bottom_top, south_north, west_east) ;
        ROOH:units = "ppb vol" ;
        ROOH:long_name = "ROOH Concentration" ;
    float HCNM(Time, bottom_top, south_north, west_east) ;
        HCNM:units = "ppb vol" ;
        HCNM:long_name = "HCNM Concentration" ;
    float pBCAR(Time, bottom_top, south_north, west_east) ;
        pBCAR:units = "ug/m3" ;
        pBCAR:long_name = "pBCAR Concentration" ;
    float pDUST(Time, bottom_top, south_north, west_east) ;
        pDUST:units = "ug/m3" ;
        pDUST:long_name = "pDUST Concentration" ;
    float pOCAR(Time, bottom_top, south_north, west_east) ;
        pOCAR:units = "ug/m3" ;
        pOCAR:long_name = "pOCAR Concentration" ;
    float pPPM(Time, bottom_top, south_north, west_east) ;
        pPPM:units = "ug/m3" ;
        pPPM:long_name = "pPPM Concentration" ;
    float pSALT(Time, bottom_top, south_north, west_east) ;
        pSALT:units = "ug/m3" ;
        pSALT:long_name = "pSALT Concentration" ;
    float pAnA1D(Time, bottom_top, south_north, west_east) ;
        pAnA1D:units = "ug/m3" ;
        pAnA1D:long_name = "pAnA1D Concentration" ;
    float pAnBmP(Time, bottom_top, south_north, west_east) ;
        pAnBmP:units = "ug/m3" ;
        pAnBmP:long_name = "pAnBmP Concentration" ;
    float pBiA1D(Time, bottom_top, south_north, west_east) ;
        pBiA1D:units = "ug/m3" ;
        pBiA1D:long_name = "pBiA1D Concentration" ;
    float pBiBmP(Time, bottom_top, south_north, west_east) ;
        pBiBmP:units = "ug/m3" ;
        pBiBmP:long_name = "pBiBmP Concentration" ;
    float pISOPA1(Time, bottom_top, south_north, west_east) ;
        pISOPA1:units = "ug/m3" ;
        pISOPA1:long_name = "pISOPA1 Concentration" ;
    float pH2SO4(Time, bottom_top, south_north, west_east) ;
        pH2SO4:units = "ug/m3" ;
        pH2SO4:long_name = "pH2SO4 Concentration" ;
    float pHNO3(Time, bottom_top, south_north, west_east) ;
        pHNO3:units = "ug/m3" ;
        pHNO3:long_name = "pHNO3 Concentration" ;
    float pNH3(Time, bottom_top, south_north, west_east) ;
        pNH3:units = "ug/m3" ;
        pNH3:long_name = "pNH3 Concentration" ;
    float pWATER(Time, bottom_top, south_north, west_east) ;
        pWATER:units = "ug/m3" ;
        pWATER:long_name = "pWATER Concentration" ;
    float PM25(Time, bottom_top, south_north, west_east) ;
        PM25:units = "ug/m3" ;
        PM25:long_name = "PM25 Concentration" ;
    float PM25bio(Time, bottom_top, south_north, west_east) ;
        PM25bio:units = "ug/m3" ;
        PM25bio:long_name = "PM25bio Concentration" ;
    float PM25ant(Time, bottom_top, south_north, west_east) ;
        PM25ant:units = "ug/m3" ;
        PM25ant:long_name = "PM25ant Concentration" ;
    float PM10(Time, bottom_top, south_north, west_east) ;
        PM10:units = "ug/m3" ;
        PM10:long_name = "PM10 Concentration" ;
    float PM10bio(Time, bottom_top, south_north, west_east) ;
        PM10bio:units = "ug/m3" ;
        PM10bio:long_name = "PM10bio Concentration" ;
    float PM10ant(Time, bottom_top, south_north, west_east) ;
        PM10ant:units = "ug/m3" ;
        PM10ant:long_name = "PM10ant Concentration" ;
    double bnum(Time, nbins, bottom_top, south_north, west_east) ;
        bnum:units = "#/cm3" ;
        bnum:long_name = "Particle concentration per size bin" ;
    double chem_regime(Time, bottom_top, south_north, west_east) ;
        chem_regime:units = "ratio" ;
        chem_regime:long_name = "Chemical regime" ;
    double hlay(Time, bottom_top, south_north, west_east) ;
        hlay:units = "m" ;
        hlay:long_name = "Layer top altitude" ;
    double thlay(Time, bottom_top, south_north, west_east) ;
        thlay:units = "m" ;
```

```
                thlay:long_name = "Layer thickness" ;
        double airm(Time, bottom_top, south_north, west_east) ;
                airm:units = "molecule/cm**3" ;
                airm:long_name = "Air density" ;
        double relh(Time, bottom_top, south_north, west_east) ;
                relh:units = "%/100" ;
                relh:long_name = "Relative Humidity" ;
        double temp(Time, bottom_top, south_north, west_east) ;
                temp:units = "K" ;
                temp:long_name = "Temperature" ;
        double winz(Time, bottom_top, south_north, west_east) ;
                winz:units = "m/s" ;
                winz:long_name = "Zonal wind" ;
        double winm(Time, bottom_top, south_north, west_east) ;
                winm:units = "m/s" ;
                winm:long_name = "Meridional wind" ;
        double winw(Time, bottom_top, south_north, west_east) ;
                winw:units = "m/s" ;
                winw:long_name = "Vertical wind" ;
        double sphu(Time, bottom_top, south_north, west_east) ;
                sphu:units = "kg/kg" ;
                sphu:long_name = "Specific humidity" ;
        double kzzz(Time, bottom_top, south_north, west_east) ;
                kzzz:units = "m2/s2" ;
                kzzz:long_name = "Kz diffusion coefficient" ;
        double cliq(Time, bottom_top, south_north, west_east) ;
                cliq:units = "kg/kg" ;
                cliq:long_name = "Cloud liquid water mixing ratio" ;
        double rain(Time, bottom_top, south_north, west_east) ;
                rain:units = "kg/kg" ;
                rain:long_name = "Rain mixing ratio" ;
        double snow(Time, bottom_top, south_north, west_east) ;
                snow:units = "kg/kg" ;
                snow:long_name = "Snow mixing ratio" ;
        double cice(Time, bottom_top, south_north, west_east) ;
                cice:units = "kg/kg" ;
                cice:long_name = "Ice liquid water mixing ratio" ;
        double pres(Time, bottom_top, south_north, west_east) ;
                pres:units = "Pa" ;
                pres:long_name = "Pressure" ;
        double dpeu(Time, bottom_top, south_north, west_east) ;
                dpeu:units = "kg/m2/s" ;
                dpeu:long_name = "Entrainment in updraft" ;
        double dped(Time, bottom_top, south_north, west_east) ;
                dped:units = "kg/m2/s" ;
                dped:long_name = "Entrainment in downdraft" ;
        double dpdu(Time, bottom_top, south_north, west_east) ;
                dpdu:units = "kg/m2/s" ;
                dpdu:long_name = "Detrainment in updraft" ;
        double dpdd(Time, bottom_top, south_north, west_east) ;
                dpdd:units = "kg/m2/s" ;
                dpdd:long_name = "Detrainment in downdraft" ;
        double flxu(Time, bottom_top, south_north, west_east) ;
                flxu:units = "kg/m2/s" ;
                flxu:long_name = "Updraft mass flux" ;
        double flxd(Time, bottom_top, south_north, west_east) ;
                flxd:units = "kg/m2/s" ;
                flxd:long_name = "Downdraft mass flux" ;
        double jO3(Time, bottom_top, south_north, west_east) ;
                jO3:units = "s-1" ;
                jO3:long_name = "jO3" ;
        double jNO2(Time, bottom_top, south_north, west_east) ;
                jNO2:units = "s-1" ;
                jNO2:long_name = "jNO2" ;
        double optdtotal(Time, wavelength_dim, south_north, west_east) ;
                optdtotal:units = "ad" ;
                optdtotal:long_name = "Total optical depth" ;
        double optdcloud(Time, wavelength_dim, south_north, west_east) ;
                optdcloud:units = "ad" ;
                optdcloud:long_name = "Cloud optical depth" ;
        double optdaero(Time, wavelength_dim, south_north, west_east) ;
                optdaero:units = "ad" ;
                optdaero:long_name = "Aerosol optical depth" ;
        double dry_to_wet(Time, nbins, bottom_top, south_north, west_east) ;
                dry_to_wet:units = "-" ;
                dry_to_wet:long_name = "Ratio between dry and wet diameter" ;
        double wet_density(Time, nbins, bottom_top, south_north, west_east) ;
                wet_density:units = "kg/m3" ;
                wet_density:long_name = "Wet dennsity" ;
        double tem2(Time, south_north, west_east) ;
                tem2:units = "K" ;
```

```
                tem2:long_name = "2m temperature" ;
        double hght(Time, south_north, west_east) ;
                hght:units = "m" ;
                hght:long_name = "Boundary layer height" ;
        double u10m(Time, south_north, west_east) ;
                u10m:units = "m/s" ;
                u10m:long_name = "10m zonal wind speed" ;
        double v10m(Time, south_north, west_east) ;
                v10m:units = "m/s" ;
                v10m:long_name = "10m meridional wind speed" ;
        double w10m(Time, south_north, west_east) ;
                w10m:units = "m/s" ;
                w10m:long_name = "10m vertical wind speed" ;
        double usta(Time, south_north, west_east) ;
                usta:units = "m/s" ;
                usta:long_name = "Friction velocity u*" ;
        double wsta(Time, south_north, west_east) ;
                wsta:units = "m/s" ;
                wsta:long_name = "Convection velocity w*" ;
        double soim(Time, south_north, west_east) ;
                soim:units = "m^3/m^3" ;
                soim:long_name = "Soil moisture" ;
        double sreh(Time, south_north, west_east) ;
                sreh:units = "%" ;
                sreh:long_name = "Surface relative humidity" ;
        double sshf(Time, south_north, west_east) ;
                sshf:units = "W/m2" ;
                sshf:long_name = "Surface sensible heat flux" ;
        double slhf(Time, south_north, west_east) ;
                slhf:units = "W/m2" ;
                slhf:long_name = "Surface latent heat flux" ;
        double copc(Time, south_north, west_east) ;
                copc:units = "mm/h" ;
                copc:long_name = "Convective precipitation" ;
        double lspc(Time, south_north, west_east) ;
                lspc:units = "kg/m^2/h" ;
                lspc:long_name = "Large scale precipitation" ;
        double topc(Time, south_north, west_east) ;
                topc:units = "kg/m^2/h" ;
                topc:long_name = "Precipitation" ;
        double aerr(Time, south_north, west_east) ;
                aerr:units = "s/m" ;
                aerr:long_name = "Aerodynamic resistance" ;
        double atte(Time, south_north, west_east) ;
                atte:units = "[0-1]" ;
                atte:long_name = "Radiation attenuation factor" ;
        double swrd(Time, south_north, west_east) ;
                swrd:units = "W/m2" ;
                swrd:long_name = "Shortwave radiation flux" ;
        double albedo(Time, south_north, west_east) ;
                albedo:units = "-" ;
                albedo:long_name = "Surface albedo" ;
        double lastrain(Time, south_north, west_east) ;
                lastrain:units = "h" ;
                lastrain:long_name = "Time since last rain" ;
        double flashnum(Time, south_north, west_east) ;
                flashnum:units = "#/mn/km2" ;
                flashnum:long_name = "Number of lighning flashes" ;

// global attributes:
                :_NCProperties = "version=1|netcdflibversion=4.6.0|hdf5libversion=1.8.20" ;
                :Title = "CHIMERE SUITE" ;
                :Sub-title = "Hourly output concentrations file" ;
                :Chimere_type = "out" ;
                :Generating_process = "Generated by chimere" ;
                :Conventions = "None" ;
                :Domain = "MED60" ;
}
```

E.4 The results file for deposition fluxes "dep.[label].nc"

```
netcdf dep.2014072900_2014080300_MED60o1_nc4 {
dimensions:
        Time = UNLIMITED ; // (20 currently)
        DateStrLen = 19 ;
        SpStrLen = 23 ;
        west_east = 159 ;
```

```
        south_north = 109 ;
        bottom_top = 20 ;
        nbins = 10 ;
        ninterv = 11 ;
        nmineralchem = 8 ;
        nhcumdepo = 6 ;
variables:
        double ninterv(ninterv) ;
                ninterv:units = "meters" ;
                ninterv:long_name = "ninterv" ;
        double mmd(nbins) ;
                mmd:units = "meters" ;
                mmd:long_name = "Mean_Mass_Median_Diameters" ;
        double lon(south_north, west_east) ;
                lon:units = "degrees_east" ;
                lon:long_name = "Longitude" ;
        double lat(south_north, west_east) ;
                lat:units = "degrees_north" ;
                lat:long_name = "Latitude" ;
        char Times(Time, DateStrLen) ;
                Times:units = "Date string" ;
                Times:long_name = "WRF-style Times" ;
        double O3_gasVd(Time, south_north, west_east) ;
                O3_gasVd:units = "cm s-1" ;
                O3_gasVd:long_name = "O3_gasDRY dry deposition velocity" ;
        double SO2_gasVd(Time, south_north, west_east) ;
                SO2_gasVd:units = "cm s-1" ;
                SO2_gasVd:long_name = "SO2_gasDRY dry deposition velocity" ;
        double NO2_gasVd(Time, south_north, west_east) ;
                NO2_gasVd:units = "cm s-1" ;
                NO2_gasVd:long_name = "NO2_gasDRY dry deposition velocity" ;
        double NO_gasVd(Time, south_north, west_east) ;
                NO_gasVd:units = "cm s-1" ;
                NO_gasVd:long_name = "NO_gasDRY dry deposition velocity" ;
        double NH3_gasVd(Time, south_north, west_east) ;
                NH3_gasVd:units = "cm s-1" ;
                NH3_gasVd:long_name = "NH3_gasDRY dry deposition velocity" ;
        double HCHO_gasVd(Time, south_north, west_east) ;
                HCHO_gasVd:units = "cm s-1" ;
                HCHO_gasVd:long_name = "HCHO_gasDRY dry deposition velocity" ;
        double CH3CHO_gasVd(Time, south_north, west_east) ;
                CH3CHO_gasVd:units = "cm s-1" ;
                CH3CHO_gasVd:long_name = "CH3CHO_gasDRY dry deposition velocity" ;
        double CH3COE_gasVd(Time, south_north, west_east) ;
                CH3COE_gasVd:units = "cm s-1" ;
                CH3COE_gasVd:long_name = "CH3COE_gasDRY dry deposition velocity" ;
        double CH3COY_gasVd(Time, south_north, west_east) ;
                CH3COY_gasVd:units = "cm s-1" ;
                CH3COY_gasVd:long_name = "CH3COY_gasDRY dry deposition velocity" ;
        double GLYOX_gasVd(Time, south_north, west_east) ;
                GLYOX_gasVd:units = "cm s-1" ;
                GLYOX_gasVd:long_name = "GLYOX_gasDRY dry deposition velocity" ;
        double MGLYOX_gasVd(Time, south_north, west_east) ;
                MGLYOX_gasVd:units = "cm s-1" ;
                MGLYOX_gasVd:long_name = "MGLYOX_gasDRY dry deposition velocity" ;
        double MEMALD_gasVd(Time, south_north, west_east) ;
                MEMALD_gasVd:units = "cm s-1" ;
                MEMALD_gasVd:long_name = "MEMALD_gasDRY dry deposition velocity" ;
        double MVK_gasVd(Time, south_north, west_east) ;
                MVK_gasVd:units = "cm s-1" ;
                MVK_gasVd:long_name = "MVK_gasDRY dry deposition velocity" ;
        double MAC_gasVd(Time, south_north, west_east) ;
                MAC_gasVd:units = "cm s-1" ;
                MAC_gasVd:long_name = "MAC_gasDRY dry deposition velocity" ;
        double PAN_gasVd(Time, south_north, west_east) ;
                PAN_gasVd:units = "cm s-1" ;
                PAN_gasVd:long_name = "PAN_gasDRY dry deposition velocity" ;
        double toPAN_gasVd(Time, south_north, west_east) ;
                toPAN_gasVd:units = "cm s-1" ;
                toPAN_gasVd:long_name = "toPAN_gasDRY dry deposition velocity" ;
        double HONO_gasVd(Time, south_north, west_east) ;
                HONO_gasVd:units = "cm s-1" ;
                HONO_gasVd:long_name = "HONO_gasDRY dry deposition velocity" ;
        double HNO3_gasVd(Time, south_north, west_east) ;
                HNO3_gasVd:units = "cm s-1" ;
                HNO3_gasVd:long_name = "HNO3_gasDRY dry deposition velocity" ;
        double H2O2_gasVd(Time, south_north, west_east) ;
                H2O2_gasVd:units = "cm s-1" ;
                H2O2_gasVd:long_name = "H2O2_gasDRY dry deposition velocity" ;
        double CH3O2H_gasVd(Time, south_north, west_east) ;
                CH3O2H_gasVd:units = "cm s-1" ;
```

```
        CH3O2H_gasVd:long_name = "CH3O2H_gasDRY dry deposition velocity" ;
    double PPA_gasVd(Time, south_north, west_east) ;
        PPA_gasVd:units = "cm s-1" ;
        PPA_gasVd:long_name = "PPA_gasDRY dry deposition velocity" ;
    double obioH_gasVd(Time, south_north, west_east) ;
        obioH_gasVd:units = "cm s-1" ;
        obioH_gasVd:long_name = "obioH_gasDRY dry deposition velocity" ;
    double oROOH_gasVd(Time, south_north, west_east) ;
        oROOH_gasVd:units = "cm s-1" ;
        oROOH_gasVd:long_name = "oROOH_gasDRY dry deposition velocity" ;
    double PANH_gasVd(Time, south_north, west_east) ;
        PANH_gasVd:units = "cm s-1" ;
        PANH_gasVd:long_name = "PANH_gasDRY dry deposition velocity" ;
    double AnA1D_gasVd(Time, south_north, west_east) ;
        AnA1D_gasVd:units = "cm s-1" ;
        AnA1D_gasVd:long_name = "AnA1D_gasDRY dry deposition velocity" ;
    double ISOPA1_gasVd(Time, south_north, west_east) ;
        ISOPA1_gasVd:units = "cm s-1" ;
        ISOPA1_gasVd:long_name = "ISOPA1_gasDRY dry deposition velocity" ;
    double BiA1D_gasVd(Time, south_north, west_east) ;
        BiA1D_gasVd:units = "cm s-1" ;
        BiA1D_gasVd:long_name = "BiA1D_gasDRY dry deposition velocity" ;
    double AnBmP_gasVd(Time, south_north, west_east) ;
        AnBmP_gasVd:units = "cm s-1" ;
        AnBmP_gasVd:long_name = "AnBmP_gasDRY dry deposition velocity" ;
    double BiBmP_gasVd(Time, south_north, west_east) ;
        BiBmP_gasVd:units = "cm s-1" ;
        BiBmP_gasVd:long_name = "BiBmP_gasDRY dry deposition velocity" ;
    double O3_gasDRY(Time, south_north, west_east) ;
        O3_gasDRY:units = "g/cm2/6hours" ;
        O3_gasDRY:long_name = "O3_gasDRY deposition flux" ;
    double SO2_gasDRY(Time, south_north, west_east) ;
        SO2_gasDRY:units = "g/cm2/6hours" ;
        SO2_gasDRY:long_name = "SO2_gasDRY deposition flux" ;
    double NO2_gasDRY(Time, south_north, west_east) ;
        NO2_gasDRY:units = "g/cm2/6hours" ;
        NO2_gasDRY:long_name = "NO2_gasDRY deposition flux" ;
    double NO_gasDRY(Time, south_north, west_east) ;
        NO_gasDRY:units = "g/cm2/6hours" ;
        NO_gasDRY:long_name = "NO_gasDRY deposition flux" ;
    double NH3_gasDRY(Time, south_north, west_east) ;
        NH3_gasDRY:units = "g/cm2/6hours" ;
        NH3_gasDRY:long_name = "NH3_gasDRY deposition flux" ;
    double HCHO_gasDRY(Time, south_north, west_east) ;
        HCHO_gasDRY:units = "g/cm2/6hours" ;
        HCHO_gasDRY:long_name = "HCHO_gasDRY deposition flux" ;
    double CH3CHO_gasDRY(Time, south_north, west_east) ;
        CH3CHO_gasDRY:units = "g/cm2/6hours" ;
        CH3CHO_gasDRY:long_name = "CH3CHO_gasDRY deposition flux" ;
    double CH3COE_gasDRY(Time, south_north, west_east) ;
        CH3COE_gasDRY:units = "g/cm2/6hours" ;
        CH3COE_gasDRY:long_name = "CH3COE_gasDRY deposition flux" ;
    double CH3COY_gasDRY(Time, south_north, west_east) ;
        CH3COY_gasDRY:units = "g/cm2/6hours" ;
        CH3COY_gasDRY:long_name = "CH3COY_gasDRY deposition flux" ;
    double GLYOX_gasDRY(Time, south_north, west_east) ;
        GLYOX_gasDRY:units = "g/cm2/6hours" ;
        GLYOX_gasDRY:long_name = "GLYOX_gasDRY deposition flux" ;
    double MGLYOX_gasDRY(Time, south_north, west_east) ;
        MGLYOX_gasDRY:units = "g/cm2/6hours" ;
        MGLYOX_gasDRY:long_name = "MGLYOX_gasDRY deposition flux" ;
    double MEMALD_gasDRY(Time, south_north, west_east) ;
        MEMALD_gasDRY:units = "g/cm2/6hours" ;
        MEMALD_gasDRY:long_name = "MEMALD_gasDRY deposition flux" ;
    double MVK_gasDRY(Time, south_north, west_east) ;
        MVK_gasDRY:units = "g/cm2/6hours" ;
        MVK_gasDRY:long_name = "MVK_gasDRY deposition flux" ;
    double MAC_gasDRY(Time, south_north, west_east) ;
        MAC_gasDRY:units = "g/cm2/6hours" ;
        MAC_gasDRY:long_name = "MAC_gasDRY deposition flux" ;
    double PAN_gasDRY(Time, south_north, west_east) ;
        PAN_gasDRY:units = "g/cm2/6hours" ;
        PAN_gasDRY:long_name = "PAN_gasDRY deposition flux" ;
    double toPAN_gasDRY(Time, south_north, west_east) ;
        toPAN_gasDRY:units = "g/cm2/6hours" ;
        toPAN_gasDRY:long_name = "toPAN_gasDRY deposition flux" ;
    double HONO_gasDRY(Time, south_north, west_east) ;
        HONO_gasDRY:units = "g/cm2/6hours" ;
        HONO_gasDRY:long_name = "HONO_gasDRY deposition flux" ;
    double HNO3_gasDRY(Time, south_north, west_east) ;
        HNO3_gasDRY:units = "g/cm2/6hours" ;
```

```
                HNO3_gasDRY:long_name = "HNO3_gasDRY deposition flux" ;
        double H2O2_gasDRY(Time, south_north, west_east) ;
                H2O2_gasDRY:units = "g/cm2/6hours" ;
                H2O2_gasDRY:long_name = "H2O2_gasDRY deposition flux" ;
        double CH3O2H_gasDRY(Time, south_north, west_east) ;
                CH3O2H_gasDRY:units = "g/cm2/6hours" ;
                CH3O2H_gasDRY:long_name = "CH3O2H_gasDRY deposition flux" ;
        double PPA_gasDRY(Time, south_north, west_east) ;
                PPA_gasDRY:units = "g/cm2/6hours" ;
                PPA_gasDRY:long_name = "PPA_gasDRY deposition flux" ;
        double obioH_gasDRY(Time, south_north, west_east) ;
                obioH_gasDRY:units = "g/cm2/6hours" ;
                obioH_gasDRY:long_name = "obioH_gasDRY deposition flux" ;
        double oROOH_gasDRY(Time, south_north, west_east) ;
                oROOH_gasDRY:units = "g/cm2/6hours" ;
                oROOH_gasDRY:long_name = "oROOH_gasDRY deposition flux" ;
        double PANH_gasDRY(Time, south_north, west_east) ;
                PANH_gasDRY:units = "g/cm2/6hours" ;
                PANH_gasDRY:long_name = "PANH_gasDRY deposition flux" ;
        double AnA1D_gasDRY(Time, south_north, west_east) ;
                AnA1D_gasDRY:units = "g/cm2/6hours" ;
                AnA1D_gasDRY:long_name = "AnA1D_gasDRY deposition flux" ;
        double ISOPA1_gasDRY(Time, south_north, west_east) ;
                ISOPA1_gasDRY:units = "g/cm2/6hours" ;
                ISOPA1_gasDRY:long_name = "ISOPA1_gasDRY deposition flux" ;
        double BiA1D_gasDRY(Time, south_north, west_east) ;
                BiA1D_gasDRY:units = "g/cm2/6hours" ;
                BiA1D_gasDRY:long_name = "BiA1D_gasDRY deposition flux" ;
        double AnBmP_gasDRY(Time, south_north, west_east) ;
                AnBmP_gasDRY:units = "g/cm2/6hours" ;
                AnBmP_gasDRY:long_name = "AnBmP_gasDRY deposition flux" ;
        double BiBmP_gasDRY(Time, south_north, west_east) ;
                BiBmP_gasDRY:units = "g/cm2/6hours" ;
                BiBmP_gasDRY:long_name = "BiBmP_gasDRY deposition flux" ;
        double HNO3_gasWET(Time, south_north, west_east) ;
                HNO3_gasWET:units = "g/cm2/6hours" ;
                HNO3_gasWET:long_name = "HNO3_gasWET deposition flux" ;
        double NH3_gasWET(Time, south_north, west_east) ;
                NH3_gasWET:units = "g/cm2/6hours" ;
                NH3_gasWET:long_name = "NH3_gasWET deposition flux" ;
        double H2O2_gasWET(Time, south_north, west_east) ;
                H2O2_gasWET:units = "g/cm2/6hours" ;
                H2O2_gasWET:long_name = "H2O2_gasWET deposition flux" ;
        double H2SO4_gasWET(Time, south_north, west_east) ;
                H2SO4_gasWET:units = "g/cm2/6hours" ;
                H2SO4_gasWET:long_name = "H2SO4_gasWET deposition flux" ;
        double H2SO4AQ_gasWET(Time, south_north, west_east) ;
                H2SO4AQ_gasWET:units = "g/cm2/6hours" ;
                H2SO4AQ_gasWET:long_name = "H2SO4AQ_gasWET deposition flux" ;
        double SO2_gasWET(Time, south_north, west_east) ;
                SO2_gasWET:units = "g/cm2/6hours" ;
                SO2_gasWET:long_name = "SO2_gasWET deposition flux" ;
        double BiA1D_gasWET(Time, south_north, west_east) ;
                BiA1D_gasWET:units = "g/cm2/6hours" ;
                BiA1D_gasWET:long_name = "BiA1D_gasWET deposition flux" ;
        double BiBmP_gasWET(Time, south_north, west_east) ;
                BiBmP_gasWET:units = "g/cm2/6hours" ;
                BiBmP_gasWET:long_name = "BiBmP_gasWET deposition flux" ;
        double O3_gasWET(Time, south_north, west_east) ;
                O3_gasWET:units = "g/cm2/6hours" ;
                O3_gasWET:long_name = "O3_gasWET deposition flux" ;
        double NO_gasWET(Time, south_north, west_east) ;
                NO_gasWET:units = "g/cm2/6hours" ;
                NO_gasWET:long_name = "NO_gasWET deposition flux" ;
        double NO2_gasWET(Time, south_north, west_east) ;
                NO2_gasWET:units = "g/cm2/6hours" ;
                NO2_gasWET:long_name = "NO2_gasWET deposition flux" ;
        double NO3_gasWET(Time, south_north, west_east) ;
                NO3_gasWET:units = "g/cm2/6hours" ;
                NO3_gasWET:long_name = "NO3_gasWET deposition flux" ;
        double HONO_gasWET(Time, south_north, west_east) ;
                HONO_gasWET:units = "g/cm2/6hours" ;
                HONO_gasWET:long_name = "HONO_gasWET deposition flux" ;
        double OH_gasWET(Time, south_north, west_east) ;
                OH_gasWET:units = "g/cm2/6hours" ;
                OH_gasWET:long_name = "OH_gasWET deposition flux" ;
        double HCHO_gasWET(Time, south_north, west_east) ;
                HCHO_gasWET:units = "g/cm2/6hours" ;
                HCHO_gasWET:long_name = "HCHO_gasWET deposition flux" ;
        double PPA_gasWET(Time, south_north, west_east) ;
                PPA_gasWET:units = "g/cm2/6hours" ;
```

```
        PPA_gasWET:long_name = "PPA_gasWET deposition flux" ;
    double GLYOX_gasWET(Time, south_north, west_east) ;
        GLYOX_gasWET:units = "g/cm2/6hours" ;
        GLYOX_gasWET:long_name = "GLYOX_gasWET deposition flux" ;
    double MGLYOX_gasWET(Time, south_north, west_east) ;
        MGLYOX_gasWET:units = "g/cm2/6hours" ;
        MGLYOX_gasWET:long_name = "MGLYOX_gasWET deposition flux" ;
    double CH3CHO_gasWET(Time, south_north, west_east) ;
        CH3CHO_gasWET:units = "g/cm2/6hours" ;
        CH3CHO_gasWET:long_name = "CH3CHO_gasWET deposition flux" ;
    double AnBmP_gasWET(Time, south_north, west_east) ;
        AnBmP_gasWET:units = "g/cm2/6hours" ;
        AnBmP_gasWET:long_name = "AnBmP_gasWET deposition flux" ;
    double DUSTAQ_gasWET(Time, south_north, west_east) ;
        DUSTAQ_gasWET:units = "g/cm2/6hours" ;
        DUSTAQ_gasWET:long_name = "DUSTAQ_gasWET deposition flux" ;
    double aerr(Time, south_north, west_east) ;
        aerr:units = "s/m" ;
        aerr:long_name = "Aerodynamic resistance" ;
    double topc(Time, south_north, west_east) ;
        topc:units = "mm" ;
        topc:long_name = "Cumulated precipitation" ;
    double vsed(Time, nbins, bottom_top, south_north, west_east) ;
        vsed:units = "cm s-1" ;
        vsed:long_name = "Settling velocity" ;
    double vaero(Time, nbins, south_north, west_east) ;
        vaero:units = "cm s-1" ;
        vaero:long_name = "Aerosol bins deposition velocity" ;
    double BCAR_aerbinDRY(Time, nbins, south_north, west_east) ;
        BCAR_aerbinDRY:units = "g/cm2/6hours" ;
        BCAR_aerbinDRY:long_name = "BCAR_aerbinDRY aero deposition flux" ;
    double DUST_aerbinDRY(Time, nbins, south_north, west_east) ;
        DUST_aerbinDRY:units = "g/cm2/6hours" ;
        DUST_aerbinDRY:long_name = "DUST_aerbinDRY aero deposition flux" ;
    double OCAR_aerbinDRY(Time, nbins, south_north, west_east) ;
        OCAR_aerbinDRY:units = "g/cm2/6hours" ;
        OCAR_aerbinDRY:long_name = "OCAR_aerbinDRY aero deposition flux" ;
    double PPM_aerbinDRY(Time, nbins, south_north, west_east) ;
        PPM_aerbinDRY:units = "g/cm2/6hours" ;
        PPM_aerbinDRY:long_name = "PPM_aerbinDRY aero deposition flux" ;
    double SALT_aerbinDRY(Time, nbins, south_north, west_east) ;
        SALT_aerbinDRY:units = "g/cm2/6hours" ;
        SALT_aerbinDRY:long_name = "SALT_aerbinDRY aero deposition flux" ;
    double AnA1D_aerbinDRY(Time, nbins, south_north, west_east) ;
        AnA1D_aerbinDRY:units = "g/cm2/6hours" ;
        AnA1D_aerbinDRY:long_name = "AnA1D_aerbinDRY aero deposition flux" ;
    double AnBmP_aerbinDRY(Time, nbins, south_north, west_east) ;
        AnBmP_aerbinDRY:units = "g/cm2/6hours" ;
        AnBmP_aerbinDRY:long_name = "AnBmP_aerbinDRY aero deposition flux" ;
    double BiA1D_aerbinDRY(Time, nbins, south_north, west_east) ;
        BiA1D_aerbinDRY:units = "g/cm2/6hours" ;
        BiA1D_aerbinDRY:long_name = "BiA1D_aerbinDRY aero deposition flux" ;
    double BiBmP_aerbinDRY(Time, nbins, south_north, west_east) ;
        BiBmP_aerbinDRY:units = "g/cm2/6hours" ;
        BiBmP_aerbinDRY:long_name = "BiBmP_aerbinDRY aero deposition flux" ;
    double ISOPA1_aerbinDRY(Time, nbins, south_north, west_east) ;
        ISOPA1_aerbinDRY:units = "g/cm2/6hours" ;
        ISOPA1_aerbinDRY:long_name = "ISOPA1_aerbinDRY aero deposition flux" ;
    double H2SO4_aerbinDRY(Time, nbins, south_north, west_east) ;
        H2SO4_aerbinDRY:units = "g/cm2/6hours" ;
        H2SO4_aerbinDRY:long_name = "H2SO4_aerbinDRY aero deposition flux" ;
    double HNO3_aerbinDRY(Time, nbins, south_north, west_east) ;
        HNO3_aerbinDRY:units = "g/cm2/6hours" ;
        HNO3_aerbinDRY:long_name = "HNO3_aerbinDRY aero deposition flux" ;
    double NH3_aerbinDRY(Time, nbins, south_north, west_east) ;
        NH3_aerbinDRY:units = "g/cm2/6hours" ;
        NH3_aerbinDRY:long_name = "NH3_aerbinDRY aero deposition flux" ;
    double WATER_aerbinDRY(Time, nbins, south_north, west_east) ;
        WATER_aerbinDRY:units = "g/cm2/6hours" ;
        WATER_aerbinDRY:long_name = "WATER_aerbinDRY aero deposition flux" ;
    double BCAR_aerbinWET(Time, nbins, south_north, west_east) ;
        BCAR_aerbinWET:units = "g/cm2/6hours" ;
        BCAR_aerbinWET:long_name = "BCAR_aerbinWET aero deposition flux" ;
    double DUST_aerbinWET(Time, nbins, south_north, west_east) ;
        DUST_aerbinWET:units = "g/cm2/6hours" ;
        DUST_aerbinWET:long_name = "DUST_aerbinWET aero deposition flux" ;
    double OCAR_aerbinWET(Time, nbins, south_north, west_east) ;
        OCAR_aerbinWET:units = "g/cm2/6hours" ;
        OCAR_aerbinWET:long_name = "OCAR_aerbinWET aero deposition flux" ;
    double PPM_aerbinWET(Time, nbins, south_north, west_east) ;
        PPM_aerbinWET:units = "g/cm2/6hours" ;
```

```
        PPM_aerbinWET:long_name = "PPM_aerbinWET aero deposition flux" ;
    double SALT_aerbinWET(Time, nbins, south_north, west_east) ;
        SALT_aerbinWET:units = "g/cm2/6hours" ;
        SALT_aerbinWET:long_name = "SALT_aerbinWET aero deposition flux" ;
    double AnA1D_aerbinWET(Time, nbins, south_north, west_east) ;
        AnA1D_aerbinWET:units = "g/cm2/6hours" ;
        AnA1D_aerbinWET:long_name = "AnA1D_aerbinWET aero deposition flux" ;
    double AnBmP_aerbinWET(Time, nbins, south_north, west_east) ;
        AnBmP_aerbinWET:units = "g/cm2/6hours" ;
        AnBmP_aerbinWET:long_name = "AnBmP_aerbinWET aero deposition flux" ;
    double BiA1D_aerbinWET(Time, nbins, south_north, west_east) ;
        BiA1D_aerbinWET:units = "g/cm2/6hours" ;
        BiA1D_aerbinWET:long_name = "BiA1D_aerbinWET aero deposition flux" ;
    double BiBmP_aerbinWET(Time, nbins, south_north, west_east) ;
        BiBmP_aerbinWET:units = "g/cm2/6hours" ;
        BiBmP_aerbinWET:long_name = "BiBmP_aerbinWET aero deposition flux" ;
    double ISOPA1_aerbinWET(Time, nbins, south_north, west_east) ;
        ISOPA1_aerbinWET:units = "g/cm2/6hours" ;
        ISOPA1_aerbinWET:long_name = "ISOPA1_aerbinWET aero deposition flux" ;
    double H2SO4_aerbinWET(Time, nbins, south_north, west_east) ;
        H2SO4_aerbinWET:units = "g/cm2/6hours" ;
        H2SO4_aerbinWET:long_name = "H2SO4_aerbinWET aero deposition flux" ;
    double HNO3_aerbinWET(Time, nbins, south_north, west_east) ;
        HNO3_aerbinWET:units = "g/cm2/6hours" ;
        HNO3_aerbinWET:long_name = "HNO3_aerbinWET aero deposition flux" ;
    double NH3_aerbinWET(Time, nbins, south_north, west_east) ;
        NH3_aerbinWET:units = "g/cm2/6hours" ;
        NH3_aerbinWET:long_name = "NH3_aerbinWET aero deposition flux" ;
    double WATER_aerbinWET(Time, nbins, south_north, west_east) ;
        WATER_aerbinWET:units = "g/cm2/6hours" ;
        WATER_aerbinWET:long_name = "WATER_aerbinWET aero deposition flux" ;
    double BCAR_aerDRY(Time, south_north, west_east) ;
        BCAR_aerDRY:units = "g/cm2/6hours" ;
        BCAR_aerDRY:long_name = "BCAR_aerDRY aero total deposition flux" ;
    double DUST_aerDRY(Time, south_north, west_east) ;
        DUST_aerDRY:units = "g/cm2/6hours" ;
        DUST_aerDRY:long_name = "DUST_aerDRY aero total deposition flux" ;
    double OCAR_aerDRY(Time, south_north, west_east) ;
        OCAR_aerDRY:units = "g/cm2/6hours" ;
        OCAR_aerDRY:long_name = "OCAR_aerDRY aero total deposition flux" ;
    double PPM_aerDRY(Time, south_north, west_east) ;
        PPM_aerDRY:units = "g/cm2/6hours" ;
        PPM_aerDRY:long_name = "PPM_aerDRY aero total deposition flux" ;
    double SALT_aerDRY(Time, south_north, west_east) ;
        SALT_aerDRY:units = "g/cm2/6hours" ;
        SALT_aerDRY:long_name = "SALT_aerDRY aero total deposition flux" ;
    double AnA1D_aerDRY(Time, south_north, west_east) ;
        AnA1D_aerDRY:units = "g/cm2/6hours" ;
        AnA1D_aerDRY:long_name = "AnA1D_aerDRY aero total deposition flux" ;
    double AnBmP_aerDRY(Time, south_north, west_east) ;
        AnBmP_aerDRY:units = "g/cm2/6hours" ;
        AnBmP_aerDRY:long_name = "AnBmP_aerDRY aero total deposition flux" ;
    double BiA1D_aerDRY(Time, south_north, west_east) ;
        BiA1D_aerDRY:units = "g/cm2/6hours" ;
        BiA1D_aerDRY:long_name = "BiA1D_aerDRY aero total deposition flux" ;
    double BiBmP_aerDRY(Time, south_north, west_east) ;
        BiBmP_aerDRY:units = "g/cm2/6hours" ;
        BiBmP_aerDRY:long_name = "BiBmP_aerDRY aero total deposition flux" ;
    double ISOPA1_aerDRY(Time, south_north, west_east) ;
        ISOPA1_aerDRY:units = "g/cm2/6hours" ;
        ISOPA1_aerDRY:long_name = "ISOPA1_aerDRY aero total deposition flux" ;
    double H2SO4_aerDRY(Time, south_north, west_east) ;
        H2SO4_aerDRY:units = "g/cm2/6hours" ;
        H2SO4_aerDRY:long_name = "H2SO4_aerDRY aero total deposition flux" ;
    double HNO3_aerDRY(Time, south_north, west_east) ;
        HNO3_aerDRY:units = "g/cm2/6hours" ;
        HNO3_aerDRY:long_name = "HNO3_aerDRY aero total deposition flux" ;
    double NH3_aerDRY(Time, south_north, west_east) ;
        NH3_aerDRY:units = "g/cm2/6hours" ;
        NH3_aerDRY:long_name = "NH3_aerDRY aero total deposition flux" ;
    double WATER_aerDRY(Time, south_north, west_east) ;
        WATER_aerDRY:units = "g/cm2/6hours" ;
        WATER_aerDRY:long_name = "WATER_aerDRY aero total deposition flux" ;
    double BCAR_aerWET(Time, south_north, west_east) ;
        BCAR_aerWET:units = "g/cm2/6hours" ;
        BCAR_aerWET:long_name = "BCAR_aerWET aero total deposition flux" ;
    double DUST_aerWET(Time, south_north, west_east) ;
        DUST_aerWET:units = "g/cm2/6hours" ;
        DUST_aerWET:long_name = "DUST_aerWET aero total deposition flux" ;
    double OCAR_aerWET(Time, south_north, west_east) ;
        OCAR_aerWET:units = "g/cm2/6hours" ;
```

```
                OCAR_aerWET:long_name = "OCAR_aerWET aero total deposition flux" ;
        double PPM_aerWET(Time, south_north, west_east) ;
                PPM_aerWET:units = "g/cm2/6hours" ;
                PPM_aerWET:long_name = "PPM_aerWET aero total deposition flux" ;
        double SALT_aerWET(Time, south_north, west_east) ;
                SALT_aerWET:units = "g/cm2/6hours" ;
                SALT_aerWET:long_name = "SALT_aerWET aero total deposition flux" ;
        double AnA1D_aerWET(Time, south_north, west_east) ;
                AnA1D_aerWET:units = "g/cm2/6hours" ;
                AnA1D_aerWET:long_name = "AnA1D_aerWET aero total deposition flux" ;
        double AnBmP_aerWET(Time, south_north, west_east) ;
                AnBmP_aerWET:units = "g/cm2/6hours" ;
                AnBmP_aerWET:long_name = "AnBmP_aerWET aero total deposition flux" ;
        double BiA1D_aerWET(Time, south_north, west_east) ;
                BiA1D_aerWET:units = "g/cm2/6hours" ;
                BiA1D_aerWET:long_name = "BiA1D_aerWET aero total deposition flux" ;
        double BiBmP_aerWET(Time, south_north, west_east) ;
                BiBmP_aerWET:units = "g/cm2/6hours" ;
                BiBmP_aerWET:long_name = "BiBmP_aerWET aero total deposition flux" ;
        double ISOPA1_aerWET(Time, south_north, west_east) ;
                ISOPA1_aerWET:units = "g/cm2/6hours" ;
                ISOPA1_aerWET:long_name = "ISOPA1_aerWET aero total deposition flux" ;
        double H2SO4_aerWET(Time, south_north, west_east) ;
                H2SO4_aerWET:units = "g/cm2/6hours" ;
                H2SO4_aerWET:long_name = "H2SO4_aerWET aero total deposition flux" ;
        double HNO3_aerWET(Time, south_north, west_east) ;
                HNO3_aerWET:units = "g/cm2/6hours" ;
                HNO3_aerWET:long_name = "HNO3_aerWET aero total deposition flux" ;
        double NH3_aerWET(Time, south_north, west_east) ;
                NH3_aerWET:units = "g/cm2/6hours" ;
                NH3_aerWET:long_name = "NH3_aerWET aero total deposition flux" ;
        double WATER_aerWET(Time, south_north, west_east) ;
                WATER_aerWET:units = "g/cm2/6hours" ;
                WATER_aerWET:long_name = "WATER_aerWET aero total deposition flux" ;
        double BCAR_aerDRYandWET(Time, south_north, west_east) ;
                BCAR_aerDRYandWET:units = "g/cm2/6hours" ;
                BCAR_aerDRYandWET:long_name = "BCAR_aerDRYandWET aero total deposition flux" ;
        double DUST_aerDRYandWET(Time, south_north, west_east) ;
                DUST_aerDRYandWET:units = "g/cm2/6hours" ;
                DUST_aerDRYandWET:long_name = "DUST_aerDRYandWET aero total deposition flux" ;
        double OCAR_aerDRYandWET(Time, south_north, west_east) ;
                OCAR_aerDRYandWET:units = "g/cm2/6hours" ;
                OCAR_aerDRYandWET:long_name = "OCAR_aerDRYandWET aero total deposition flux" ;
        double PPM_aerDRYandWET(Time, south_north, west_east) ;
                PPM_aerDRYandWET:units = "g/cm2/6hours" ;
                PPM_aerDRYandWET:long_name = "PPM_aerDRYandWET aero total deposition flux" ;
        double SALT_aerDRYandWET(Time, south_north, west_east) ;
                SALT_aerDRYandWET:units = "g/cm2/6hours" ;
                SALT_aerDRYandWET:long_name = "SALT_aerDRYandWET aero total deposition flux" ;
        double AnA1D_aerDRYandWET(Time, south_north, west_east) ;
                AnA1D_aerDRYandWET:units = "g/cm2/6hours" ;
                AnA1D_aerDRYandWET:long_name = "AnA1D_aerDRYandWET aero total deposition flux" ;
        double AnBmP_aerDRYandWET(Time, south_north, west_east) ;
                AnBmP_aerDRYandWET:units = "g/cm2/6hours" ;
                AnBmP_aerDRYandWET:long_name = "AnBmP_aerDRYandWET aero total deposition flux" ;
        double BiA1D_aerDRYandWET(Time, south_north, west_east) ;
                BiA1D_aerDRYandWET:units = "g/cm2/6hours" ;
                BiA1D_aerDRYandWET:long_name = "BiA1D_aerDRYandWET aero total deposition flux" ;
        double BiBmP_aerDRYandWET(Time, south_north, west_east) ;
                BiBmP_aerDRYandWET:units = "g/cm2/6hours" ;
                BiBmP_aerDRYandWET:long_name = "BiBmP_aerDRYandWET aero total deposition flux" ;
        double ISOPA1_aerDRYandWET(Time, south_north, west_east) ;
                ISOPA1_aerDRYandWET:units = "g/cm2/6hours" ;
                ISOPA1_aerDRYandWET:long_name = "ISOPA1_aerDRYandWET aero total deposition flux" ;
        double H2SO4_aerDRYandWET(Time, south_north, west_east) ;
                H2SO4_aerDRYandWET:units = "g/cm2/6hours" ;
                H2SO4_aerDRYandWET:long_name = "H2SO4_aerDRYandWET aero total deposition flux" ;
        double HNO3_aerDRYandWET(Time, south_north, west_east) ;
                HNO3_aerDRYandWET:units = "g/cm2/6hours" ;
                HNO3_aerDRYandWET:long_name = "HNO3_aerDRYandWET aero total deposition flux" ;
        double NH3_aerDRYandWET(Time, south_north, west_east) ;
                NH3_aerDRYandWET:units = "g/cm2/6hours" ;
                NH3_aerDRYandWET:long_name = "NH3_aerDRYandWET aero total deposition flux" ;
        double WATER_aerDRYandWET(Time, south_north, west_east) ;
                WATER_aerDRYandWET:units = "g/cm2/6hours" ;
                WATER_aerDRYandWET:long_name = "WATER_aerDRYandWET aero total deposition flux" ;

// global attributes:
                :_NCProperties = "version=1|netcdflibversion=4.6.0|hdf5libversion=1.8.20" ;
                :Title = "CHIMERE SUITE" ;
                :Sub-title = "Dry and wet deposition fluxes" ;
```

```
                :Chimere_type = "dep" ;
                :Generating_process = "Generated by chimere" ;
                :Conventions = "None" ;
                :Domain = "MED60" ;
}
```

E.5 The results file for restart "end.[label].nc"

```
netcdf end.2012010100_2012010200_GEMSHCL {
dimensions:
        Time = UNLIMITED ; // (7 currently)
        DateStrLen = 19 ;
        SpStrLen = 23 ;
        Species = 296 ;
        west_east = 202 ;
        south_north = 182 ;
        bottom_top = 8 ;
        wavelength_dim = 5 ;
        wvlgthlw_dim = 16 ;
        nlev_wrf = 1 ;
variables:
        double lon(south_north, west_east) ;
                lon:units = "degrees_east" ;
                lon:long_name = "Longitude" ;
        double lat(south_north, west_east) ;
                lat:units = "degrees_north" ;
                lat:long_name = "Latitude" ;
        char species(Species, SpStrLen) ;
                species:units =  ;
                species:long_name = "CHIMERE Species" ;
        char Times(Time, DateStrLen) ;
                Times:units = "Date string" ;
                Times:long_name = "WRF-style Times" ;
        double AACD(Time, bottom_top, south_north, west_east) ;
                AACD:units = "molecules/cm3" ;
                AACD:long_name = "AACD concentration" ;
        double ACET(Time, bottom_top, south_north, west_east) ;
                ACET:units = "molecules/cm3" ;
                ACET:long_name = "ACET concentration" ;
        double ACYE(Time, bottom_top, south_north, west_east) ;
                ACYE:units = "molecules/cm3" ;
                ACYE:long_name = "ACYE concentration" ;
        double AFG1(Time, bottom_top, south_north, west_east) ;
                AFG1:units = "molecules/cm3" ;
                AFG1:long_name = "AFG1 concentration" ;
        double AFG2(Time, bottom_top, south_north, west_east) ;
                AFG2:units = "molecules/cm3" ;
                AFG2:long_name = "AFG2 concentration" ;
        double AFG3(Time, bottom_top, south_north, west_east) ;
                AFG3:units = "molecules/cm3" ;
                AFG3:long_name = "AFG3 concentration" ;
        double ALK1(Time, bottom_top, south_north, west_east) ;
                ALK1:units = "molecules/cm3" ;
                ALK1:long_name = "ALK1 concentration" ;
        double ALK2(Time, bottom_top, south_north, west_east) ;
                ALK2:units = "molecules/cm3" ;
                ALK2:long_name = "ALK2 concentration" ;
        double ALK3(Time, bottom_top, south_north, west_east) ;
                ALK3:units = "molecules/cm3" ;
                ALK3:long_name = "ALK3 concentration" ;
        double ALK4(Time, bottom_top, south_north, west_east) ;
                ALK4:units = "molecules/cm3" ;
                ALK4:long_name = "ALK4 concentration" ;
        double ALK5(Time, bottom_top, south_north, west_east) ;
                ALK5:units = "molecules/cm3" ;
                ALK5:long_name = "ALK5 concentration" ;
        double APINEN(Time, bottom_top, south_north, west_east) ;
                APINEN:units = "molecules/cm3" ;
                APINEN:long_name = "APINEN concentration" ;
        double ARO1(Time, bottom_top, south_north, west_east) ;
                ARO1:units = "molecules/cm3" ;
                ARO1:long_name = "ARO1 concentration" ;
        double ARO2(Time, bottom_top, south_north, west_east) ;
                ARO2:units = "molecules/cm3" ;
                ARO2:long_name = "ARO2 concentration" ;
        double AnA1D(Time, bottom_top, south_north, west_east) ;
                AnA1D:units = "molecules/cm3" ;
```

```
        AnA1D:long_name = "AnA1D concentration" ;
    double AnA1DAQ(Time, bottom_top, south_north, west_east) ;
        AnA1DAQ:units = "molecules/cm3" ;
        AnA1DAQ:long_name = "AnA1DAQ concentration" ;
    double AnBmP(Time, bottom_top, south_north, west_east) ;
        AnBmP:units = "molecules/cm3" ;
        AnBmP:long_name = "AnBmP concentration" ;
    double AnBmPAQ(Time, bottom_top, south_north, west_east) ;
        AnBmPAQ:units = "molecules/cm3" ;
        AnBmPAQ:long_name = "AnBmPAQ concentration" ;
    double BACL(Time, bottom_top, south_north, west_east) ;
        BACL:units = "molecules/cm3" ;
        BACL:long_name = "BACL concentration" ;
    double BALD(Time, bottom_top, south_north, west_east) ;
        BALD:units = "molecules/cm3" ;
        BALD:long_name = "BALD concentration" ;
    double BCARAQ(Time, bottom_top, south_north, west_east) ;
        BCARAQ:units = "molecules/cm3" ;
        BCARAQ:long_name = "BCARAQ concentration" ;
    double BENZ(Time, bottom_top, south_north, west_east) ;
        BENZ:units = "molecules/cm3" ;
        BENZ:long_name = "BENZ concentration" ;
    double BPINEN(Time, bottom_top, south_north, west_east) ;
        BPINEN:units = "molecules/cm3" ;
        BPINEN:long_name = "BPINEN concentration" ;
    double BZCO3(Time, bottom_top, south_north, west_east) ;
        BZCO3:units = "molecules/cm3" ;
        BZCO3:long_name = "BZCO3 concentration" ;
    double BZO(Time, bottom_top, south_north, west_east) ;
        BZO:units = "molecules/cm3" ;
        BZO:long_name = "BZO concentration" ;
    double BiA1D(Time, bottom_top, south_north, west_east) ;
        BiA1D:units = "molecules/cm3" ;
        BiA1D:long_name = "BiA1D concentration" ;
    double BiA1DAQ(Time, bottom_top, south_north, west_east) ;
        BiA1DAQ:units = "molecules/cm3" ;
        BiA1DAQ:long_name = "BiA1DAQ concentration" ;
    double BiBmP(Time, bottom_top, south_north, west_east) ;
        BiBmP:units = "molecules/cm3" ;
        BiBmP:long_name = "BiBmP concentration" ;
    double BiBmPAQ(Time, bottom_top, south_north, west_east) ;
        BiBmPAQ:units = "molecules/cm3" ;
        BiBmPAQ:long_name = "BiBmPAQ concentration" ;
    double C2H4(Time, bottom_top, south_north, west_east) ;
        C2H4:units = "molecules/cm3" ;
        C2H4:long_name = "C2H4 concentration" ;
    double C5H8(Time, bottom_top, south_north, west_east) ;
        C5H8:units = "molecules/cm3" ;
        C5H8:long_name = "C5H8 concentration" ;
    double CCHO(Time, bottom_top, south_north, west_east) ;
        CCHO:units = "molecules/cm3" ;
        CCHO:long_name = "CCHO concentration" ;
    double CH3O2H(Time, bottom_top, south_north, west_east) ;
        CH3O2H:units = "molecules/cm3" ;
        CH3O2H:long_name = "CH3O2H concentration" ;
    double CH4(Time, bottom_top, south_north, west_east) ;
        CH4:units = "molecules/cm3" ;
        CH4:long_name = "CH4 concentration" ;
    double CL(Time, bottom_top, south_north, west_east) ;
        CL:units = "molecules/cm3" ;
        CL:long_name = "CL concentration" ;
    double CL2(Time, bottom_top, south_north, west_east) ;
        CL2:units = "molecules/cm3" ;
        CL2:long_name = "CL2 concentration" ;
    double CLACET(Time, bottom_top, south_north, west_east) ;
        CLACET:units = "molecules/cm3" ;
        CLACET:long_name = "CLACET concentration" ;
    double CLCCHO(Time, bottom_top, south_north, west_east) ;
        CLCCHO:units = "molecules/cm3" ;
        CLCCHO:long_name = "CLCCHO concentration" ;
    double CLNO(Time, bottom_top, south_north, west_east) ;
        CLNO:units = "molecules/cm3" ;
        CLNO:long_name = "CLNO concentration" ;
    double CLNO2(Time, bottom_top, south_north, west_east) ;
        CLNO2:units = "molecules/cm3" ;
        CLNO2:long_name = "CLNO2 concentration" ;
    double CLO(Time, bottom_top, south_north, west_east) ;
        CLO:units = "molecules/cm3" ;
        CLO:long_name = "CLO concentration" ;
    double CLONO(Time, bottom_top, south_north, west_east) ;
        CLONO:units = "molecules/cm3" ;
```

```
        CLONO:long_name = "CLONO concentration" ;
    double CLONO2(Time, bottom_top, south_north, west_east) ;
        CLONO2:units = "molecules/cm3" ;
        CLONO2:long_name = "CLONO2 concentration" ;
    double CO(Time, bottom_top, south_north, west_east) ;
        CO:units = "molecules/cm3" ;
        CO:long_name = "CO concentration" ;
    double COOH(Time, bottom_top, south_north, west_east) ;
        COOH:units = "molecules/cm3" ;
        COOH:long_name = "COOH concentration" ;
    double CRES(Time, bottom_top, south_north, west_east) ;
        CRES:units = "molecules/cm3" ;
        CRES:long_name = "CRES concentration" ;
    double DUSTAQ(Time, bottom_top, south_north, west_east) ;
        DUSTAQ:units = "molecules/cm3" ;
        DUSTAQ:long_name = "DUSTAQ concentration" ;
    double FACD(Time, bottom_top, south_north, west_east) ;
        FACD:units = "molecules/cm3" ;
        FACD:long_name = "FACD concentration" ;
    double GLY(Time, bottom_top, south_north, west_east) ;
        GLY:units = "molecules/cm3" ;
        GLY:long_name = "GLY concentration" ;
    double H2(Time, bottom_top, south_north, west_east) ;
        H2:units = "molecules/cm3" ;
        H2:long_name = "H2 concentration" ;
    double H2O2(Time, bottom_top, south_north, west_east) ;
        H2O2:units = "molecules/cm3" ;
        H2O2:long_name = "H2O2 concentration" ;
    double H2SO4(Time, bottom_top, south_north, west_east) ;
        H2SO4:units = "molecules/cm3" ;
        H2SO4:long_name = "H2SO4 concentration" ;
    double H2SO4AQ(Time, bottom_top, south_north, west_east) ;
        H2SO4AQ:units = "molecules/cm3" ;
        H2SO4AQ:long_name = "H2SO4AQ concentration" ;
    double HCHO(Time, bottom_top, south_north, west_east) ;
        HCHO:units = "molecules/cm3" ;
        HCHO:long_name = "HCHO concentration" ;
    double HCL(Time, bottom_top, south_north, west_east) ;
        HCL:units = "molecules/cm3" ;
        HCL:long_name = "HCL concentration" ;
    double HCLAQ(Time, bottom_top, south_north, west_east) ;
        HCLAQ:units = "molecules/cm3" ;
        HCLAQ:long_name = "HCLAQ concentration" ;
    double HNO3(Time, bottom_top, south_north, west_east) ;
        HNO3:units = "molecules/cm3" ;
        HNO3:long_name = "HNO3 concentration" ;
    double HNO3AQ(Time, bottom_top, south_north, west_east) ;
        HNO3AQ:units = "molecules/cm3" ;
        HNO3AQ:long_name = "HNO3AQ concentration" ;
    double HNO4(Time, bottom_top, south_north, west_east) ;
        HNO4:units = "molecules/cm3" ;
        HNO4:long_name = "HNO4 concentration" ;
    double HO2(Time, bottom_top, south_north, west_east) ;
        HO2:units = "molecules/cm3" ;
        HO2:long_name = "HO2 concentration" ;
    double HOCL(Time, bottom_top, south_north, west_east) ;
        HOCL:units = "molecules/cm3" ;
        HOCL:long_name = "HOCL concentration" ;
    double HONO(Time, bottom_top, south_north, west_east) ;
        HONO:units = "molecules/cm3" ;
        HONO:long_name = "HONO concentration" ;
    double HUMULE(Time, bottom_top, south_north, west_east) ;
        HUMULE:units = "molecules/cm3" ;
        HUMULE:long_name = "HUMULE concentration" ;
    double IPRD(Time, bottom_top, south_north, west_east) ;
        IPRD:units = "molecules/cm3" ;
        IPRD:long_name = "IPRD concentration" ;
    double ISOPA1(Time, bottom_top, south_north, west_east) ;
        ISOPA1:units = "molecules/cm3" ;
        ISOPA1:long_name = "ISOPA1 concentration" ;
    double ISOPA1AQ(Time, bottom_top, south_north, west_east) ;
        ISOPA1AQ:units = "molecules/cm3" ;
        ISOPA1AQ:long_name = "ISOPA1AQ concentration" ;
    double LIMONE(Time, bottom_top, south_north, west_east) ;
        LIMONE:units = "molecules/cm3" ;
        LIMONE:long_name = "LIMONE concentration" ;
    double MACO3(Time, bottom_top, south_north, west_east) ;
        MACO3:units = "molecules/cm3" ;
        MACO3:long_name = "MACO3 concentration" ;
    double MACR(Time, bottom_top, south_north, west_east) ;
        MACR:units = "molecules/cm3" ;
```

```
                MACR:long_name = "MACR concentration" ;
        double MAPAN(Time, bottom_top, south_north, west_east) ;
                MAPAN:units = "molecules/cm3" ;
                MAPAN:long_name = "MAPAN concentration" ;
        double MECO3(Time, bottom_top, south_north, west_east) ;
                MECO3:units = "molecules/cm3" ;
                MECO3:long_name = "MECO3 concentration" ;
        double MEK(Time, bottom_top, south_north, west_east) ;
                MEK:units = "molecules/cm3" ;
                MEK:long_name = "MEK concentration" ;
        double MEO2(Time, bottom_top, south_north, west_east) ;
                MEO2:units = "molecules/cm3" ;
                MEO2:long_name = "MEO2 concentration" ;
        double MEOH(Time, bottom_top, south_north, west_east) ;
                MEOH:units = "molecules/cm3" ;
                MEOH:long_name = "MEOH concentration" ;
        double MGLY(Time, bottom_top, south_north, west_east) ;
                MGLY:units = "molecules/cm3" ;
                MGLY:long_name = "MGLY concentration" ;
        double MVK(Time, bottom_top, south_north, west_east) ;
                MVK:units = "molecules/cm3" ;
                MVK:long_name = "MVK concentration" ;
        double N2O5(Time, bottom_top, south_north, west_east) ;
                N2O5:units = "molecules/cm3" ;
                N2O5:long_name = "N2O5 concentration" ;
        double NAAQ(Time, bottom_top, south_north, west_east) ;
                NAAQ:units = "molecules/cm3" ;
                NAAQ:long_name = "NAAQ concentration" ;
        double NH3(Time, bottom_top, south_north, west_east) ;
                NH3:units = "molecules/cm3" ;
                NH3:long_name = "NH3 concentration" ;
        double NH3AQ(Time, bottom_top, south_north, west_east) ;
                NH3AQ:units = "molecules/cm3" ;
                NH3AQ:long_name = "NH3AQ concentration" ;
        double NO(Time, bottom_top, south_north, west_east) ;
                NO:units = "molecules/cm3" ;
                NO:long_name = "NO concentration" ;
        double NO2(Time, bottom_top, south_north, west_east) ;
                NO2:units = "molecules/cm3" ;
                NO2:long_name = "NO2 concentration" ;
        double NO3(Time, bottom_top, south_north, west_east) ;
                NO3:units = "molecules/cm3" ;
                NO3:long_name = "NO3 concentration" ;
        double NPHE(Time, bottom_top, south_north, west_east) ;
                NPHE:units = "molecules/cm3" ;
                NPHE:long_name = "NPHE concentration" ;
        double O3(Time, bottom_top, south_north, west_east) ;
                O3:units = "molecules/cm3" ;
                O3:long_name = "O3 concentration" ;
        double OCARAQ(Time, bottom_top, south_north, west_east) ;
                OCARAQ:units = "molecules/cm3" ;
                OCARAQ:long_name = "OCARAQ concentration" ;
        double OCIMEN(Time, bottom_top, south_north, west_east) ;
                OCIMEN:units = "molecules/cm3" ;
                OCIMEN:long_name = "OCIMEN concentration" ;
        double OH(Time, bottom_top, south_north, west_east) ;
                OH:units = "molecules/cm3" ;
                OH:long_name = "OH concentration" ;
        double OLE1(Time, bottom_top, south_north, west_east) ;
                OLE1:units = "molecules/cm3" ;
                OLE1:long_name = "OLE1 concentration" ;
        double OLE2(Time, bottom_top, south_north, west_east) ;
                OLE2:units = "molecules/cm3" ;
                OLE2:long_name = "OLE2 concentration" ;
        double PACD(Time, bottom_top, south_north, west_east) ;
                PACD:units = "molecules/cm3" ;
                PACD:long_name = "PACD concentration" ;
        double PAN(Time, bottom_top, south_north, west_east) ;
                PAN:units = "molecules/cm3" ;
                PAN:long_name = "PAN concentration" ;
        double PAN2(Time, bottom_top, south_north, west_east) ;
                PAN2:units = "molecules/cm3" ;
                PAN2:long_name = "PAN2 concentration" ;
        double PBZN(Time, bottom_top, south_north, west_east) ;
                PBZN:units = "molecules/cm3" ;
                PBZN:long_name = "PBZN concentration" ;
        double PPMAQ(Time, bottom_top, south_north, west_east) ;
                PPMAQ:units = "molecules/cm3" ;
                PPMAQ:long_name = "PPMAQ concentration" ;
        double PRD2(Time, bottom_top, south_north, west_east) ;
                PRD2:units = "molecules/cm3" ;
```

```
        PRD2:long_name = "PRD2 concentration" ;
    double R2O2(Time, bottom_top, south_north, west_east) ;
        R2O2:units = "molecules/cm3" ;
        R2O2:long_name = "R2O2 concentration" ;
    double RCHO(Time, bottom_top, south_north, west_east) ;
        RCHO:units = "molecules/cm3" ;
        RCHO:long_name = "RCHO concentration" ;
    double RCO3(Time, bottom_top, south_north, west_east) ;
        RCO3:units = "molecules/cm3" ;
        RCO3:long_name = "RCO3 concentration" ;
    double RNO3(Time, bottom_top, south_north, west_east) ;
        RNO3:units = "molecules/cm3" ;
        RNO3:long_name = "RNO3 concentration" ;
    double RO2CL(Time, bottom_top, south_north, west_east) ;
        RO2CL:units = "molecules/cm3" ;
        RO2CL:long_name = "RO2CL concentration" ;
    double RO2N(Time, bottom_top, south_north, west_east) ;
        RO2N:units = "molecules/cm3" ;
        RO2N:long_name = "RO2N concentration" ;
    double RO2R(Time, bottom_top, south_north, west_east) ;
        RO2R:units = "molecules/cm3" ;
        RO2R:long_name = "RO2R concentration" ;
    double SO2(Time, bottom_top, south_north, west_east) ;
        SO2:units = "molecules/cm3" ;
        SO2:long_name = "SO2 concentration" ;
    double TBUO(Time, bottom_top, south_north, west_east) ;
        TBUO:units = "molecules/cm3" ;
        TBUO:long_name = "TBUO concentration" ;
    double TERPEN(Time, bottom_top, south_north, west_east) ;
        TERPEN:units = "molecules/cm3" ;
        TERPEN:long_name = "TERPEN concentration" ;
    double TMB(Time, bottom_top, south_north, west_east) ;
        TMB:units = "molecules/cm3" ;
        TMB:long_name = "TMB concentration" ;
    double TOL(Time, bottom_top, south_north, west_east) ;
        TOL:units = "molecules/cm3" ;
        TOL:long_name = "TOL concentration" ;
    double XN(Time, bottom_top, south_north, west_east) ;
        XN:units = "molecules/cm3" ;
        XN:long_name = "XN concentration" ;
    double XOOH(Time, bottom_top, south_north, west_east) ;
        XOOH:units = "molecules/cm3" ;
        XOOH:long_name = "XOOH concentration" ;
    double phot_cln(Time, bottom_top, south_north, west_east) ;
        phot_cln:units = "molecules/cm3" ;
        phot_cln:long_name = "phot_cln concentration" ;
    double p01BCAR(Time, bottom_top, south_north, west_east) ;
        p01BCAR:units = "molecules/cm3" ;
        p01BCAR:long_name = "p01BCAR concentration" ;
    double p01DUST(Time, bottom_top, south_north, west_east) ;
        p01DUST:units = "molecules/cm3" ;
        p01DUST:long_name = "p01DUST concentration" ;
    double p01NA(Time, bottom_top, south_north, west_east) ;
        p01NA:units = "molecules/cm3" ;
        p01NA:long_name = "p01NA concentration" ;
    double p01OCAR(Time, bottom_top, south_north, west_east) ;
        p01OCAR:units = "molecules/cm3" ;
        p01OCAR:long_name = "p01OCAR concentration" ;
    double p01PPM(Time, bottom_top, south_north, west_east) ;
        p01PPM:units = "molecules/cm3" ;
        p01PPM:long_name = "p01PPM concentration" ;
    double p01AnA1D(Time, bottom_top, south_north, west_east) ;
        p01AnA1D:units = "molecules/cm3" ;
        p01AnA1D:long_name = "p01AnA1D concentration" ;
    double p01AnBmP(Time, bottom_top, south_north, west_east) ;
        p01AnBmP:units = "molecules/cm3" ;
        p01AnBmP:long_name = "p01AnBmP concentration" ;
    double p01BiA1D(Time, bottom_top, south_north, west_east) ;
        p01BiA1D:units = "molecules/cm3" ;
        p01BiA1D:long_name = "p01BiA1D concentration" ;
    double p01BiBmP(Time, bottom_top, south_north, west_east) ;
        p01BiBmP:units = "molecules/cm3" ;
        p01BiBmP:long_name = "p01BiBmP concentration" ;
    double p01ISOPA1(Time, bottom_top, south_north, west_east) ;
        p01ISOPA1:units = "molecules/cm3" ;
        p01ISOPA1:long_name = "p01ISOPA1 concentration" ;
    double p01H2SO4(Time, bottom_top, south_north, west_east) ;
        p01H2SO4:units = "molecules/cm3" ;
        p01H2SO4:long_name = "p01H2SO4 concentration" ;
    double p01HCL(Time, bottom_top, south_north, west_east) ;
        p01HCL:units = "molecules/cm3" ;
```

```
        p01HCL:long_name = "p01HCL concentration" ;
    double p01HNO3(Time, bottom_top, south_north, west_east) ;
        p01HNO3:units = "molecules/cm3" ;
        p01HNO3:long_name = "p01HNO3 concentration" ;
    double p01NH3(Time, bottom_top, south_north, west_east) ;
        p01NH3:units = "molecules/cm3" ;
        p01NH3:long_name = "p01NH3 concentration" ;
    double p01WATER(Time, bottom_top, south_north, west_east) ;
        p01WATER:units = "molecules/cm3" ;
        p01WATER:long_name = "p01WATER concentration" ;
    double p02BCAR(Time, bottom_top, south_north, west_east) ;
        p02BCAR:units = "molecules/cm3" ;
        p02BCAR:long_name = "p02BCAR concentration" ;
    double p02DUST(Time, bottom_top, south_north, west_east) ;
        p02DUST:units = "molecules/cm3" ;
        p02DUST:long_name = "p02DUST concentration" ;
    double p02NA(Time, bottom_top, south_north, west_east) ;
        p02NA:units = "molecules/cm3" ;
        p02NA:long_name = "p02NA concentration" ;
    double p02OCAR(Time, bottom_top, south_north, west_east) ;
        p02OCAR:units = "molecules/cm3" ;
        p02OCAR:long_name = "p02OCAR concentration" ;
    double p02PPM(Time, bottom_top, south_north, west_east) ;
        p02PPM:units = "molecules/cm3" ;
        p02PPM:long_name = "p02PPM concentration" ;
    double p02AnA1D(Time, bottom_top, south_north, west_east) ;
        p02AnA1D:units = "molecules/cm3" ;
        p02AnA1D:long_name = "p02AnA1D concentration" ;
    double p02AnBmP(Time, bottom_top, south_north, west_east) ;
        p02AnBmP:units = "molecules/cm3" ;
        p02AnBmP:long_name = "p02AnBmP concentration" ;
    double p02BiA1D(Time, bottom_top, south_north, west_east) ;
        p02BiA1D:units = "molecules/cm3" ;
        p02BiA1D:long_name = "p02BiA1D concentration" ;
    double p02BiBmP(Time, bottom_top, south_north, west_east) ;
        p02BiBmP:units = "molecules/cm3" ;
        p02BiBmP:long_name = "p02BiBmP concentration" ;
    double p02ISOPA1(Time, bottom_top, south_north, west_east) ;
        p02ISOPA1:units = "molecules/cm3" ;
        p02ISOPA1:long_name = "p02ISOPA1 concentration" ;
    double p02H2SO4(Time, bottom_top, south_north, west_east) ;
        p02H2SO4:units = "molecules/cm3" ;
        p02H2SO4:long_name = "p02H2SO4 concentration" ;
    double p02HCL(Time, bottom_top, south_north, west_east) ;
        p02HCL:units = "molecules/cm3" ;
        p02HCL:long_name = "p02HCL concentration" ;
    double p02HNO3(Time, bottom_top, south_north, west_east) ;
        p02HNO3:units = "molecules/cm3" ;
        p02HNO3:long_name = "p02HNO3 concentration" ;
    double p02NH3(Time, bottom_top, south_north, west_east) ;
        p02NH3:units = "molecules/cm3" ;
        p02NH3:long_name = "p02NH3 concentration" ;
    double p02WATER(Time, bottom_top, south_north, west_east) ;
        p02WATER:units = "molecules/cm3" ;
        p02WATER:long_name = "p02WATER concentration" ;
    double p03BCAR(Time, bottom_top, south_north, west_east) ;
        p03BCAR:units = "molecules/cm3" ;
        p03BCAR:long_name = "p03BCAR concentration" ;
    double p03DUST(Time, bottom_top, south_north, west_east) ;
        p03DUST:units = "molecules/cm3" ;
        p03DUST:long_name = "p03DUST concentration" ;
    double p03NA(Time, bottom_top, south_north, west_east) ;
        p03NA:units = "molecules/cm3" ;
        p03NA:long_name = "p03NA concentration" ;
    double p03OCAR(Time, bottom_top, south_north, west_east) ;
        p03OCAR:units = "molecules/cm3" ;
        p03OCAR:long_name = "p03OCAR concentration" ;
    double p03PPM(Time, bottom_top, south_north, west_east) ;
        p03PPM:units = "molecules/cm3" ;
        p03PPM:long_name = "p03PPM concentration" ;
    double p03AnA1D(Time, bottom_top, south_north, west_east) ;
        p03AnA1D:units = "molecules/cm3" ;
        p03AnA1D:long_name = "p03AnA1D concentration" ;
    double p03AnBmP(Time, bottom_top, south_north, west_east) ;
        p03AnBmP:units = "molecules/cm3" ;
        p03AnBmP:long_name = "p03AnBmP concentration" ;
    double p03BiA1D(Time, bottom_top, south_north, west_east) ;
        p03BiA1D:units = "molecules/cm3" ;
        p03BiA1D:long_name = "p03BiA1D concentration" ;
    double p03BiBmP(Time, bottom_top, south_north, west_east) ;
        p03BiBmP:units = "molecules/cm3" ;
```

```
        p03BiBmP:long_name = "p03BiBmP concentration" ;
    double p03ISOPA1(Time, bottom_top, south_north, west_east) ;
        p03ISOPA1:units = "molecules/cm3" ;
        p03ISOPA1:long_name = "p03ISOPA1 concentration" ;
    double p03H2SO4(Time, bottom_top, south_north, west_east) ;
        p03H2SO4:units = "molecules/cm3" ;
        p03H2SO4:long_name = "p03H2SO4 concentration" ;
    double p03HCL(Time, bottom_top, south_north, west_east) ;
        p03HCL:units = "molecules/cm3" ;
        p03HCL:long_name = "p03HCL concentration" ;
    double p03HNO3(Time, bottom_top, south_north, west_east) ;
        p03HNO3:units = "molecules/cm3" ;
        p03HNO3:long_name = "p03HNO3 concentration" ;
    double p03NH3(Time, bottom_top, south_north, west_east) ;
        p03NH3:units = "molecules/cm3" ;
        p03NH3:long_name = "p03NH3 concentration" ;
    double p03WATER(Time, bottom_top, south_north, west_east) ;
        p03WATER:units = "molecules/cm3" ;
        p03WATER:long_name = "p03WATER concentration" ;
    double p04BCAR(Time, bottom_top, south_north, west_east) ;
        p04BCAR:units = "molecules/cm3" ;
        p04BCAR:long_name = "p04BCAR concentration" ;
    double p04DUST(Time, bottom_top, south_north, west_east) ;
        p04DUST:units = "molecules/cm3" ;
        p04DUST:long_name = "p04DUST concentration" ;
    double p04NA(Time, bottom_top, south_north, west_east) ;
        p04NA:units = "molecules/cm3" ;
        p04NA:long_name = "p04NA concentration" ;
    double p04OCAR(Time, bottom_top, south_north, west_east) ;
        p04OCAR:units = "molecules/cm3" ;
        p04OCAR:long_name = "p04OCAR concentration" ;
    double p04PPM(Time, bottom_top, south_north, west_east) ;
        p04PPM:units = "molecules/cm3" ;
        p04PPM:long_name = "p04PPM concentration" ;
    double p04AnA1D(Time, bottom_top, south_north, west_east) ;
        p04AnA1D:units = "molecules/cm3" ;
        p04AnA1D:long_name = "p04AnA1D concentration" ;
    double p04AnBmP(Time, bottom_top, south_north, west_east) ;
        p04AnBmP:units = "molecules/cm3" ;
        p04AnBmP:long_name = "p04AnBmP concentration" ;
    double p04BiA1D(Time, bottom_top, south_north, west_east) ;
        p04BiA1D:units = "molecules/cm3" ;
        p04BiA1D:long_name = "p04BiA1D concentration" ;
    double p04BiBmP(Time, bottom_top, south_north, west_east) ;
        p04BiBmP:units = "molecules/cm3" ;
        p04BiBmP:long_name = "p04BiBmP concentration" ;
    double p04ISOPA1(Time, bottom_top, south_north, west_east) ;
        p04ISOPA1:units = "molecules/cm3" ;
        p04ISOPA1:long_name = "p04ISOPA1 concentration" ;
    double p04H2SO4(Time, bottom_top, south_north, west_east) ;
        p04H2SO4:units = "molecules/cm3" ;
        p04H2SO4:long_name = "p04H2SO4 concentration" ;
    double p04HCL(Time, bottom_top, south_north, west_east) ;
        p04HCL:units = "molecules/cm3" ;
        p04HCL:long_name = "p04HCL concentration" ;
    double p04HNO3(Time, bottom_top, south_north, west_east) ;
        p04HNO3:units = "molecules/cm3" ;
        p04HNO3:long_name = "p04HNO3 concentration" ;
    double p04NH3(Time, bottom_top, south_north, west_east) ;
        p04NH3:units = "molecules/cm3" ;
        p04NH3:long_name = "p04NH3 concentration" ;
    double p04WATER(Time, bottom_top, south_north, west_east) ;
        p04WATER:units = "molecules/cm3" ;
        p04WATER:long_name = "p04WATER concentration" ;
    double p05BCAR(Time, bottom_top, south_north, west_east) ;
        p05BCAR:units = "molecules/cm3" ;
        p05BCAR:long_name = "p05BCAR concentration" ;
    double p05DUST(Time, bottom_top, south_north, west_east) ;
        p05DUST:units = "molecules/cm3" ;
        p05DUST:long_name = "p05DUST concentration" ;
    double p05NA(Time, bottom_top, south_north, west_east) ;
        p05NA:units = "molecules/cm3" ;
        p05NA:long_name = "p05NA concentration" ;
    double p05OCAR(Time, bottom_top, south_north, west_east) ;
        p05OCAR:units = "molecules/cm3" ;
        p05OCAR:long_name = "p05OCAR concentration" ;
    double p05PPM(Time, bottom_top, south_north, west_east) ;
        p05PPM:units = "molecules/cm3" ;
        p05PPM:long_name = "p05PPM concentration" ;
    double p05AnA1D(Time, bottom_top, south_north, west_east) ;
        p05AnA1D:units = "molecules/cm3" ;
```

```
        p05AnA1D:long_name = "p05AnA1D concentration" ;
    double p05AnBmP(Time, bottom_top, south_north, west_east) ;
        p05AnBmP:units = "molecules/cm3" ;
        p05AnBmP:long_name = "p05AnBmP concentration" ;
    double p05BiA1D(Time, bottom_top, south_north, west_east) ;
        p05BiA1D:units = "molecules/cm3" ;
        p05BiA1D:long_name = "p05BiA1D concentration" ;
    double p05BiBmP(Time, bottom_top, south_north, west_east) ;
        p05BiBmP:units = "molecules/cm3" ;
        p05BiBmP:long_name = "p05BiBmP concentration" ;
    double p05ISOPA1(Time, bottom_top, south_north, west_east) ;
        p05ISOPA1:units = "molecules/cm3" ;
        p05ISOPA1:long_name = "p05ISOPA1 concentration" ;
    double p05H2SO4(Time, bottom_top, south_north, west_east) ;
        p05H2SO4:units = "molecules/cm3" ;
        p05H2SO4:long_name = "p05H2SO4 concentration" ;
    double p05HCL(Time, bottom_top, south_north, west_east) ;
        p05HCL:units = "molecules/cm3" ;
        p05HCL:long_name = "p05HCL concentration" ;
    double p05HNO3(Time, bottom_top, south_north, west_east) ;
        p05HNO3:units = "molecules/cm3" ;
        p05HNO3:long_name = "p05HNO3 concentration" ;
    double p05NH3(Time, bottom_top, south_north, west_east) ;
        p05NH3:units = "molecules/cm3" ;
        p05NH3:long_name = "p05NH3 concentration" ;
    double p05WATER(Time, bottom_top, south_north, west_east) ;
        p05WATER:units = "molecules/cm3" ;
        p05WATER:long_name = "p05WATER concentration" ;
    double p06BCAR(Time, bottom_top, south_north, west_east) ;
        p06BCAR:units = "molecules/cm3" ;
        p06BCAR:long_name = "p06BCAR concentration" ;
    double p06DUST(Time, bottom_top, south_north, west_east) ;
        p06DUST:units = "molecules/cm3" ;
        p06DUST:long_name = "p06DUST concentration" ;
    double p06NA(Time, bottom_top, south_north, west_east) ;
        p06NA:units = "molecules/cm3" ;
        p06NA:long_name = "p06NA concentration" ;
    double p06OCAR(Time, bottom_top, south_north, west_east) ;
        p06OCAR:units = "molecules/cm3" ;
        p06OCAR:long_name = "p06OCAR concentration" ;
    double p06PPM(Time, bottom_top, south_north, west_east) ;
        p06PPM:units = "molecules/cm3" ;
        p06PPM:long_name = "p06PPM concentration" ;
    double p06AnA1D(Time, bottom_top, south_north, west_east) ;
        p06AnA1D:units = "molecules/cm3" ;
        p06AnA1D:long_name = "p06AnA1D concentration" ;
    double p06AnBmP(Time, bottom_top, south_north, west_east) ;
        p06AnBmP:units = "molecules/cm3" ;
        p06AnBmP:long_name = "p06AnBmP concentration" ;
    double p06BiA1D(Time, bottom_top, south_north, west_east) ;
        p06BiA1D:units = "molecules/cm3" ;
        p06BiA1D:long_name = "p06BiA1D concentration" ;
    double p06BiBmP(Time, bottom_top, south_north, west_east) ;
        p06BiBmP:units = "molecules/cm3" ;
        p06BiBmP:long_name = "p06BiBmP concentration" ;
    double p06ISOPA1(Time, bottom_top, south_north, west_east) ;
        p06ISOPA1:units = "molecules/cm3" ;
        p06ISOPA1:long_name = "p06ISOPA1 concentration" ;
    double p06H2SO4(Time, bottom_top, south_north, west_east) ;
        p06H2SO4:units = "molecules/cm3" ;
        p06H2SO4:long_name = "p06H2SO4 concentration" ;
    double p06HCL(Time, bottom_top, south_north, west_east) ;
        p06HCL:units = "molecules/cm3" ;
        p06HCL:long_name = "p06HCL concentration" ;
    double p06HNO3(Time, bottom_top, south_north, west_east) ;
        p06HNO3:units = "molecules/cm3" ;
        p06HNO3:long_name = "p06HNO3 concentration" ;
    double p06NH3(Time, bottom_top, south_north, west_east) ;
        p06NH3:units = "molecules/cm3" ;
        p06NH3:long_name = "p06NH3 concentration" ;
    double p06WATER(Time, bottom_top, south_north, west_east) ;
        p06WATER:units = "molecules/cm3" ;
        p06WATER:long_name = "p06WATER concentration" ;
    double p07BCAR(Time, bottom_top, south_north, west_east) ;
        p07BCAR:units = "molecules/cm3" ;
        p07BCAR:long_name = "p07BCAR concentration" ;
    double p07DUST(Time, bottom_top, south_north, west_east) ;
        p07DUST:units = "molecules/cm3" ;
        p07DUST:long_name = "p07DUST concentration" ;
    double p07NA(Time, bottom_top, south_north, west_east) ;
        p07NA:units = "molecules/cm3" ;
```

```
        p07NA:long_name = "p07NA concentration" ;
    double p07OCAR(Time, bottom_top, south_north, west_east) ;
        p07OCAR:units = "molecules/cm3" ;
        p07OCAR:long_name = "p07OCAR concentration" ;
    double p07PPM(Time, bottom_top, south_north, west_east) ;
        p07PPM:units = "molecules/cm3" ;
        p07PPM:long_name = "p07PPM concentration" ;
    double p07AnA1D(Time, bottom_top, south_north, west_east) ;
        p07AnA1D:units = "molecules/cm3" ;
        p07AnA1D:long_name = "p07AnA1D concentration" ;
    double p07AnBmP(Time, bottom_top, south_north, west_east) ;
        p07AnBmP:units = "molecules/cm3" ;
        p07AnBmP:long_name = "p07AnBmP concentration" ;
    double p07BiA1D(Time, bottom_top, south_north, west_east) ;
        p07BiA1D:units = "molecules/cm3" ;
        p07BiA1D:long_name = "p07BiA1D concentration" ;
    double p07BiBmP(Time, bottom_top, south_north, west_east) ;
        p07BiBmP:units = "molecules/cm3" ;
        p07BiBmP:long_name = "p07BiBmP concentration" ;
    double p07ISOPA1(Time, bottom_top, south_north, west_east) ;
        p07ISOPA1:units = "molecules/cm3" ;
        p07ISOPA1:long_name = "p07ISOPA1 concentration" ;
    double p07H2SO4(Time, bottom_top, south_north, west_east) ;
        p07H2SO4:units = "molecules/cm3" ;
        p07H2SO4:long_name = "p07H2SO4 concentration" ;
    double p07HCL(Time, bottom_top, south_north, west_east) ;
        p07HCL:units = "molecules/cm3" ;
        p07HCL:long_name = "p07HCL concentration" ;
    double p07HNO3(Time, bottom_top, south_north, west_east) ;
        p07HNO3:units = "molecules/cm3" ;
        p07HNO3:long_name = "p07HNO3 concentration" ;
    double p07NH3(Time, bottom_top, south_north, west_east) ;
        p07NH3:units = "molecules/cm3" ;
        p07NH3:long_name = "p07NH3 concentration" ;
    double p07WATER(Time, bottom_top, south_north, west_east) ;
        p07WATER:units = "molecules/cm3" ;
        p07WATER:long_name = "p07WATER concentration" ;
    double p08BCAR(Time, bottom_top, south_north, west_east) ;
        p08BCAR:units = "molecules/cm3" ;
        p08BCAR:long_name = "p08BCAR concentration" ;
    double p08DUST(Time, bottom_top, south_north, west_east) ;
        p08DUST:units = "molecules/cm3" ;
        p08DUST:long_name = "p08DUST concentration" ;
    double p08NA(Time, bottom_top, south_north, west_east) ;
        p08NA:units = "molecules/cm3" ;
        p08NA:long_name = "p08NA concentration" ;
    double p08OCAR(Time, bottom_top, south_north, west_east) ;
        p08OCAR:units = "molecules/cm3" ;
        p08OCAR:long_name = "p08OCAR concentration" ;
    double p08PPM(Time, bottom_top, south_north, west_east) ;
        p08PPM:units = "molecules/cm3" ;
        p08PPM:long_name = "p08PPM concentration" ;
    double p08AnA1D(Time, bottom_top, south_north, west_east) ;
        p08AnA1D:units = "molecules/cm3" ;
        p08AnA1D:long_name = "p08AnA1D concentration" ;
    double p08AnBmP(Time, bottom_top, south_north, west_east) ;
        p08AnBmP:units = "molecules/cm3" ;
        p08AnBmP:long_name = "p08AnBmP concentration" ;
    double p08BiA1D(Time, bottom_top, south_north, west_east) ;
        p08BiA1D:units = "molecules/cm3" ;
        p08BiA1D:long_name = "p08BiA1D concentration" ;
    double p08BiBmP(Time, bottom_top, south_north, west_east) ;
        p08BiBmP:units = "molecules/cm3" ;
        p08BiBmP:long_name = "p08BiBmP concentration" ;
    double p08ISOPA1(Time, bottom_top, south_north, west_east) ;
        p08ISOPA1:units = "molecules/cm3" ;
        p08ISOPA1:long_name = "p08ISOPA1 concentration" ;
    double p08H2SO4(Time, bottom_top, south_north, west_east) ;
        p08H2SO4:units = "molecules/cm3" ;
        p08H2SO4:long_name = "p08H2SO4 concentration" ;
    double p08HCL(Time, bottom_top, south_north, west_east) ;
        p08HCL:units = "molecules/cm3" ;
        p08HCL:long_name = "p08HCL concentration" ;
    double p08HNO3(Time, bottom_top, south_north, west_east) ;
        p08HNO3:units = "molecules/cm3" ;
        p08HNO3:long_name = "p08HNO3 concentration" ;
    double p08NH3(Time, bottom_top, south_north, west_east) ;
        p08NH3:units = "molecules/cm3" ;
        p08NH3:long_name = "p08NH3 concentration" ;
    double p08WATER(Time, bottom_top, south_north, west_east) ;
        p08WATER:units = "molecules/cm3" ;
```

```
        p08WATER:long_name = "p08WATER concentration" ;
    double p09BCAR(Time, bottom_top, south_north, west_east) ;
        p09BCAR:units = "molecules/cm3" ;
        p09BCAR:long_name = "p09BCAR concentration" ;
    double p09DUST(Time, bottom_top, south_north, west_east) ;
        p09DUST:units = "molecules/cm3" ;
        p09DUST:long_name = "p09DUST concentration" ;
    double p09NA(Time, bottom_top, south_north, west_east) ;
        p09NA:units = "molecules/cm3" ;
        p09NA:long_name = "p09NA concentration" ;
    double p09OCAR(Time, bottom_top, south_north, west_east) ;
        p09OCAR:units = "molecules/cm3" ;
        p09OCAR:long_name = "p09OCAR concentration" ;
    double p09PPM(Time, bottom_top, south_north, west_east) ;
        p09PPM:units = "molecules/cm3" ;
        p09PPM:long_name = "p09PPM concentration" ;
    double p09AnA1D(Time, bottom_top, south_north, west_east) ;
        p09AnA1D:units = "molecules/cm3" ;
        p09AnA1D:long_name = "p09AnA1D concentration" ;
    double p09AnBmP(Time, bottom_top, south_north, west_east) ;
        p09AnBmP:units = "molecules/cm3" ;
        p09AnBmP:long_name = "p09AnBmP concentration" ;
    double p09BiA1D(Time, bottom_top, south_north, west_east) ;
        p09BiA1D:units = "molecules/cm3" ;
        p09BiA1D:long_name = "p09BiA1D concentration" ;
    double p09BiBmP(Time, bottom_top, south_north, west_east) ;
        p09BiBmP:units = "molecules/cm3" ;
        p09BiBmP:long_name = "p09BiBmP concentration" ;
    double p09ISOPA1(Time, bottom_top, south_north, west_east) ;
        p09ISOPA1:units = "molecules/cm3" ;
        p09ISOPA1:long_name = "p09ISOPA1 concentration" ;
    double p09H2SO4(Time, bottom_top, south_north, west_east) ;
        p09H2SO4:units = "molecules/cm3" ;
        p09H2SO4:long_name = "p09H2SO4 concentration" ;
    double p09HCL(Time, bottom_top, south_north, west_east) ;
        p09HCL:units = "molecules/cm3" ;
        p09HCL:long_name = "p09HCL concentration" ;
    double p09HNO3(Time, bottom_top, south_north, west_east) ;
        p09HNO3:units = "molecules/cm3" ;
        p09HNO3:long_name = "p09HNO3 concentration" ;
    double p09NH3(Time, bottom_top, south_north, west_east) ;
        p09NH3:units = "molecules/cm3" ;
        p09NH3:long_name = "p09NH3 concentration" ;
    double p09WATER(Time, bottom_top, south_north, west_east) ;
        p09WATER:units = "molecules/cm3" ;
        p09WATER:long_name = "p09WATER concentration" ;
    double p10BCAR(Time, bottom_top, south_north, west_east) ;
        p10BCAR:units = "molecules/cm3" ;
        p10BCAR:long_name = "p10BCAR concentration" ;
    double p10DUST(Time, bottom_top, south_north, west_east) ;
        p10DUST:units = "molecules/cm3" ;
        p10DUST:long_name = "p10DUST concentration" ;
    double p10NA(Time, bottom_top, south_north, west_east) ;
        p10NA:units = "molecules/cm3" ;
        p10NA:long_name = "p10NA concentration" ;
    double p10OCAR(Time, bottom_top, south_north, west_east) ;
        p10OCAR:units = "molecules/cm3" ;
        p10OCAR:long_name = "p10OCAR concentration" ;
    double p10PPM(Time, bottom_top, south_north, west_east) ;
        p10PPM:units = "molecules/cm3" ;
        p10PPM:long_name = "p10PPM concentration" ;
    double p10AnA1D(Time, bottom_top, south_north, west_east) ;
        p10AnA1D:units = "molecules/cm3" ;
        p10AnA1D:long_name = "p10AnA1D concentration" ;
    double p10AnBmP(Time, bottom_top, south_north, west_east) ;
        p10AnBmP:units = "molecules/cm3" ;
        p10AnBmP:long_name = "p10AnBmP concentration" ;
    double p10BiA1D(Time, bottom_top, south_north, west_east) ;
        p10BiA1D:units = "molecules/cm3" ;
        p10BiA1D:long_name = "p10BiA1D concentration" ;
    double p10BiBmP(Time, bottom_top, south_north, west_east) ;
        p10BiBmP:units = "molecules/cm3" ;
        p10BiBmP:long_name = "p10BiBmP concentration" ;
    double p10ISOPA1(Time, bottom_top, south_north, west_east) ;
        p10ISOPA1:units = "molecules/cm3" ;
        p10ISOPA1:long_name = "p10ISOPA1 concentration" ;
    double p10H2SO4(Time, bottom_top, south_north, west_east) ;
        p10H2SO4:units = "molecules/cm3" ;
        p10H2SO4:long_name = "p10H2SO4 concentration" ;
    double p10HCL(Time, bottom_top, south_north, west_east) ;
        p10HCL:units = "molecules/cm3" ;
```

```
        p10HCL:long_name = "p10HCL concentration" ;
    double p10HNO3(Time, bottom_top, south_north, west_east) ;
        p10HNO3:units = "molecules/cm3" ;
        p10HNO3:long_name = "p10HNO3 concentration" ;
    double p10NH3(Time, bottom_top, south_north, west_east) ;
        p10NH3:units = "molecules/cm3" ;
        p10NH3:long_name = "p10NH3 concentration" ;
    double p10WATER(Time, bottom_top, south_north, west_east) ;
        p10WATER:units = "molecules/cm3" ;
        p10WATER:long_name = "p10WATER concentration" ;
    double M(Time, bottom_top, south_north, west_east) ;
        M:units = "molecules/cm3" ;
        M:long_name = "M concentration" ;
    double O2(Time, bottom_top, south_north, west_east) ;
        O2:units = "molecules/cm3" ;
        O2:long_name = "O2 concentration" ;
    double N2(Time, bottom_top, south_north, west_east) ;
        N2:units = "molecules/cm3" ;
        N2:long_name = "N2 concentration" ;
    double H2O(Time, bottom_top, south_north, west_east) ;
        H2O:units = "molecules/cm3" ;
        H2O:long_name = "H2O concentration" ;
    double NOX(Time, bottom_top, south_north, west_east) ;
        NOX:units = "molecules/cm3" ;
        NOX:long_name = "NOX concentration" ;
    double TOTPAN(Time, bottom_top, south_north, west_east) ;
        TOTPAN:units = "molecules/cm3" ;
        TOTPAN:long_name = "TOTPAN concentration" ;
    double NOY(Time, bottom_top, south_north, west_east) ;
        NOY:units = "molecules/cm3" ;
        NOY:long_name = "NOY concentration" ;
    double HOX(Time, bottom_top, south_north, west_east) ;
        HOX:units = "molecules/cm3" ;
        HOX:long_name = "HOX concentration" ;
    double ROX(Time, bottom_top, south_north, west_east) ;
        ROX:units = "molecules/cm3" ;
        ROX:long_name = "ROX concentration" ;
    double ANT(Time, bottom_top, south_north, west_east) ;
        ANT:units = "molecules/cm3" ;
        ANT:long_name = "ANT concentration" ;
    double BIO(Time, bottom_top, south_north, west_east) ;
        BIO:units = "molecules/cm3" ;
        BIO:long_name = "BIO concentration" ;
    double COVS(Time, bottom_top, south_north, west_east) ;
        COVS:units = "molecules/cm3" ;
        COVS:long_name = "COVS concentration" ;
    double ANTS(Time, bottom_top, south_north, west_east) ;
        ANTS:units = "molecules/cm3" ;
        ANTS:long_name = "ANTS concentration" ;
    double OX(Time, bottom_top, south_north, west_east) ;
        OX:units = "molecules/cm3" ;
        OX:long_name = "OX concentration" ;
    double pBCAR(Time, bottom_top, south_north, west_east) ;
        pBCAR:units = "molecules/cm3" ;
        pBCAR:long_name = "pBCAR concentration" ;
    double pDUST(Time, bottom_top, south_north, west_east) ;
        pDUST:units = "molecules/cm3" ;
        pDUST:long_name = "pDUST concentration" ;
    double pNA(Time, bottom_top, south_north, west_east) ;
        pNA:units = "molecules/cm3" ;
        pNA:long_name = "pNA concentration" ;
    double pOCAR(Time, bottom_top, south_north, west_east) ;
        pOCAR:units = "molecules/cm3" ;
        pOCAR:long_name = "pOCAR concentration" ;
    double pPPM(Time, bottom_top, south_north, west_east) ;
        pPPM:units = "molecules/cm3" ;
        pPPM:long_name = "pPPM concentration" ;
    double pAnA1D(Time, bottom_top, south_north, west_east) ;
        pAnA1D:units = "molecules/cm3" ;
        pAnA1D:long_name = "pAnA1D concentration" ;
    double pAnBmP(Time, bottom_top, south_north, west_east) ;
        pAnBmP:units = "molecules/cm3" ;
        pAnBmP:long_name = "pAnBmP concentration" ;
    double pBiA1D(Time, bottom_top, south_north, west_east) ;
        pBiA1D:units = "molecules/cm3" ;
        pBiA1D:long_name = "pBiA1D concentration" ;
    double pBiBmP(Time, bottom_top, south_north, west_east) ;
        pBiBmP:units = "molecules/cm3" ;
        pBiBmP:long_name = "pBiBmP concentration" ;
    double pISOPA1(Time, bottom_top, south_north, west_east) ;
        pISOPA1:units = "molecules/cm3" ;
```

```
                pISOPA1:long_name = "pISOPA1 concentration" ;
        double pH2SO4(Time, bottom_top, south_north, west_east) ;
                pH2SO4:units = "molecules/cm3" ;
                pH2SO4:long_name = "pH2SO4 concentration" ;
        double pHCL(Time, bottom_top, south_north, west_east) ;
                pHCL:units = "molecules/cm3" ;
                pHCL:long_name = "pHCL concentration" ;
        double pHNO3(Time, bottom_top, south_north, west_east) ;
                pHNO3:units = "molecules/cm3" ;
                pHNO3:long_name = "pHNO3 concentration" ;
        double pNH3(Time, bottom_top, south_north, west_east) ;
                pNH3:units = "molecules/cm3" ;
                pNH3:long_name = "pNH3 concentration" ;
        double pWATER(Time, bottom_top, south_north, west_east) ;
                pWATER:units = "molecules/cm3" ;
                pWATER:long_name = "pWATER concentration" ;
        double PM25(Time, bottom_top, south_north, west_east) ;
                PM25:units = "molecules/cm3" ;
                PM25:long_name = "PM25 concentration" ;
        double PM25bio(Time, bottom_top, south_north, west_east) ;
                PM25bio:units = "molecules/cm3" ;
                PM25bio:long_name = "PM25bio concentration" ;
        double PM25ant(Time, bottom_top, south_north, west_east) ;
                PM25ant:units = "molecules/cm3" ;
                PM25ant:long_name = "PM25ant concentration" ;
        double PM10(Time, bottom_top, south_north, west_east) ;
                PM10:units = "molecules/cm3" ;
                PM10:long_name = "PM10 concentration" ;
        double PM10bio(Time, bottom_top, south_north, west_east) ;
                PM10bio:units = "molecules/cm3" ;
                PM10bio:long_name = "PM10bio concentration" ;
        double PM10ant(Time, bottom_top, south_north, west_east) ;
                PM10ant:units = "molecules/cm3" ;
                PM10ant:long_name = "PM10ant concentration" ;
        double airm(Time, bottom_top, south_north, west_east) ;
                airm:units = "molecule/m**3" ;
                airm:long_name = "Air density" ;
        double hlay(Time, bottom_top, south_north, west_east) ;
                hlay:units = "m" ;
                hlay:long_name = "Layer top altitude" ;
        double lastrain(Time, south_north, west_east) ;
                lastrain:units = "h" ;
                lastrain:long_name = "Hours since last rain" ;

// global attributes:
                :Title = "CHIMERE SUITE" ;
                :Sub-title = "Final concentrations file" ;
                :Generating_process = "Generated by chimere" ;
                :Conventions = "None" ;
                :Domain = "GEMSHCL" ;
                :history = "File /DATA/MODEL/SIOUR/CLNO2/OUTPUTS/CLNO2/end.2012010100_2012010200_GEMSHCL.nc was generated on Tue
}
```

E.6 The results file "exdomout.[label].nc"

```
netcdf exdomout.2012010100_2012010200_GEMSHCL {
dimensions:
        Time = UNLIMITED ; // (25 currently)
        DateStrLen = 19 ;
        west_east = 202 ;
        south_north = 182 ;
        bottom_top = 51 ;
variables:
        char Times(Time, DateStrLen) ;
        double lon(south_north, west_east) ;
                lon:units = "degrees_east" ;
                lon:long_name = "Longitude" ;
        double lat(south_north, west_east) ;
                lat:units = "degrees_north" ;
                lat:long_name = "Latitude" ;
        double temp(Time, bottom_top, south_north, west_east) ;
                temp:units = "K" ;
                temp:long_name = "Temperature" ;
        double sphu(Time, bottom_top, south_north, west_east) ;
                sphu:units = "kg/kg" ;
                sphu:long_name = "Specific humidity" ;
        double cliq(Time, bottom_top, south_north, west_east) ;
```

```
        cliq:units = "kg/kg" ;
        cliq:long_name = "Cloud liquid water content" ;
    double cice(Time, bottom_top, south_north, west_east) ;
        cice:units = "kg/kg" ;
        cice:long_name = "Cloud ice water content" ;
    double winz(Time, bottom_top, south_north, west_east) ;
        winz:units = "m/s" ;
        winz:long_name = "Zonal wind" ;
    double winm(Time, bottom_top, south_north, west_east) ;
        winm:units = "m/s" ;
        winm:long_name = "Meridian wind" ;
    double pres(Time, bottom_top, south_north, west_east) ;
        pres:units = "Pa" ;
        pres:long_name = "Pressure" ;
    double alti(Time, bottom_top, south_north, west_east) ;
        alti:units = "m" ;
        alti:long_name = "Altitude of the middle of layer" ;
    double tem2(Time, south_north, west_east) ;
        tem2:units = "K" ;
        tem2:long_name = "2 metre temperature" ;
    double psfc(Time, south_north, west_east) ;
        psfc:units = "Pa" ;
        psfc:long_name = "Surf ace pressure" ;
    double clol(Time, south_north, west_east) ;
        clol:units = "Fraction" ;
        clol:long_name = "Low cloud cover" ;
    double clom(Time, south_north, west_east) ;
        clom:units = "Fraction" ;
        clom:long_name = "Medium cloud cover" ;
    double cloh(Time, south_north, west_east) ;
        cloh:units = "Fraction" ;
        cloh:long_name = "High cloud cover" ;
    double lspc(Time, south_north, west_east) ;
        lspc:units = "mm/hr" ;
        lspc:long_name = "Large scale precipitation" ;
    double copc(Time, south_north, west_east) ;
        copc:units = "mm/hr" ;
        copc:long_name = "Convective precipitation" ;
    double hght(Time, south_north, west_east) ;
        hght:units = "m" ;
        hght:long_name = "Boundary layer height" ;
    double sshf(Time, south_north, west_east) ;
        sshf:units = "W/m^2" ;
        sshf:long_name = "Surf. sens. heat flux" ;
    double slhf(Time, south_north, west_east) ;
        slhf:units = "W/m^2" ;
        slhf:long_name = "Surf. lat. heat flux" ;
    double lsm(Time, south_north, west_east) ;
        lsm:units = "Fraction" ;
        lsm:long_name = "Land-Sea Mask" ;
    double oro(Time, south_north, west_east) ;
        oro:units = "m" ;
        oro:long_name = "Orography" ;
    double stl1(Time, south_north, west_east) ;
        stl1:units = "K" ;
        stl1:long_name = "Soil temperature 0-7 cm" ;
    double stl2(Time, south_north, west_east) ;
        stl2:units = "K" ;
        stl2:long_name = "Soil temperature 7-28 cm" ;
    double stl3(Time, south_north, west_east) ;
        stl3:units = "K" ;
        stl3:long_name = "Soil temperature 28-100 cm" ;
    double stl4(Time, south_north, west_east) ;
        stl4:units = "K" ;
        stl4:long_name = "Soil temperature 100-255 cm" ;
    double snowh(Time, south_north, west_east) ;
        snowh:units = "m" ;
        snowh:long_name = "Snow depth m (of water equivalent)" ;
    double seaice(Time, south_north, west_east) ;
        seaice:units = "Fraction" ;
        seaice:long_name = "Sea ice cover (0-1)" ;
    double soim(Time, south_north, west_east) ;
        soim:units = "m^3/m^3" ;
        soim:long_name = "Volumetric soil water 0-7 cm" ;
    double swvl2(Time, south_north, west_east) ;
        swvl2:units = "m^3/m^3" ;
        swvl2:long_name = "Volumetric soil water 7-28 cm" ;
    double swvl3(Time, south_north, west_east) ;
        swvl3:units = "m^3/m^3" ;
        swvl3:long_name = "Volumetric soil water 28-100 cm" ;
    double swvl4(Time, south_north, west_east) ;
```

```
                swvl4:units = "m^3/m^3" ;
                swvl4:long_name = "Volumetric soil water 100-255 cm" ;
        double rh2m(Time, south_north, west_east) ;
                rh2m:units = "fraction" ;
                rh2m:long_name = "Relative Humidity at 2m" ;
        double u10m(Time, south_north, west_east) ;
                u10m:units = "m/s" ;
                u10m:long_name = "Zonal wind at 10 m" ;
        double v10m(Time, south_north, west_east) ;
                v10m:units = "m/s" ;
                v10m:long_name = "Zonal wind at 10 m" ;
        double swrd(Time, south_north, west_east) ;
                swrd:units = "W/m^2" ;
                swrd:long_name = "Surf.solar rad. downwards" ;

// global attributes:
                :Title = "CHIMERE SUITE" ;
                :Sub-title = "Interpolated Meteo file" ;
                :Generating_process = "Generated by meteo-prepecm.F90" ;
                :Conventions =  ;
                :Domain = "GEMSHCL" ;
                :history = "File /DATA/MODEL/SIOUR/CLNO2/OUTPUTS/CLNO2/data_GEMSHCL_GEMSHCL/exdomout.2012010100_2012010200_GEMSHC
                        "" ;
                :ecm_nzonal = 241 ;
                :ecm_nmerid = 221 ;
                :ecm_nlevs = 91 ;
}
```

F VOC speciation

SNAP sector (%)										Real compound	Name
S1	S2	S3	S4	S5	S6	S7	S8	S9	S10		
0.0	0.0	0.0	0.0	0.0	0.2	0.0	0.0	0.0	0.0	1MECHX	(1-methylethyl)cyclohexane
0.0	0.0	0.0	0.0	0.0	0.3	0.0	0.0	0.0	0.0	1MPCHX	(1-methylpropyl)cyclohexane
0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.0	0.0	0.0	2M1PAC	(2-methyl-1-propyl)acetate
0.0	0.0	0.0	0.0	0.0	0.2	0.0	0.0	0.0	0.0	2MPCHX	(2-methylpropyl)cyclohexane
0.0	0.0	0.0	0.0	0.0	2.2	0.0	0.0	0.4	0.0	111TCE	1,1,1-trichloroethane
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.3	0.0	11DCEA	1,1-dichloroethane
0.0	0.0	0.0	0.5	0.0	0.0	0.0	0.0	0.0	0.0	11DCEE	1,1-dichloroethene
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.3	0.0	11DMCP	1,1-dimethylcyclopentane
0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.0	0.0	0.0	1234TB	1,2,3,4-tetramethylbenzene
0.0	0.0	0.0	0.0	0.0	0.2	0.0	0.0	0.0	0.0	1235TB	1,2,3,5-tetramethylbenzene
0.0	0.0	0.0	0.0	0.0	0.5	0.5	0.3	0.0	0.0	123TMB	1,2,3-trimethylbenzene
0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.0	0.0	0.0	123TCH	1,2,3-trimethylcyclohexane
0.0	0.0	0.0	0.0	0.0	0.2	0.0	0.0	0.0	0.0	1245TB	1,2,4,5-tetramethylbenzene
0.0	0.0	0.0	0.2	0.0	1.7	2.1	0.7	0.0	0.0	124TMB	1,2,4-trimethylbenzene
0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	12DCEA	1,2-dichloroethane
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.7	0.0	12DCEE	1,2-dichloroethene
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.3	0.0	12DMCP	1,2-dimethylcyclopentane
0.0	0.0	0.0	0.0	0.0	0.2	0.0	0.0	0.0	0.0	12PDIO	1,2-propanediol
0.0	0.0	0.0	0.0	0.0	0.5	0.8	0.3	0.0	0.0	135TMB	1,3,5-trimethylbenzene
0.0	0.0	0.0	0.2	0.0	0.0	1.1	0.8	0.0	0.0	13BDIE	1,3-butadiene
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.3	0.0	13DMCP	1,3-dimethylcyclopentane
0.0	0.0	0.0	0.0	0.0	0.0	0.4	0.0	0.0	0.0	13HDIE	1,3-hexadiene
0.0	0.0	0.0	0.0	0.0	0.3	0.0	0.0	0.0	0.0	14DCBZ	1,4-dichlorobenzene
0.0	0.0	0.0	0.0	0.0	0.2	0.0	0.0	0.0	0.0	1BME2P	1-(2-butoxy-1-methyl-ethoxy)-2-propanol
0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.0	0.0	0.0	1EME2P	1-(2-ethoxy-1-methyl-ethoxy)-2-propanol
0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.0	0.0	0.0	1MME2P	1-(2-methoxy-1-methyl-ethoxy)-2-propanol
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.3	0.0	0.0	1BUTAL	1-butanal
0.0	0.0	0.0	0.0	0.0	1.2	0.0	0.0	0.0	0.0	1BUTOL	1-butanol
0.0	0.0	0.6	0.2	0.2	0.0	0.6	0.4	0.0	0.0	1BUTEN	1-butene
				...						...	...

SNAP sector (%)										Real compound	Name
S1	S2	S3	S4	S5	S6	S7	S8	S9	S10		
				...						...	...
0.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1DECEN	1-decene
0.0	0.0	0.0	0.0	0.0	0.3	0.0	0.0	0.0	0.0	1E2POL	1-ethoxy-2-propanol
0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.0	0.0	0.0	1E2PAC	1-ethoxy-2-propyl acetate
0.0	0.0	0.0	0.0	0.0	0.3	0.0	0.0	0.0	0.0	1E3MCH	1-ethyl-3-methylcyclohexane
0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.0	0.0	0.0	1E4MCH	1-ethyl-4-methylcyclohexane
0.4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1HPTEN	1-heptene
0.4	1.5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1HXENE	1-hexene
0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.0	0.0	0.0	1M2EOL	1-methoxy-2-ethanol
0.0	0.0	0.0	0.0	0.0	0.4	0.0	0.0	0.0	0.0	1M2POL	1-methoxy-2-propanol
0.0	0.0	0.0	0.0	0.0	0.2	0.0	0.0	0.0	0.0	1M2PAC	1-methoxy-2-propyl acetate
0.0	0.0	0.0	0.0	0.0	0.2	0.0	0.0	0.0	0.0	1M3PBZ	1-methyl-3-propylbenzene
0.0	0.0	0.0	0.0	0.0	0.3	0.0	0.0	0.0	0.0	1M4PBZ	1-methyl-4-isopropylbenzene
0.0	0.0	0.0	0.0	0.0	0.3	0.0	0.0	0.0	0.0	1M4IPC	1-methyl-4-isopropylcyclohexane
0.0	0.5	0.0	0.0	0.2	0.0	0.6	0.4	0.0	0.0	1PTENE	1-pentene
0.0	0.0	0.0	0.0	0.0	0.0	0.2	0.3	0.0	0.0	1PRPAL	1-propanal
0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.3	0.0	1PRPOL	1-propanol
0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.0	0.0	0.0	2233TH	2,2,3,3-tetramethylhexane
0.0	0.0	0.0	0.0	0.0	0.2	0.0	0.0	0.0	0.0	TMPMIB	2,2,4-trimethyl-1,3-pentanediol monoisobutyrate
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.2	0.0	0.0	22DMBA	2,2-dimethylbutane
0.0	0.0	0.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	22DMPA	2,2-dimethylpropane
0.0	0.0	0.0	0.2	0.0	0.0	0.0	0.2	0.0	0.0	23DMBA	2,3-dimethylbutane
0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.0	0.0	0.0	23DMHP	2,3-dimethylheptane
0.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	24DMPT	2,4-dimethylpentane
0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.0	0.0	0.0	25DMOC	2,5-dimethyloctane
0.0	0.0	0.0	0.0	0.0	0.4	0.0	0.0	0.0	0.0	26DMOC	2,6-dimethyloctane
0.0	0.0	0.0	0.0	0.0	0.4	0.0	0.0	0.0	0.0	2BEEOL	2-(2-butoxyethoxy)ethanol
0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.0	0.0	0.0	2BEEAC	2-(2-butoxyethoxy)ethyl acetate
0.0	0.0	0.0	0.0	0.0	0.4	0.0	0.0	0.0	0.0	2EEEOL	2-(2-ethoxyethoxy)ethanol
0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.0	0.0	0.0	2EEEAC	2-(2-ethoxyethoxy)ethyl acetate
0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.0	0.0	0.0	2MEEOL	2-(methoxyethoxy)ethanol
0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.0	0.0	0.0	2EEEEO	2-[2-(2-ethoxy-ethoxy)-ethoxy]ethanol
0.0	0.0	0.0	0.0	0.0	0.8	0.0	0.0	0.5	0.0	2BUTOL	2-butanol
0.0	0.0	0.0	0.0	0.0	3.0	0.0	0.0	0.0	0.0	2BUTON	2-butanone
0.0	0.0	0.0	0.0	0.0	0.0	1.4	0.4	0.0	0.0	2BUTEN	2-butene
0.0	0.0	0.0	0.0	0.0	0.4	0.0	0.0	0.0	0.0	2BOEOL	2-butoxyethanol
0.0	0.0	0.0	0.0	0.0	0.2	0.0	0.0	0.0	0.0	2BOEAC	2-butoxyethyl acetate
				...						...	...

SNAP sector (%)										Real compound	Name
S1	S2	S3	S4	S5	S6	S7	S8	S9	S10		
				...						...	...
0.0	0.0	0.0	0.0	0.0	0.2	0.0	0.0	0.0	0.0	2EOEOL	2-ethoxyethanol
0.0	0.0	0.0	0.0	0.0	0.2	0.0	0.0	0.0	0.0	2EOEAC	2-ethoxyethyl acetate
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.3	0.0	0.0	2ETOLU	2-ethyltoluene
0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.0	0.0	0.0	2IPEOL	2-isopropoxyethanol
0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.0	0.0	0.0	2MOEOL	2-methoxyethanol
0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.0	0.0	0.0	2MOEAC	2-methoxyethyl acetate
0.0	0.0	0.0	0.0	0.0	0.0	0.2	0.0	0.0	0.0	2MBZAL	2-methyl benzaldehyde
0.0	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2M1BEN	2-methyl-1-butene
0.0	0.0	0.0	0.0	0.0	0.5	0.0	0.0	0.0	0.0	2M1POL	2-methyl-1-propanol
0.0	0.3	0.0	0.0	0.1	0.0	0.0	0.2	0.0	0.0	2M2BEN	2-methyl-2-butene
0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.0	0.0	0.0	2M5EOC	2-methyl-5-ethyloctane
0.8	7.4	3.5	1.2	6.1	0.0	9.9	2.1	0.0	0.0	2MBANE	2-methylbutane
0.0	0.0	0.0	0.0	0.0	0.3	0.0	0.0	0.0	0.0	2MDCAN	2-methyldecane
0.0	0.0	0.0	0.0	0.0	0.0	0.2	0.4	0.0	0.0	2MHPAN	2-methylheptane
0.0	0.0	0.0	0.0	0.0	0.3	0.7	0.3	0.0	0.0	2MHXAN	2-methylhexane
0.0	0.0	0.0	0.0	0.0	0.4	0.0	0.0	0.0	0.0	2MNNAN	2-methylnonane
0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.0	0.0	0.0	2MOCAN	2-methyloctane
0.0	0.0	0.3	0.4	0.8	0.4	0.0	1.0	0.3	0.0	2MPTAN	2-methylpentane
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.2	0.0	0.0	2MPRAL	2-methylpropanal
0.5	2.8	0.2	1.3	7.3	0.2	3.5	0.9	0.0	0.0	2MPANE	2-methylpropane
0.0	0.0	0.0	0.0	0.0	0.0	2.4	0.0	0.0	0.0	2MPENE	2-methylpropene
0.0	1.1	0.0	0.0	0.4	0.0	0.9	0.2	0.0	0.0	2PTENE	2-pentene
0.0	0.0	0.0	0.4	0.0	1.6	0.0	0.0	0.0	0.0	2PRPOL	2-propanol
0.0	0.0	0.0	0.0	0.0	0.7	0.0	0.0	0.0	0.0	2PRACE	2-propyl acetate
0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.0	0.0	0.0	33DMOC	3,3-dimethyloctane
0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.0	0.0	0.0	34DMHP	3,4-dimethylheptane
0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.0	0.0	0.0	37DMNN	3,7-dimethylnonane
0.0	0.0	0.0	0.0	0.0	0.4	0.0	0.0	0.0	0.0	3E2MHP	3-ethyl-2-methylheptane
0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.0	0.0	0.0	3EHPAN	3-ethylheptane
0.0	0.0	0.0	0.0	0.0	0.3	0.0	0.4	0.0	0.0	3ETOLU	3-ethyltoluene
0.0	0.0	0.0	0.0	0.0	0.0	0.2	0.0	0.0	0.0	3MBZAL	3-methyl benzaldehyde
0.0	0.3	0.0	0.0	0.0	0.0	0.0	0.2	0.0	0.0	3M1BEN	3-methyl-1-butene
0.0	0.0	0.0	0.0	0.0	0.3	0.0	0.0	0.0	0.0	3MDCAN	3-methyldecane
0.0	0.0	0.0	0.0	0.0	0.0	0.3	0.4	0.0	0.0	3MHPAN	3-methylheptane
0.0	0.0	0.0	0.0	0.0	0.3	0.5	0.2	0.0	0.0	3MHXAN	3-methylhexane
0.0	0.0	0.0	0.0	0.0	0.4	0.0	0.0	0.0	0.0	3MNNAN	3-methylnonane
				...						...	...

SNAP sector (%)										Real compound	Name
S1	S2	S3	S4	S5	S6	S7	S8	S9	S10		
				...						...	...
0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.0	0.0	0.0	3MOCAN	3-methyloctane
0.0	0.0	0.2	0.3	0.4	0.4	0.0	0.6	0.0	0.0	3MPTAN	3-methylpentane
0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.4	0.0	0.0	4ETOLU	4-ethyltoluene
0.0	0.0	0.0	0.0	0.0	2.5	0.0	0.0	0.0	0.0	4M2PON	4-methyl-2-pentanone
0.0	0.0	0.0	0.0	0.0	0.4	0.0	0.0	0.0	0.0	4M4HPO	4-methyl-4-hydroxy-2-pentanone
0.0	0.0	0.0	0.1	0.0	0.5	0.0	0.0	0.0	0.0	4MDCAN	4-methyldecane
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.4	0.0	0.0	4MHPAN	4-methylheptane
0.0	0.0	0.0	0.0	0.0	0.3	0.0	0.0	0.0	0.0	4MNNAN	4-methylnonane
0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.0	0.0	0.0	4MOCAN	4-methyloctane
0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.0	0.0	0.0	5MDCAN	5-methyldecane
0.0	0.0	0.0	0.1	0.0	0.3	0.2	0.0	1.0	0.0	ALKA10	C10 alkanes
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.5	0.0	ALKE10	C10 alkenes
0.0	0.0	0.0	0.0	0.0	0.0	0.7	0.0	0.0	0.0	AROM10	C10 aromatic hydrocarbons
0.0	0.0	0.0	0.0	0.0	0.3	0.0	0.0	0.0	0.0	CALK10	C10 cycloalkanes
0.0	0.0	0.0	0.0	0.0	0.3	0.2	0.0	0.4	0.0	ALKA11	C11 alkanes
0.0	0.0	0.0	0.0	0.0	0.1	0.2	0.0	0.0	0.0	ALKA12	C12 alkanes
0.0	0.0	0.0	0.0	0.0	0.0	3.1	0.0	0.0	0.0	ALKA13	C13+ alkanes
0.0	0.0	0.0	0.0	0.0	0.0	5.3	0.0	0.0	0.0	AROM13	C13+ aromatic hydrocarbons
0.0	0.0	0.0	0.0	0.2	0.0	0.0	0.0	0.3	0.0	ALKE06	C6 alkenes
0.0	0.0	0.0	0.3	1.2	0.0	0.0	0.0	2.5	0.0	ALKA07	C7 alkanes
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.2	0.0	ALKE07	C7 alkenes
0.0	0.0	0.0	0.6	0.1	0.0	0.0	0.0	0.0	0.0	CALK07	C7 cycloalkanes
0.0	0.0	0.1	0.1	0.9	0.0	0.0	0.0	1.6	0.0	ALKA08	C8 alkanes
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.3	0.0	ALKE08	C8 alkenes
0.0	0.0	0.0	0.2	0.0	0.0	0.0	0.0	0.0	0.0	CALK08	C8 cycloalkanes
0.0	0.0	0.0	0.1	0.0	0.0	0.0	0.0	0.6	0.0	ALKA09	C9 alkanes
0.0	0.0	0.0	0.0	0.0	0.0	2.4	0.0	0.0	0.0	AROM09	C9 aromatic hydrocarbons
0.0	0.0	0.0	0.1	0.0	0.0	0.9	0.6	0.0	0.0	ACETAL	acetaldehyde
0.0	0.0	0.0	0.3	0.0	0.0	0.0	0.0	0.0	0.0	ACETIC	acetic acid
0.5	0.0	0.8	1.1	0.0	5.0	0.3	0.6	0.0	0.0	ACETON	acetone
0.2	0.0	0.6	0.0	0.0	0.0	3.0	3.0	0.0	0.0	ACETYL	acetylene
0.0	0.0	0.0	0.0	0.0	0.0	0.3	0.0	0.0	0.0	ACROLN	acrolein
0.0	0.0	0.0	0.3	0.0	0.0	0.0	0.0	0.0	0.0	ACRYLA	acrylic acid
0.0	0.0	0.0	0.2	0.0	0.0	0.0	0.0	0.0	0.0	ACRYLN	acrylonitrile
0.0	0.0	0.0	0.0	0.0	0.0	0.4	0.0	0.0	0.0	BENZAL	benzaldehyde
1.6	6.8	12.7	0.8	0.4	0.0	4.4	3.0	0.3	0.0	BENZEN	benzene
				...						...	...

SNAP sector (%)										Real compound	Name
S1	S2	S3	S4	S5	S6	S7	S8	S9	S10		
				...						...	...
0.0	0.0	0.0	0.0	0.0	0.3	0.0	0.0	0.0	0.0	BZACOL	benzyl alcohol
4.1	8.0	9.3	5.3	30.0	4.4	7.5	1.8	0.0	0.0	BUTANE	butane
0.0	1.4	0.0	0.2	0.0	0.0	0.0	0.0	0.0	0.0	BUTENE	butene
0.0	0.0	0.0	0.0	0.0	1.4	0.0	0.0	0.0	0.0	BUTACE	butyl acetate
0.0	0.0	0.0	0.0	0.0	0.3	0.0	0.0	0.0	0.0	BCHXAN	butylcyclohexane
0.0	0.0	0.0	0.3	0.0	0.0	0.0	0.0	0.0	0.0	CARBSU	carbonyl sulphide
0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	CLEANE	chloroethane
0.0	0.0	0.0	0.3	0.0	0.0	0.0	0.0	0.0	0.0	CLEENE	chloroethene
0.0	0.0	0.0	0.8	0.0	0.0	0.0	0.0	0.0	0.0	CLMANE	chloromethane
0.0	0.0	0.0	0.0	0.3	0.0	0.0	0.0	0.0	0.0	C2BENE	cis-2-butene
0.0	0.0	0.0	0.0	0.2	0.0	0.0	0.0	0.0	0.0	C2PTEN	cis-2-pentene
0.0	0.0	0.0	0.0	0.0	0.0	0.2	0.0	0.0	0.0	CROTAL	crotonaldehyde
0.0	4.2	0.9	0.1	0.0	0.4	0.0	0.0	0.2	0.0	CYHXAN	cyclohexane
0.0	0.0	0.0	0.1	0.0	0.2	0.0	0.0	0.0	0.0	CYHXON	cyclohexanone
0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.0	0.0	0.0	DECALI	decalin
0.0	0.0	0.0	0.6	0.0	1.8	0.3	0.0	0.0	0.0	DECANE	decane
0.0	0.0	0.0	1.7	0.0	2.0	0.0	0.0	0.4	0.0	DCMANE	dichloromethane
0.0	0.0	0.0	0.3	0.0	0.0	0.0	0.0	0.3	0.0	DIIPET	diisopropyl ether
0.0	0.0	0.0	0.5	0.0	0.1	0.0	0.0	0.0	0.0	DIMETH	dimethyl ether
0.0	0.0	0.0	0.0	0.0	0.2	0.0	0.0	0.0	0.0	DMCPTA	dimethylcyclopentane
15.1	5.1	17.3	23.6	6.6	6.8	1.4	11.3	5.6	100.0	OTHERS	dimethylformamide
0.0	0.0	0.0	0.0	0.0	0.6	0.0	0.0	0.0	0.0	DIPTEN	dipentene
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.3	0.0	DIPETH	dipropyl ether
0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.2	0.0	0.0	DODCAN	dodecane
2.7	11.5	3.1	2.1	8.4	0.0	1.4	0.8	26.2	0.0	ETHANE	ethane
0.0	2.8	4.1	24.4	0.0	9.4	0.0	0.0	1.9	0.0	ETHNOL	ethanol
0.0	0.0	0.0	0.3	0.0	2.8	0.0	0.0	0.0	0.0	ETACET	ethyl acetate
0.0	0.0	0.0	0.2	0.0	0.0	0.0	0.0	0.0	0.0	EACRYL	ethyl acrylate
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.8	0.0	EBUTAT	ethyl butanoate
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.3	0.0	EPROAT	ethyl propionate
0.0	0.0	0.0	0.2	0.0	0.0	0.0	0.0	0.0	0.0	ETAMIN	ethylamine
1.3	0.0	0.2	0.2	0.0	1.2	2.4	1.1	0.8	0.0	ETBZEN	ethylbenzene
0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.0	0.0	0.0	ECHXAN	ethylcyclohexane
0.0	0.0	0.0	0.0	0.0	0.6	0.0	0.0	0.0	0.0	ETDMBZ	ethyldimethylbenzene
1.6	7.3	8.8	4.3	0.0	0.0	6.1	6.8	0.0	0.0	ETHYLE	ethylene
0.0	0.0	0.0	0.0	0.0	0.3	0.0	0.0	0.0	0.0	ETGLOL	ethylene glycol
				...						...	...

SNAP sector (%)										Real compound	Name
S1	S2	S3	S4	S5	S6	S7	S8	S9	S10		
				...						...	...
55.4	4.0	17.7	0.0	0.0	0.0	2.2	2.5	17.5	0.0	FORMAL	formaldehyde
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.2	0.0	0.0	GLYXAL	glyoxal
0.0	3.0	0.0	0.8	5.2	0.4	0.9	0.2	0.0	0.0	HPTANE	heptane
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.5	0.0	0.0	HMCTSX	hexamethylcyclotrisiloxane
1.4	2.7	1.1	3.0	5.0	1.6	4.7	0.6	0.7	0.0	HEXANE	hexane
0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.0	0.0	0.0	INDANE	indan
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.2	0.4	0.0	IPROBZ	isopropylbenzene
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.6	0.0	LIMONE	limonene
4.5	0.0	0.4	0.5	0.0	3.2	2.4	1.6	0.5	0.0	MXYLEN	m-xylene
0.0	0.0	0.0	0.6	0.0	6.4	0.0	0.0	0.5	0.0	METNOL	methanol
0.0	0.0	0.0	2.2	0.0	0.0	0.0	0.0	0.0	0.0	MEACET	methyl acetate
0.0	0.0	0.0	0.2	0.0	0.0	0.0	0.0	0.0	0.0	MEFORM	methyl formate
0.0	2.4	0.0	0.0	0.0	0.2	0.0	0.0	0.7	0.0	MCHXAN	methylcyclohexane
0.0	0.0	0.0	0.1	0.0	0.3	0.0	0.0	0.0	0.0	MCPTAN	methylcyclopentane
0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	ETOLUE	methylethylbenzene
0.0	0.0	0.0	0.0	0.2	0.0	0.0	0.2	0.0	0.0	MEPENE	methylpropene
0.0	0.0	0.0	0.0	0.0	0.3	0.0	0.0	0.0	0.0	MEPRBZ	methylpropylbenzene
0.0	0.2	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	NAPHTA	naphthalene
0.0	0.0	0.0	0.4	0.0	1.0	0.0	0.0	0.0	0.0	NONANE	nonane
1.0	0.0	0.0	0.2	0.0	0.8	2.4	1.3	0.3	0.0	OXYLEN	o-xylene
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.2	0.0	0.0	OMCTSX	octamethylcyclotetrasiloxane
0.0	0.0	0.0	0.5	4.6	0.0	0.4	0.1	0.0	0.0	OCTANE	octane
0.0	0.0	0.0	0.2	0.0	0.8	1.8	1.6	0.4	0.0	PXYLEN	p-xylene
1.9	13.8	10.1	3.0	10.2	0.0	5.1	1.1	0.0	0.0	PENTAN	pentane
0.0	0.0	0.0	0.2	0.0	0.0	0.0	0.0	0.0	0.0	PHENOL	phenol
2.7	6.2	4.2	4.2	10.3	0.9	0.6	0.5	26.2	0.0	PROPAN	propane
0.0	0.0	0.0	0.0	0.0	0.3	0.0	0.0	0.0	0.0	PRACET	propyl acetate
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.2	0.0	PBUTAT	propyl butanoate
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.5	0.0	PPROAT	propyl propionate
0.0	0.0	0.0	0.0	0.0	0.3	0.0	0.2	0.4	0.0	PROPBZ	propylbenzene
0.0	0.0	0.0	0.0	0.0	0.4	0.0	0.0	0.0	0.0	PCHXAN	propylcyclohexane
1.7	3.7	1.2	3.8	0.0	0.0	3.0	2.5	0.0	0.0	PROPYL	propylene
0.0	0.0	0.0	0.0	0.0	0.0	0.3	0.0	0.0	0.0	PROPYN	propyne
0.0	0.0	0.0	0.2	0.0	0.3	0.5	0.0	0.0	0.0	STYREN	styrene
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.8	0.0	TERPEN	terpenes
0.0	0.0	0.0	0.2	0.0	1.5	0.0	0.0	0.8	0.0	TECEEN	tetrachloroethene
				...						...	...

SNAP sector (%)										Real compound	Name
S1	S2	S3	S4	S5	S6	S7	S8	S9	S10		
				...						...	...
0.0	0.0	0.0	0.0	0.0	0.2	0.0	0.0	0.0	0.0	TEDCAN	tetradecane
0.0	0.0	0.0	0.3	0.0	0.0	0.0	0.0	0.0	0.0	THFRAN	tetrahydrofuran
1.1	3.0	2.0	1.1	0.2	5.2	8.0	4.7	1.1	0.0	TOLUEN	toluene
0.0	0.0	0.0	0.0	0.3	0.0	0.0	0.0	0.0	0.0	T2BENE	trans-2-butene
0.0	0.0	0.0	0.0	0.3	0.0	0.0	0.0	0.0	0.0	T2PTEN	trans-2-pentene
0.0	0.0	0.0	1.0	0.0	4.0	0.0	0.0	0.4	0.0	TCEENE	trichloroethene
0.0	0.0	0.0	0.2	0.0	0.0	0.0	0.0	0.0	0.0	TCMANE	trichloromethane
0.0	0.0	0.0	0.2	0.0	0.0	0.0	0.0	0.0	0.0	TFMANE	trifluoromethane
0.0	0.0	0.0	0.2	0.0	0.9	0.0	0.2	0.0	0.0	UNDCAN	undecane
0.0	0.0	0.0	0.0	0.0	0.0	0.7	33.2	0.0	0.0	ALKAUN	unspeciated alkanes
0.0	0.0	0.0	0.0	0.0	0.0	0.0	4.1	0.0	0.0	ALKEUN	unspeciated alkenes
0.0	0.0	0.0	0.0	0.0	0.4	0.0	2.8	0.0	0.0	AROMUN	unspeciated aromatic hydrocarbons

Table F.1: split of VOC in real species according to the SNAP sector [Passant, 2002].

G Chemical reactions

G.1 Reaction list of the reduced MELCHIOR1 chemical mechanism

Reactions	Kinetic constants
Inorganic	
O3+NO→NO2	k(T)=Ae$^{-B/T}$,A=1.8e-12,B=1370
O3+NO2→NO3	k(T)=Ae$^{-B/T}$,A=1.2e-13,B=2450
O3+OH→HO2	k(T)=Ae$^{-B/T}$,A=1.9e-12,B=1000
O3+HO2→OH	k(T)=Ae$^{-B/T}$,A=1.4e-14,B=600
NO+HO2→OH+NO2	k(T)=Ae$^{-B/T}$,A=3.7e-12,B=-240
NO2+OH+M→HNO3	k(T,M)=mtroe(3.4e-30,0,3.2,4.77e-11,0,1.4,0.30)
HO2+OH→H2O	k(T)=Ae$^{-B/T}$,A=4.8e-11,B=-250
H2O2+OH→HO2	k(T)=Ae$^{-B/T}$,A=2.9e-12,B=160
HNO3+OH→NO3	k(T)=Ae$^{-B/T}$,A=7.2e-15,B=-785
HNO3+OH+M→NO3	k(T,M)=troe(1.9e-33,-725,0,4.1e-16,-1440,0,1)
CO+OH→HO2+CO2	k(T)=Ae$^{-B/T}$(300/T)^{N},A=1.3e-13,B=0,N=1
CO+OH+M→HO2+CO2	k=3.21e-33
H2+OH→HO2	k(T)=Ae$^{-B/T}$,A=7.7e-12,B=2100
HO2+HO2→H2O2	k(T)=Ae$^{-B/T}$,A=2.2e-13,B=-600
HO2+HO2+M→H2O2	k(T)=Ae$^{-B/T}$,A=1.7e-33,B=-1000
HO2+HO2+H2O→H2O2	k(T)=Ae$^{-B/T}$,A=3.22e-34,B=-2800
HO2+HO2+H2O+M→H2O2	k(T)=Ae$^{-B/T}$,A=2.38e-54,B=-3200
NO3+HO2→NO2+OH	k=4e-12
NO3+H2O2→HNO3+HO2	k=2e-15
NO3+O3→NO2	k=1e-17
NO3+NO→2*NO2	k(T)=Ae$^{-B/T}$,A=1.8e-11,B=-110
NO3+NO3→2*NO2	k(T)=Ae$^{-B/T}$,A=8.5e-13,B=2450
NO2+NO3→NO+NO2	k(T)=Ae$^{-B/T}$,A=4.5e-14,B=1260
NO2+NO3+M→N2O5	k(T,M)=troe(2.7e-30,0,3.4,2e-12,0,-0.2,0.33)
N2O5+M→NO3+NO2	k(T,M)=troe(1e-3,11000,3.5,9.7e14,11080,-0.1,0.33)
NO2+HO2+M→HNO4	k(T,M)=troe(1.8e-31,0,3.2,4.7e-12,0,0,0.6)
HNO4+M→HO2+NO2	k(T,M)=troe(5e-6,10000,0,2.6e15,10900,0,0.6)
HNO4+OH→NO2	k(T)=Ae$^{-B/T}$,A=1.5e-12,B=-360
NO+OH+M→HONO	k(T,M)=troe(7.e-31,0,2.6,1.5e-11,0,0.5,0.6)
HONO+OH→NO2	k(T)=Ae$^{-B/T}$,A=1.8e-11,B=390
NO2→HONO+NO2	ks=0.5*depo(NO2)
SOx	
SO2+CH3O2→H2SO4+HCHO+HO2	k=4e-17
SO2+OH+M→H2SO4+HO2	k(T,M)=troe(4e-31,0,3.3,2e-12,0,0,0.45)
OH	
CH4+OH→CH3O2	k(T)=Ae$^{-B/T}$,A=2.3e-12,B=1765
C2H6+OH→C2H5O2	k(T)=Ae$^{-B/T}$,A=7.9e-12,B=1030
C2H4+OH+M→CH2O2C	k(T,M)=troe(7e-29,0,3.1,9e-12,0,0,0.7)
C3H6+OH+M→CH3CHX	k(T,M)=troe(8e-27,0,3.5,3e-11,0,0,0.5)
CH3OH+OH→HCHO+HO2	k(T)=Ae$^{-B/T}$,A=3.1e-12,B=360
C2H5OH+OH→CH3CHO+HO2	k(T)=Ae$^{-B/T}$,A=4.1e-12,B=70
OH	
HCHO+OH→CO+HO2	k(T)=Ae$^{-B/T}$,A=8.6e-12,B=-20
CH3CHO+OH→CH3COO	k(T)=Ae$^{-B/T}$,A=5.6e-12,B=-310
CH3COE+OH→CH3COX	k(T)=Ae$^{-B/T}$(300/T)^{N},A=2.92e-13,B=-414,N=-2
MEMALD+OH→MEMAL1	k=5.6e-11
GLYOX+OH→2*CO+HO2	k=1.1e-11
MGLYOX+OH→CH3COO+CO	k=1.5e-11
OFURAN+OH→MVKO2	k=6.9e-11

Reactions	Kinetic constants
MAC+OH→0.5*MACO2	k(T)=A$e^{-B/T}$,A=1.86e-11,B=-175
MVK+OH→MVKO2	k(T)=A$e^{-B/T}$,A=4.1e-12,B=-453
CH3O2H+OH→CH3O2	k(T)=A$e^{-B/T}$,A=1.9e-12,B=-190
ETO2H+OH→C2H5O2	k(T)=A$e^{-B/T}$,A=1.9e-12,B=-190
PPA+OH→CH3COO	k(T)=A$e^{-B/T}$,A=1.9e-12,B=-190
BUO2H+OH→SECC4H	k(T)=A$e^{-B/T}$,A=1.9e-12,B=-190
CH4COX+OH→CH3COX	k(T)=A$e^{-B/T}$,A=1.9e-12,B=-190
CH3O2C+OH→CH2O2C	k(T)=A$e^{-B/T}$,A=1.9e-12,B=-190
CH4CHX+OH→CH3CHX	k(T)=A$e^{-B/T}$,A=1.9e-12,B=-190
OXYLH1+OH→OXYL1	k(T)=A$e^{-B/T}$,A=1.9e-12,B=-190
MEMALH+OH→MEMAL1	k(T)=A$e^{-B/T}$,A=1.9e-12,B=-190
MEMAH2+OH→MEMAL2	k(T)=A$e^{-B/T}$,A=1.9e-12,B=-190
PEROX+OH→RO2IP1	k(T)=A$e^{-B/T}$,A=1.9e-12,B=-190
MACO2H+OH→MACO2	k(T)=A$e^{-B/T}$,A=1.9e-12,B=-190
MVKO2H+OH→MVKO2	k(T)=A$e^{-B/T}$,A=1.9e-12,B=-190
CH4COZ+OH→CH3COZ	k(T)=A$e^{-B/T}$,A=1.9e-12,B=-190
APIO2H+OH→APIOH	k(T)=A$e^{-B/T}$,A=1.9e-12,B=-190
CH3O2H+OH→HCHO+OH	k(T)=A$e^{-B/T}$,A=1.e-12,B=-190
ETO2H+OH→CH3CHO+OH	k(T)=A$e^{-B/T}$(300/T)^{N},A=1.30e-13,B=-1176,N=-2
BUO2H+OH→CH3COE+OH	k(T)=A$e^{-B/T}$(300/T)^{N},A=6.1e-14,B=-1681,N=-2
CH4COX+OH→CH3COY+OH	k(T)=A$e^{-B/T}$(300/T)^{N},A=6.1e-14,B=-1533,N=-2
CH3O2C+OH→2*HCHO+OH	k(T)=A$e^{-B/T}$(300/T)^{N},A=1.30e-13,B=-1238,N=-2
CH3O2C+OH→2*HCHO+OH	k(T)=A$e^{-B/T}$(300/T)^{N},A=4.05e-13,B=-688,N=-2
CH4CHX+OH→HCHO+CH3CHO+OH	k(T)=A$e^{-B/T}$(300/T)^{N},A=4.05e-13,B=-688,N=-2
CH4CHX+OH→HCHO+CH3CHO+OH	k(T)=A$e^{-B/T}$(300/T)^{N},A=6.1e-14,B=-1681,N=-2
OXYLH1+OH→MEMALD+MGLYOX+OH	k=1.44e-10
MEMALH+OH→GLYOX+MGLYOX+OH	k(T)=A$e^{-B/T}$(300/T)^{N},A=1.71e-13,B=-1403,N=-2
MEMAH2+OH→MALEIC+CH3O2	k=1.27e-11
PEROX+OH→0.43*MAC+0.57*MVK+HCHO+OH	k=8.22e-11
MACO2H+OH→CH3COZ	k=1.28e-11
MVKO2H+OH→0.28*HCHO+0.28*MGLYOX+0.28*OH+0.72*CH3CHO +0.72*CH3COO	k(T)=A$e^{-B/T}$(300/T)^{N},A=4.05e-13,B=-1032,N=-2
MVKO2H+OH→0.28*HCHO+0.28*MGLYOX+0.28*OH+0.72*CH3CHO +0.72*CH3COO	k(T)=A$e^{-B/T}$(300/T)^{N},A=6.1e-14,B=-1595,N=-2
CH4COZ+OH→CH3COE+OH	k=5.14e-11
APIO2H+OH→CH3CHO+CH3COE+OH	k(T)=A$e^{-B/T}$(300/T)^{N},A=1.91e-13,B=-1193,N=-2
APIO2H+OH→CH3CHO+CH3COE+OH	k(T)=A$e^{-B/T}$(300/T)^{N},A=3.82e-13,B=-882,N=-2
NO3	
CH4+NO3→CH3O2+HNO3	k=4e-19
C2H6+NO3→C2H5O2+HNO3	k=8e-18
NC4H10+NO3→SECC4H+HNO3	k=5.5e-17
C2H4+NO3→RNC2H4	k=2e-16
C3H6+NO3→RNC3H6	k=9.45e-15
OXYL+NO3→ROXYL1+HNO3	k=3.7e-16
C5H8+NO3→RNC5H8	k=7.8e-13
CH3OH+NO3→CH3O2+HNO3	k(T)=A$e^{-B/T}$,A=1.3e-12,B=2560
C2H5OH+NO3→C2H5O2+HNO3	k=9e-16
HCHO+NO3→CO+HNO3+HO2	k=5.8e-16
CH3CHO+NO3→CH3COO+HNO3	k=2.8e-15
CH3O2+NO3→HCHO+HO2+NO2	k=1.2e-12
C2H5O2+NO3→CH3CHO+NO2+HO2	k=1.2e-12
CH3COO+NO3→CH3O2+NO2+CO2	k=4e-12
CH2O2C+NO3→2*HCHO+NO2+HO2	k=1.2e-12
SECC4H+NO3→SECBUO+NO2	k=1.2e-12
CH3COX+NO3→CH3COY+NO2+HO2	k=1.2e-12
CH3CHX+NO3→HCHO+CH3CHO+NO2+HO2	k=1.2e-12
OXYL1+NO3→MEMALD+MGLYOX+NO2+HO2	k=1.2e-12
MEMAL1+NO3→GLYOX+MGLYOX+NO2+HO2	k=1.2e-12
MEMAL2+NO3→MALEIC+NO2+CH3O2	k=4e-12
MACO2+NO3→CH3COZ+NO2+CO2	k=4e-12
RO2IP1+NO3→0.38*MAC+0.50*MVK+0.88*HCHO+0.12*RO2IP1 +0.88*HO2+NO2	k=1.2e-12
MVKO2+NO3→0.28*MGLYOX+0.28*HCHO+0.72*CH3CHO +0.72*CH3COO+0.28*HO2+NO2	k=1.2e-12
CH3COZ+NO3→CH3COE+HO2+NO2	k=1.2e-12

Reactions	Kinetic constants
APIOH+NO3→CH3CHO+CH3COE+NO2+HO2	k=1.2e-12
RNC2H4+NO3→0.5*CARNIT+HCHO +1.5*NO2+0.5*HO2	k=1.2e-12
RNC3H6+NO3→0.5*CARNIT+0.5*CH3CHO+0.5*HCHO +1.5*NO2+0.5*HO2	k=1.2e-12
ROXYL1+NO3→BENZAL+NO2+HO2	k=1.2e-12
ISNIR+NO3→0.95*CH3CHO+0.475*CH3COE+0.475*MGLYOX +0.05*ISNI+0.05*HO2+1.9*NO2	k=1.2e-12
RNC5H8+NO3→0.85*ISNI+0.1*MAC+0.05*MVK+0.15*HCHO +1.1*NO2+0.8*HO2	k=1.2e-12
APINO3+NO3→CH3CHO+CH3COE+2*NO2	k=1.2e-12
O3	
C2H4+O3→HCHO+0.12*HO2+0.13*H2+0.44*CO	k(T)=Ae$^{-B/T}$,A=9.1e-15,B=2580
C3H6+O3→0.53*HCHO+0.5*CH3CHO+0.31*CH3O2+0.28*HO2 +0.15*OH+0.065*H2+0.4*CO+0.7*CH4	k(T)=Ae$^{-B/T}$,A=5.5e-15,B=1880
C5H8+O3→0.67*MAC+0.26*MVK+0.55*OH+0.07*C3H6+0.8*HCHO +0.06*HO2+0.05*CO+0.3*O3	k(T)=Ae$^{-B/T}$,A=1.2e-14,B=2013
MAC+O3→0.8*MGLYOX+0.7*HCHO+0.215*OH+0.275*HO2 +0.2*CO+0.2*O3	k(T)=Ae$^{-B/T}$,A=5.3e-15,B=2520
MVK+O3→0.82*MGLYOX+0.8*HCHO+0.04*CH3CHO+0.08*OH +0.06*HO2+0.05*CO+0.2*O3	k(T)=Ae$^{-B/T}$,A=4.3e-15,B=2016
PEROX+O3→0.7*HCHO	k=8e-18
organic	
CH3O2+NO→HCHO+NO2+HO2	k(T)=Ae$^{-B/T}$,A=4.2e-12,B=-180
C2H5O2+NO→CH3CHO+NO2+HO2	k=8.7e-12
CH3COO+NO→CH3O2+NO2+CO2	k=2e-11
CH2O2C+NO→2*HCHO+NO2+HO2	k=9e-12
SECC4H+NO→SECBUO+NO2	k=4e-12
SECC4H+NO+M→SECBUO+NO2	k(T,M)=troe(-3.76e-32,0,0,-3.3e-12,0,8.1,0.41)
SECBUO+M→CH3COE+HO2	k(T)=Ae$^{-B/T}$,A=3.0e-14,B=200
SECBUO→CH3CHO+C2H5O2	k(T)=Ae$^{-B/T}$,A=7.20e+12,B=5780
CH3COX+NO→CH3COY+NO2+HO2	k=4e-12
CH3CHX+NO→HCHO+CH3CHO+NO2+HO2	k=4e-12
OXYL1+NO→MEMALD+MGLYOX+NO2+HO2	k=4e-12
MEMAL1+NO→GLYOX+MGLYOX+NO2+HO2	k=4e-12
MEMAL2+NO→MALEIC+NO2+CH3O2	k=1.4e-11
MACO2+NO→CH3COZ+NO2+CO2	k=1.4e-11
RO2IP1+NO→0.32*MAC+0.42*MVK+0.74*HCHO+0.14*ISNI +0.12*RO2IP1+0.78*HO2+0.86*NO2	k(T)=Ae$^{-B/T}$,A=1.4e-11,B=180
MVKO2+NO→0.266*MGLYOX+0.266*HCHO+0.684*CH3CHO +0.684*CH3COO+0.05*ISNI+0.266*HO2+0.95*NO2	k(T)=Ae$^{-B/T}$,A=1.4e-11,B=180
CH3COZ+NO→CH3COE+HO2+NO2	k(T)=Ae$^{-B/T}$,A=1.4e-11,B=180
APIOH+NO→0.8*CH3CHO+0.8*CH3COE+0.8*NO2+0.8*HO2+0.2*ISNI	k=4e-12
organic	
CH3O2+HO2→CH3O2H	k(T)=Ae$^{-B/T}$,A=4.1e-13,B=-790
C2H5O2+HO2→ETO2H	k(T)=Ae$^{-B/T}$,A=2.7e-13,B=-1000
CH3COO+HO2→0.67*PPA+0.33*O3	k(T)=Ae$^{-B/T}$,A=4.3e-13,B=-1040
SECC4H+HO2→BUO2H	k(T)=Ae$^{-B/T}$,A=2.7e-13,B=-1000
CH3COX+HO2→CH4COX	k(T)=Ae$^{-B/T}$,A=2.7e-13,B=-1000
CH2O2C+HO2→CH3O2C	k(T)=Ae$^{-B/T}$,A=2.7e-13,B=-1000
CH3CHX+HO2→CH4CHX	k(T)=Ae$^{-B/T}$,A=2.7e-13,B=-1000
OXYL1+HO2→OXYLH1	k(T)=Ae$^{-B/T}$,A=2.7e-13,B=-1000
MEMAL1+HO2→MEMALH	k(T)=Ae$^{-B/T}$,A=2.7e-13,B=-1000
MEMAL2+HO2→MEMAH2	k(T)=Ae$^{-B/T}$,A=2.7e-13,B=-1000
RO2IP1+HO2→PEROX	k(T)=Ae$^{-B/T}$,A=2.7e-13,B=-1000
MACO2+HO2→MACO2H	k(T)=Ae$^{-B/T}$,A=2.7e-13,B=-1000
MVKO2+HO2→MVKO2H	k(T)=Ae$^{-B/T}$,A=2.7e-13,B=-1000
CH3COZ+HO2→CH4COZ	k(T)=Ae$^{-B/T}$,A=2.7e-13,B=-1000
APIOH+HO2→APIO2H	k(T)=Ae$^{-B/T}$,A=2.7e-13,B=-1000
CH3O2+CH3O2→2*HCHO+2*HO2	k(T)=SPECIAL_1(1.1e-13,-365,25,1165)
CH3O2+CH3O2→HCHO+CH3OH	k(T)=SPECIAL_2(1.1e-13,-365,25,1165)
C2H5O2+CH3O2→CH3CHO+HCHO+2*HO2	k(T)=SPECIAL_3(6.4e-14,0,1.1e-13,-365,10.2,533,25,1165)
C2H5O2+CH3O2→CH3CHO+CH3OH	k(T)=SPECIAL_4(6.4e-14,0,1.1e-13,-365,10.2,533,25,1165)
C2H5O2+CH3O2→C2H5OH+HCHO	k(T)=SPECIAL_4(6.4e-14,0,1.1e-13,-365,25,1165,10.2,533)
CH3COO+CH3O2→CH3O2+CO2+HCHO+HO2	k(T)=SPECIAL_1(5.1e-12,-272,4.4e5,3910)
CH3COO+CH3O2→MECOOH+HCHO	k(T)=SPECIAL_2(5.1e-12,-272,4.4e5,3910)
SECC4H+CH3O2→SECBUO+HCHO+HO2	k(T)=SPECIAL_3(1.2e-12,1500,1.1e-13,-365,50,1165,25,1165)

Reactions	Kinetic constants
SECC4H+CH3O2→CH3COE+CH3OH	k(T)=SPECIAL_4(1.2e-12,1500,1.1e-13,-365,50,1165,25,1165)
SECC4H+CH3O2→CH3COE+HCHO	k(T)=SPECIAL_4(1.2e-12,1500,1.1e-13,-365,25,1165,50,1165)
CH3COX+CH3O2→CH3COY+HCHO+2*HO2	k(T)=SPECIAL_3(5.5e-11,1500,1.1e-13,-365,50,1165,25,1165)
CH3COX+CH3O2→CH3COY+CH3OH	k(T)=SPECIAL_4(5.5e-11,1500,1.1e-13,-365,50,1165,25,1165)
CH3COX+CH3O2→CH3COY+HCHO	k(T)=SPECIAL_4(5.5e-11,1500,1.1e-13,-365,25,1165,50,1165)
CH2O2C+CH3O2→3*HCHO+2*HO2	k(T)=SPECIAL_3(2.3e-12,0,1.1e-13,-365,50,1165,25,1165)
CH2O2C+CH3O2→2*HCHO+CH3OH	k(T)=SPECIAL_4(2.3e-12,0,1.1e-13,-365,50,1165,25,1165)
CH2O2C+CH3O2→3*HCHO	k(T)=SPECIAL_4(2.3e-12,0,1.1e-13,-365,25,1165,50,1165)
CH3CHX+CH3O2→CH3CHO+2*HCHO+2*HO2	k(T)=SPECIAL_3(1.3e-10,2200,1.1e-13,-365,50,1165,25,1165)
CH3CHX+CH3O2→CH3CHO+HCHO+CH3OH	k(T)=SPECIAL_4(1.3e-10,2200,1.1e-13,-365,50,1165,25,1165)
CH3CHX+CH3O2→CH3CHO+2*HCHO	k(T)=SPECIAL_4(1.3e-10,2200,1.1e-13,-365,25,1165,50,1165)
RNC3H6+CH3O2→0.5*CARNIT+1.5*HCHO+0.5*CH3CHO +0.5*NO2+1.5*HO2	k(T)=SPECIAL_3(1.3e-10,2200,1.1e-13,-365,50,1165,25,1165)
RNC3H6+CH3O2→CARNIT+CH3OH	k(T)=SPECIAL_4(1.3e-10,2200,1.1e-13,-365,50,1165,25,1165)
RNC3H6+CH3O2→ORNIT1+HCHO	k(T)=SPECIAL_4(1.3e-10,2200,1.1e-13,-365,25,1165,50,1165)
OXYL1+CH3O2→MEMALD+MGLYOX+HCHO+2*HO2	k(T)=SPECIAL_3(2.1e-13,-600,1.1e-13,-365,50,1165,25,1165)
OXYL1+CH3O2→MEMALD+MGLYOX+CH3OH	k(T)=SPECIAL_4(2.1e-13,-600,1.1e-13,-365,50,1165,25,1165)
OXYL1+CH3O2→MEMALD+MGLYOX+HCHO	k(T)=SPECIAL_4(2.1e-13,-600,1.1e-13,-365,25,1165,50,1165)
MEMAL1+CH3O2→GLYOX+MGLYOX+HCHO+2*HO2	k(T)=SPECIAL_3(1.3e-11,800,1.1e-13,-365,50,1165,25,1165)
MEMAL1+CH3O2→GLYOX+MGLYOX+CH3OH	k(T)=SPECIAL_4(1.3e-11,800,1.1e-13,-365,50,1165,25,1165)
MEMAL1+CH3O2→GLYOX+MGLYOX+HCHO	k(T)=SPECIAL_4(1.3e-11,800,1.1e-13,-365,25,1165,50,1165)
MEMAL2+CH3O2→MALEIC+CH3O2+HCHO+HO2	k(T)=SPECIAL_1(5.1e-12,-272,4.4e5,3910)
MEMAL2+CH3O2→MAA+HCHO	k(T)=SPECIAL_2(5.1e-12,-272,4.4e5,3910)
RO2IP1+CH3O2→0.3*MVK+0.3*MAC+1.6*HCHO+2*HO2	k(T)=SPECIAL_3(1.3e-11,800,1.1e-13,-365,50,1165,25,1165)
RO2IP1+CH3O2→0.3*MVK+0.3*MAC+0.6*HCHO+CH3OH	k(T)=SPECIAL_4(1.3e-11,800,1.1e-13,-365,50,1165,25,1165)
RO2IP1+CH3O2→0.3*MVK+0.3*MAC+1.6*HCHO	k(T)=SPECIAL_4(1.3e-11,800,1.1e-13,-365,25,1165,50,1165)
RNC5H8+CH3O2→0.6*ISNI+HCHO+2*HO2	k(T)=SPECIAL_3(1.3e-11,800,1.1e-13,-365,50,1165,25,1165)
RNC5H8+CH3O2→0.6*ISNI+CH3OH	k(T)=SPECIAL_4(1.3e-11,800,1.1e-13,-365,50,1165,25,1165)
RNC5H8+CH3O2→0.6*ISNI+HCHO	k(T)=SPECIAL_4(1.3e-11,800,1.1e-13,-365,25,1165,50,1165)
MACO2+CH3O2→CH3COZ+CO2+HCHO+HO2	k(T)=SPECIAL_1(5.1e-12,-272,4.4e5,3910)
MACO2+CH3O2→MAA+HCHO	k(T)=SPECIAL_2(5.1e-12,-272,4.4e5,3910)
MVKO2+CH3O2→MGLYOX+2*HCHO+2*HO2	k(T)=SPECIAL_3(5.5e-11,1500,1.1e-13,-365,50,1165,25,1165)
MVKO2+CH3O2→MGLYOX+HCHO+CH3OH	k(T)=SPECIAL_4(5.5e-11,1500,1.1e-13,-365,50,1165,25,1165)
MVKO2+CH3O2→MGLYOX+2*HCHO	k(T)=SPECIAL_4(5.5e-11,1500,1.1e-13,-365,25,1165,50,1165)
APIOH+CH3O2→CH3CHO+CH3COE+HCHO+2*HO2	k(T)=SPECIAL_3(1.8e-11,2200,1.1e-13,-365,1e9,0,25,1165)
APIOH+CH3O2→CH3CHO+CH3COE+HCHO	k(T)=SPECIAL_4(1.8e-11,2200,1.1e-13,-365,1e9,0,25,1165)
APINO3+CH3O2→CH3CHO+CH3COE+NO2+HCHO+HO2	k(T)=SPECIAL_3(1.8e-11,2200,1.1e-13,-365,1e9,0,25,1165)
APINO3+CH3O2→ORNIT2+HCHO	k(T)=SPECIAL_4(1.8e-11,2200,1.1e-13,-365,1e9,0,25,1165)
CH3COO+CH3COO→2.*CH3O2+2.*CO2	k(T)=Ae$^{-B/T}$,A=2.8e-12,B=-530
C2H5O2+CH3COO→CH3CHO+CH3O2+CO2+HO2	k(T)=SPECIAL_3(6.4e-14,0,2.8e-12,-530,10.2,533,1e9,0)
C2H5O2+CH3COO→CH3CHO+MECOOH	k(T)=SPECIAL_4(6.4e-14,0,2.8e-12,-530,1e9,0,10.2,533)
SECC4H+CH3COO→SECBUO+CH3O2+CO2	k(T)=SPECIAL_3(1.2e-12,1500,2.8e-12,-530,1e9,0,50,1165)
SECC4H+CH3COO→CH3COE+MECOOH	k(T)=SPECIAL_4(1.2e-12,1500,2.8e-12,-530,1e9,0,50,1165)
CH3COX+CH3COO→CH3COY+CH3O2+CO2+HO2	k(T)=SPECIAL_3(5.5e-11,1500,2.8e-12,-530,1e9,0,50,1165)
CH3COX+CH3COO→CH3COY+MECOOH	k(T)=SPECIAL_4(5.5e-11,1500,2.8e-12,-530,1e9,0,50,1165)
CH2O2C+CH3COO→2*HCHO+CH3O2+CO2+HO2	k(T)=SPECIAL_3(2.3e-12,0,2.8e-12,-530,1e9,0,50,1165)
CH2O2C+CH3COO→2*HCHO+MECOOH	k(T)=SPECIAL_4(2.3e-12,0,2.8e-12,-530,1e9,0,50,1165)
CH3CHX+CH3COO→CH3CHO+HCHO+CH3O2+CO2+HO2	k(T)=SPECIAL_3(1.2e-10,2200,2.8e-12,-530,1e9,0,50,1165)
CH3CHX+CH3COO→CH3CHO+HCHO+MECOOH	k(T)=SPECIAL_4(1.2e-10,2200,2.8e-12,-530,1e9,0,50,1165)
RNC3H6+CH3COO→0.5*CARNIT+0.5*HCHO+0.5*CH3CHO+0.5*NO2 +CH3O2+CO2+0.5*HO2	k(T)=SPECIAL_3(1.2e-10,2200,2.8e-12,-530,1e9,0,50,1165)
RNC3H6+CH3COO→ORNIT1+MECOOH	k(T)=SPECIAL_4(1.2e-10,2200,2.8e-12,-530,1e9,0,50,1165)
OXYL1+CH3COO→MEMALD+MGLYOX+CH3O2+CO2+HO2	k(T)=SPECIAL_3(2.1e-13,-600,2.8e-12,-530,1e9,0,50,1165)
OXYL1+CH3COO→MEMALD+MGLYOX+MECOOH	k(T)=SPECIAL_4(2.1e-13,-600,2.8e-12,-530,1e9,0,50,1165)
MEMAL1+CH3COO→GLYOX+MGLYOX+CH3O2+CO2+HO2	k(T)=SPECIAL_3(1.3e-11,800,2.8e-12,-530,1e9,0,50,1165)
MEMAL1+CH3COO→GLYOX+MGLYOX+MECOOH	k(T)=SPECIAL_4(1.3e-11,800,2.8e-12,-530,1e9,0,50,1165)
MEMAL2+CH3COO→MALEIC+2*CH3O2+CO2	k(T)=Ae$^{-B/T}$,A=5.6e-12,B=-530
RO2IP1+CH3COO→0.3*MVK+0.3*MAC+0.6*HCHO+HO2 +CH3O2+CO2	k(T)=SPECIAL_3(1.3e-11,800,2.8e-12,-530,1e9,0,50,1165)
RO2IP1+CH3COO→0.3*MVK+0.3*MAC+0.6*HCHO+MECOOH	k(T)=SPECIAL_4(1.3e-11,800,2.8e-12,-530,1e9,0,50,1165)
RNC5H8+CH3COO→0.6*ISNI+CH3O2+CO2	k(T)=SPECIAL_3(1.3e-11,800,2.8e-12,-530,1e9,0,50,1165)
RNC5H8+CH3COO→0.6*ISNI*+MECOOH	k(T)=SPECIAL_4(1.3e-11,800,2.8e-12,-530,1e9,0,50,1165)
MACO2+CH3COO→CH3O2+CH3COZ+2*CO2	k(T)=Ae$^{-B/T}$,A=5.6e-12,B=-530
MVKO2+CH3COO→MGLYOX+HCHO+HO2+CH3O2+CO2	k(T)=SPECIAL_3(5.5e-11,1500,2.8e-12,-530,1e9,0,50,1165)
MVKO2+CH3COO→MGLYOX+HCHO+MECOOH	k(T)=SPECIAL_4(5.5e-11,1500,2.8e-12,-530,1e9,0,50,1165)
APIOH+CH3COO→CH3CHO+CH3COE+HO2+CH3O2+CO2	k(T)=Ae$^{-B/T}$,A=1.4e-11,B=835
APINO3+CH3COO→CH3CHO+CH3COE+NO2+CH3O2+CO2	k(T)=Ae$^{-B/T}$,A=1.4e-11,B=835
C2H5O2+C2H5O2→2*CH3CHO+2*HO2	k(T)=SPECIAL_1(6.4e-14,0,10.2,533)

Reactions	Kinetic constants
C2H5O2+C2H5O2→CH3CHO+C2H5OH	k(T)=SPECIAL_2(6.4e-14,0,10.2,533)
SECC4H+SECC4H→2*SECBUO	k(T)=SPECIAL_1(1.2e-12,1500,50,1165)
SECC4H+SECC4H→2*CH3COE	k(T)=SPECIAL_2(1.2e-12,1500,50,1165)
CH3COX+CH3COX→2*CH3COY+2*HO2	k(T)=SPECIAL_1(5.5e-11,1500,50,1165)
CH3COX+CH3COX→2*CH3COY	k(T)=SPECIAL_2(5.5e-11,1500,50,1165)
CH2O2C+CH2O2C→4*HCHO+2*HO2	k(T)=SPECIAL_1(2.3e-12,0,50,1165)
CH2O2C+CH2O2C→4*HCHO	k(T)=SPECIAL_2(2.3e-12,0,50,1165)
CH3CHX+CH3CHX→2*HCHO+2*CH3CHO+2*HO2	k(T)=SPECIAL_1(1.3e-10,2200,50,1165)
CH3CHX+CH3CHX→2*HCHO+2*CH3CHO	k(T)=SPECIAL_2(1.3e-10,2200,50,1165)
RNC3H6+RNC3H6→CARNIT+HO2+HCHO+CH3CHO+NO2	k(T)=SPECIAL_1(1.3e-10,2200,50,1165)
RNC3H6+RNC3H6→CARNIT+ORNIT1	k(T)=SPECIAL_2(1.3e-10,2200,50,1165)
OXYL1+OXYL1→2*MEMALD+2*MGLYOX+2*HO2	k(T)=SPECIAL_1(2.1e-13,-600.,50,1165)
OXYL1+OXYL1→2*MEMALD+2*MGLYOX	k(T)=SPECIAL_2(2.1e-13,-600.,50,1165)
MEMAL1+MEMAL1→2*GLYOX+2*MGLYOX+2*HO2	k(T)=SPECIAL_1(1.3e-11,800,50,1165)
MEMAL1+MEMAL1→2*GLYOX+2*MGLYOX	k(T)=SPECIAL_2(1.3e-11,800,50,1165)
MEMAL2+MEMAL2→2*MALEIC+2*CH3O2	k(T)=$Ae^{-B/T}$,A=2.8e-12,B=-530
RO2IP1+RO2IP1→0.6*MVK+0.6*MAC+1.2*HCHO+1.2*HO2	k(T)=SPECIAL_1(1.3e-11,800,50,1165)
RO2IP1+RO2IP1→0.6*MVK+0.6*MAC+1.2*HCHO	k(T)=SPECIAL_2(1.3e-11,800,50,1165)
RNC5H8+RNC5H8→1.2*ISNI+1.2*HO2	k(T)=SPECIAL_1(1.3e-11,800,50,1165)
RNC5H8+RNC5H8→1.2*ISNI	k(T)=SPECIAL_2(1.3e-11,800,50,1165)
MACO2+MACO2→2*CH3COZ+2*CO2	k(T)=$Ae^{-B/T}$,A=2.8e-12,B=-530
MVKO2+MVKO2→2*MGLYOX+2*HCHO+2*HO2	k(T)=SPECIAL_1(5.5e-11,1500,50,1165)
MVKO2+MVKO2→2*MGLYOX+2*HCHO	k(T)=SPECIAL_2(5.5e-11,1500,50,1165)
APIOH+APIOH→2*CH3CHO+2*CH3COE+2*HO2	k(T)=$Ae^{-B/T}$,A=1.8e-11,B=2200
APINO3+APINO3→2*CH3CHO+2*CH3COE+2*NO2	k(T)=$Ae^{-B/T}$,A=1.8e-11,B=2200
organic	
CH3O2+NO→CH3NO3	k(T)=$Ae^{-B/T}$,A=2.1e-14,B=-180
CH3NO3+OH→HCHO+HNO3+HO2	k(T)=$Ae^{-B/T}$,A=1.e-14,B=-1060
C2H5O2+NO→ETNO3	k=1.25e-13
ETNO3+OH→CH3CHO+HNO3+HO2	k(T)=$Ae^{-B/T}$,A=4.4e-14,B=-720
SECC4H+NO+M→BUNO3	k(T,M)=troe(3.76e-32,0,0,3.3e-12,0,8.1,0.41)
BUNO3+OH→RNC3H6	k=6.7e-13
RNC2H4+NO→0.5*CARNIT+HCHO+1.5*NO2+0.5*HO2	k=8.9e-12
RNC3H6+NO→0.5*CARNIT+0.5*CH3CHO+0.5*HCHO +1.5*NO2+0.5*HO2	k=4e-12
ROXYL1+NO→BENZAL+NO2+HO2	k=4e-12
ISNI+OH→ISNIR	k=3.4e-11
ISNI+O3→0.5*ISNI+0.5*HCHO+0.5*MGLYOX+0.5*NO2 +0.08*OH+0.2*O3	k=5e-18
ISNIR+NO→0.95*CH3CHO+0.475*CH3COE+0.475*MGLYOX +0.05*ISNI+0.05*HO2+1.9*NO2	k(T)=$Ae^{-B/T}$,A=1.4e-11,B=180
RNC5H8+NO→0.85*ISNI+0.1*MAC+0.05*MVK+0.15*HCHO +1.1*NO2+0.8*HO2	k(T)=$Ae^{-B/T}$,A=1.4e-11,B=180
APINO3+NO→CH3CHO+CH3COE+2*NO2	k=4e-12
RNC5H8+HO2→ORNIT2	k(T)=$Ae^{-B/T}$,A=2.7e-13,B=-1000
RNC2H4+HO2→ORNIT1	k(T)=$Ae^{-B/T}$,A=2.7e-13,B=-1000
RNC3H6+HO2→ORNIT1	k(T)=$Ae^{-B/T}$,A=2.7e-13,B=-1000
ROXYL1+HO2→BENZYL	k(T)=$Ae^{-B/T}$,A=2.7e-13,B=-1000
ISNIR+HO2→ORNIT2	k(T)=$Ae^{-B/T}$,A=2.7e-13,B=-1000
APINO3+HO2→ORNIT2	k(T)=$Ae^{-B/T}$,A=2.7e-13,B=-1000
CH3COO+NO2+M→PAN	k(T,M)=troe(2.7e-28,0,7.1,1.2e-11,0,0.9,0.3)
PAN+M→CH3COO+NO2	k(T,M)=troe(4.9e-3,12100,0,5.4e16,13830,0,0.3)
PAN+OH→HCHO+NO3+CO2	k(T)=$Ae^{-B/T}$,A=9.5e-13,B=650
MACO2+NO2+M→MACPAN	k(T,M)=troe(2.7e-28,0,7.1,1.2e-11,0,0.9,0.3)
MACPAN+M→MACO2+NO2	k(T,M)=troe(4.9e-3,12100,0,5.4e16,13830,0,0.3)
MACPAN+OH→CH3COE+NO3+O3+CO2	k(T)=$Ae^{-B/T}$,A=3.25e-13,B=-500
MEMAL2+NO2+M→MEMPAN	k(T,M)=troe(2.7e-28,0,7.1,1.2e-11,0,0.9,0.3)
MEMPAN+M→MEMAL2+NO2	k(T,M)=troe(4.9e-3,12100,0,5.4e16,13830,0,0.3)
MEMPAN+OH→CH3COE+NO3+CO2	k(T)=$Ae^{-B/T}$,A=3.25e-13,B=-500
CH3O2+NO2+M→CH3NO4	k(T,M)=troe(2.5e-30,0,5.5,7.5e-12,0,0,0.36)
CH3NO4+M→CH3O2+NO2	k(T,M)=troe(9e-5,9690,0,1.1e16,10560,0,0.36)
ORNIT1+OH→CARNIT+OH	k(T)=$Ae^{-B/T}(300/T)^N$,A=6.1e-14,B=-1743,N=-2
ORNIT1+OH→RNC3H6	k(T)=$Ae^{-B/T}$,A=1.9e-12,B=-190
ORNIT2+OH→0.95*CH3CHO+0.475*CH3COE+0.475*MGLYOX +0.05*ISNI+0.05*OH+0.95*NO2	k(T)=$Ae^{-B/T}(300/T)^N$,A=6.1e-14,B=-1743,N=-2
ORNIT2+OH→ISNIR	k(T)=$Ae^{-B/T}$,A=1.9e-12,B=-190

Reactions	Kinetic constants
photolytic	
O3→2*OH	J(T,Z,H2O)=photorate(O3)
NO2→NO+O3	J(Z)=photorate(NO2)
NO3→NO2+O3	J(Z)=photorate(NO3-2)
H2O2→2*OH	J(Z)=photorate(H2O2)
HNO3→NO2+OH	J(Z)=photorate(HNO3)
HCHO→CO+2*HO2	J(Z)=photorate(HCHO-1)
HCHO→CO+H2	J(Z)=photorate(HCHO-2)
CH3CHO→CH3O2+HO2+CO	J(Z)=photorate(CH3CHO)
CH3COE→CH3COO+C2H5O2	J(Z)=photorate(CH3COE)
CH3COY→2*CH3COO	J(Z)=photorate(CH3COY)
MGLYOX→CH3COO+HO2+CO	J(Z)=photorate(MGLYOX)
GLYOX→0.6*HO2+2*CO+1.4*H2	J(Z)=photorate(GLYOX)
MEMALD→0.5*OFURAN+0.5*MEMAL2+0.5*HO2	J(Z)=photorate(MEMALD)
NO3→NO	J(Z)=photorate(NO3-1)
N2O5→NO2+NO3	J(Z)=photorate(N2O5)
CH3O2H→HCHO+OH+HO2	J(Z)=photorate(CH3O2H)
ETO2H→CH3CHO+OH+HO2	J(Z)=photorate(CH3O2H)
BUO2H→CH3COE+OH+HO2	J(Z)=photorate(CH3O2H)
CH4COX→CH3COY+OH+HO2	J(Z)=photorate(CH3O2H)
CH3O2C→2*HCHO+OH+HO2	J(Z)=photorate(CH3O2H)
CH4CHX→HCHO+CH3CHO+OH+HO2	J(Z)=photorate(CH3O2H)
OXYLH1→MEMALD+MGLYOX+OH+HO2	J(Z)=photorate(CH3O2H)
MEMALH→GLYOX+MGLYOX+OH+HO2	J(Z)=photorate(CH3O2H)
MEMAH2→MALEIC+OH+CH3O2	J(Z)=photorate(CH3O2H)
PEROX→0.14*RO2IP1+0.488*CH3COE+0.372*MAC+0.86*HCHO +0.9*HO2+OH	J(Z)=photorate(CH3O2H)
MACO2H→CH3COZ+OH+CO2	J(Z)=photorate(CH3O2H)
MVKO2H→0.28*HCHO+0.28*MGLYOX+0.28*HO2+0.72*CH3CHO +0.72*CH3COO+OH	J(Z)=photorate(CH3O2H)
CH4COZ→CH3COE+OH+HO2	J(Z)=photorate(CH3O2H)
APIO2H→CH3CHO+CH3COE+OH+HO2	J(Z)=photorate(CH3O2H)
PPA→CH3O2+CO2+OH	J(Z)=photorate(PPA)
HNO4→HO2+NO2	J(Z)=photorate(HNO4)
HONO→NO+OH	J(Z)=photorate(HONO)
CH3NO4→CH3O2+NO2	J(Z)=photorate(HNO4)
CH3NO3→HCHO+HO2+NO2	J(Z)=photorate(CH3NO3)
ETNO3→CH3CHO+HO2+NO2	J(Z)=photorate(ETNO3)
BUNO3→CH3COE+HO2+NO2	J(Z)=photorate(BUNO3)
PAN→CH3COO+NO2	J(Z)=photorate(PAN)
CARNIT→CH3COO+HCHO+NO2	J(Z)=photorate(CH3CHO)
ORNIT1→CH3CHO+HCHO+NO2+OH	J(Z)=photorate(CH3O2H)
ORNIT2→0.95*CH3CHO+0.475*CH3COE+0.475*MGLYOX+0.05*ISNI +0.05*HO2+0.95*NO2+OH	J(Z)=photorate(CH3O2H)
APINEN+NO3→APINO3	k(T)=A$e^{-B/T}$,A=1.19e-12,B=-490
BPINEN+NO3→APINO3	k(T)=A$e^{-B/T}$,A=1.19e-12,B=-490
LIMONE+NO3→APINO3	k(T)=A$e^{-B/T}$,A=1.19e-12,B=-490
OCIMEN+NO3→APINO3	k(T)=A$e^{-B/T}$,A=1.19e-12,B=-490
TERPEN+NO3→APINO3	k(T)=A$e^{-B/T}$,A=1.19e-12,B=-490
HUMULE+NO3→APINO3	k(T)=A$e^{-B/T}$,A=1.19e-12,B=-490
APINEN+OH→APIOH	k(T)=A$e^{-B/T}$,A=1.21e-11,B=-444
BPINEN+OH→APIOH	k(T)=A$e^{-B/T}$,A=1.21e-11,B=-444
LIMONE+OH→APIOH	k(T)=A$e^{-B/T}$,A=1.21e-11,B=-444
OCIMEN+OH→APIOH	k(T)=A$e^{-B/T}$,A=1.21e-11,B=-444
TERPEN+OH→APIOH	k(T)=A$e^{-B/T}$,A=1.21e-11,B=-444
HUMULE+OH→APIOH	k(T)=A$e^{-B/T}$,A=1.21e-11,B=-444
APINEN+O3→0.65*CH3CHO+0.53*CH3COE+0.14*CO+0.2*C2H5O2 +0.42*CH3CHX+0.85*OH+0.1*HO2	k(T)=A$e^{-B/T}$,A=1.0e-15,B=736
BPINEN+O3→0.65*CH3CHO+0.53*CH3COE+0.14*CO+0.2*C2H5O2 +0.42*CH3CHX+0.85*OH+0.1*HO2	k(T)=A$e^{-B/T}$,A=1.0e-15,B=736
LIMONE+O3→0.65*CH3CHO+0.53*CH3COE+0.14*CO+0.2*C2H5O2 +0.42*CH3CHX+0.85*OH+0.1*HO2	k(T)=A$e^{-B/T}$,A=1.0e-15,B=736
TERPEN+O3→0.65*CH3CHO+0.53*CH3COE+0.14*CO+0.2*C2H5O2 +0.42*CH3CHX+0.85*OH+0.1*HO2	k(T)=A$e^{-B/T}$,A=1.0e-15,B=736
OCIMEN+O3→0.65*CH3CHO+0.53*CH3COE+0.14*CO+0.2*C2H5O2 +0.42*CH3CHX+0.85*OH+0.1*HO2	k(T)=A$e^{-B/T}$,A=1.0e-15,B=736

Reactions	Kinetic constants
HUMULE+O3→0.65*CH3CHO+0.53*CH3COE+0.14*CO+0.2*C2H5O2 +0.42*CH3CHX+0.85*OH+0.1*HO2	k(T)=Ae$^{-B/T}$,A=1.0e-15,B=736
C5H8+OH→RO2IP1	k(T)=Ae$^{-B/T}$,A=2.55e-11,B=-410
NC4H10+OH→SECC4H	k(T)=Ae$^{-B/T}$(300/T)^{N},A=1.36e-12,B=-190,N=-2
OXYL+OH→OXYL1	k=1.37e-11

- The reactions O_3+hν→O(1D), O(1D)+H_2O→2OH, O(1D)+M→O(3P)+M and O(3P)+O_2+M→O_3+M have been combined to reaction O_3+hν→ O(1D) by taking into account H_2O and M concentrations in order to adjust the photolysis frequency.
- Due to lack of data, photolysis frequencies of higher organic peroxides are taken as that of CH_3OOH.
- Three body Troe reactions [1], given in the form :

$$k = \frac{k_0[M]}{1+\frac{k_0[M]}{k_\infty}} f^Z$$

with

$$Z = \frac{1}{1+\left(log_{10}\left(\frac{k_0(T)[M]}{k_\infty(T)}\right)\right)^2}$$

$$k_0 = A_0 e^{\left(-\frac{B_0}{T}\right)}\left(\frac{T}{300}\right)^{-n}, \text{ and } k_\infty = A_\infty e^{\left(-\frac{B_\infty}{T}\right)}\left(\frac{T}{300}\right)^{-m},$$

The parameters are given in the order $A_0, B_0, n, A_\infty, B_\infty, m, f$

- The reaction NO2+OH :

$$k = \frac{k_0[M]}{1+\frac{k_0[M]}{k_\infty}} f^Z$$

with

$$Z = \frac{1}{1+\left(\frac{log_{10}\left(\frac{k_0(T)[M]}{k_\infty(T)}-0.12\right)}{1.2}\right)^2}$$

$$k_0 = A_0 e^{\left(-\frac{B_0}{T}\right)}\left(\frac{T}{300}\right)^{-n}, \text{ and } k_\infty = A_\infty e^{\left(-\frac{B_\infty}{T}\right)}\left(\frac{T}{300}\right)^{-m},$$

The parameters are given in the order $A_0, B_0, n, A_\infty, B_\infty, m, f$

- Radical recombinations SPECIAL_1 :

$$k(T) = k_1(T) \times \frac{k_2(T)}{1+k_2(T)}$$

The parameters are given in the order A_1, B_1, A_2, B_2

- Radical recombinations SPECIAL_2 :

$$k(T) = \frac{k_1(T)}{1+k_2(T)}$$

The parameters are given in the order A_1, B_1, A_2, B_2

- Radical recombinations SPECIAL_3 :

$$k(T) = 2 \times \sqrt{\frac{k_1(T) \times k_2(T) \times k_3(T) \times k_4(T)}{(1+k_3(T)) \times (1+k_4(T))}}$$

The parameters are given in the order $A_1, B_1, A_2, B_2, A_3, B_3, A_4, B_4$

- Radical recombinations SPECIAL_4 :

$$k(T) = 2 \times \sqrt{k_1(T) \times k_2(T)} \times \frac{1-\sqrt{\frac{k_3(T)}{1+k_3(T)} \times \frac{k_4(T)}{1+k_4(T)}} \times \left(1-\frac{k_4(T)}{1+k_4(T)}\right)}{2-\frac{k_3(T)}{1+k_3(T)}-\frac{k_4(T)}{1+k_4(T)}}$$

The parameters are given in the order $A_1, B_1, A_2, B_2, A_3, B_3, A_4, B_4$

G.2 Reaction list of the reduced MELCHIOR2 chemical mechanism

Reactions	Kinetic constants	Ref.
Inorganic chemistry		
O3+NO→NO2	$Ae^{-B/T}$,A=1.8e-12,B=1370	[1]
O3+NO2→NO3	$Ae^{-B/T}$,A=1.2e-13,B=2450	[1]
O3+OH→HO2	$Ae^{-B/T}$,A=1.9e-12,B=1000	[1]
O3+HO2→OH	$Ae^{-B/T}$,A=1.4e-14,B=600	[1]
NO+HO2→OH+NO2	$Ae^{-B/T}$,A=3.7e-12,B=-240	[1]
NO2+OH+M→HNO3	troe(3.4e-30,0,3.2,4.77e-11,0,1.4,0.30)	[2], n9
HO2+OH→H2O	$Ae^{-B/T}$,A=4.8e-11,B=-250	[1]
H2O2+OH→HO2	$Ae^{-B/T}$,A=2.9e-12,B=160	[1]
HNO3+OH→NO3	$Ae^{-B/T}$,A=5.5e-15,B=-985	[3], n1
CO+OH→HO2+CO2	$Ae^{-B/T}(300/T)^{N}$, A=2e-13, B=0, N=1	[3], n1
HO2+HO2→H2O2	$Ae^{-B/T}$,A=2.2e-13,B=-740	[1], n1
HO2+HO2+H2O→H2O2	$Ae^{-B/T}$,A=4.52e-34,B=-2827	[1], n1
NO3+HO2→NO2+OH	4e-12	[1]
NO3+H2O2→HNO3+HO2	2e-15	[3]
NO3+NO→2*NO2	$Ae^{-B/T}$,A=1.8e-11,B=-110	[1]
NO2+NO3→NO+NO2	$Ae^{-B/T}$,A=4.5e-14,B=1260	[3]
NO2+NO3+M→N2O5	troe(2.7e-30,0,3.4,2e-12,0,-0.2,0.33)	[1], n9
N2O5+M→NO3+NO2	troe(1e-3,11000,3.5,9.7e14,11080,-0.1,0.33)	[1], n9
N2O5+H2O→2*HNO3	2.6e-22	[4]
N2O5+H2O+H2O→2*HNO3	2e-39	[4]
NO+OH+M→HONO	troe(7.e-31,0,2.6,1.5e-11,0,0.5,0.6)	[1], n9
HONO+OH→NO2	$Ae^{-B/T}$,A=1.8e-11,B=390	[1]
NO+NO+O2→2*NO2	$Ae^{-B/T}$,A=3.30E-39,B=-530.0	[1]
NO2→HONO	0.5*depo(NO2)	[18]
SOx chemistry		
SO2+CH3O2→H2SO4+HCHO+HO2	4e-17	[3]
SO2+OH+M→H2SO4+HO2	troe(4e-31,0,3.3,2e-12,0,0,0.45)	[3], n9
OH attack to hydrocarbons		
CH4+OH→CH3O2	$Ae^{-B/T}$,A=2.3e-12,B=1765	[1]
C2H6+OH→CH3CHO+oRO2	$Ae^{-B/T}$,A=7.9e-12,B=1030	[1]
NC4H10+OH→0.9*CH3COE+0.1*CH3CHO+0.1*CH3COO+0.9*oRO2	$Ae^{-B/T}(300/T)^{N}$,A=1.36e-12, B=-190, N=-2	[5]
C2H4+OH+M→2*HCHO+oRO2	troe(7e-29,0,3.1,9e-12,0,0,0.7)	[1], n9
C3H6+OH+M→HCHO+CH3CHO+oRO2	troe(8e-27,0,3.5,3e-11,0,0,0.5)	[1], n9
OXYL+OH→ MEMALD+MGLYOX+oRO2	1.37e-11	[5]
C5H8+OH→0.32*MAC +0.42*MVK +0.74*HCHO +obio	$Ae^{-B/T}$, A=2.55e-11, B=-410	[1]
APINEN+OH→0.8*CH3CHO+0.8*CH3COE+obio	$Ae^{-B/T}$, A=1.21e-11, B=-444	[17]
BPINEN+OH→0.8*CH3CHO+0.8*CH3COE+obio	$Ae^{-B/T}$, A=1.21e-11, B=-444	[17]
LIMONE+OH→0.8*CH3CHO+0.8*CH3COE+obio	$Ae^{-B/T}$, A=1.21e-11, B=-444	[17]
HUMULE+OH→0.8*CH3CHO+0.8*CH3COE+obio	$Ae^{-B/T}$, A=1.21e-11, B=-444	[17]
OCIMEN+OH→0.8*CH3CHO+0.8*CH3COE+obio	$Ae^{-B/T}$, A=1.21e-11, B=-444	[17]
TERPEN+OH→0.8*CH3CHO+0.8*CH3COE+obio	$Ae^{-B/T}$, A=1.21e-11, B=-444	[17]
OH attack to carbonyls and peroxides		
HCHO+OH→CO+HO2	$Ae^{-B/T}$,A=8.6e-12,B=-20	[1]
CH3CHO+OH →CH3COO	$Ae^{-B/T}$,A=5.6e-12,B=-310	[1]
MEMALD+OH →GLYOX+MGLYOX+oRO2	5.6e-11	[1]
CH3COE+OH →CH3COY+oRO2	$Ae^{-B/T}(300/T)^{N}$, A=2.92e-13, B=-414, N=-2	[1]
GLYOX+OH→2*CO+HO2	1.1e-11	[1]
MGLYOX+OH→CH3COO+CO	1.5e-11	[1]
MVK+OH→0.266*MGLYOX +0.266*HCHO+0.684*CH3CHO +0.684*CH3COO+0.05*ISNI+0.95*oRO2	$Ae^{-B/T}$,A=4.1e-12,B=-453	[6]
MAC+OH→0.5*CH3COE+0.5*CO2+0.5*oPAN	$Ae^{-B/T}$,A=1.86e-11,B=-175	[6]
CH3O2H+OH→CH3O2	$Ae^{-B/T}$,A=1.9e-12,B=-190	[1]

Reactions	Kinetic constants	Ref.
PPA+OH→CH3COO	$Ae^{-B/T}$,A=1.9e-12,B=-190	n2
CH3O2H+OH→HCHO+OH	$Ae^{-B/T}$,A=1.e-12,B=-190	[1]
oROOH+OH→0.8*OH+0.2*oRO2	$Ae^{-B/T}$,A=4.35e-12,B=-455	n2
obioH+OH→OH	8e-11	n2
PANH+OH→0.2*oPAN	1.64e-11	n2
CARNIT+OH→CH3CHO+CO+NO2	k(T)=$Ae^{-B/T}$,A=5.6e-12,B=-310	
NO3 attack to hydrocarbons and aldehydes		
C2H4+NO3→0.5*CARNIT+HCHO+oRN1	2e-16 [1]	
C3H6+NO3→0.5*CARNIT+1.5*HCHO+0.5*CH3CHO +0.5*HO2+oRN1	9.45e-15	[1]
APINEN+NO3→CH3CHO+CH3COE+oRN1	$Ae^{-B/T}$,A=1.19e-12,B=490	[8]
BPINEN+NO3→CH3CHO+CH3COE+oRN1	$Ae^{-B/T}$,A=1.19e-12,B=490	[8]
LIMONE+NO3→CH3CHO+CH3COE+oRN1	$Ae^{-B/T}$,A=1.19e-12,B=490	[8]
OCIMEN+NO3→CH3CHO+CH3COE+oRN1	$Ae^{-B/T}$,A=1.19e-12,B=490	[8]
HUMULE+NO3→CH3CHO+CH3COE+oRN1	$Ae^{-B/T}$,A=1.19e-12,B=490	[8]
TERPEN+NO3→CH3CHO+CH3COE+oRN1	$Ae^{-B/T}$,A=1.19e-12,B=490	[8]
C5H8+NO3→0.85*ISNI+0.1*MAC+0.05*MVK+0.15*HCHO +0.8*HO2+oRN1	7.8e-13	[6]
HCHO+NO3→CO+HNO3+HO2	5.8e-16	[1]
CH3CHO+NO3→CH3COO+HNO3	2.8e-15	[9]
CH3O2+NO3→HCHO+HO2+NO2	1.2e-12	[10]
CH3COO+NO3→CH3O2+NO2+CO2	4e-12	[10]
oRO2+NO3→NO2+HO2	1.2e-12	[10]
obio+NO3→NO2+HO2	1.2e-12	[10]
oPAN+NO3→NO2+HO2	4e-12	[10]
oRN1+NO3→1.5*NO2	1.2e-12	[10]
O3 attack to unsaturated hydrocarbons		
C2H4+O3→HCHO+0.12*HO2+0.13*H2+0.44*CO	$Ae^{-B/T}$,A=9.1e-15,B=2580	[1]
C3H6+O3→0.53*HCHO+0.5*CH3CHO+0.31*CH3O2 +0.28*HO2+0.15*OH+0.065*H2+0.4*CO+0.7*CH4	$Ae^{-B/T}$,A=5.5e-15,B=1880	[1]
C5H8+O3→0.67*MAC+0.26*MVK+0.55*OH+0.07*C3H6 +0.8*HCHO +0.06*HO2+0.05*CO+0.3*O3	$Ae^{-B/T}$,A=1.2e-14,B=2013	[6]
MAC+O3→0.8*MGLYOX+0.7*HCHO+0.215*OH+0.275*HO2 +0.2*CO+0.2*O3	$Ae^{-B/T}$,A=5.3e-15,B=2520	[6]
MVK+O3→0.82*MGLYOX+0.8*HCHO+0.04*CH3CHO +0.08*OH+0.06*HO2+0.05*CO+0.2*O3	$Ae^{-B/T}$,A=4.3e-15,B=2016	[6]
APINEN+O3→1.27*CH3CHO+0.53*CH3COE+0.14*CO +0.62*oRO2+0.42*HCHO+0.85*OH+0.1*HO2	$Ae^{-B/T}$,A=1.0e-15,B=736	[8]
BPINEN+O3→1.27*CH3CHO+0.53*CH3COE+0.14*CO +0.62*oRO2+0.42*HCHO+0.85*OH+0.1*HO2	$Ae^{-B/T}$,A=1.0e-15,B=736	[8]
LIMONE+O3→1.27*CH3CHO+0.53*CH3COE+0.14*CO +0.62*oRO2+0.42*HCHO+0.85*OH+0.1*HO2	$Ae^{-B/T}$,A=1.0e-15,B=736	[8]
TERPEN+O3→1.27*CH3CHO+0.53*CH3COE+0.14*CO +0.62*oRO2+0.42*HCHO+0.85*OH+0.1*HO2	$Ae^{-B/T}$,A=1.0e-15,B=736	[8]
OCIMEN+O3→1.27*CH3CHO+0.53*CH3COE+0.14*CO +0.62*oRO2+0.42*HCHO+0.85*OH+0.1*HO2	$Ae^{-B/T}$,A=1.0e-15,B=736	[8]
HUMULE+O3→1.27*CH3CHO+0.53*CH3COE+0.14*CO +0.62*oRO2+0.42*HCHO+0.85*OH+0.1*HO2	$Ae^{-B/T}$,A=1.0e-15,B=736	[8]
organic radical conversion		
CH3O2+NO→HCHO+NO2+HO2	$Ae^{-B/T}$,A=4.2e-12,B=-180	[1]
CH3COO+NO→CH3O2+NO2+CO2	2e-11	[1]
oRO2+NO→NO2+HO2	4e-12	n3
obio+NO→0.86*NO2+0.78*HO2+0.14*ISNI	$Ae^{-B/T}$,A=1.4e-11,B=180	[6]
oPAN+NO→NO2+HO2	1.4e-11	n3
oRN1+NO→1.5*NO2	4e-11	n3
organic radical recombination		
CH3O2+HO2→CH3O2H	$Ae^{-B/T}$,A=4.1e-13,B=-790	[1]
CH3COO+HO2→0.67*PPA+0.33*O3	$Ae^{-B/T}$,A=4.3e-13,B=-1040	[1]

Reactions	Kinetic constants	Ref.
oRO2+HO2→ oROOH	$Ae^{-B/T}$,A=2.7e-13,B=-1000	n4
obio+HO2→obioH	$Ae^{-B/T}$,A=2.7e-13,B=-1000	n4
oPAN+HO2→PANH	$Ae^{-B/T}$,A=2.7e-13,B=-1000	n4
oRN1+HO2→X	$Ae^{-B/T}$,A=2.7e-13,B=-1000	n4
CH3O2+CH3O2→1.35*HCHO+0.7*HO2	$Ae^{-B/T}$,A=1.13e-13,B=-356	[1], n5
CH3COO+CH3O2→0.5*CH3O2+0.5*CO2+HCHO+0.5*HO2	$Ae^{-B/T}$,A=3.34e-12,B=-400	[1], n5
oRO2+CH3O2→0.65*HCHO+0.8*HO2+0.35*CH3OH	$Ae^{-B/T}$,A=1.5e-13,B=-220	n5
obio+CH3O2→0.8*HO2+0.5*HCHO	$Ae^{-B/T}$,A=2.44e-11,B=223	n5
oPAN+CH3O2→HCHO+0.5*HO2	$Ae^{-B/T}$,A=7.9e-12,B=-140	n5
CH3COO+CH3COO→2.*CH3O2+2.*CO2	$Ae^{-B/T}$,A=2.8e-12,B=-530	[1], n5
oRO2+CH3COO→0.8*CH3O2+0.8*CO2+0.8*HO2	$Ae^{-B/T}$,A=8.6e-13,B=-260	n5
obio+CH3COO→0.5*HCHO+1.5*HO2+0.7*CO2	$Ae^{-B/T}$,A=1.18e-11,B=127	n5
oPAN+CH3COO→CH3O2+CO2+HO2	$Ae^{-B/T}$,A=3.34e-12,B=-400	n5
oRO2+oRO2→1.3*HO2	6.4e-14	n5
organic nitrates and pernitrates		
CH3COO+NO2+M→PAN	troe(2.7e-28,0,7.1,1.2e-11,0,0.9,0.3)	[1], n9
oPAN+NO2+M→toPAN	troe(2.7e-28,0,7.1,1.2e-11,0,0.9,0.3)	n6 , n9
PAN+M→CH3COO+NO2	troe(4.9e-3,12100,0,5.4e16,13830,0,0.3)	[1], n9
toPAN+M→oPAN+NO2	troe(4.9e-3,12100,0,5.4e16,13830,0,0.3)	n6 , n9
PAN+OH→HCHO+NO3+CO2	$Ae^{-B/T}$,A=9.5e-13,B=650	[1]
toPAN+OH→NO3+CO2	$Ae^{-B/T}$,A=3.25e-13,B=-500	n6
ISNI+OH→0.95*CH3CHO+0.475*CH3COE+0.475*MGLYOX +0.05*ISNI+0.05*HO2+oRN1	3.4e-11	[6]
photolytic reactions		
O3→2*OH	photorate	[3], n7
NO2→NO+O3	photorate	[3]
NO3→NO2+O3	photorate	[3], [11]
NO3→NO	photorate	[3], [11]
N2O5→NO2+NO3	photorate	[3]
H2O2→2*OH	photorate	[3]
HNO3→NO2+OH	photorate	[3]
HONO→NO+OH	photorate	[3]
HCHO→CO+2*HO2	photorate	[3]
HCHO→CO+H2	photorate	[3]
CH3CHO→CH3O2+HO2+CO	photorate	[1]
CH3COE→CH3COO+CH3CHO+oRO2	photorate	[11], [12]
CH3COY→2*CH3COO	photorate	[13], [14]
MGLYOX→CH3COO+HO2+CO	photorate	[1]
GLYOX→0.6*HO2+2*CO+0.7*H2	photorate	[13], [14]
MEMALD→0.5*MVK+0.5*MALEIC+0.5*oPAN+0.5*HCHO +0.5*HO2	photorate	[15]
CH3O2H→HCHO+OH+HO2	photorate	[3]
PPA→CH3O2+CO2+OH	photorate	n8
oROOH→OH+HO2	photorate	n8
obioH→OH+HO2	photorate	n8
PANH→OH+HO2	photorate	n8
PAN→CH3COO+NO2	photorate	[3]

- [1] Atkinson et al., 1997
- [2] Donahue et al., 1997
- [3] DeMore et al., 1997
- [4] Mentel et al., 1996
- [5] Atkinson, 1990
- [6] Paulson and Seinfeld, 1992
- [7] Le Bras, 1997
- [8] Atkinson et al., 1994
- [9] DeMore et al., 1990
- [10] Canosa-Mas et al., 1996
- [11] Johnston et al., 1996
- [12] Martinez et al., 1992
- [13] Jenkins et al., 1997
- [14] Plum et al., 1983
- [15] Bierbach et al., 1994
- [16] Lightfoot et al., 1996
- [17] Kirchner and Stockwell, 1996
- [18] Aumont, 2002

- n1 : Rate constants containing a density dependent term in a minor reaction channel have been simplified by putting an average boundary layer molecular density of 2.5*10^{19} molecules/cm^3.
- n2 : For operator reactions with OH, oROOH is interpreted as 2-butyl hydrogen peroxide, obioH as a peroxide from isoprene degradation, PANH represents unsaturated organic peroxides; due to lack of data, the rate constants of these compounds and of peroxy acetyl acid have been estimated using structure reactivity relationships in Kwok and Atkinson (1995).
- n3 : The rate constant of the operator reaction oRO2+NO has been set to that of the i-C3H7O2+NO reaction given in [16], that of oPAN+NO to that of CH3COO2+NO [1], the rate constant for oRO2NO+NO is taken from that of ISNIR (nitrate peroxy radicals from isoprene degradation) + NO [6].
- n4 : The rate constants for the operator reactions oRO2, obio, RO2NI+HO2 have been set to that of the reaction C2H5O2+HO2 [1], that of oPAN+HO2 has been taken from CH3COO2+HO2 [1].
- n5 : Radical conserving and terminating pathways are combined to single reactions; rate constants in operator reactions are affected by interpreting oRO2 as 2-butyl peroxy radical [16], obio as a peroxy radical from isoprene degradation [6] and oPAN as CH3COO2.
- n6 : Reaction rates of the operators oPAN and OPAN have been set to those of the peroxy acetyl radical and PAN, respectively.
- n7 : The reactions O3+hν→O(1D), O(1D)+H2O→2OH, O(1D)+M→O(3P)+M and O(3P)+O2+M→O3+M have been combined to reaction O3+hν→O(1D) by taking into account H2O and M concentrations in order to adjust the photolysis frequency.
- n8 : Due to lack of data, photolysis frequencies of higher organic peroxides are taken as that of CH3OOH.
- n9 : Three body Troe reactions [1], given in the form:

$$k = \frac{k_0[M]}{1 + \frac{k_0[M]}{k_\infty}} f^Z$$

with

$$Z = \frac{1}{1 + \left(log_{10}\left(\frac{k_0(T)[M]}{k_\infty(T)}\right)\right)^2}$$

$$k_0 = A_0 e^{\left(-\frac{B_0}{T}\right)} \left(\frac{T}{300}\right)^{-n}, \text{ and } k_\infty = A_\infty e^{\left(-\frac{B_\infty}{T}\right)} \left(\frac{T}{300}\right)^{-m},$$

The parameters are given in the order $A_0, B_0, n, A_\infty, B_\infty, m, f$

G.3 Reaction list of the SAPRC-07-A chemical mechanism

Reactions	Kinetic constants
Inorganic chemistry	
O3+NO→NO2	k(T)=A$e^{-B/T}$,A=3.00e-12,B=1500
O3+NO2→NO3	k(T)=A$e^{-B/T}$,A=1.40e-13,B=2470
NO+NO3→2*NO2	k(T)=A$e^{-B/T}$,A=1.80e-11,B=-110
NO+NO+O2→2*NO2	k(T)=A$e^{-B/T}$,A=3.30e-39,B=-530

Reactions	Kinetic constants
NO2+NO3+M→N2O5	k(T,M)=saprc_troe(3.60e-30,-4.10,0.,1.90e-12,0.20,0.,0.35,1.33)
N2O5+M→NO2+NO3	k(T,M)=saprc_troe(1.30e-03,-3.50,11000,9.70e+14,0.10,11080,0.35,1.33)
N2O5+H2O→2*HNO3	k=2.50e-22
N2O5+H2O+H2O→2*HNO3	k=1.80e-39
NO2+NO3→NO+NO2	k(T)=Ae$^{-B/T}$,A=4.50e-14,B=1260
OH+NO+M→HONO	k(T,M)=saprc_troe(7.00e-31,-2.60,0.,3.60e-11,-0.10,0.,0.60,1.0)
OH+HONO→NO2	k(T)=Ae$^{-B/T}$,A=2.50e-12,B=-260
OH+NO2+M→HNO3	k(T,M)=saprc_troe(1.80e-30,-3.00,0.,2.80e-11,0.,0.,0.60,1.0)
OH+NO3→HO2+NO2	k=2.00e-11
OH+HNO3→NO3	k(T)=saprc_2(2.40e-14,-460,2.70e-17,-2199,6.50e-34,-1335)
OH+CO→HO2+CO2	k(T)=saprc_1(1.44e-13,0.,3.43e-33,0.)
OH+O3→HO2	k(T)=Ae$^{-B/T}$,A=1.70e-12,B=940
HO2+NO→OH+NO2	k(T)=Ae$^{-B/T}$,A=3.60e-12,B=-270
HO2+NO2+M→HNO4	k(T,M)=saprc_troe(2.00e-31,-3.40,0.,2.90e-12,-1.10,0.,0.60,1.0)
HNO4+M→HO2+NO2	k(T,M)=saprc_troe(3.72e-05,-2.40,10650,5.42e+15,-2.30,11170,0.60,1.0)
HNO4+OH→NO2	k(T)=Ae$^{-B/T}$,A=1.30e-12,B=-380
HO2+O3→OH	k(T)=Ae$^{-B/T}$(300/T)^{N},A=2.03e-16,B=-693,N=-4.57
HO2+HO2→H2O2	k(T)=saprc_1(2.20e-13,-600,1.90e-33,-980)
HO2+HO2+H2O→H2O2	k(T)=saprc_1(3.08e-34,-2800,2.66e-54,-3180)
NO3+HO2→0.8*OH+0.8*NO2+0.2*HNO3	k=4.00e-12
NO3+NO3→2*NO2	k(T)=Ae$^{-B/T}$,A=8.50e-13,B=2450
H2O2+OH→HO2	k=1.80e-12
OH+HO2→H2O	k(T)=Ae$^{-B/T}$,A=4.80e-11,B=-250
OH+H2→HO2	k(T)=Ae$^{-B/T}$,A=7.70e-12,B=2100
SOx chemistry	
OH+SO2+M→HO2+H2SO4	k(T,M)=saprc_troe(3.30e-31,-4.30,0.,1.60e-12,0.,0,0.60,1.0)
RO_2X operator	
RO2R+NO→NO2+HO2	k(T)=Ae$^{-B/T}$,A=2.60e-12,B=-380
RO2R+HO2→XOOH	k(T)=Ae$^{-B/T}$,A=3.80e-13,B=-900
RO2R+NO3→NO2+HO2	k=2.30e-12
RO2R+MEO2→HO2+0.75*HCHO+0.25*MEOH	k=2e-13
RO2R+RO2R→HO2	k=3.50e-14
RO2N+NO→RNO3	k(T)=Ae$^{-B/T}$,A=2.60e-12,B=-380
RO2N+HO2→XOOH+PRD2	k(T)=Ae$^{-B/T}$,A=3.80e-13,B=-900
RO2N+NO3→NO2+HO2+PRD2	k=2.30e-12
RO2N+MEO2→HO2+0.25*MEOH+PRD2+0.75*HCHO	k=2e-13
RO2N+RO2R→HO2+PRD2	k=3.50e-14
RO2N+RO2N→HO2+2*PRD2	k=3.50e-14
MECO3+RO2R→MEO2+CO2	k(T)=Ae$^{-B/T}$,A=4.40e-13,B=-1070
MECO3+RO2N→MEO2+CO2+HO2+PRD2	k(T)=Ae$^{-B/T}$,A=4.40e-13,B=-1070
MACO3+RO2R→CO2+HCHO+MECO3	k(T)=Ae$^{-B/T}$,A=4.40e-13,B=-1070
MACO3+RO2N→CO2+HCHO+MECO3+HO2+PRD2	k(T)=Ae$^{-B/T}$,A=4.40e-13,B=-1070
R2O2+RO2R→RO2R	k=3.50e-14
R2O2+RO2N→RO2N	k=3.50e-14
R2O2+R2O2→X	k=3.50e-14
R2O2+MECO3→MECO3	k(T)=Ae$^{-B/T}$,A=4.40e-13,B=-1070
R2O2+RCO3→RCO3	k(T)=Ae$^{-B/T}$,A=4.40e-13,B=-1070
R2O2+BZCO3→BZCO3	k(T)=Ae$^{-B/T}$,A=4.40e-13,B=-1070
R2O2+MACO3→MACO3	k(T)=Ae$^{-B/T}$,A=4.40e-13,B=-1070
RCO3+RO2R→CO2+RO2R+CCHO	k(T)=Ae$^{-B/T}$,A=4.40e-13,B=-1070
RCO3+RO2N→RO2R+CCHO+CO2+HO2+PRD2	k(T)=Ae$^{-B/T}$,A=4.40e-13,B=-1070
BZCO3+RO2R→CO2+BZO+R2O2	k(T)=Ae$^{-B/T}$,A=4.40e-13,B=-1070
BZCO3+RO2N→CO2+BZO+R2O2+HO2+PRD2	k(T)=Ae$^{-B/T}$,A=4.40e-13,B=-1070
OH attack to hydrocarbons	
AFG1+OH→0.521*RO2R+0.201*MECO3+0.217*MACO3 +0.06*RO2N+0.202*R2O2+0.334*CO+0.407*RCHO+0.129*MEK +0.107*GLY+0.267*MGLY+0.284*XC	k=7.40e-11
AFG2+OH→0.521*RO2R+0.201*MECO3+0.217*MACO3 +0.06*RO2N+0.202*R2O2+0.334*CO+0.407*RCHO+0.129*MEK +0.107*GLY+0.267*MGLY+0.284*XC	k=7.40e-11
AFG3+OH→0.561*RO2R+0.117*MECO3+0.206*MACO3 +0.117*RO2N+0.172*R2O2+0.114*CO+0.274*GLY+0.153*MGLY +0.019*BACL+0.231*IPRD+0.195*AFG1+0.195*AFG2+0.938*XC	k=9.35e-11

Reactions	Kinetic constants
CH4+OH→MEO2	k(T)=A$e^{-B/T}$,A=1.85e-12,B=1690
C2H4+OH+M→RO2R+1.61*HCHO+0.195*CCHO	k(T,M)=saprc_troe(1.00e-28,-4.50,0.,8.80e-12,-0.85,0.,0.60,1.0)
C5H8+OH→0.907*RO2R+0.093*RO2N+0.079*R2O2+0.624*HCHO +0.23*MACR+0.32*MVK+0.357*IPRD+-0.167*XC	k(T)=A$e^{-B/T}$,A=2.54e-11,B=-410
ACYE+OH+M→0.3*HO2+0.7*OH+0.3*CO+0.3*FACD+0.7*GLY	k(T,M)=saprc_troe(5.50e-30,-2.00,0.,8.30e-13,0.,0.,0.60,1.0)
BENZ+OH→0.57*HO2+0.29*RO2R+0.116*OH+0.024*RO2N +0.29*GLY+0.57*CRES+0.029*AFG1+0.261*AFG2+0.116*AFG3+-0.976*XC	k(T)=A$e^{-B/T}$,A=2.33e-12,B=193
ALK1+OH→RO2R+CCHO	k(T)=A$e^{-B/T}$(300/T)^N,A=1.34e-12,B=499,N=-2
ALK2+OH→0.965*RO2R+0.035*RO2N+0.261*RCHO+0.704*ACET+- 0.105*XC	k(T)=A$e^{-B/T}$(300/T)^N,A=1.49e-12,B=87,N=-2
ALK3+OH→0.695*RO2R+0.236*TBUO+0.07*RO2N+0.558*R2O2 +0.026*HCHO+0.445*CCHO+0.122*RCHO+0.024*ACET+0.332*MEK+- 0.046*XC	k(T)=A$e^{-B/T}$,A=1.51e-12,B=-126
ALK4+OH→0.83*RO2R+0.01*MEO2+0.011*MECO3+0.149*RO2N +0.933*R2O2+0.002*CO+0.029*HCHO+0.438*CCHO+0.236*RCHO +0.426*ACET+0.106*MEK+0.146*PRD2+-0.119*XC	k(T)=A$e^{-B/T}$,A=3.75e-12,B=-44
ALK5+OH→0.647*RO2R+0.353*RO2N+0.958*R2O2+0.04*HCHO +0.106*CCHO+0.209*RCHO+0.071*ACET+0.086*MEK +0.407*PRD2+2.004*XC	k(T)=A$e^{-B/T}$,A=2.70e-12,B=-374
OLE1+OH→0.904*RO2R+0.001*MEO2+0.095*RO2N+0.234*R2O2 +0.7*HCHO+0.301*CCHO+0.47*RCHO+0.005*ACET+0.119*PRD2 +0.026*MACR+0.008*MVK+0.006*IPRD+0.822*XC	k(T)=A$e^{-B/T}$,A=6.18e-12,B=-501
OLE2+OH→0.914*RO2R+0.086*RO2N+0.052*R2O2+0.209*HCHO +0.788*CCHO+0.481*RCHO+0.136*ACET+0.076*MEK+0.022*PRD2 +0.027*MACR+0.002*MVK+0.037*IPRD+0.111*XC	k(T)=A$e^{-B/T}$,A=1.26e-11,B=-488
ARO1+OH→0.166*HO2+0.482*RO2R+0.284*OH+0.068*RO2N +0.077*PRD2+0.218*GLY+0.138*MGLY+0.166*CRES+0.049*BALD +0.164*AFG1+0.193*AFG2+0.284*AFG3+0.002*XC	k=6.15e-12
ARO2+OH→0.108*HO2+0.58*RO2R+0.202*OH+0.11*RO2N +0.035*PRD2+0.116*GLY+0.286*MGLY+0.104*BACL+0.108*CRES +0.039*BALD+0.217*AFG1+0.21*AFG2+0.282*AFG3+1.486*XC	k=2.57e-11
OH attack to carbonyls and peroxides	
GLY+OH→0.63*HO2+0.37*RCO3+1.26*CO+-0.37*XC	k=1.10e-11
MGLY+OH→MECO3+CO	k=1.50e-11
CRES+OH→0.8*RO2R+0.2*BZO+0.25*MGLY+5.05*XC	k(T)=A$e^{-B/T}$,A=1.70e-12,B=-950
NPHE+OH→BZO+XN	k=3.50e-12
BALD+OH→BZCO3	k=1.20e-11
MACR+OH→0.5*RO2R+0.5*MACO3+0.416*CO+0.084*HCHO +0.416*MEK+0.084*MGLY+-0.416*XC	k(T)=A$e^{-B/T}$,A=8.00e-12,B=-380
MVK+OH→0.3*RO2R+0.675*MECO3+0.025*RO2N+0.675*R2O2 +0.3*HCHO+0.675*RCHO+0.3*MGLY+-0.725*XC	k(T)=A$e^{-B/T}$,A=2.60e-12,B=-610
IPRD+OH→0.67*RO2R+0.289*MACO3+0.041*RO2N+0.336*CO +0.055*HCHO+0.129*CCHO+0.013*RCHO+0.15*MEK+0.332*PRD2 +0.15*GLY+0.174*MGLY+-0.504*XC	k=6.19e-11
PRD2+OH→0.472*HO2+0.379*RO2R+0.029*MECO3+0.049*RCO3 +0.071*RO2N+0.094*R2O2+0.213*HCHO+0.084*CCHO+0.545*RCHO +0.115*MEK+0.336*PRD2+0.877*XC	k=1.55e-11
MEOH+OH→HO2+HCHO	k(T)=A$e^{-B/T}$,A=2.85e-12,B=345
FACD+OH→HO2+CO2	k=4.50e-13
AACD+OH→0.491*RO2R+0.509*MEO2+0.509*CO2+0.491*MGLY+-0.491*XC	k(T)=A$e^{-B/T}$,A=4.20e-14,B=-855
PACD+OH→RO2R+0.143*CO2+0.142*CCHO+0.4*RCHO +0.457*BACL+- 0.455*XC	k=1.20e-12
COOH+OH→0.3*OH+0.7*MEO2+0.3*HCHO	k(T)=A$e^{-B/T}$,A=3.80e-12,B=-200
ACET+OH→MECO3+R2O2+HCHO	k(T)=A$e^{-B/T}$(300/T)^N,A=4.56e-14,B=-429,N=-3.65
RCHO+OH→0.035*RO2R+0.965*RCO3+0.035*CO+0.035*CCHO	k(T)=A$e^{-B/T}$,A=5.10e-12,B=-405
CCHO+OH→MECO3	k(T)=A$e^{-B/T}$,A=4.4e-12,B=-365
HCHO+OH→HO2+CO	k(T)=A$e^{-B/T}$,A=5.4e-12,B=-135
MEK+OH→0.376*RO2R+0.51*MECO3+0.074*RCO3+0.039*RO2N +0.591*R2O2+0.088*HCHO+0.504*CCHO+0.376*RCHO+0.3*XC	k(T)=A$e^{-B/T}$(300/T)^N,A=1.3e-12,B=25,N=-2
XOOH+OH→OH	k=2.50e-11
organic nitrates and pernitrates	
RNO3+OH→0.189*HO2+0.305*RO2R+0.332*NO2+0.175*RO2N +0.671*R2O2+0.011*HCHO+0.429*CCHO+0.037*RCHO+0.004*ACET +0.18*MEK+0.039*PRD2+0.494*RNO3+0.174*XN+0.04*XC	k=7.20e-12
PAN+M→MECO3+NO2	k(T,M)=saprc_troe(4.90e-03,0.,12100,4.00e+16,0.,13600,0.30,1.41)

Reactions	Kinetic constants
PAN2→RCO3+NO2	k(T)=A$e^{-B/T}$,A=8.30e+16,B=13940
PBZN→BZCO3+NO2	k(T)=A$e^{-B/T}$,A=7.9e+16,B=14000
MAPAN→MACO3+NO2	k(T)=A$e^{-B/T}$,A=1.6e+16,B=13486
MECO3+NO2+M→PAN	k(T,M)=saprc_troe(2.70e-28,-7.10,0.,1.21e-11,-0.90,0.,0.30,1.41)
RCO3+NO2→PAN2	k(T)=A$e^{-B/T}$(300/T)^N,A=1.21e-11,B=0,N=1.07
BZCO3+NO2→PBZN	k=1.37e-11
MACO3+NO2→MAPAN	k(T)=A$e^{-B/T}$(300/T)^N,A=1.21e-11,B=0,N=1.07
TBUO+NO2→RNO3+-2*XC	k=2.4e-11
BZO+NO2→NPHE	k(T)=A$e^{-B/T}$,A=2.3e-11,B=-150
NO_3 attack	
MEO2+NO3→HCHO+HO2+NO2	k=1.30e-12
MECO3+NO3→MEO2+CO2+NO2	k=2.3e-12
RCO3+NO3→NO2+CCHO+RO2R+CO2	k=2.30e-12
BZCO3+NO3→NO2+CO2+BZO+R2O2	k=2.30e-12
MACO3+NO3→NO2+CO2+HCHO+MECO3	k=2.30e-12
R2O2+NO3→NO2	k=2.30e-12
HCHO+NO3→HNO3+HO2+CO	k(T)=A$e^{-B/T}$,A=2e-12,B=2431
CCHO+NO3→MECO3+HNO3	k(T)=A$e^{-B/T}$,A=1.40e-12,B=1860
RCHO+NO3→RCO3+HNO3	k(T)=A$e^{-B/T}$,A=1.40e-12,B=1601
GLY+NO3→0.63*HO2+0.37*RCO3+HNO3+1.26*CO+-0.37*XC	k(T)=A$e^{-B/T}$,A=2.80e-12,B=2376
MGLY+NO3→MECO3+HNO3+CO	k(T)=A$e^{-B/T}$,A=1.40e-12,B=1895
CRES+NO3→BZO+HNO3+XC	k=1.40e-11
BALD+NO3→BZCO3+HNO3	k(T)=A$e^{-B/T}$,A=1.34e-12,B=1860
MACR+NO3→0.5*RO2R+0.5*MACO3+0.5*HNO3+0.5*CO +0.5*XN+1.5*XC	k(T)=A$e^{-B/T}$,A=1.50e-12,B=1815
IPRD+NO3→0.799*RO2R+0.15*MACO3+0.051*RO2N+0.15*HNO3 +0.572*CO+0.227*HCHO+0.218*RCHO+0.008*MGLY+0.572*RNO3 +0.278*XN+-0.815*XC	k=1.00e-13
C2H4+NO3→RO2R+RCHO+XN+-1*XC	k(T)=A$e^{-B/T}$(300/T)^N,A=3.3e-12,B=2880,N=-2
OLE1+NO3→0.824*RO2R+0.176*RO2N+0.488*R2O2+0.009*CCHO +0.002*RCHO+0.024*ACET+0.546*RNO3+0.454*XN+0.572*XC	k(T)=A$e^{-B/T}$,A=4.73e-13,B=1047
OLE2+NO3→0.423*RO2R+0.409*NO2+0.033*MEO2+0.136*RO2N +0.762*R2O2+0.074*HCHO+0.546*CCHO+0.154*RCHO+0.11*ACET +0.002*MEK+0.026*MVK+0.007*IPRD+0.322*RNO3 +0.269*XN+0.114*XC	k(T)=A$e^{-B/T}$,A=2.20e-13,B=-382
ozonolysis	
AFG1+O3→0.522*HO2+0.826*OH+0.652*RCO3+0.652*R2O2 +0.522*CO+0.174*CO2+0.652*HCHO+0.432*GLY+0.568*MGLY+-0.872*XC	k=9.66e-18
AFG2+O3→0.522*HO2+0.826*OH+0.652*RCO3+0.652*R2O2 +0.522*CO+0.174*CO2+0.652*HCHO+0.432*GLY+0.568*MGLY+-0.872*XC	k=9.66e-18
AFG3+O3→0.554*HO2+0.095*RO2R+0.471*OH+0.013*MECO3 +0.163*RCO3+0.007*RO2N+0.163*R2O2+0.58*CO+0.19*CO2 +0.163*HCHO+0.366*GLY+0.279*MGLY+0.003*MACR+0.004*MVK +0.003*IPRD+0.35*AFG1+0.35*AFG2+0.139*AFG3+-0.575*XC	k=1.43e-17
MVK+O3→0.064*HO2+0.05*RO2R+0.164*OH+0.05*RCO3 +0.475*CO+0.124*CO2+0.1*HCHO+0.351*FACD+0.95*MGLY+-0.05*XC	k(T)=A$e^{-B/T}$,A=8.50e-16,B=1520
IPRD+O3→0.4*HO2+0.285*OH+0.048*RCO3+0.048*R2O2 +0.498*CO+0.14*CO2+0.125*HCHO+0.047*CCHO+0.21*MEK +0.1*FACD+0.372*PACD+0.023*GLY+0.742*MGLY+-0.329*XC	k=4.18e-18
C2H4+O3→0.16*HO2+0.16*OH+0.51*CO+0.12*CO2+HCHO +0.37*FACD	k(T)=A$e^{-B/T}$,A=9.14e-15,B=2580
ACYE+O3→1.5*HO2+0.5*OH+1.5*CO+0.5*CO2	k(T)=A$e^{-B/T}$,A=1.00e-14,B=4100
OLE1+O3→0.116*HO2+0.04*RO2R+0.193*OH+0.104*MEO2 +0.004*RO2N+0.023*R2O2+0.368*CO+0.125*CO2+0.5*HCHO +0.154*CCHO+0.384*RCHO+0.002*ACET+0.006*MEK+0.189*PRD2 +0.185*FACD+0.022*AACD+0.112*PACD+0.69*XC	k(T)=A$e^{-B/T}$,A=3.15e-15,B=1701
OLE2+O3→0.093*HO2+0.039*RO2R+0.423*OH+0.29*MEO2 +0.147*MECO3+0.008*RCO3+0.003*RO2N+0.161*R2O2+0.297*CO +0.162*CO2+0.26*HCHO+0.495*CCHO+0.333*RCHO+0.048*ACET +0.032*MEK+0.042*PRD2+0.033*FACD+0.061*AACD +0.222*PACD+0.028*MACR+0.021*MVK+0.125*XC	k(T)=A$e^{-B/T}$,A=8.14e-15,B=1255
MACR+O3→0.108*HO2+0.208*OH+0.1*RCO3+0.1*R2O2 +0.45*CO+0.117*CO2+0.2*HCHO+0.333*FACD+0.9*MGLY+-0.1*XC	k(T)=A$e^{-B/T}$,A=1.40e-15,B=2100
organic radical conversion	
MEO2+NO→NO2+HCHO+HO2	k(T)=A$e^{-B/T}$,A=2.30e-12,B=-360
MECO3+NO→MEO2+CO2+NO2	k(T)=A$e^{-B/T}$,A=7.50e-12,B=-290
RCO3+NO→NO2+CCHO+RO2R+CO2	k(T)=A$e^{-B/T}$,A=6.70e-12,B=-340

Reactions	Kinetic constants
BZCO3+NO→NO2+CO2+BZO+R2O2	k(T)=A$e^{-B/T}$,A=6.70e-12,B=-340
MACO3+NO→NO2+CO2+HCHO+MECO3	k(T)=A$e^{-B/T}$,A=6.70e-12,B=-340
R2O2+NO→NO2	k(T)=A$e^{-B/T}$,A=2.6e-12,B=-380
organic radical recombinaison	
MEO2+HO2→COOH	k(T)=A$e^{-B/T}$(300/T)^N,A=3.46e-13,B=-780,N=-0.36
MEO2+HO2→HCHO	k(T)=A$e^{-B/T}$(300/T)^N,A=3.34e-14,B=-780,N=3.53
MEO2+MEO2→MEOH+HCHO	k(T)=A$e^{-B/T}$(300/T)^N,A=6.39e-14,B=-365,N=1.80
MEO2+MEO2→2*HCHO+2*HO2	k(T)=A$e^{-B/T}$,A=7.40e-13,B=520
MECO3+HO2→AACD+0.3*O3	k(T)=A$e^{-B/T}$,A=5.20e-13,B=-980
MECO3+MEO2→0.1*AACD+0.1*HCHO+0.9*HCHO+0.9*HO2 +0.9*MEO2+0.9*CO2	k(T)=A$e^{-B/T}$,A=2.00e-12,B=-500
MECO3+MECO3→2*MEO2+2*CO2	k(T)=A$e^{-B/T}$,A=2.90e-12,B=-500
RCO3+HO2→PACD+0.25*O3	k(T)=A$e^{-B/T}$,A=5.20e-13,B=-980
RCO3+MEO2→CO2+RO2R+CCHO+HCHO+HO2	k(T)=A$e^{-B/T}$,A=2.00e-12,B=-500
RCO3+MECO3→2*CO2+MEO2+CCHO+RO2R	k(T)=A$e^{-B/T}$,A=2.90e-12,B=-500
RCO3+RCO3→2*CCHO+2*RO2R+2*CO2	k(T)=A$e^{-B/T}$,A=2.90e-12,B=-500
BZCO3+HO2→PACD+0.25*O3+4*XC	k(T)=A$e^{-B/T}$,A=5.20e-13,B=-980
BZCO3+MEO2→CO2+BZO+R2O2+HCHO+HO2	k(T)=A$e^{-B/T}$,A=2.00e-12,B=-500
BZCO3+MECO3→2*CO2+MEO2+BZO+R2O2	k(T)=A$e^{-B/T}$,A=2.90e-12,B=-500
BZCO3+RCO3→2*CO2+CCHO+RO2R+BZO+R2O2	k(T)=A$e^{-B/T}$,A=2.90e-12,B=-500
BZCO3+BZCO3→2*BZO+2*R2O2+2*CO2	k(T)=A$e^{-B/T}$,A=2.90e-12,B=-500
MACO3+HO2→PACD+0.25*O3+XC	k(T)=A$e^{-B/T}$,A=5.20e-13,B=-980
MACO3+MEO2→2*HCHO+HO2+CO2+MECO3	k(T)=A$e^{-B/T}$,A=2.00e-12,B=-500
MACO3+MECO3→2*CO2+MEO2+HCHO+MECO3	k(T)=A$e^{-B/T}$,A=2.90e-12,B=-500
MACO3+RCO3→HCHO+MECO3+CCHO+RO2R+2*CO2	k(T)=A$e^{-B/T}$,A=2.90e-12,B=-500
MACO3+BZCO3→HCHO+MECO3+BZO+R2O2+2*CO2	k(T)=A$e^{-B/T}$,A=2.90e-12,B=-500
MACO3+MACO3→2*HCHO+2*MECO3+2*CO2	k(T)=A$e^{-B/T}$,A=2.90e-12,B=-500
BZO+HO2→CRES+-1*XC	k(T)=A$e^{-B/T}$,A=3.80e-13,B=-900
R2O2+HO2→HO2	k(T)=A$e^{-B/T}$,A=3.80e-13,B=-900
R2O2+MEO2→MEO2	k=2e-13
BZO→CRES+RO2R+-1*XC	k=1.e-3
TBUO→ACET+MEO2	k(T)=A$e^{-B/T}$,A=7.5e+14,B=8152
photolytic reactions	
NO2→NO+O3	J(Z)=photorate(NO2)
NO3→NO	J(Z)=photorate(NO3-1)
NO3→NO2+O3	J(Z)=photorate(NO3-2)
O3→2*OH	J(T,Z,H2O)=photorate(O3-1)
HONO→OH+NO	J(Z)=photorate(HONO)
HNO3→OH+NO2	J(Z)=photorate(HNO3)
HNO4→0.61*HO2+0.61*NO2+0.39*OH+0.39*NO3	J(Z)=photorate(HNO4)
H2O2→2*OH	J(Z)=photorate(H2O2)
PAN→0.6*MECO3+0.6*NO2+0.4*MEO2+0.4*CO2+0.4*NO3	J(Z)=photorate(PAN)
PAN2→0.6*RCO3+0.6*NO2+0.4*CCHO+0.4*RO2R+0.4*CO2+0.4*NO3	J(Z)=photorate(PAN2)
PBZN→0.6*BZCO3+0.6*NO2+0.4*CO2+0.4*BZO+0.4*R2O2+0.4*NO3	J(Z)=photorate(PBZN)
MAPAN→0.6*MACO3+0.6*NO2+0.4*CO2+0.4*HCHO+0.4*MECO3 +0.4*NO3	J(Z)=photorate(MAPAN)
HCHO→2*HO2+CO	J(Z)=photorate(HCHO-1)
HCHO→CO	J(Z)=photorate(HCHO-2)
CCHO→HO2+MEO2+CO	J(Z)=photorate(CCHO)
RCHO→HO2+RO2R+CO+CCHO	J(Z)=photorate(RCHO)
ACET→1.38*MEO2+0.62*MECO3+0.38*CO	J(Z)=photorate(ACET)
MEK→RO2R+MECO3+CCHO	J(Z)=photorate(MEK)
COOH→HO2+OH+HCHO	J(Z)=photorate(COOH)
XOOH→HO2+OH	J(Z)=photorate(XOOH)
GLY→2*HO2+2*CO	J(Z)=photorate(GLY-07R)
GLY→HCHO+CO	J(Z)=photorate(GLY-07M)
MGLY→HO2+MECO3+CO	J(Z)=photorate(MGLY)
BACL→2*MECO3	J(Z)=photorate(BACL)
NPHE→HONO+6*XC	J(Z)=photorate(NPHE)
NPHE→XN+6*XC	J(Z)=photorate(NPHE-2)
BALD→7*XC	J(Z)=photorate(BALD)
AFG1→1.023*HO2+0.173*MEO2+0.305*MECO3+0.5*MACO3 +0.695*CO+0.195*GLY+0.305*MGLY+0.217*XC	J(Z)=photorate(AFG1)
AFG2→PRD2+-1*XC	J(Z)=photorate(AFG2)

Reactions	Kinetic constants
MACR→0.67*HO2+0.33*OH+0.67*MECO3+0.33*MACO3 +0.33*R2O2+0.67*CO+0.67*HCHO	J(Z)=photorate(MACR)
MVK→0.4*MEO2+0.4*MACO3+0.6*CO+0.6*PRD2+-2.2*XC	J(Z)=photorate(MVK)
IPRD→1.233*HO2+0.467*MECO3+0.3*RCO3+1.233*CO+0.3*HCHO +0.467*CCHO+0.233*MEK+-0.233*XC	J(Z)=photorate(IPRD)
PRD2→0.913*RO2R+0.4*MECO3+0.6*RCO3+0.087*RO2N +0.677*R2O2+0.303*HCHO+0.163*CCHO+0.78*RCHO+-0.091*XC	J(Z)=photorate(PRD2)
RNO3→0.344*HO2+0.554*RO2R+NO2+0.102*RO2N+0.167*R2O2 +0.135*HCHO+0.444*CCHO+0.137*RCHO+0.008*ACET+0.207*MEK +0.451*PRD2+0.396*XC	J(Z)=photorate(RNO3)
O_3 attack to unsaturated hydrocarbons	
C5H8+O3→0.066*HO2+0.266*OH+0.192*MACO3+0.008*RO2N +0.192*R2O2+0.275*CO+0.122*CO2+0.592*HCHO+0.1*PRD2 +0.204*FACD+0.39*MACR+0.16*MVK+0.15*IPRD+-0.559*XC	k(T)=A$e^{-B/T}$,A=7.86e-15,B=1912
C5H8+NO3→0.749*RO2R+0.187*NO2+0.064*RO2N+0.187*R2O2 +0.936*IPRD+0.813*XN+-0.064*XC	k(T)=A$e^{-B/T}$,A=3.03e-12,B=448
TERPEN+OH→0.759*RO2R+0.042*RCO3+0.2*RO2N+0.388*R2O2 +0.001*CO+0.264*HCHO+0.533*RCHO+0.036*ACET+0.005*MEK +0.255*PRD2+0.009*MGLY+0.014*BACL+0.002*MVK +0.001*IPRD+5.056*XC	k(T)=A$e^{-B/T}$,A=1.87e-11,B=-435
TERPEN+O3→0.052*HO2+0.067*RO2R+0.585*OH+0.126*MECO3 +0.149*RCO3+0.203*RO2N+0.808*R2O2+0.185*CO+0.045*CO2 +0.229*HCHO+0.22*RCHO+0.165*ACET+0.004*MEK+0.409*PRD2 +0.107*FACD+0.043*PACD+0.001*GLY+0.002*MGLY+0.055*BACL +0.001*MACR+0.001*IPRD+3.526*XC	k(T)=A$e^{-B/T}$,A=9.57e-16,B=785
TERPEN+NO3→0.162*RO2R+0.421*NO2+0.019*RCO3+0.397*RO2N +1.347*R2O2+0.01*CO+0.017*HCHO+0.001*CCHO+0.509*RCHO +0.175*ACET+0.001*MGLY+0.003*MACR+0.001*MVK+0.002*IPRD +0.163*RNO3+0.416*XN+4.473*XC	k(T)=A$e^{-B/T}$,A=1.28e-12,B=-490
APINEN+OH→0.759*RO2R+0.042*RCO3+0.2*RO2N+0.388*R2O2 +0.001*CO+0.264*HCHO+0.533*RCHO+0.036*ACET+0.005*MEK +0.255*PRD2+0.009*MGLY+0.014*BACL+0.002*MVK +0.001*IPRD+5.056*XC	k(T)=A$e^{-B/T}$,A=1.87e-11,B=-435
APINEN+O3→0.052*HO2+0.067*RO2R+0.585*OH+0.126*MECO3 +0.149*RCO3+0.203*RO2N+0.808*R2O2+0.185*CO+0.045*CO2 +0.229*HCHO+0.22*RCHO+0.165*ACET+0.004*MEK+0.409*PRD2 +0.107*FACD+0.043*PACD+0.001*GLY+0.002*MGLY+0.055*BACL +0.001*MACR+0.001*IPRD+3.526*XC	k(T)=A$e^{-B/T}$,A=9.57e-16,B=785
APINEN+NO3→0.162*RO2R+0.421*NO2+0.019*RCO3+0.397*RO2N +1.347*R2O2+0.01*CO+0.017*HCHO+0.001*CCHO+0.509*RCHO +0.175*ACET+0.001*MGLY+0.003*MACR+0.001*MVK+0.002*IPRD +0.163*RNO3+0.416*XN+4.473*XC	k(T)=A$e^{-B/T}$,A=1.28e-12,B=-490
BPINEN+OH→0.759*RO2R+0.042*RCO3+0.2*RO2N+0.388*R2O2 +0.001*CO+0.264*HCHO+0.533*RCHO+0.036*ACET+0.005*MEK +0.255*PRD2+0.009*MGLY+0.014*BACL+0.002*MVK +0.001*IPRD+5.056*XC	k(T)=A$e^{-B/T}$,A=1.87e-11,B=-435
BPINEN+O3→0.052*HO2+0.067*RO2R+0.585*OH+0.126*MECO3 +0.149*RCO3+0.203*RO2N+0.808*R2O2+0.185*CO+0.045*CO2 +0.229*HCHO+0.22*RCHO+0.165*ACET+0.004*MEK+0.409*PRD2 +0.107*FACD+0.043*PACD+0.001*GLY+0.002*MGLY+0.055*BACL +0.001*MACR+0.001*IPRD+3.526*XC	k(T)=A$e^{-B/T}$,A=9.57e-16,B=785
BPINEN+NO3→0.162*RO2R+0.421*NO2+0.019*RCO3+0.397*RO2N +1.347*R2O2+0.01*CO+0.017*HCHO+0.001*CCHO+0.509*RCHO +0.175*ACET+0.001*MGLY+0.003*MACR+0.001*MVK+0.002*IPRD +0.163*RNO3+0.416*XN+4.473*XC	k(T)=A$e^{-B/T}$,A=1.28e-12,B=-490
LIMONE+OH→0.759*RO2R+0.042*RCO3+0.2*RO2N+0.388*R2O2 +0.001*CO+0.264*HCHO+0.533*RCHO+0.036*ACET+0.005*MEK +0.255*PRD2+0.009*MGLY+0.014*BACL+0.002*MVK +0.001*IPRD+5.056*XC	k(T)=A$e^{-B/T}$,A=1.87e-11,B=-435
LIMONE+O3→0.052*HO2+0.067*RO2R+0.585*OH+0.126*MECO3 +0.149*RCO3+0.203*RO2N+0.808*R2O2+0.185*CO+0.045*CO2 +0.229*HCHO+0.22*RCHO+0.165*ACET+0.004*MEK+0.409*PRD2 +0.107*FACD+0.043*PACD+0.001*GLY+0.002*MGLY+0.055*BACL +0.001*MACR+0.001*IPRD+3.526*XC	k(T)=A$e^{-B/T}$,A=9.57e-16,B=785

Reactions	Kinetic constants
LIMONE+NO3→0.162*RO2R+0.421*NO2+0.019*RCO3+0.397*RO2N +1.347*R2O2+0.01*CO+0.017*HCHO+0.001*CCHO+0.509*RCHO +0.175*ACET+0.001*MGLY+0.003*MACR+0.001*MVK+0.002*IPRD +0.163*RNO3+0.416*XN+4.473*XC	k(T)=Ae$^{-B/T}$,A=1.28e-12,B=-490
OCIMEN+OH→0.759*RO2R+0.042*RCO3+0.2*RO2N+0.388*R2O2 +0.001*CO+0.264*HCHO+0.533*RCHO+0.036*ACET+0.005*MEK +0.255*PRD2+0.009*MGLY+0.014*BACL+0.002*MVK +0.001*IPRD+5.056*XC	k(T)=Ae$^{-B/T}$,A=1.87e-11,B=-435
OCIMEN+O3→0.052*HO2+0.067*RO2R+0.585*OH+0.126*MECO3 +0.149*RCO3+0.203*RO2N+0.808*R2O2+0.185*CO+0.045*CO2 +0.229*HCHO+0.22*RCHO+0.165*ACET+0.004*MEK+0.409*PRD2 +0.107*FACD+0.043*PACD+0.001*GLY+0.002*MGLY+0.055*BACL +0.001*MACR+0.001*IPRD+3.526*XC	k(T)=Ae$^{-B/T}$,A=9.57e-16,B=785
OCIMEN+NO3→0.162*RO2R+0.421*NO2+0.019*RCO3+0.397*RO2N +1.347*R2O2+0.01*CO+0.017*HCHO+0.001*CCHO+0.509*RCHO +0.175*ACET+0.001*MGLY+0.003*MACR+0.001*MVK+0.002*IPRD +0.163*RNO3+0.416*XN+4.473*XC	k(T)=Ae$^{-B/T}$,A=1.28e-12,B=-490
HUMULE+OH→0.759*RO2R+0.042*RCO3+0.2*RO2N+0.388*R2O2 +0.001*CO+0.264*HCHO+0.533*RCHO+0.036*ACET+0.005*MEK +0.255*PRD2+0.009*MGLY+0.014*BACL+0.002*MVK+0.001*IPRD +5.056*XC	k(T)=Ae$^{-B/T}$,A=1.87e-11,B=-435
HUMULE+O3→0.052*HO2+0.067*RO2R+0.585*OH+0.126*MECO3 +0.149*RCO3+0.203*RO2N+0.808*R2O2+0.185*CO+0.045*CO2 +0.229*HCHO+0.22*RCHO+0.165*ACET+0.004*MEK+0.409*PRD2 +0.107*FACD+0.043*PACD+0.001*GLY+0.002*MGLY+0.055*BACL +0.001*MACR+0.001*IPRD+3.526*XC	k(T)=Ae$^{-B/T}$,A=9.57e-16,B=785
HUMULE+NO3→0.162*RO2R+0.421*NO2+0.019*RCO3+0.397*RO2N +1.347*R2O2+0.01*CO+0.017*HCHO+0.001*CCHO+0.509*RCHO +0.175*ACET+0.001*MGLY+0.003*MACR+0.001*MVK+0.002*IPRD +0.163*RNO3+0.416*XN+4.473*XC	k(T)=Ae$^{-B/T}$,A=1.28e-12,B=-490

- The reactions O_3+hν→O(1D), O(1D)+H_2O→2OH, O(1D)+M→O(3P)+M and O(3P)+O_2+M→O_3+M have been combined to reaction O_3+hν→ O(1D) by taking into account H_2O and M concentrations in order to adjust the photolysis frequency.
- The reactions OH+CO→HO2+CO2, HO2+HO2→H2O2 et HO2+HO2+H2O→H2O2 :

$$k\left(T,M\right)=k_1\left(T\right)+k_2\left(T\right)\left[M\right]$$

  The parameters are given in the order A_1, B_1, A_2, B_2
- The reactions OH+HNO3→NO3 :

$$k\left(T,M\right)=k_0\left(T\right)+\frac{k_3\left(T\right)\left[M\right]}{\left(1+\frac{k_3(T)[M]}{k_2(T)}\right)}$$

  The parameters are given in the order $A_0, B_0, A_2, B_2, A_3, B_3$
- Three body Troe reactions :

$$k(T,M)=\frac{k_0(T)[M]}{1+\frac{k_0(T)[M]}{k_\infty(T)}}\times F^Z$$

  with

$$Z=\frac{1}{1+\left(\frac{log_{10}\left(\frac{k_0(T)[M]}{k_\infty(T)}\right)}{N}\right)^2}$$

$$k_0=A_0e^{\left(-\frac{B_0}{T}\right)}\left(\frac{T}{300}\right)^{-n},\ \text{and}\ k_\infty=A_\infty e^{\left(-\frac{B_\infty}{T}\right)}\left(\frac{T}{300}\right)^{-m},$$

  The parameters are given in the order $A_0, n, B_0, A_\infty, m, B_\infty, F, N$

G.4 Reaction list of the chlorine SAPRC-07-A chemical mechanism

Reactions	Kinetic constants
Base Chlorine Mechanism	
CL2→2*CL	J(Z)=photorate(CL2)
CL+NO→CLNO	k(T)=Aexp$^{-B/T}$(300/T)^{N},A=7.60e-32,B=0.,N=-1.80
CLNO→CL+NO	J(Z)=photorate(CLNO)
CL+NO2+M→CLONO	k(T,M)=saprc_troe(1.30e-30,-2.00,0.,1.00e-10,-1.00,0.,0.60,1.00)
CL+NO2+M→CLNO2	k(T,M)=saprc_troe(1.80e-31,-2.00,0.,1.00e-10,-1.00,0.,0.60,1.00)
CLONO→CL+NO2	J(Z)=photorate(CLONO)
CLNO2→CL+NO2	J(Z)=photorate(CLNO2)
CL+HO2→HCL	k(T)=Aexp$^{-B/T}$(300/T)^{N},A=3.44e-11,B=0.,N=-0.56
CL+HO2→CLO+OH	k(T)=Aexp$^{-B/T}$(300/T)^{N},A=9.41e-12,B=0.,N=2.10
CL+O3→CLO	k(T)=Aexp$^{-B/T}$,A=2.80e-11,B=250
CL+NO3→CLO+NO2	k=2.40e-11
CLO+NO→CL+NO2	k(T)=Aexp$^{-B/T}$,A=6.20e-12,B=-295
CLO+NO2+M→CLONO2	k(T,M)=saprc_troe(1.80e-31,-3.40,0.,1.50e-11,-1.90,0.,0.60,1.00)
CLONO2→CLO+NO2	J(Z)=photorate(CLONO2-1)
CLONO2→CL+NO3	J(Z)=photorate(CLONO2-2)
CLONO2+M→CLO+NO2	k(T,M)=saprc_troe(4.85e-5,-1.,1253,3.71e15,3.5,12530,0.60,1.00)
CL+CLONO2→CL2+NO3	k(T)=Aexp$^{-B/T}$,A=6.20e-12,B=-145
CLO+HO2→HOCL	k(T)=Aexp$^{-B/T}$,A=2.20e-12,B=-340
HOCL→OH+CL	J(Z)=photorate(HOCL)
CLO+CLO→0.29*CL2+1.42*CL	k(T)=Aexp$^{-B/T}$,A=1.25e-11,B=1960
OH+HCL→CL	k(T)=Aexp$^{-B/T}$,A=1.70e-12,B=230
CL+H2→HCL+HO2	k(T)=Aexp$^{-B/T}$,A=3.90e-11,B=2310
HCHO+CL→HCL+HO2+CO	k(T)=Aexp$^{-B/T}$,A=8.10e-11,B=30
CCHO+CL→HCL+MECO3	k=8.00e-11
MEOH+CL→HCL+HO2+HCHO	k=5.50e-11
RCHO+CL→HCL+0.1*RO2R+0.9*RCO3+0.1*CO+0.1*CCHO	k=1.23e-10
ACET+CL→HCL+MECO3+R2O2+HCHO	k(T)=Aexp$^{-B/T}$,A=7.70e-11,B=1000
MEK+CL→HCL+0.84*RO2R+0.085*MECO3+0.036*RCO3 +0.039*RO2N+0.135*R2O2+0.065*HCHO+0.07*CCHO +0.84*RCHO+0.763*XC	k=3.60e-11
RNO3+CL→HCL+0.055*HO2+0.547*RO2R+0.197*NO2 +0.202*RO2N+0.735*R2O2+0.045*HCHO+0.3*CCHO +0.029*RCHO+0.003*ACET+0.059*MEK+0.058*PRD2 +0.602*RNO3+0.201*XN+-0.149*XC	k=1.92e-10
PRD2+CL→HCL+0.314*HO2+0.541*RO2R+0.007*MECO3 +0.022*RCO3+0.116*RO2N+0.138*R2O2+0.237*HCHO +0.109*CCHO+0.789*RCHO+0.051*MEK+0.156*PRD2+1.262*XC	k=2.00e-10
GLY+CL→HCL+0.63*HO2+0.37*RCO3+1.26*CO+-0.37*XC	k(T)=Aexp$^{-B/T}$,A=8.10e-11,B=30
MGLY+CL→HCL+MECO3+CO	k=8.00e-11
CRES+CL→HCL+RO2R+-1*R2O2+BALD	k=6.20e-11
BALD+CL→HCL+BZCO3	k=8.00e-11
XOOH+CL→CL	k=1.66e-10
MACR+CL→0.25*HCL+0.802*RO2R+0.165*MACO3+0.033*RO2N +0.541*CO+0.082*IPRD+0.18*CLCCHO+0.541*CLACET +0.208*XC	k=3.85e-10
MVK+CL→0.322*RO2R+0.625*MECO3+0.053*RO2N +0.961*R2O2+0.947*CLCCHO+0.538*XC	k=2.32e-10
IPRD+CL→0.401*HCL+0.084*HO2+0.712*RO2R +0.154*MACO3+0.051*RO2N+0.018*R2O2+0.498*CO +0.195*HCHO+0.017*MGLY+0.115*IPRD+0.051*AFG1 +0.051*AFG2+0.14*CLCCHO+0.42*CLACET+0.709*XC	k=4.12e-10
CLCCHO→HO2+RO2CL+CO+HCHO	J(Z)=photorate(CLCCHO)
CLCCHO+OH→RCO3+-1*XC	k=3.10e-12
CLCCHO+CL→HCL+RCO3+-1*XC	k=1.29e-11
CLACET→RO2CL+MECO3+HCHO	J(Z)=photorate(CLACET)
RO2CL+NO→NO2+CL	k(T)=Aexp$^{-B/T}$,A=2.60e-12,B=-380
RO2CL+HO2→XOOH	k(T)=Aexp$^{-B/T}$,A=3.80e-13,B=-900
RO2CL+NO3→NO2+CL	k=2.30e-12
RO2CL+MEO2→0.5*CL+0.5*HO2+0.75*HCHO+0.25*MEOH	k=2e-13
RO2CL+RO2R→0.5*CL+0.5*HO2	k=3.50e-14
RO2CL+R2O2→RO2CL	k=3.50e-14
RO2CL+RO2N→0.5*CL+0.5*HO2+0.5*MEK+0.5*PRD2+XC	k=3.50e-14
RO2CL+RO2CL→CL	k=3.50e-14
RO2CL+MECO3→MEO2+CO2	k(T)=Aexp$^{-B/T}$,A=4.40e-13,B=-1070
RO2CL+RCO3→CO2+RO2R+CCHO	k(T)=Aexp$^{-B/T}$,A=4.40e-13,B=-1070
RO2CL+BZCO3→CO2+BZO+R2O2	k(T)=Aexp$^{-B/T}$,A=4.40e-13,B=-1070

Reactions	Kinetic constants
RO2CL+MACO3→CO2+HCHO+MECO3	k(T)=Aexp$^{-B/T}$,A=4.40e-13,B=-1070
CH4+CL→HCL+MEO2	k(T)=Aexp$^{-B/T}$,A=7.30e-12,B=1280
C2H4+CL+M→RO2R+R2O2+HCHO+CLCHO	k(T,M)=saprc_troe(1.60e-29,-3.30,0.,3.10e-10,-1.00,0.,0.60,1.00)
C5H8+CL→0.15*HCL+0.738*RO2R+0.177*RO2CL+0.085*RO2N +0.253*R2O2+0.275*HCHO+0.177*MVK+0.671*IPRD+0.067*CLCCHO +0.018*XC	k=4.80e-10
ACYE+CL+M→HO2+CO+XC	k(T,M)=saprc_troe(5.20e-30,-2.40,0.,2.20e-10,0.,0.,0.60,1.00)
Chlorine + Atmospheric Lumped VOC Mechanism	
ALK1+CL→HCL+RO2R+CCHO	k=5.95e-11
ALK2+CL→HCL+0.97*RO2R+0.03*RO2N+0.482*RCHO+0.488*ACET+- 0.09*XC	k=1.37e-10
ALK3+CL→HCL+0.835*RO2R+0.094*TBUO+0.07*RO2N +0.526*R2O2+0.078*HCHO+0.34*CCHO+0.343*RCHO+0.075*ACET +0.253*MEK+0.18*XC	k=1.86e-10
ALK4+CL→HCL+0.827*RO2R+0.003*MEO2+0.004*MECO3 +0.165*RO2N+0.91*R2O2+0.003*CO+0.034*HCHO0.287*CCHO +0.412*RCHO+0.247*ACET+0.076*MEK+0.13*PRD2+0.327*XC	k=2.63e-10
ALK5+CL→HCL+0.647*RO2R+0.352*RO2N+0.894*R2O2 +0.022*HCHO+0.08*CCHO+0.258*RCHO+0.044*ACET +0.041*MEK+0.378*PRD2+2.368*XC	k=4.21e-10
OLE1+CL→0.308*HCL+0.902*RO2R+0.098*RO2N+0.518*R2O2 +0.025*HCHO+0.146*CCHO+0.051*RCHO+0.188*MACR+0.014*MVK +0.027*IPRD+0.225*CLCCHO+0.396*CLACET+1.361*XC	k=3.55e-10
OLE2+CL→0.263*HCL+0.447*RO2R+0.448*RO2CL+0.001*MEO2 +0.104*RO2N+0.619*R2O2+0.228*HCHO+0.361*CCHO+0.3*RCHO +0.081*ACET+0.04*MEK+0.049*MACR+0.055*MVK+0.179*IPRD +0.012*CLCCHO+0.18*CLACET+0.247*XC	k=3.83e-10
ARO1+CL→0.88*RO2R+0.12*RO2N+0.21*PRD2+0.671*BALD +0.323*XC	k=1.00e-10
ARO2+CL→0.842*RO2R+0.158*RO2N+0.224*PRD2+0.618*BALD +2.382*XC	k=2.18e-10
TERPEN+CL→0.548*HCL+0.252*RO2R+0.068*RO2CL+0.034*MECO3 +0.05*RCO3+0.016*MACO3+0.582*RO2N+1.938*R2O2+0.035*CO +0.158*HCHO+0.185*RCHO+0.274*ACET+0.007*GLY+0.003*BACL +0.003*MVK+0.158*IPRD+0.006*AFG1+0.006*AFG2+0.001*AFG3 +0.109*CLCCHO+3.543*XC	k=5.46e-10
HUMULE+CL→0.548*HCL+0.252*RO2R+0.068*RO2CL+0.034*MECO3 +0.05*RCO3+0.016*MACO3+0.582*RO2N+1.938*R2O2+0.035*CO +0.158*HCHO+0.185*RCHO+0.274*ACET+0.007*GLY+0.003*BACL +0.003*MVK+0.158*IPRD+0.006*AFG1+0.006*AFG2+0.001*AFG3 +0.109*CLCCHO+3.543*XC	k=5.46e-10
OCIMEN+CL→0.548*HCL+0.252*RO2R+0.068*RO2CL+0.034*MECO3 +0.05*RCO3+0.016*MACO3+0.582*RO2N+1.938*R2O2+0.035*CO +0.158*HCHO+0.185*RCHO+0.274*ACET+0.007*GLY+0.003*BACL +0.003*MVK+0.158*IPRD+0.006*AFG1+0.006*AFG2+0.001*AFG3 +0.109*CLCCHO+3.543*XC	k=5.46e-10
LIMONE+CL→0.548*HCL+0.252*RO2R+0.068*RO2CL+0.034*MECO3 +0.05*RCO3+0.016*MACO3+0.582*RO2N+1.938*R2O2+0.035*CO +0.158*HCHO+0.185*RCHO+0.274*ACET+0.007*GLY+0.003*BACL +0.003*MVK+0.158*IPRD+0.006*AFG1+0.006*AFG2+0.001*AFG3 +0.109*CLCCHO+3.543*XC	k=5.46e-10
BPINEN+CL→0.548*HCL+0.252*RO2R+0.068*RO2CL+0.034*MECO3 +0.05*RCO3+0.016*MACO3+0.582*RO2N+1.938*R2O2+0.035*CO +0.158*HCHO+0.185*RCHO+0.274*ACET+0.007*GLY+0.003*BACL +0.003*MVK+0.158*IPRD+0.006*AFG1+0.006*AFG2+0.001*AFG3 +0.109*CLCCHO+3.543*XC	k=5.46e-10
APINEN+CL→0.548*HCL+0.252*RO2R+0.068*RO2CL+0.034*MECO3 +0.05*RCO3+0.016*MACO3+0.582*RO2N+1.938*R2O2+0.035*CO +0.158*HCHO+0.185*RCHO+0.274*ACET+0.007*GLY+0.003*BACL +0.003*MVK+0.158*IPRD+0.006*AFG1+0.006*AFG2+0.001*AFG3 +0.109*CLCCHO+3.543*XC	k=5.46e-10

G.5 Reaction list of the VBS scheme with functionalization process for POA

Reactions	kinetic rates ($molec.cm^{-3}.s^{-1}$)
POA2+OH→OH+1.075*OPOA1	k=4.00e-11
POA3+OH→OH+1.075*OPOA2	k=4.00e-11
POA4+OH→OH+1.075*OPOA3	k=4.00e-11
POA5+OH→OH+1.075*OPOA4	k=4.00e-11
POA6+OH→OH+1.075*OPOA5	k=4.00e-11
POA7+OH→OH+1.075*OPOA6	k=4.00e-11
POA8+OH→OH+1.075*OPOA7	k=4.00e-11
POA9+OH→OH+1.075*OPOA8	k=4.00e-11
OPOA2+OH→OH+1.075*OPOA1	k=4.00e-11
OPOA3+OH→OH+1.075*OPOA2	k=4.00e-11
OPOA4+OH→OH+1.075*OPOA3	k=4.00e-11
OPOA5+OH→OH+1.075*OPOA4	k=4.00e-11
OPOA6+OH→OH+1.075*OPOA5	k=4.00e-11
OPOA7+OH→OH+1.075*OPOA6	k=4.00e-11
OPOA8+OH→OH+1.075*OPOA7	k=4.00e-11

Table G.5: *Gas phase chemical scheme for VBS scheme used in CHIMERE by [Zhang et al., 2013]*

G.6 Reaction list of the VBS scheme with functionalization and fragmentation process for POA

Reactions	kinetic rates ($molec.cm^{-3}.s^{-1}$)
POA2+OH→OH+1.075*OPOA1	k=4.00e-11
POA3+OH→OH+1.075*OPOA2	k=4.00e-11
POA4+OH→OH+1.075*OPOA3	k=4.00e-11
POA5+OH→OH+1.075*OPOA4	k=4.00e-11
POA6+OH→OH+1.075*OPOA5	k=4.00e-11
POA7+OH→OH+1.075*OPOA6	k=4.00e-11
POA8+OH→OH+1.075*OPOA7	k=4.00e-11
POA9+OH→OH+1.075*OPOA8	k=4.00e-11
OPOA2+OH→OH+1.075*O2POA1	k=4.00e-11
OPOA3+OH→OH+1.075*O2POA2	k=4.00e-11
OPOA4+OH→OH+1.075*O2POA3	k=4.00e-11
OPOA5+OH→OH+1.075*O2POA4	k=4.00e-11
OPOA6+OH→OH+1.075*O2POA5	k=4.00e-11
OPOA7+OH→OH+1.075*O2POA6	k=4.00e-11
OPOA8+OH→OH+1.075*O2POA7	k=4.00e-11
O2POA2+OH→OH+1.075*O3POA1	k=4.00e-11
O2POA3+OH→OH+1.075*O3POA2	k=4.00e-11
O2POA4+OH→OH+1.075*O3POA3	k=4.00e-11
O2POA5+OH→OH+1.075*O3POA4	k=4.00e-11
O2POA6+OH→OH+1.075*O3POA5	k=4.00e-11
O2POA7+OH→OH+1.075*O3POA6	k=4.00e-11
O2POA8+OH→OH+1.075*O3POA7	k=4.00e-11
Fragmentation process	
O3POA1+OH→OH+0.16125*O3POA1+0.75*O3POA6	k=4.00e-11
O3POA2+OH→OH+0.16125*O3POA1+0.75*O3POA6	k=4.00e-11
O3POA3+OH→OH+0.16125*O3POA2+0.75*O3POA6	k=4.00e-11
O3POA4+OH→OH+0.16125*O3POA3+0.75*O3POA6	k=4.00e-11
O3POA5+OH→OH+0.16125*O3POA4+0.75*O3POA6	k=4.00e-11
O3POA6+OH→OH+0.16125*O3POA5+0.75*O3POA7	k=4.00e-11
O3POA7+OH→OH+0.16125*O3POA6+0.75*O3POA8	k=4.00e-11
O3POA8+OH→OH+0.16125*O3POA7+0.75*O3POA8	k=4.00e-11
Production of non volatile secondary aerosol from organic aerosols	
OPOA1→ONVSOA	k=3e-4
OPOA2→ONVSOA	k=3e-4
OPOA3→ONVSOA	k=3e-4
OPOA4→ONVSOA	k=3e-4
OPOA5→ONVSOA	k=3e-4
OPOA6→ONVSOA	k=3e-4
O2POA1→O2NVSOA	k=3e-4
O2POA2→O2NVSOA	k=3e-4
O2POA3→O2NVSOA	k=3e-4
O2POA4→O2NVSOA	k=3e-4
O2POA5→O2NVSOA	k=3e-4
O2POA6→O2NVSOA	k=3e-4
O3POA1→O3NVSOA	k=3e-4
O3POA2→O3NVSOA	k=3e-4
O3POA3→O3NVSOA	k=3e-4
O3POA4→O3NVSOA	k=3e-4
O3POA5→O3NVSOA	k=3e-4
O3POA6→O3NVSOA	k=3e-4

Table G.6: *Gas phase chemical scheme for VBS scheme used in CHIMERE by [Cholakian et al., 2018]*

G.7 The SOA simple scheme with single-step oxidation

Reactions	kinetic rates ($molec.cm^{-3}.s^{-1}$)
NC4H10+OH → 0.07*SOA	k(T)=Aexp(-B/T)(300/T)**N,A=1.36e-12,B=-190,N=-2
OXYL+OH → 0.2*SOA	k=1.37e-11
APINEN+OH → 0.07*SOA	k(T)=Aexp(-B/T),A=1.21e-11,B=-444
BPINEN+OH → 0.07*SOA	k(T)=Aexp(-B/T),A=1.21e-11,B=-444
LIMONE+OH → 0.07*SOA	k(T)=Aexp(-B/T),A=1.21e-11,B=-444
OCIMEN+OH → 0.07*SOA	k(T)=Aexp(-B/T),A=1.21e-11,B=-444
TERPEN+OH → 0.07*SOA	k(T)=Aexp(-B/T),A=1.21e-11,B=-444
HUMULE+OH → 0.07*SOA	k(T)=Aexp(-B/T),A=1.21e-11,B=-444

Table G.7: *Gas phase chemical scheme for SOA formation in CHIMERE with simple scheme.*

G.8 The SOA medium scheme with single-step oxidation

Reactions	kinetic rates ($molec.cm^{-3}.s^{-1}$)
TOL+OH → 0.005×AnA1D + 0.2×AnBmP	1.81×10^{-12}exp(355/T)
TMB+OH → 0.005×AnA1D + 0.2×AnBmP	1.81×10^{-12}exp(355/T)
NC4H10+OH → 0.07×AnBmP	1.36×10^{-12}exp(190/T)$^{-2}$
APINEN+NO3 → 0.28×BiBmP + 0.27×BiA1D	1.19×10^{-12}exp(490/T)
BPINEN+NO3 → 0.28×BiBmP + 0.27×BiA1D	1.19×10^{-12}exp(490/T)
OCINEN+NO3 → 0.28×BiBmP + 0.27×BiA1D	1.19×10^{-12}exp(490/T)
TERPEN+NO3 → 0.28×BiBmP + 0.27×BiA1D	1.19×10^{-12}exp(490/T)
HUMULE+NO3 → 0.28×BiBmP + 0.27×BiA1D	1.19×10^{-12}exp(490/T)
LIMONE+NO3 → 0.28×BiBmP + 0.27×BiA1D	1.19×10^{-12}exp(490/T)
APINEN+OH → 0.18×BiBmP + 0.41×BiA1D	1.19×3.9^{-11}exp(400/T)
BPINEN+OH → 0.18×BiBmP + 0.41×BiA1D	1.19×3.9^{-11}exp(400/T)
OCIMEN+OH → 0.18×BiBmP + 0.41×BiA1D	1.19×3.9^{-11}exp(400/T)
TERPEN+OH → 0.18×BiBmP + 0.41×BiA1D	1.19×3.9^{-11}exp(400/T)
HUMULE+OH → 0.18×BiBmP + 0.41×BiA1D	1.19×3.9^{-11}exp(400/T)
LIMONE+OH → 0.18×BiBmP + 0.41×BiA1D	1.19×3.9^{-11}exp(400/T)
APINEN+O3 → 0.24×BiA1D	1.19×1.51^{-15}exp(-732/T)
BPINEN+O3 → 0.24×BiA1D	1.19×1.51^{-15}exp(-732/T)
OCIMEN+O3 → 0.24×BiA1D	1.19×1.51^{-15}exp(-732/T)
TERPEN+O3 → 0.24×BiA1D	1.19×1.51^{-15}exp(-732/T)
HUMULE+O3 → 0.24×BiA1D	1.19×1.51^{-15}exp(-732/T)
LIMONE+O3 → 0.24×BiA1D	1.19×1.51^{-15}exp(-732/T)
C5H8+OH → 0.232×ISOPA1	2.55×10^{-11}exp(410/T)

Table G.8: *Gas phase chemical scheme for SOA formation in CHIMERE with medium scheme.*

G.9 The SOA complex scheme with single-step oxidation

Reactions	kinetic rates ($molec.cm^{-3}.s^{-1}$)
TOL+OH → 0.004×AnA0D + 0.001×AnA1D + 0.084×AnBmP + 0.013×AnBlP	1.81×10^{-12}exp(355/T)
TMB+OH → 0.002×AnA0D + 0.002× AnA1D + 0.001×AnA2D + 0.088×AnBmP + 0.006×AnBlP	9.80×10^{-9}/T
NC4H10+OH → 0.07×AnBmP	1.36×10^{-12}exp(190/T)$^{-2}$
APINEN+OH → 0.30×BiA0D + 0.17×BiA1D + 0.10×BiA2D	1.21×10^{-11}exp(444/T)
APINEN+O3 → 0.18×BiA0D + 0.16×BiA1D + 0.05×BiA2D	1.01×10^{-15}exp (-732/T)
APINEN+NO3 → 0.80×BiBmP	1.19×10^{-12}exp(490/T)
BPINEN+OH → 0.07×BiA0D + 0.08×BiA1D + 0.06×BiA2D	2.38×10^{-11}exp(357/T)
BPINEN+O3 → 0.09×BiA0D + 0.13×BiA1D + 0.04×BiA2D	1.50×10^{-17}
BPINEN+NO3 → 0.80×BiBmP	2.51×10^{-12}
LIMONE+OH → 0.20×BiA0D + 0.25×BiA1D + 0.005×BiA2D	1.71×10^{-10}
LIMONE+O3 → 0.09×BiA0D + 0.10×BiA1D	2×10^{-16}
OCIMEN+OH → 0.70×BiA0D + 0.075×BiA1D	$5.10\times^{-8}$/T
OCIMEN+O3 → 0.50×BiA0D + 0.055×BiA1D	7.50×10^{-14}/T
OCIMEN+NO3 → 0.70×BiA0D + 0.075×BiA1D	4.30×10^{-9}/T
HUMULE+OH → 0.74×BiBmP + 0.26×BiBlP	2.93×10^{-10}
C5H8+OH → 0.232×ISOPA1 + 0.0288×ISOPA2	2.55×10^{-11}exp(410/T)

Table G.9: *Gas phase chemical scheme for SOA formation in CHIMERE. The surrogate SOA compounds consist of six hydrophilic species that include an anthropogenic nondissociative species (AnA0D), an anthropogenic once-dissociative species (AnA1D), an anthropogenic twice-dissociative species (AnA2D), a biogenic non dissociative species (BiA0D), a biogenic once-dissociative species (BiA1D) and a biogenic twice-dissociative species (BiA2D), three hydrophobic species that include an anthropogenic species with moderate saturation vapor pressure (AnBmP), an anthropogenic species with low saturation vapor pressure (AnBlP) and a biogenic species with moderate saturation vapor pressure (BiBmP), and two surrogate compounds for the isoprene oxidation products.*

G.10 Reaction list of the VBS scheme with functionalization process

Reactions	kinetic rates ($molec.cm^{-3}.s^{-1}$)
BIOGENIC LOW NOX	
APINEN+NO3->0.1073*BSOA1+0.0918*BSOA2+0.3587*BSOA3+0.6075*BSOA4	k(T)=Aexp(-B/T),A=1.19e-12,B=-490
BPINEN+NO3->0.1073*BSOA1+0.0918*BSOA2+0.3587*BSOA3+0.6075*BSOA4	k=2.51e-12
OCIMEN+NO3->0.1073*BSOA1+0.0918*BSOA2+0.3587*BSOA3+0.6075*BSOA4	k(1/T)=A/T,A=4.30e-9
APINEN+OH->0.1073*BSOA1+0.0918*BSOA2+0.3587*BSOA3+0.6075*BSOA4	k(T)=Aexp(-B/T),A=1.21e-11,B=-444
BPINEN+OH->0.1073*BSOA1+0.0918*BSOA2+0.3587*BSOA3+0.6075*BSOA4	k(T)=Aexp(-B/T),A=2.38e-11,B=-357
OCIMEN+OH->0.1073*BSOA1+0.0918*BSOA2+0.3587*BSOA3+0.6075*BSOA4	k(1/T)=A/T,A=5.1e-8
HUMULE+OH->0.0750*BSOA1+0.1500*BSOA2+0.7500*BSOA3+0.9000*BSOA4	k=2.93e-10
C5H8+OH->0.0090*BSOA1+0.0300*BSOA2+0.0150*BSOA3	k(T)=Aexp(-B/T),A=2.55e-11,B=-410
APINEN+O3->0.1073*BSOA1+0.0918*BSOA2+0.3587*BSOA3+0.6075*BSOA4	k(T)=Aexp(-B/T),A=1.01e-15,B=732
BPINEN+O3->0.1073*BSOA1+0.0918*BSOA2+0.3587*BSOA3+0.6075*BSOA4	k=1.5e-17
OCIMEN+O3->0.1073*BSOA1+0.0918*BSOA2+0.3587*BSOA3+0.6075*BSOA4	k(1/T)=A/T,A=7.5e-14
LIMONE+O3->0.1073*BSOA1+0.0918*BSOA2+0.3587*BSOA3+0.6075*BSOA4	k=2e-16
BIOGENIC HIGH NOX	
APINEN+NO3->0.0120*BSOA1+0.1215*BSOA2+0.2010*BSOA3+0.5070*BSOA4	k(T)=Aexp(-B/T),A=1.19e-12,B=-490
BPINEN+NO3->0.0120*BSOA1+0.1215*BSOA2+0.2010*BSOA3+0.5070*BSOA4	k=2.51e-12
OCIMEN+NO3->0.0120*BSOA1+0.1215*BSOA2+0.2010*BSOA3+0.5070*BSOA4	k(1/T)=A/T,A=4.30e-9
APINEN+OH->0.0120*BSOA1+0.1215*BSOA2+0.2010*BSOA3+0.5070*BSOA4	k(T)=Aexp(-B/T),A=1.21e-11,B=-444
BPINEN+OH->0.0120*BSOA1+0.1215*BSOA2+0.2010*BSOA3+0.5070*BSOA4	k(T)=Aexp(-B/T),A=2.38e-11,B=-357
OCIMEN+OH->0.0120*BSOA1+0.1215*BSOA2+0.2010*BSOA3+0.5070*BSOA4	k(1/T)=A/T,A=5.1e-8
HUMULE+OH->0.0750*BSOA1+0.1500*BSOA2+0.7500*BSOA3+0.9000*BSOA4	k=2.93e-10
C5H8+OH->0.0003*BSOA1+0.0225*BSOA2+0.0150*BSOA3	k(T)=Aexp(-B/T),A=2.55e-11,B=-410
APINEN+O3->0.012*BSOA1+0.1215*BSOA2+0.2010*BSOA3+0.5070*BSOA4	k(T)=Aexp(-B/T),A=1.01e-15,B=732
BPINEN+O3->0.012*BSOA1+0.1215*BSOA2+0.2010*BSOA3+0.5070*BSOA4	k=1.5e-17
OCIMEN+O3->0.012*BSOA1+0.1215*BSOA2+0.2010*BSOA3+0.5070*BSOA4	k(1/T)=A/T,A=7.5e-14
LIMONE+O3->0.012*BSOA1+0.1215*BSOA2+0.2010*BSOA3+0.5070*BSOA4	k=2e-16
ANTHROPOGENIC LOW NOX	
ARO1+OH->0.0750*ASOA1+0.2250*ASOA2+0.3750*ASOA3+0.5250*ASOA4	k=6.15e-12
ARO2+OH->0.0750*ASOA1+0.3000*ASOA2+0.3750*ASOA3+0.5250*ASOA4	k=2.57e-11
ALK4+OH->0.0750*ASOA2	k(T)=Aexp(-B/T),A=3.75e-12,B=-44
ALK5+OH->0.3000*ASOA2	k(T)=Aexp(-B/T),A=2.70e-12,B=-374
OLE1+OH->0.0045*ASOA1+0.0090*ASOA2+0.0600*ASOA3+0.2250*ASOA4	k(T)=Aexp(-B/T),A=6.18e-12,B=-501
OLE2+OH->0.0225*ASOA1+0.0435*ASOA2+0.1290*ASOA3+0.3750*ASOA4	k(T)=Aexp(-B/T),A=1.26e-11,B=-488
ANTHROPOGENIC HIGH NOX	
ARO1+OH->0.0030*ASOA1+0.1650*ASOA2+0.3000*ASOA3+0.4350*ASOA4	k=6.15e-12
ARO2+OH->0.0015*ASOA1+0.1950*ASOA2+0.3000*ASOA3+0.4350*ASOA4	k=2.57e-11
ALK4+OH->0.0375*ASOA2	k(T)=Aexp(-B/T),A=3.75e-12,B=-44
ALK5+OH->0.1500*ASOA2	k(T)=Aexp(-B/T),A=2.70e-12,B=-374
OLE1+OH->0.0008*ASOA1+0.0045*ASOA2+0.0375*ASOA3+0.1500*ASOA4	k(T)=Aexp(-B/T),A=6.18e-12,B=-501
OLE2+OH->0.0030*ASOA1+0.0225*ASOA2+0.0825*ASOA3+0.2700*ASOA4	k(T)=Aexp(-B/T),A=1.26e-11,B=-488
AGING BSOA	
BSOA4+OH->1.075*BSOA3	k=1e-11
BSOA3+OH->1.075*BSOA2	k=1e-11
BSOA2+OH->1.075*BSOA1	k=1e-11
AGING ASOA	
ASOA4+OH->1.075*ASOA3	k=1e-11
ASOA3+OH->1.075*ASOA2	k=1e-11
ASOA2+OH->1.075*ASOA1	k=1e-11

Table G.10: *Gas phase chemical scheme for SOA formation with VBS sheme.*

G.11 Reaction list of the VBS scheme with functionalization and fragmentation process

Reactions	kinetic rates ($molec.cm^{-3}.s^{-1}$)
BIOGENIC LOW NOX	
APINEN+NO3->0.1073*BSOA1+0.0918*BSOA2+0.3587*BSOA3+0.6075*BSOA4	k(T)=Aexp(-B/T),A=1.19e-12,B=-490
BPINEN+NO3->0.1073*BSOA1+0.0918*BSOA2+0.3587*BSOA3+0.6075*BSOA4	k=2.51e-12
OCIMEN+NO3->0.1073*BSOA1+0.0918*BSOA2+0.3587*BSOA3+0.6075*BSOA4	k(1/T)=A/T,A=4.30e-9
APINEN+OH->0.1073*BSOA1+0.0918*BSOA2+0.3587*BSOA3+0.6075*BSOA4	k(T)=Aexp(-B/T),A=1.21e-11,B=-444
BPINEN+OH->0.1073*BSOA1+0.0918*BSOA2+0.3587*BSOA3+0.6075*BSOA4	k(T)=Aexp(-B/T),A=2.38e-11,B=-357
OCIMEN+OH->0.1073*BSOA1+0.0918*BSOA2+0.3587*BSOA3+0.6075*BSOA4	k(1/T)=A/T,A=5.1e-8
HUMULE+OH->0.0750*BSOA1+0.1500*BSOA2+0.7500*BSOA3+0.9000*BSOA4	k=2.93e-10
C5H8+OH->0.0090*BSOA1+0.0300*BSOA2+0.0150*BSOA3	k(T)=Aexp(-B/T),A=2.55e-11,B=-410
APINEN+O3->0.1073*BSOA1+0.0918*BSOA2+0.3587*BSOA3+0.6075*BSOA4	k(T)=Aexp(-B/T),A=1.01e-15,B=732
BPINEN+O3->0.1073*BSOA1+0.0918*BSOA2+0.3587*BSOA3+0.6075*BSOA4	k=1.5e-17
OCIMEN+O3->0.1073*BSOA1+0.0918*BSOA2+0.3587*BSOA3+0.6075*BSOA4	k(1/T)=A/T,A=7.5e-14
LIMONE+O3->0.1073*BSOA1+0.0918*BSOA2+0.3587*BSOA3+0.6075*BSOA4	k=2e-16
BIOGENIC HIGH NOX	
APINEN+NO3->0.0120*BSOA1+0.1215*BSOA2+0.2010*BSOA3+0.5070*BSOA4	k(T)=Aexp(-B/T),A=1.19e-12,B=-490
BPINEN+NO3->0.0120*BSOA1+0.1215*BSOA2+0.2010*BSOA3+0.5070*BSOA4	k=2.51e-12
OCIMEN+NO3->0.0120*BSOA1+0.1215*BSOA2+0.2010*BSOA3+0.5070*BSOA4	k(1/T)=A/T,A=4.30e-9
APINEN+OH->0.0120*BSOA1+0.1215*BSOA2+0.2010*BSOA3+0.5070*BSOA4	k(T)=Aexp(-B/T),A=1.21e-11,B=-444
BPINEN+OH->0.0120*BSOA1+0.1215*BSOA2+0.2010*BSOA3+0.5070*BSOA4	k(T)=Aexp(-B/T),A=2.38e-11,B=-357
OCIMEN+OH->0.0120*BSOA1+0.1215*BSOA2+0.2010*BSOA3+0.5070*BSOA4	k(1/T)=A/T,A=5.1e-8
HUMULE+OH->0.0750*BSOA1+0.1500*BSOA2+0.7500*BSOA3+0.9000*BSOA4	k=2.93e-10
C5H8+OH->0.0003*BSOA1+0.0225*BSOA2+0.0150*BSOA3	k(T)=Aexp(-B/T),A=2.55e-11,B=-410
APINEN+O3->0.012*BSOA1+0.1215*BSOA2+0.2010*BSOA3+0.5070*BSOA4	k(T)=Aexp(-B/T),A=1.01e-15,B=732
BPINEN+O3->0.012*BSOA1+0.1215*BSOA2+0.2010*BSOA3+0.5070*BSOA4	k=1.5e-17
OCIMEN+O3->0.012*BSOA1+0.1215*BSOA2+0.2010*BSOA3+0.5070*BSOA4	k(1/T)=A/T,A=7.5e-14
LIMONE+O3->0.012*BSOA1+0.1215*BSOA2+0.2010*BSOA3+0.5070*BSOA4	k=2e-16
ANTHROPOGENIC LOW NOX	
ARO1+OH->0.0750*ASOA1+0.2250*ASOA2+0.3750*ASOA3+0.5250*ASOA4	k=6.15e-12
ARO2+OH->0.0750*ASOA1+0.3000*ASOA2+0.3750*ASOA3+0.5250*ASOA4	k=2.57e-11
ALK4+OH->0.0750*ASOA2	k(T)=Aexp(-B/T),A=3.75e-12,B=-44
ALK5+OH->0.3000*ASOA2	k(T)=Aexp(-B/T),A=2.70e-12,B=-374
OLE1+OH->0.0045*ASOA1+0.0090*ASOA2+0.0600*ASOA3+0.2250*ASOA4	k(T)=Aexp(-B/T),A=6.18e-12,B=-501
OLE2+OH->0.0225*ASOA1+0.0435*ASOA2+0.1290*ASOA3+0.3750*ASOA4	k(T)=Aexp(-B/T),A=1.26e-11,B=-488
ANTHROPOGENIC HIGH NOX	
ARO1+OH->0.0030*ASOA1+0.1650*ASOA2+0.3000*ASOA3+0.4350*ASOA4	k=6.15e-12
ARO2+OH->0.0015*ASOA1+0.1950*ASOA2+0.3000*ASOA3+0.4350*ASOA4	k=2.57e-11
ALK4+OH->0.0375*ASOA2	k(T)=Aexp(-B/T),A=3.75e-12,B=-44
ALK5+OH->0.1500*ASOA2	k(T)=Aexp(-B/T),A=2.70e-12,B=-374
OLE1+OH->0.0008*ASOA1+0.0045*ASOA2+0.0375*ASOA3+0.1500*ASOA4	k(T)=Aexp(-B/T),A=6.18e-12,B=-501
OLE2+OH->0.0030*ASOA1+0.0225*ASOA2+0.0825*ASOA3+0.2700*ASOA4	k(T)=Aexp(-B/T),A=1.26e-11,B=-488
AGING BSOA	
BSOA4+OH->0.16125*BSOA3+0.75*BSOA4	k=1e-11
BSOA3+OH->0.16125*BSOA2+0.75*BSOA4	k=1e-11
BSOA2+OH->0.16125*BSOA1+0.75*BSOA4	k=1e-11
BSOA1+OH->0.16125*BSOA1+0.75*BSOA4	k=1e-11
AGING ASOA	
ASOA4+OH->0.16125*ASOA3+0.75*ASOA4	k=1e-11
ASOA3+OH->0.16125*ASOA2+0.75*ASOA4	k=1e-11
ASOA2+OH->0.16125*ASOA1+0.75*ASOA4	k=1e-11
ASOA1+OH->0.16125*ASOA1+0.75*ASOA4	k=1e-11
Production of non volatile secondary aerosol from organic aerosols	
ASOA1->ANVSOA	k=3e-4
ASOA2->ANVSOA	k=3e-4
ASOA3->ANVSOA	k=3e-4
ASOA4->ANVSOA	k=3e-4
BSOA1->BNVSOA	k=3e-4
BSOA2->BNVSOA	k=3e-4
BSOA3->BNVSOA	k=3e-4
BSOA4->BNVSOA	k=3e-4

Table G.11: *Gas phase chemical scheme for SOA formation with VBS scheme with functionalization and fragmentation process.*