Vertical cross-section of emissions and transport of Chernozemic dust in March 2007
(Bessagnet et al., 2008, JGR)

• This documentation and the model are freely available at the following internet adresse:
  http://www.lmd.polytechnique.fr/chimere/

• CHIMERE is distributed under the GNU General Public License
• Copyright (C) 2007 Institut Pierre-Simon Laplace (C.N.R.S.), INERIS, LISA (C.N.R.S.)
• For questions, send an e-mail to chimere@lmd.polytechnique.fr
• Last update of this documentation: June 4, 2009
Contents

1 Model overview and last changes 11
  1.1 Documentation overview 11
  1.2 Last changes: version chimere2008 11
  1.3 Short Description of the model 12
    1.3.1 Purpose 12
    1.3.2 Input data required 12
    1.3.3 OS, performance and software 13
    1.3.4 The physics and numerics in short 13
    1.3.5 What can be done with 14
  1.4 History: from previous version to V200606 14
    1.4.1 Parallelization 14
    1.4.2 Toolbox 15
    1.4.3 Scripts 15
    1.4.4 makefiles changes 15

2 First CHIMERE simulation 17
  2.1 Main sources installation 17
  2.2 The first run tutorial 18
    2.2.1 Download required data and programs 18
    2.2.2 Example of a directories tree 19
    2.2.3 Simulation configuration 19
    2.2.4 Specific case of single node environment 21
  2.3 Description of the top-calling script and options 21
  2.4 Direct run outputs 28
    2.4.1 The output on screen 28
    2.4.2 The output files 28
  2.5 Post-processing the results 29
    2.5.1 Dump data in ASCII form 30
    2.5.2 Use data for further numerical processing 30
    2.5.3 Plot data 30
    2.5.4 Interactively display data 32

3 The CHIMERE model source code 37
  3.1 The CHIMERE time integration 38
  3.2 Directories 39
  3.3 Model Files 40
    3.3.1 Scripts 40
    3.3.2 Makefiles 41
    3.3.3 Meteo interface - 1st stage 41
CONTENTS

3.3.4 Source directories ......................................................... 42
3.3.5 Dynamic files generated at runtime ..................................... 44
3.4 Routine calls sequence ...................................................... 44
3.5 Parallelism ................................................................. 45
  3.5.1 Domains division ....................................................... 45
  3.5.2 Recommendations ..................................................... 47
3.6 Core model input and output files ......................................... 48
  3.6.1 Input files ............................................................... 48
  3.6.2 Output files ............................................................ 49
3.7 Running real cases ......................................................... 49
3.8 How to add a variable in the code? ........................................ 50
  3.8.1 From the master to workers ........................................... 50
  3.8.2 From workers to the master .......................................... 51

4 [domains]: Domains and landuse ............................................. 53
  4.1 The domains/domainlist.nml file ........................................ 53
  4.2 The main scripts and programs ......................................... 53
  4.3 Details on output files .................................................. 55
    4.3.1 Horizontal domain COORD ....................................... 55
    4.3.2 Vertical grid VCOORD ........................................... 55
    4.3.3 Land-use types ................................................... 56

5 [chemprep]: The chemical preprocessor .................................. 57
  5.1 Options available in the main script chimere.sh ...................... 57
  5.2 The first script make-chemgeom.sh .................................... 58
  5.3 The second script chemprep.univ.sh ................................... 58
  5.4 Manual changes of the mechanism ...................................... 58
  5.5 Syntax of the REACTIONS file ......................................... 60
  5.6 Syntax of the PHOTORATES file ....................................... 60

6 Emissions and depositions ................................................ 63
  6.1 Emissions .................................................................. 63
    6.1.1 Anthropogenic surface emissions .................................. 63
    6.1.2 Biogenic emissions ................................................. 67
    6.1.3 Tracer emissions .................................................... 68
    6.1.4 Natural emissions .................................................. 69
  6.2 Dry deposition and resuspension ....................................... 72
    6.2.1 For gases : ........................................................... 72
    6.2.2 For aerosols: ........................................................ 73
  6.3 Wet scavenging ................................................................ 74
    6.3.1 Resuspension of particulate matter ............................... 74

7 Initial and boundary conditions ............................................ 77
  7.1 Chemical boundary conditions .......................................... 77
    7.1.1 Internal boundary conditions ..................................... 77
    7.1.2 Gas boundary conditions from global models .................. 77
    7.1.3 Aerosols boundary conditions from global models .......... 78
    7.1.4 Boundary conditions interface .................................... 79
  7.2 Initial concentrations .................................................... 79
    7.2.1 Initial concentrations species .................................... 79
7.2.2 Initial concentrations interface ........................................ 79

8 Meteorology 81
  8.1 Meteorological input data .............................................. 81
  8.1.1 First stage: the "interf" programs ................................ 81
  8.1.2 Second stage: the diagmet diagnostic model ..................... 85
  8.2 Transport and mixing .................................................. 87
    8.2.1 Horizontal transport ............................................ 87
    8.2.2 Optional correction of surface wind speed to account for urban effects ............................................ 87
    8.2.3 Vertical transport .............................................. 88
    8.2.4 Turbulent mixing ................................................. 88
    8.2.5 Deep convection .................................................. 89
    8.2.6 Parameterization ................................................ 90
    8.2.7 Pre-processing in diagmet ....................................... 90
    8.2.8 Implementation in CHIMERE ..................................... 91

9 Chemistry 93
  9.1 The TWO-STEP time numerical solver ............................... 93
  9.2 Gas-phase chemistry .................................................. 94
    9.2.1 Chemical mechanisms (gas phase) ................................ 94
    9.2.2 Species list of the reduced MELCHIOR2 gas-phase chemical mechanism ........................................... 95
  9.3 Aerosol processes ..................................................... 99
    9.3.1 Overview .......................................................... 99
    9.3.2 Composition and mathematical representation of aerosols .................................................. 100
    9.3.3 Coagulation ....................................................... 101
    9.3.4 Absorption ........................................................ 101
    9.3.5 Nucleation ........................................................ 102
    9.3.6 ISORROPIA ......................................................... 102
    9.3.7 Multiphase chemistry ............................................ 103
    9.3.8 Secondary organic aerosol chemistry ......................... 104
  9.4 Impact of clouds ...................................................... 105
    9.4.1 Impact on photolysis ............................................. 105
    9.4.2 Wet scavenging .................................................. 107

Bibliography 107

A References using CHIMERE 115

B Some scripts used in the CHIMERE suite 119
  B.1 The scripts/make-chemgeom.sh for preprocessing of domain and chemistry input data ...................... 119
  B.2 The domains/makeCOORDdomains script for horizontal grid definition ........................................... 122
  B.3 The util/define_geom script for vertical grid definition ............................................................... 123
  B.4 The chemprep/chemprep.univ.sh for chemistry preprocessing ...................................................... 124
  B.5 The ./install.sh for Chimere download and installation ................................................................. 132

C How To install netCDF under GNU/Linux 137
  C.1 Background ............................................................. 137
  C.2 Download ............................................................... 137
  C.3 Configure netCDF ....................................................... 137
    C.3.1 ifort ............................................................... 138
    C.3.2 g95 ................................................................. 138
List of Figures

1.1 The simulation of the 2003 heat wave: surface ozone concentrations (µg.m⁻³) .......................... 13
2.1 The directory tree after Chimere installation using install.sh script ................................. 19
2.2 Exachim screenshot .............................................................................................................. 33
3.1 Schematic structure of the CHIMERE input data ............................................................... 37
3.2 General CHIMERE structure for time integration of all processes ..................................... 38
3.3 Calculation of the number of integration steps per hour to respect the CFL over a complete 120 hours simulation. ........................................................................................................ 39
3.4 Schematic general structure of the scripts, data files of CHIMERE .................................... 41
3.5 Detailed list of CHIMERE routines and time loops ............................................................. 45
3.6 Time processing in CHIMERE ............................................................................................ 46
4.1 Examples of horizontal domains. The dots represent the MM5 EUR1 grid cells centers and the rectangle represents the boundaries of the CHIMERE CONT3 domain. .......................................................... 54
6.1 Flowchart of the several steps to produce CHIMERE compliant anthropogenic surface emissions fluxes ........................................................................................................................................ 63
6.2 Injection height of the tracer ................................................................................................. 69
8.1 How to prepare MM5 coordinates files for CHIMERE, from the original MM5 terrain file ........ 84
8.2 [left] Temperatures changes due to shallow and deep convection and [right] Schematic view of a convective cloud ....................................................................................................................................... 89
8.3 Entrainment and detrainment fluxes in the updrafts and downdrafts. .................................... 90
8.4 Convective atmospheric column and its environment ............................................................. 91
9.1 Principle of ‘operator-splitting’ versus Chimere integration ................................................... 93
# List of Tables

2.1 Useful URL adresses for the development and the use of the CHIMERE model and its modules .......................... 18  
2.2 The parameters model outputs on file. ........................................................................................................... 29  
2.3 The predefined color tables to be used with the GMT software for concentrations maps .......................... 32  
2.4 Contents of the "Slice" mode additional windows depending on the mode and mouse button pressed ........................................ 34  

4.1 GLCF land cover types ...................................................................................................................................... 55  
4.2 Landuse categories used in CHIMERE ............................................................................................................ 56  

8.1 Mandatory and optional variables in exdomout.nc meteo file. The parameters quoted with $^1$ are mandatory only for the aerosol option ........................................................................................................ 82  

9.2 List of aerosol species. (*): ions, molecules, crystals .......................................................................................... 100  
9.3 Characteristics of the look-up table used for the calculation of the thermodynamic equilibrium with ISORROPIA. ........................................................................................................................................ 103  
9.4 Relative humidity at deliquescent points for the species used by ISORROPIA in the system sulfate-nitrate-ammonium-water. .................................................................................................................. 103  
9.5 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. .................................................................................................................. 106
Acknowledgments

The CHIMERE developers acknowledge:

- Gabriele Curci for the biogenic emissions interface
- 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,
- the EMEP-MSCWEST team (http://www.emep.int) for anthropogenic emissions database,
- the EMEP-MSCEAST team for benzo(a)pyrene, benzo(k)fluoranthene and benzo(b)fluoranthene data (http://www.msceast.org).
- José Antonio Souto for section 3.8.2 on how to add a new variable in the code.

chimere@lmd.polytechnique.fr

1Bertrand BESSAGNET, Dmitry KHVOROSTYANOV, Laurent MENUT, Jean-Louis MONGE and Robert VAUTARD. You can contact Jean-Louis MONGE only for CHIMERE expertise for numerical purpose at the following address: jean-louis.monge@free.fr
Chapter 1

Model overview and last changes

1.1 Documentation overview

This documentation describes the last version of the CHIMERE chemistry-transport model named chimere2008. The model and associated source codes are freely available on a web site:

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

- The chapter 2 explains how to install the model from scratch and perform a first run using the prepared test case 'Heat Wave summer 2003 over Europe'.
- The chapter 3 (p.37) presents the model sources documentation including the pre- and post-processing interfaces.
- The chapter 4 (p.53) and 5 (p.57), explain how the user may define its own model geometry (domain and landuse) and its own configuration for chemistry (gas and/or aerosols, tracers etc.)
- The chapters 6 (p.63), 7 (p.77), 8 (p.81), 9 (p.93) describe the treatment (processes and numerics) of the emissions, boundary conditions, meteorology and chemistry.
- The Appendix present "how-to" files explaining the way to install libraries such as netCDF and LAM/MPI, and the format of all files used by the model.
- A list of published papers about studies using CHIMERE is displayed in the Chapter A, p.115.

For further questions, please send an e-mail to the developers at chimere@lmd.polytechnique.fr.

1.2 Last changes: version chimere2008

This version chimere2008 contains numerous improved programs in order to have an easier model to use. Some important physical and chemical processes were also added in the model.

- New functionality:
  - Due to integration of the MEGAN model biogenic emission potentials data and emission rates parameterizations, Chimere can now be used outside Europe if the user provides anthropogenic emissions in the region of interest
  - Automatic preparation of the file LANDUSE_<domain> from the original GLCF global land use database for the desired Chimere domain
  - Meteorological interface to the WRF model prepwrf has been added
  - Possibility to use aerosol boundary and initial conditions from the LMDz model output
  - A new mode of exachim to display various data slices by a simple mouse click

- Process changes:
Updated Secondary Organic Aerosols (SOA) chemistry: the full MEGAN SOA scheme is now used

- Computational changes:
  - New format of the emissions interface emiSURF and prepemis. The emission files are now separated by individual species, which makes the prepemis interface simpler and easier to modify. If a simulation spans two consequent months, emissions for each day are now taken from a corresponding month rather than from the first month for all days. One can also choose emission types to use independently: either surface, point, or fire emissions. Hourly changes of fire emissions are now allowed. If emissions are not available for a given species, they are set to zero, and the simulation continues
  - Reorganized boundary and initial conditions interface for more convenience and code simplicity. User can now choose a global database for each species
  - Revised vertical interpolations in the boundary and initial conditions interface
  - Running Open MPI or LAM depending on user system settings. If both realizations are found, preference is given to Open MPI
  - Simulation parameters are now organized in the form of an array in the top calling script chimere.sh. This allows more convenient parameter specification, especially for simulations for several, possibly nested, model domains
  - The script chimere-run.sh has been split into parts corresponding to Chimere components: compilation, meteorological interfaces, emissions, initial and boundary conditions, and Chimere core. It is now possible to run individually one or more of these components.

1.3 Short Description of the model

1.3.1 Purpose

The CHIMERE multi-scale model is primarily designed to produce daily forecasts of ozone, aerosols and other pollutants and make long-term simulations for emission control scenarios. CHIMERE runs over a range of spatial scales from the regional scale (several thousand kilometers) to the urban scale (100-200 Km) with resolutions from 1-2 Km to 100 Km. On CHIMERE server, documentation and source codes are proposed for the complete multi-scale model. However most data are valid only for Europe and should be revisited for applications on other continents.

CHIMERE proposes many different options for simulations which make it also a powerful research tool for testing parameterizations, hypotheses. Its use is relatively simple so long as input data is correctly provided. It can run with several vertical resolutions, and with a wide range of complexity. It can run with several chemical mechanisms, simplified or more complete, with or without aerosols.

The Figure 1.1 shows the example of the ozone simulation for August 8th, 2003 at 14h UT, during the exceptional heat wave that occurred over Western Europe. The gray dots show monitoring stations where the 180 $\mu$g.m$^{-3}$ exceedance is reached at this exact time.

1.3.2 Input data required

CHIMERE is a portable model which has been adapted to various types of input data. It requires meteorological data, boundary conditions, land-use information, emissions. In all cases but emissions, a default solution is proposed for the user. Refinements on input data have to be made by the user.

- Meteorology: Meteo data is not provided on the server, but a CHIMERE interface for the MM5 mesoscale model is proposed. MM5 (free software\(^1\)) can be used combined with AVN/NCEP (as forcing) public weather forecasts.

\(^1\)http://box.mmm.ucar.edu/mm5
Boundary conditions: A set of boundary conditions from the MOZART and/or LMDz-INCA models is proposed as a default solution. Data are kindly provided by Max-Planck Institut, Hamburg, thanks to M. Schultz, G. Brasseur, C. Granier and D. Niehl, and IPSL/LSCE, thanks to Sophie Szopa and Didier Hauglustaine. This allows tropospheric simulations below 200 hPa. For aerosols, a set of boundary conditions is proposed based on GOCART global simulations, thanks to Mian Chin (NASA).

Land-use: CHIMERE needs input landuse information, as well as biogenic emission potentials, based on land cover. The default proposed land use is the GLCF data base\(^2\). The data for deriving biogenic emission potentials only cover Europe. It needs to be completed for other continents.

Emissions: Emissions are usually coming from local, nonpublic databases. Here, for model testing, emissions are proposed for Western Europe at a resolution of 1/2 degree, and for the region (150Kmx150Km) around Paris.

1.3.3 OS, performance and software

The model has only been tested on PCs under GNU/Linux. It should also easily work on SUN and other UNIX systems with minor changes. Most changes should be due to the unformatted binary data interfaces. Please report to us all problems. A typical time for simulation on a single-processor PC, using the European version with a 65x33x8 grid, is 5 minutes for one simulated day with the gas-phase version, and 30 minutes with the aerosol version. It can then easily simulate entire seasons or years. A minimal setting of 1 Gb RAM is necessary. Software required is a Fortran 95 compiler (g95 is really great and free!) or ifort, and an MPI-1 compatible library. Simple graphical interfaces are available using either the GMT\(^3\) or the GrADS\(^4\) free software (to be installed).

1.3.4 The physics and numerics in short

- The chemical mechanism (MELCHIOR) is adapted from the original EMEP mechanism.
- Photolytic rates are attenuated using liquid water or relative humidity

---

\(^2\)http://glcf.umiacs.umd.edu
\(^3\)http://gmt.soest.hawaii.edu/
\(^4\)http://grads.iges.org/grads/grads.html
Boundary layer turbulence is represented as a diffusion (Troen and Mahrt, 1986, BLM)
Vertical wind is diagnosed through a bottom-up mass balance scheme.
Dry deposition is as in Wesely (1989). Wet deposition is included.
Six aerosol sizes represented as "bins" in the model.
Aerosol thermodynamic equilibrium is achieved using the ISORROPIA model.
Several aqueous-phase reactions considered
Secondary organic aerosols formation considered
Advection is performed either by a first upwind scheme, the Van Leer scheme or by the PPM (Piecewise Parabolic Method) 3d order scheme for slow species.
The numerical time solver is the TWOSTEP method.

1.3.5 What can be done with
- Use the model for simulating episodes or long-term periods.
- Use the model for operational forecasting.
- Use the model for studying emission scenarios.
- Sensitivity tests: CHIMERE is an easy-to-modify code which allows for testing of various processes and making sensitivity tests. For instance the influence of new chemical reactions can be tested.
- Model development: It is the wish of the authors that CHIMERE be an open model, hence all model modifications in the direction of larger realism are welcome. There are a number of such improvements still to be done.

1.4 History: from previous version to V200606

1.4.1 Parallelization
The previous version V200606 represented a major breakthrough in the model computational speed, since it has been parallelized and can now run on distributed memory clusters and shared memory machines, ranging from the local home-made cluster of PCs, to large supercomputers. However, there are no changes neither in the scientific core of the model, nor in the I/O file formats.

The rationale of the parallelization was:
- the increasing computation time of the aerosol version and of nextcoming, more sophisticated versions,
- the user requirements for higher resolution domains,
- the requirements of air pollution forecast agencies, in terms of forecast computation time.

CHIMERE uses the distributed memory scheme, and an MPI message passing library. It has been tested with Open MPI and LAM/MPI, but should work, with minor changes in the scripts, with MPICH or other MPI compatible parallel environments.
There is one master program, which calls several worker subroutines, each worker been in charge of the processing of a subdomain. Subdomains are the result of a Cartesian division of the main geographical domain by the number of processors.
The master program is in charge of the initializations and file I/O. The workers are in charge of the scientific processing and inter-subdomains boundary conditions updates.

As described above, CHIMERE is basically an MPMD (multiple program, multiple data) application. However, MPMD is only available with MPI-2 compatible environments. To increase CHIMERE portability, its interface has been tweaked to mimic the appearance of an SPMD (single program, multiple data) application, which makes it compatible with widely spread MPI-1 parallel environments.
Despite the tremendous architectural change in the code structure, most scientific subroutines, now called by the worker main subroutine, have remained unchanged.
As far as the user interface is concerned, the only apparent modifications are:
• definition of subdomains in chimere-mm5 script: nzdoms and nmdoms parameters.
• use of an MPI library: MPI must be installed for running CHIMERE, even if it runs on only one processor with only one subdomain.

1.4.2 Toolbox

There is a new processing inside exactim: the COARDS intermediate file is no longer necessary. Instead, GrADS compliant files are built directly. A new utility, chm2grads.f90, replaces chm2coards.f90. All kinds of CHIMERE grids may be used, not only rectilinear ones. More sophisticated command line options allows for displaying only selected variables and time slots.

The udunits library is no longer used.

A new tool to transform the old CHIMERE binary files into netCDF files, chmbin2cdf.f90, has been included in the distribution.

An older tool, chim2fig, which used to be distributed with pre-netCDF versions of CHIMERE, has been upgraded to accept netCDF files.

1.4.3 Scripts

New top-level scripts have been added, to run CHIMERE on PC clusters with a PBS-like batch scheduler. Despite the fact that an interface to ECMWF meteorology is not yet provided in this version, a script to run CHIMERE on ECMWF HPCD computer is provided as an example for running CHIMERE under IBM LoadLeveler and POE environments.

1.4.4 makefiles changes

The directory structure of the src/ tree has been modified, and Makefiles have been rationalized. This allows now for using the parallel compilation capability of the GNU make utility. Also, the meteo interface is run in background, during model compilation. This reduces dramatically the compilation time on multiprocessor machines, allowing for more sophisticated optimizations. Minor bugs have been corrected in Makefile headers.
Chapter 2

First CHIMERE simulation

In the case of a user who never used the CHIMERE model before, this chapter presents how to:

- Install all softwares prerequisites, §2.1
- Perform a simulation with the pre-defined test case available on the CHIMERE web site, §2.2, with a complete description of the model top-calling script, §2.3
- See the results, §2.4, post-process the concentrations fields §2.5.

2.1 Main sources installation

The CHIMERE model has only been tested on GNU/Linux systems with LAM-MPI message passing library. However it should be working on most UNIX systems provided the following software is installed. The foreseeable changes should be related to shell, awk and make syntax, to the MPI library used, and also to unformatted binary files which may have to be converted.

The model requires:

- a Fortran 95 compiler
- GNU bash Bourne shell
- GNU awk
- GNU make
- Unidata netCDF library (free)
- Open MPI (http://www.open-mpi.org) or LAM-MPI (Indiana University) software (free)
- The Generic Mapping Tools GMT (free)
- IGES/COLA GrADS graphical software (free)

The installation process of netCDF and LAM/MPI is fully described in §C, p.137 and §D, p.143, and in files HOWTO-INSTALL-NETCDF and HOWTO-INSTALL-LAMMPI in the top level CHIMERE directory. The netCDF library must be compiled with the same compiler as CHIMERE model.

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 in account some syntactic features specific to bash. The same remark apply to the awk and make utilities.

The model has been tested with the following compilers: g95, ifort and pgf90.

GMT and GraDS are only required if the user wants to use the post-processing stages proposed in the toolbox directory, i.e. ncmait, chim2fig, and exachim/exadiff.

In the following examples, the wget utility is used for downloading, because it is a robust and powerful downloading tool. However, if wget is not installed on your system, you can obviously use your favorite browser to download files.

The Table 2.1 reports useful URL adresses for the development and the use of the CHIMERE model and its modules.
2.2 The first run tutorial

For the first CHIMERE run, we propose to simulate the heat wave observed during the summer 2003. This run was already performed with CHIMERE and analyzed. Results are presented in [Vautard et al., 2005b]. Please note they were done with a previous version of CHIMERE, and, thus, a previous version of surface emissions. The real test case with a simulation over Europe of a 5-day period starts on July 30 2003 and ends on August 3 2003, i.e. the beginning of the 2003 heat wave, with MM5 forcing. The MM5 run is forced by the 6-hourly NCEP/AVN analyses, and the "nudging" option (grid FDDA) is chosen, with coefficients of 1e-4 for temperature, wind, and 2e-5 for humidity. The model domain covers Western Europe with an approximate 36km horizontal resolution and 32 vertical levels.

2.2.1 Download required data and programs

To perform this test case, you need to download the model and some input data archives. The script install.sh is provided to automate this task. You can find it on the Chimere web site in the downloads section: [http://www.lmd.polytechnique.fr/chimere/download.php](http://www.lmd.polytechnique.fr/chimere/download.php). To install the model you need to download the script from the web site and to launch it from the directory where you want to install Chimere. The script does the following:

1. Prompts you to specify the directory where you would like to store large data files. Creates the model directory and subdirectories for the big files, emissions, output, and the meteorology

2. Downloads the CHIMERE source code and data for the test case: EMEP emissions\(^1\) preprocessed for July and August and MM5 meteorological output for the 5 days of test simulation.

\(^1\)For the use of these emissions data, please acknowledge EMEP (Yearly totals), IER (Time variations), TNO (Aerosol emissions), UK Dept of Environment (VOC speciation, Passant, 2002).
3. Downloads emission data for other months and the complete emiSURF preprocessor

4. Downloads the boundary and initial conditions\(^2\): INCA for gaseous species and GOCART for aerosols species.

5. Decompresses the downloaded archives to the appropriate directories

6. Moves the archives to the ARCH directory.

7. Generates makefile header file **Makefile.hdr** for your system type (32 or 64-bit), Fortran compiler and NetCDF library version.

The listing of the installation script **install.sh** is given in section B.5, p.132.

You can also generate a new **Makefile.hdr** at any time after Chimere installation running **install.sh** and choosing not to download/install the model.

### 2.2.2 Example of a directories tree

The directory tree after Chimere installation using **install.sh** script is shown in Figure 2.1(a, b), for the current and BIGFILES directories, respectively.

![Figure 2.1: The directory tree after Chimere installation using install.sh script](image)

### 2.2.3 Simulation configuration

Once the code is downloaded and the mandatory libraries netCDF and LAM/MPI are installed, the next step is to set up some configuration files to match your system.

CHIMERE uses configuration files that have to be tailored to user’s requirements and environment. These files reside in **chimere2008** top directory:

---
\(^2\)These data were kindly provided by colleagues, thus please acknowledge D. Hauglustaine and S. Szopa (Institut Pierre-Simon Laplace) for INCA data, M. Schultz (MPI Hamburg), C. Granier, G. Brasseur and D. Niehl for MOZART2 data and M. Chin, P. Ginoux (NASA) for GOCART data. Please contact sophie.szopa@cea.fr and mian.chin@nasa.gov for any question about INCA and GOCART, respectively.
• [makefile.hdr] is a symbolic link to a preformatted file, specific to a given compiler. It contains all the informations pertinent to your execution environment: compilers options, libraries, and so on. makefile.hdr is included by all subsequent makefiles in CHIMERE source tree, to ensure a consistent compilation.

• [chimere.sh] the top calling script of CHIMERE. It contains all the information necessary to run CHIMERE: data base files locations, domain definition, physical and numerical options. A copy of this file is given in the §2.3, p.21.

• When executed, chimere sources an execution script, scripts/chimere-MM5-run.sh, which in turn launches all CHIMERE processes. Only experienced users should try to modify this latter script.

To configure CHIMERE:

• Link Makefile.hdr to your compiler specific makefile header. For instance:
  
  ln -sf makefiles.hdr/Makefile.hdr.ifort Makefile.hdr

• Edit Makefile.hdr for :
  
  □ the location of your compiler
  □ the location of your netCDF package

• Edit chimere to :
  
  □ set up the time span of a run
  □ set the cartesian division of the domain (nzdoms and nmdoms)
  □ tell CHIMERE the location of pertinent files and directories :
    □ domain geometry
    □ species list
    □ emission files
    □ boundary conditions files
    □ meteo file
    □ working directory
    □ output directory
  □ choose model options

• The current Chimere version has been tested with Open MPI. The latter does not need to be booted, you can just launch mpirun with your executable.
However, if you are still using LAM MPI, make sure that it is booted, with the right number of processors. The command to use is:

  lamnodes

  The system should respond with a list of active nodes and the number of processors per node. The total number of declared processors should be superior or equal to nzdoms*nmdoms (see p.47).
If it does not, boot your MPI environment. With LAM/MPI, to boot the default nodes list, type:

  lamhalt
  lamboot

  A file named nodelist.example is provided here as a template if you want to boot different nodes. To use this file, first edit it according to your environment, taking care to declare at least nzdoms*nmdoms processors. Once your nodelist.example file is edited, type:

  lamhalt
  lamboot nodelist.example

  If you want to have an understanding of the impacts of the hostlist file (nodelist.example) read the recommendations in p.47
2.2.4 Specific case of single node environment

In the trivial case of a single node environment, the node list file, either `/etc/lam/lam-bhosts.def` or your own `nodelist.example` should contain only this single line:

```
localhost
```

And you should set `nzdoms=1` and `nmdoms=1` in `chimere.sh`.

In the other trivial case of a single bi-processor node, the node list file, should contain only this single line:

```
localhost cpu=2
```

And you should set `nzdoms=2` and `nmdoms=1` in `chimere.sh`.

Now it is time to customize `chimere.sh` for directories and files locations depending on where you stored them after download.

Compile all programs:

```
make
```

Then you can execute the top-calling script using the command (for example and for the proposed test case):

```
chimere.sh 20030730
```

and see the messages on the screen to check that all installations were correctly done. At a later stage, if you want to run CHIMERE on a cluster with a PBS-like job scheduler, link `chimere.sh` to `scripts/linux-pbs-mm5/chimere.sh` and edit it as described before.

2.3 Description of the top-calling script and options

Notes to the following top-calling script `chimere.sh`:

- The only argument of the script is the starting date. For example, for the tutorial test case, the user has to launch the script using `chimere.sh 20030730`. The parameter ‘di’ would have the value ‘20030730’ in this case.
- Simulation parameters, such as `lab`, `nested`, `dom`, etc., are now given in arrays with names starting with ‘_’. The columns in the array correspond to different simulations that the user can launch by running `chimere.sh` only once. The loop is performed over the numbers given in the `runlist` variable, those numbers being the column numbers in the above-mentioned arrays. This is particularly useful when running nested simulations: e.g., the first column corresponds to a coarse domain, while the second one specifies the fine domain.
- Analysis or forecast mode?
  - In the analysis mode, the run duration is ‘ndays’, from ‘di’ to ‘di’+‘ndays’-1. If a previous simulation is used to force the current one (with initial conditions), it is assumed to have the same duration ‘ndays’ and to span from ‘di’-‘ndays’ to ‘di’-1.
  - In the forecast mode, ‘ndays’ refers to the maximum forecast time, and the whole simulation period is from ‘di’ to ‘di’+‘ndays’, i.e., ‘ndays’+1. The previous simulation used to force the current one spans from ‘di’-1 to ‘di’+‘ndays’-1.
- Typical ‘ndays’ value is 5 days for an analysis run and 4 days for a forecast simulation.
- Compilation and execution flags. The user specifies whether he wants to run various Chimere components with corresponding flags:

<table>
<thead>
<tr>
<th>Flag</th>
<th>Action</th>
<th>Script in the scripts/ directory</th>
<th>Chimere component(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>imakecompil</td>
<td>Model compilation</td>
<td>chimere-compil.sh</td>
<td>interf-&lt;model&gt; and diagmet.e</td>
</tr>
<tr>
<td>imakemeteo</td>
<td>Meteorological interfaces</td>
<td>chimere-meteo.sh</td>
<td>prepemis.e</td>
</tr>
<tr>
<td>imakeemis</td>
<td>Emissions interface</td>
<td>chimere-emis.sh</td>
<td>prep_bound.e/prep_chimere.e</td>
</tr>
<tr>
<td>imakebound</td>
<td>Initial and boundary conditions</td>
<td>chimere-bound.sh</td>
<td>chimere.e</td>
</tr>
<tr>
<td>imakerun</td>
<td>Running Chimere</td>
<td>chimere-run.sh</td>
<td></td>
</tr>
</tbody>
</table>
If some of the Chimere components are not to be launched, their results (e.g., the file METEO.nc for the meteorological interfaces) must be stored in the directory specified by the data dir variable in chimere.sh.

- The directory postproc is intended for post-processing after your Chimere simulation. The script postproc/postproc.sh is called after the simulation if the flag imakefigs equals 1 in chimere.sh top calling script:

```bash
./postproc.sh ${di} ${ndays} ${dom}
```

where di, ndays, dom are simulation date, duration and domain specified in chimere.sh.

- The script make-chemgeom.sh is fully described in §B.1, p.119.

Here is the listing of the top-calling script chimere.sh:

```bash
#!/bin/bash
unset LANG
ulimit -s unlimited
[ $# -eq 0 ] && { echo "Usage: $0 <date>"; exit 1; }
#==================================================================
# General script running the CHIMERE model
# ==> script documentation at the end of this page
# # http://www.lmd.polytechnique.fr/chimere
# # Questions: chimere@lmd.polytechnique.fr
# LAST REVISION : September 2008
#==================================================================
# EXAMPLE OF AUTOMATIC RUN.
#
# This script assumes an already existing data base of MM5 or WRF runs.
# This script is designed to launch nested simulations
#-----------------------------------
export chimere_root="pwd"

# Timing for the run:
forecast=0    # analysis or forecast mode
ndays=5      # run duration

# Flags for:
imakecompl=1  # code compilation. Attention: if you change nzdoms or nmdoms you must recompile!
imakemeteo=1 # meteorological interface
imakeemis=1  # emissions
imakebound=1 # initial and boundary conditions
imakerun=1   # Chimere launching
imakefigs=0  # post-processing

#-----------------------------------
# Simulation parameters
#-----------------------------------
# The runlist contains (nested) run IDs:
#  runlist="01" for only one simulation
#  runlist="01 02" for two nested simulations ("nest" means one way nesting), etc.
export runlist="03"

for runs in ${runlist} ; do
  _lab=( 01 02 HWave )
  _nested=( no yes no )
  _tonest=( yes no yes )
  _dom=( FRA10.v4 IDF3 CONT5 )
  _meteo=( WRF WRF MM5 )
  _metdom=( WRF WRF EUR2 )
```
himere

_emis=( emep airp emep )
_phys=( 6 6 6 )
_step=( 2 4 1 )
_nlayer_chimere=( 8 8 8 )
_top_chimere_pressure=( 500 500 500 )
_first_layer_pressure=( 995 995 995 )
_mecachim=( 2 2 2 )
_accour=( full low low )
_aero=( 1 1 1 )
_nbins=( 8 8 8 )
_seasalt=( 1 1 1 )
_dust=( 1 1 1 )
_pops=( 0 0 0 )
_dustout=( 0 0 0 )
_carb=( 1 1 1 )
_trc=( 0 0 0 )
_soatyp=( 2 2 2 )
_ideepconv=( 0 0 0 )
_iadv=( 2 2 2 )

#---------- Allocation of simulation parameters ---------
export runs
. ${chimere_root}/scripts/alloc-param.sh
export di=${1}
mdom=${meteo}$metdom

#--------------------------------------------------------
# Parameters common for all runs in "runlist"
#--------------------------------------------------------

# Emission flags (currently used only for EMEP)
surface_emissions=1
point_emissions=0
fire_emissions=0

# Biogenic emissions information
iopt_eros=1 # Optional wind erosion dust emissions (0/1)
iopt_resu=1 # Optional resuspension parameterization (0/1)
# The resuspension parameterization is to be considered as "in test"

# Number of layers in output files (starting from bottom), valid only for the last nested domain
levout=8

# Number of MPI subdomains in zonal and meridian directions for // use
nzdoms=2
nmdoms=2

# Number of Gauss-Seidel iterations in the TWOSTEP solver. Take 1 for model testing, 2 for higher accuracy
ngs=1

# Number of Gauss-Seidel iterations in TWOSTEP during spin up
nsu=5

# Number of hours for Spin up
irs=1

# 1 : online coupling with ISORROPIA;  0 : ISORROPIA tabulation
nequil=0

# Compute equilibrium each "npeq" physical time step : < 5
npeq=1

# Save all concentrations every ... hours (file end....)
snscons=24

# Save all cumulated deposition every ... hours (file dep....)
snsdepos=12

# Meteorological information and options:
wrfcase=base # WRF model physical case
upblmin=0 # Urban correction : Minimal urban PBL height. Set 0 if not used
uflxadd=0 # Urban correction : Additional urban heat flux. Set 0 if not used
wunfac=1    # Urban correction : Urban wind factor in the SL. Set 1 if not used

#----------------------------------------------
# Directories:
#----------------------------------------------
# Big files (location of AEROMIN.bin, AEROORG.bin, etc...):
bigfilesdir='/.../BIGFILES
# Anthropogenic Emissions:
emissdir=${bigfilesdir}/EMISSIONS/EMI_2005
emissdir=${bigfilesdir}/TEST/EMIS  # for the TEST case
# MEGAN data root:
export megan_data=${bigfilesdir}/MEGAN
# Boundary conditions:
mozart2dir=${bigfilesdir}/MOZART2
lmdzinca2dir=${bigfilesdir}/INCA
gocartdir=${bigfilesdir}/GOCART
lmdzaerodir=${bigfilesdir}/LMDZAERO
# Meteo output:
meteo_MM5=${bigfilesdir}/TEST/METEO  # MM5
meteo_WRF=/home/WRF-Analysis/${wrfcase}  # WRF
meteo_ecm=/home/IFS_JLM/${di}  # ECMWF
# Output files. Caution, require much disk space!
simuldir=${bigfilesdir}/OUTPUTS
# If this run is a continuation one, initial conditions will be obtained from the "end" file in this dir:
psimuldir=${simuldir}
# Your meteo, emission, boundary & initial condition files (if their 'imake___' options are 0):
datadir=${simuldir}/CONT5_ibc_exe_meteo  # Attention: full (absolute) path!
# Where to put make.log files (useful for debug) - usually /tmp:
garbagedir=${chimere_root}/../tmp
# Required utilities. You may have to define manually their full path
MAKE="/usr/bin/gmake"
AWK="/bin/gawk"
NCDUMP=ncdump
# Temporary directory:
tmplab='date +%y%m%d_%H-%M'
export chimere_tmp=${simuldir}/tmp${tmplab}-${dom}
clean="no"  # clean=no: The entire directory is kept intact.
# clean=light: Only small files are kept.
# clean=full: The directory is removed at the end.

#--------------------------------------------------------
. ${chimere_root}/scripts/title.sh
. ${chimere_root}/scripts/check-dates.sh ${forecast} ${ndays}
#--------------------------------------------------------
case ${meteo} in
  MM5)
    mm5de='date -d "$de 1 days" +%Y%m%d'
    meteo_file=${meteo_MM5}/${metdom}/MMOUT_${metdom}_${di}_${mm5de}
    meteo_file=${meteo_MM5}/MMOUT_${metdom}_${di}_${de}  # for the TEST case
    ;;
  WRF)
    meteo_file=${meteo_WRF}/wrfout_d$lab._'date -d "${di}" +%Y-%m-%d'/_00\:00\:00 ;;
  ecm)
    meteo_file=${meteo_ecm} ;;
  *)
    echo "Unknown meteo model ${meteo}"; exit 1
esac

#--------------------------------------------------------
### some checks

```bash
if [ "${tonest}" = "yes" ]; then
  accur=full
  levout=${nlayer_chimere}
  echo "+++ This is a tonest run:"
  echo " the output resolution must be forced to "${accur}
  echo " the number of out levels must be forced to "${levout}
fi
```

```bash
echo "--------------------------------------------------------"
```

```bash
# Initialization of common parameters
.
```

### Compilation

```bash
if [ "$imakecompil" = "1" ]; then
  . ${chimere_root}/scripts/chimere-compil.sh
  [ $? -eq 0 ] || { echo "Abnormal termination of chimere-compil.sh"; exit 1; }
else
  ln -s ${datadir}/*.e ${chimere_tmp} || { echo "Cannot link ${datadir}/*.e to ${chimere_tmp}"; exit 1; }
fi
```

### Meteo interface and diagmet

```bash
if [ "$imakemeteo" = "1" ]; then
  . ${chimere_root}/scripts/chimere-meteo.sh
  [ $? -eq 0 ] || { echo "Abnormal termination of chimere-meteo.sh"; exit 1; }
else
  ln -s ${datadir}/METINFOS.nml ${chimere_tmp} || { echo "Cannot link ${datadir}/METINFOS.nml to ${chimere_tmp}"; exit 1; }
  ln -s ${datadir}/METEO.nc ${chimere_tmp} || { echo "Cannot link ${datadir}/METEO.nc to ${chimere_tmp}"; exit 1; }
fi
```

### Boundary & initial conditions

```bash
if [ "$imakebound" = "1" ]; then
  . ${chimere_root}/scripts/chimere-bound.sh
  [ $? -eq 0 ] || { echo "Abnormal termination of chimere-bound.sh"; exit 1; }
else
  ln -s ${datadir}/*_CONCS.nc ${chimere_tmp} || { echo "Cannot link ${datadir}/*_CONCS.nc to ${chimere_tmp}"; exit 1; }
fi
```

### Emissions

```bash
if [ "${imakeemis}" = "1" ]; then
  . ${chimere_root}/scripts/chimere-emis.sh
  [ $? -eq 0 ] || { echo "Abnormal termination of chimere-emis.sh"; exit 1; }
else
  ln -s ${datadir}/?EMISSIONS.nc ${chimere_tmp} || { echo "Cannot link ${datadir}/?EMISSIONS.nc to ${chimere_tmp}"; exit 1; }
fi
```

### Chimere run

```bash
if [ "$imakerun" = "1" ]; then
```
. ${chimere_root}/scripts/chimere-run.sh
 [ $? -eq 0 ] || { echo "Abnormal termination of chimere-run.sh"; exit 1; } fi

# Postprocessing
if [ ${imakefigs} = "1" ] ; then
cd ${chimere_root}/postproc
 ./postproc.sh $(di) $(ndays) $(dom)
 [ $? -eq 0 ] || { echo "Abnormal termination of postproc.sh"; exit 1; } fi

if [ ${tonest} = "yes" ] ; then
csimuldir=${simuldir}
 clab=${lab}
fi

cd ${chimere_root}
done

#===================================================================
# CHIMERE SCRIPT DOCUMENTATION
# ----------------------------
# SOFTWARE REQUIRED: gawk, gmake, netcdf, lam-mpi
# a fortran 90 compiler: CHIMERE works without
# modifications with compilers: ifort, pgf90, g95, xlf95
#
# *** TESTED on architectures: i386/GNU-Linux and power4/AIX-5L
#
#===================================================================
# emis=std
# Anthropic emission interface: std only available at present
# User can define its own interface. This assumes the
# definition of command lines added below below the
# comment "Hourly anthropic emissions"
# std emissions are given for each model species in an ascii
# files containing 3 types of day per month and 12 months.
#===================================================================
# Management of the temporary directory: tmp
# All operations are done in a temporary directory.
# CHIMERE_TMP is the temporary directory. Give a name where
# there a lot of temporary disk space
# The size of the directory is typically 100 mo but
# can be much more in high-resolution simulations
# CHIMERE is executed in this directory.
# !!!! This name must be an absolute path (not ./mydir ...) !!!!
# By setting a CHIMERE_TMP directory name that depends
# on process number ($$), it is possible
# to perform several simulations simultaneously
# without undesired interactions
########################################################################
# Chemical parameters of the simulation#
########################################################################
# mecachim:
# [0] : no gas phase chemistry
# [1] : melchior complete
# [2] : melchior reduced
# accur-low or full can be used for nesting
# low resolution (main components with families)
# full resolution (all components with families. used for nesting)
# nbins: How many bins for aerosols
The cut off diameters 2.5 and 10 um are respected.
You can change the parameter dmax in ./chemprep/chemprep.univ.sh in order to adapt the size distribution. This script is adapted for nbins=4, 6, 12 (nbins=6 is recommended)

### Aerosols: aero flag
- [0] : no NO3, NH4, SO4, PPM, Sec. Org. Aer., WATER
- [1] : Minimal configuration with NO3, NH4, SO4, PPM, Sec. Org. Aer., WATER

### Sea salt aerosols: seasalt flag
- [0] : none
- [1] : inert sea salts
- [2] : active sea salts (reaction with chemistry), not fully validated

### dust flag:
- [0] : none
- [1] : yes, these dusts are considered as emitted within the domain
  - if _dustout_ = 0 dust boudary conditions are taken into account

### pops flag: Persistent Organic Pollutants: =1 only if mecachim>0 and aero=1
- [0] : none
- [1] : yes , only three PAHs available : BaP, BbF, BkF (not fully validated)

### dustout flag: Separate dust from outside
- (_dust_ flag corresponds to locally emitted dusts)
- [0] : if _dustout_ = 1 locally emitted dusts and long rage transported dust are cumulated
- [1] : if you want to separate the dust contribution from outside

### Output species accuracy

### soatyp flag: Secondary Organic Chemistry chemistry
- [0] : no SOA chemistry
- [1] : simple scheme
- [2] : medium scheme
- [3] : complex scheme

### carb flag: Primary carbonaceous species (OCAR : Organic material, BCAR : Black carbon)
- [0] : no
- [1] : yes

### trc flag: specify the number of tracers : trc=0 , no tracer
- [0] : no tracers
- [n] : n tracer

### soatyp flag: Secondary Organic Chemistry chemistry
- soatyp>0 only if aero = 1

### Physical parameters

### phys=6 Frequency of physics calculation (number per hour)
- It could be automatically optimized during the preprocessing stage

### step=1 Refined time step relative to physics calculation
- 1/(phys*step) is the time step in hours
- Down to 1/4 degree resolution, phys=6 and step=1 is fine
- For finer grids (5-10 kms), take at least step=4
- For grids of 2-5 kms, try step=6 or more

### nzdoms: number of MPI subdomains in zonal direction, in case of parallel execution

### nmdoms: number of MPI subdomains in meridian direction.

### lab Choose a label to uniquely identify this run

### clab Same for coarse domain if necessary

### meteo Type of meteo

### nested "yes" if your simulation needs a coarse CHIMERE run

### tonest "yes" if your simulation drives a nest run

### dom Name of the CHIMERE domain as specified in file names of the "domains" directory

### ideepconv
- 0 : No deep convection; 1 : Select deep convection automatically according resolution
- iadv=2 0 : For upwind, 1 : for PPM, 2 : for VanLeer

### simuldir = output directory
- If this run is a continuation one, initial conditions will be obtained from the "end" file in this directory in the directory "psimuldir"
2.4 Direct run outputs

2.4.1 The output on screen

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 the main parameters predefined in the routine `src/initio/outprint.f90` to give some informations about the run. The choices of outputs are:

- The current date in format `yyyyymmddhh`
- The first two species defined in the `OUTPUT_SPECIES` file
- The 2m temperature (in Kelvin)
- The surface wind speed (in m/s) (in fact, this value correspond to the first vertical levels wind speed, typically 30-40m AGL).
- The boundary layer height (in meters) diagnosed using the `src/diag/diagmet.f` pre-processor
- The radiative attenuation: from 0 to 1 with 1 for the complete clear sky.
- The $K_z$ diffusion coefficient at levels 1 and 2 ($m^2/s^2$)

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

2.4.2 The output files

Four different output files are delivered. They 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

Moreover, CHIMERE uses a netCDF convention derived from the one used in NCAR’S WRF mesoscale modelling system, simplifying the toolset for post-processing.

2.4.2.1 The chemical fields outputs on file "out.[label].nc"

Some three-dimensional chemical fields are written in an output binary file called `out.[label].nc`. The selected species correspond to those listed in the `OUTPUT_SPECIES` parameter file. Default values are expressed in ppb. Output values may be transformed into $\mu g.m^{-3}$ if the molar mass of each species is added as a second column in the parameter files `/data/output_species/OUTPUT_SPECIES...`. The default value is zero. These potential transformations are managed in the routines `src/initio/inichem.f90` and `src/initio/outprint.f90`. The user is encouraged to have a look to the routine `toolbox/readcdf/readcdf.f90` to learn how to read the model output files.

example: In the file `/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 $\mu g.m^{-3}$.

2.4.2.2 The deposition fields outputs on file "dep.[label].nc"

This file contains fluxes integrated over the chemical time-step duration. The calculations are done in `src/model/twostep_mod.f90` and written in files in `src/initio/write_depo.f90`. For the dry deposition flux, only the first vertical cell is taken into account. On the other hand, for the wet deposition, fluxes are summed 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/cm².
2.4.2.3 The restart fields outputs on file "end.[label].nc"

The structure of this file is described in Appendix E.7, p.154. The main goal of this file is to save all concentrations fields every 24h. 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.

2.4.2.4 The additional fields outputs on file "par.[label].nc"

An additional output file is named par.[label].nc. Its goal is to provide meteorological outputs, corresponding to the values employed during the run to calculate the concentration fields written in out.[label].nc. The extracted parameters are hardcoded in the routine src/initio/iniparams.f90. The possible values are displayed in the Table 2.2.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>wind</td>
<td>Wind speed</td>
</tr>
<tr>
<td>iwinhs</td>
<td>Wind speed</td>
</tr>
<tr>
<td>iwinhs2</td>
<td>Vertical wind speed (m/s 1st vertical level)</td>
</tr>
<tr>
<td>iwinhs3</td>
<td>Vertical wind speed (m/s top domain vertical level)</td>
</tr>
</tbody>
</table>

specific humidity, air density and temperature

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>isphu</td>
<td>Specific humidity</td>
</tr>
<tr>
<td>iaarm</td>
<td>Air density</td>
</tr>
<tr>
<td>itemp</td>
<td>Temperature (K)</td>
</tr>
<tr>
<td>iwinxs</td>
<td>Velocity-equivalent of mixing exchange rate (surface)</td>
</tr>
<tr>
<td>iwinxt</td>
<td>Velocity-equivalent of mixing exchange rate (top domain)</td>
</tr>
<tr>
<td>ithlay</td>
<td>Thickness of the model layer (in cm)</td>
</tr>
</tbody>
</table>

mixing height, radiation

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>ihght</td>
<td>Mixing height (in meters)</td>
</tr>
<tr>
<td>iatte</td>
<td>Radiation attenuation [0-1]</td>
</tr>
<tr>
<td>izeni</td>
<td>Zenithal angle (degrees)</td>
</tr>
<tr>
<td>item2</td>
<td>2m temperature (K)</td>
</tr>
</tbody>
</table>

diagnostic turbulent parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>iusta</td>
<td>Friction velocity $u_*$ (m/s)</td>
</tr>
<tr>
<td>iaerr</td>
<td>Aerodynamical resistance</td>
</tr>
<tr>
<td>iobuk</td>
<td>Monin-Oblukov length</td>
</tr>
<tr>
<td>iwsta</td>
<td>Convective velocity $w_*$</td>
</tr>
<tr>
<td>idepo</td>
<td>Dry deposition velocity</td>
</tr>
</tbody>
</table>

Table 2.2: The parameters model outputs on file.

2.5 Post-processing the results

We provide here some indications to help the user for displaying and processing CHIMERE raw results. The complexity of post-processing depends obviously on the user intents, and the tools proposed here should only be considered as helpers or templates for developing more sophisticated programs. Apart some configuration files and static data files written in ASCII code, all the input and output files of the model are written in the netCDF format, using conventions similar to the WRF conventions. Thus the tools described below may be used to process almost any CHIMERE netCDF file, including meteo files. CHIMERE netCDF files are suffixed with .nc.

Once you have successfully performed a CHIMERE run, you may want to:

- Dump data in ASCII form
- Use data for further numerical processing
• Plot data
• Interactively display data

Guidelines are proposed in the following paragraphs.

2.5.1 Dump data in ASCII form

If you have netCDF installed, and you do have it since you were able to run this version of CHIMERE, just use the `ncdump` utility to dump the content of any of CHIMERE netCDF file:

- `ncdump -h <ncfile>` prints the file header
- `ncdump -c <ncfile>` prints the header, plus the values of so-called "coordinate variables"
- `ncdump -b f <ncfile>` prints the whole content of the file, with fortran indexes.
- `ncdump -v <myvar> -b f <ncfile>` prints the header, followed by the selected variable

For more details, read the `ncdump` man page:

```
man ncdump
```

2.5.2 Use data for further numerical processing

**Brute force method**:
Redirect the output of ncdump to a file, and open this file with your Fortran code. This method is NOT recommended, since:

- it produces large ASCII files, with inefficient sequential access
- you loose numerical accuracy

Download the netCDF Fortran 90 manual from [http://my.unidata.ucar.edu/content/software/netcdf/index.html](http://my.unidata.ucar.edu/content/software/netcdf/index.html). Read, read again and understand this manual. Code by yourself some simple examples. Have a look at the netCDF code in toolbox/ directory, for instance toolbox/readcdf/readcdf.f90. Write your own interface to get CHIMERE data into an array in your Fortran 90 code. This is the right method, with a guaranteed Return On (intellectual) Investment.

2.5.3 Plot data

2.5.3.1 ncmapt

The toolbox/ncmapit/ncmapit.sh script takes as input a CHIMERE output file (end.*, out.*, dep.*, par.*) or the meteo file METEO.nc, and:

- launches `ghostview` to display the selected field
- produces a "publication-ready" PostScript file with Chimere-standard color palette.

`ncmapit.sh` is the netCDF counterpart of the former mapit shell script proposed in earlier versions of CHIMERE. `ncmapit.sh` provides a way to display many CHIMERE netCDF files, including OUTPUT and METEO. `ncmapit.sh` tries to use the standard palettes contained in the directory toolbox/ncmapit/palettes. If a palette is not available for a species, then `ncmapit.sh` generates it, with rounded thresholds, tailored to the actual scale of the field to be displayed.
In details, `ncmapit.sh`:

- takes a CHIMERE file as input,
- launches a Fortran program, chimere2gmt, which:
  □ scans the CHIMERE file to compute range information and palette data
generates a header file containing the required metadata
□ generates an ASCII file containing a list of X, Y, VAL triplets passed to the Generic Mapping Tools (GMT, http://gmt.soest.hawaii.edu/)

- spawns GMT to produce a PostScript file suitable for display or printed publication.
- spawns a GhostScript (www.cs.wisc.edu/ghost/) window to display this PostScript.

You can tailor the script to:

- modify the style of the figure. See GMT documentation.
- keep the PostScript file (erased by default)

ncmapit.sh cannot read AEMISSIONS, BEMISSIONS, and Boundary conditions files due to the very specific format of these files.

Usage:

- 1st: build chimere2gmt.e
  ```plaintext
  make
  ```
- 2nd: run ncmapit.sh
  ```plaintext
  ./ncmapit.sh <chimere_file> <variable> <date> <level> <map_file.ps>
  ```

Where

- `<chimere_file>`: your CHIMERE output file: dep, end, out or par
- `<variable>`: the name of the variable you want to display, like O3. Must be among the OUTPUT_SPECIES list.
- `<date>`: the date/time in Chimere format, like 2004122612
- `<level>`: the index of the layer you want to display. For 2D fields, set level=1
- `<map_file.ps>`: the name you want to give to the postscript output file this script will generate

In the case of a 2D field, set level=1 in <args>.

Predefined palette files for some pollutants are given in the palettes subdirectory. For some other pollutants the user must define a new palette file by using the GMT command `makecpt`. The predefined palettes are described in the following section.

2.5.3.2 chim2fig

The toolbox/chim2fig/chim2fig.sh script can be used to produce a wider range of plots, including horizontal maps, cross-sections, time series or vertical profiles. The users shall edit `chim2fig.sh` to define their preferences. `chim2fig.sh` contains many comments which should be sufficient to help configuring it.

`chim2fig.sh` requires meteo informations from METEO.nc, the output file of the second stage of the meteo interface. It implies that:

- The `use=` statement shall contain “meteo”.
- The CHIMERE output directory shall contain a copy of METEO.nc named meteo.xxxx.nc, where xxxx is the radix used in output files naming. This is the default behaviour now.

Once configured, you can execute `chim2fig.sh`. A self-explanatory menu will be displayed, and finally the maps will be plotted. The .ps or .gif files will be kept if you required so in `chim2fig.sh`.

Summary of usage:
- 1st: build chim2fig.e
  
  cd toolbox/chim2fig
  make

- 2nd: edit chim2fig.sh

- 3rd: run chim2fig.sh
  ./chim2fig.sh

As with ncmapit, predefined palette files for some pollutants are given in the palettes subdirectory. The user can change these predefined palettes. Their characteristics are:

<table>
<thead>
<tr>
<th>Filename</th>
<th>Min</th>
<th>Max</th>
<th>Step</th>
</tr>
</thead>
<tbody>
<tr>
<td>CH4.cpt</td>
<td>1700</td>
<td>1900</td>
<td>20</td>
</tr>
<tr>
<td>CO.cpt</td>
<td>0</td>
<td>1000</td>
<td>50</td>
</tr>
<tr>
<td>HNO3.cpt</td>
<td>0</td>
<td>20</td>
<td>5</td>
</tr>
<tr>
<td>NO.cpt</td>
<td>0</td>
<td>50</td>
<td>2.5</td>
</tr>
<tr>
<td>NO2.cpt</td>
<td>0</td>
<td>150</td>
<td>10</td>
</tr>
<tr>
<td>NOY.cpt</td>
<td>0</td>
<td>150</td>
<td>10</td>
</tr>
<tr>
<td>O3.cpt</td>
<td>0</td>
<td>150</td>
<td>10/50</td>
</tr>
<tr>
<td>PAN.cpt</td>
<td>0</td>
<td>6</td>
<td>0.3</td>
</tr>
<tr>
<td>pHNO3-reg.cpt</td>
<td>0</td>
<td>50</td>
<td>5</td>
</tr>
<tr>
<td>pH304.cpt</td>
<td>0</td>
<td>200</td>
<td>5/10/50</td>
</tr>
<tr>
<td>pH2SO4.cpt</td>
<td>0</td>
<td>200</td>
<td>5/10/50</td>
</tr>
<tr>
<td>pHNO3.cpt</td>
<td>0</td>
<td>200</td>
<td>5/10/50</td>
</tr>
<tr>
<td>PM10.cpt</td>
<td>0</td>
<td>200</td>
<td>5/10/50</td>
</tr>
<tr>
<td>PM25.cpt</td>
<td>0</td>
<td>200</td>
<td>5/10/50</td>
</tr>
<tr>
<td>pNH3.cpt</td>
<td>0</td>
<td>200</td>
<td>5/10/50</td>
</tr>
<tr>
<td>pNH3-reg.cpt</td>
<td>0</td>
<td>20</td>
<td>2</td>
</tr>
<tr>
<td>PM10.cpt</td>
<td>0</td>
<td>200</td>
<td>5/10/50</td>
</tr>
<tr>
<td>SO2.cpt</td>
<td>0</td>
<td>100</td>
<td>10</td>
</tr>
</tbody>
</table>

Table 2.3: The predefined color tables to be used with the GMT software for concentrations maps

These palettes were defined to be the most adapted to ‘usual’ regional surface concentrations. You can create a new palette by using the following command line:

```
makecpt -C TABLE -T vmin/vmax/step > OUT_PALETTE_FILE
```

with:

- **TABLE**: The GMT predefined color tables: type `makecpt` in a x-terminal to have the complete list. You can use, for example:
  - gray: Grayramp from black to white [0/1]
  - rainbow: Rainbow colors: magenta-blue-cyan-green-yellow-red
  - cool: Linear change from blue to magenta

- **vmin/vmax/step**: the minimum/maximum and step values (examples are provided in the Table 2.3)

- **OUT_PALETTE_FILE** the output file for this new color palette. Please note that if the name is not composed of the species name and the ".cpt" extension, you have to change the corresponding line in the mapit script.

Users of the pgf90 compiler are warned that chim2fig does not terminate cleanly when compiled with pgf90. It is not currently possible to restart another display without quitting chim2fig, and the only way to do it is to type `Control-C`.

### 2.5.4 Interactively display data

Exachim and examout are (maybe) what you are looking for. `exachim.sh` takes any CHIMERE netCDF file (at the exception of boundary condition files, and emission files), transforms it into a GrADS compliant file, and launches a GrADS script to display the data.

Once in the display window, you can interactively, by simple mouse clicks, change the time slot, variable, level, or print the value of the variable for a specific location. The user can also focus on a particular field, and by point clicking on the image, get the values of the triplet X, Y, VAL.
examout.sh has the same functions as exachim.sh, but it uses the standard CHIMERE palettes. Thus its use is restricted to "out." files.

While exachim.sh may work on almost any netCDF CHIMERE file, examout.sh is designed for use with "out" files. It includes prebuilt palettes which are now a de facto standard in CHIMERE, thus making the display familiar to those users who were using the former mapit display tool.

In details, exachim.sh and examout.sh:

- take as input a CHIMERE file, containing 2D and/or 3D fields.
- launch a Fortran program, chm2grads, to transform the CHIMERE file into a GrADS-compliant file.
- launch a GrADS (http://grads.iges.org) script, exachim.gs (examout.gs), to generate the interactive display.

exachim.sh and examout.sh cannot read AEMISSONS, BEMISSIONS, and Boundary conditions files due to the very specific format of these files.

exadiff.sh works like exachim.sh, but displays the difference between two files. It may be useful to check the effect of a modification you could do in the model.

Usage (same usage for exachim.sh)

- 1st : build chm2grads
  
  make

- 2nd : run ./exachim.sh

  exachim.sh [-v var] [-s start_time_index] [-n n_req_times]
  [-l altitude_file] [-a altitude_var] <netcdf_file>

var is a comma separated list of variables to be displayed (default : all displayable variables)
start_time_index is the index of the first time slot to display (default : 1)
n_req_times is the number of time slots you want to display (default : all slots)
-l altitude_file and -a altitude_var options allow you to calculate vertical averages (and vertical sums), as well as vertical profiles, taking into account the level heights. If this option is not used, all vertical profiles are plotted in terms of level number, otherwise the profiles are plotted in terms of level altitude. The plotting heights are
Table 2.4: Contents of the "Slice" mode additional windows depending on the mode and mouse button pressed

<table>
<thead>
<tr>
<th>Mode</th>
<th>Mouse Button</th>
<th>Left Window</th>
<th>Right Window</th>
</tr>
</thead>
<tbody>
<tr>
<td>1D</td>
<td>Left</td>
<td>Time series at the (lon,lat) point chosen, for the time, level, and variable selected in the &quot;Loop&quot; mode</td>
<td>Time series of the (lon,lat) average, for the time, level, and variable selected in the &quot;Loop&quot; mode</td>
</tr>
<tr>
<td></td>
<td>Right</td>
<td>Latitudinal dependence at the longitude chosen with a mouse, for the time selected in the &quot;Loop&quot; mode (white curve) and time-average (green curve). The level and variable are those selected in the &quot;Loop&quot; mode</td>
<td>The same as Left Window but for the longitudinal dependence</td>
</tr>
<tr>
<td></td>
<td>Middle</td>
<td>Vertical profiles at the current point (white - current time, green - time-mean)</td>
<td>The same as Left Window but lon-lat domain average</td>
</tr>
<tr>
<td>Geo-Time</td>
<td>Left</td>
<td>Time-longitude dependence at the current latitude</td>
<td>Time-longitude dependence, latitude average</td>
</tr>
<tr>
<td></td>
<td>Right</td>
<td>Time-latitude dependence at the current longitude</td>
<td>Time-latitude dependence, longitude average</td>
</tr>
<tr>
<td>Z 2D</td>
<td>Left</td>
<td>Z-longitude slice at the current latitude</td>
<td>Z-longitude slice, latitude average</td>
</tr>
<tr>
<td></td>
<td>Right</td>
<td>Z-latitude slice at the current longitude</td>
<td>Z-latitude slice, longitude average</td>
</tr>
<tr>
<td></td>
<td>Middle</td>
<td>Z-Time slice, current lon, lat</td>
<td>Z-Time slice, (lon, lat) domain average</td>
</tr>
<tr>
<td>Z 2D TA</td>
<td>Left</td>
<td>The same as Z 2D, Left Button, but for time means</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Right</td>
<td>The same as Z 2D, Right Button, but for time means</td>
<td></td>
</tr>
</tbody>
</table>

obtained by interpolating the altitudes given in altitude_file in the netcdf variable altitude_var. If `-l` option is absent but `-a` is present, the corresponding variable is taken from the input file. If `-a` is absent but `-l` is present, hlay is used as a default variable name.

The display window has three modes:

**Loop**. You are in this mode when you start (Figure 2.2a).
- a left-click on the picture increases the time slot index
- a right-click decreases the time slot. If you click fast enough, you can simulate a movie loop. The palette remains constant when you loop along time axis. It may vary when you loop along level axis.
- you can change of variable or level by clicking on one of the four bottom-left buttons.
- click on the "Loop" button to enter the "Slice" mode

**Slice**. The main window becomes smaller, and four new buttons appear: "1D", "Geo-Time", "Z 2D", "Z 2D TA" (Figure 2.2b). Those buttons correspond to the four modes of slice plotting. The default mode is "1D". When you click somewhere in the main window, two additional windows below show data slices corresponding to the (longitude, latitude) point where you clicked. The data slices displayed in the two additional windows, depend on the mode and the mouse button clicked, as described in Table 2.4.
- if you click on the "Slice" button you enter the "Data" mode

**Data**.
- a click on the picture prints on the xterm window, depending on what was previously selected in the "Loop" mode:
  - the name of the variable
When browsing data in GrADS using `exachim.sh` you can view vertically averaged and vertically integrated fields. To do so, click on "Prev Lev" while viewing Level 1. All operations are then performed with vertical means, in both "Slice" and "Data" modes. Exception are vertical profiles, both 1D and 2D, which are always plotted using vertically varying data. Another click on "Prev Lev" button will display vertical integrals. Use "Next Lev" to go back to the means and then to individual levels.

**Notes.** A few things to keep in mind:

- You cannot view 2D vertical profiles while working with vertical mean or vertically integrated fields.
- The World coordinates that you obtain by clicking in the main window in the "Slice" mode are more or less approximate, and precision depends on the window size. You might want to adjust the parameters `dwrccd_x` and `dwrccd_y` in the beginning of the GrADS script `exachim.gs`. They are optimized for the default GrADS window sizes, so if you plan to change it (either by mouse-dragging or by start-up parameters), you are encouraged to tune these parameters.

A click on `Quit` exits the application.
Chapter 3

The CHIMERE model source code

The CHIMERE model consists of two separate modules. The first one is the meteorological interface (in the `meteo/` directory). It prepares the meteorological data in the CHIMERE format, but also diagnosed turbulent parameters, calculate biogenic emissions (meteorologically dependent). The second one is the CHIMERE code itself (in the `src/` directory).

The general structure contains a central chemistry-transport calculation unit with interfaces to several databases (emissions, meteorology, boundary conditions, land use, as shown in Figure 3.1. Each interface transforms original or processed data to CHIMERE-input formatted data adapted to the time period and region to simulate.

![Figure 3.1: Schematic structure of the CHIMERE input data](image)

The most complex interface is the meteorological interface because it uses two steps. The first one, `meteo`, transforms original meteorological data (standard variables) given on the meteo model grid, at a frequency that is not necessarily one hour, to the same variables but on the CHIMERE grid (space interpolation) at the hourly sampling rate (time interpolation) in a specific file format. An example, the `interf-mm5` interface, is provided in the `meteo/` directory for data coming from MM5. The `interf-mm5` interface can be directly applied to a standard output MM5 file (MMOUT files).

The second step for the meteorological interface is the diagnostic model `diagmet`, used to take standard meteorological variables on the CHIMERE grid and transform them into variables necessary for input to CHIMERE. For instance, the boundary layer height can be diagnosed from the vertical profile of temperature, humidity and wind. The boundary conditions interface takes either a "coarse run" produced by Chimere and interpolates hourly pollutant concentrations at the domain boundaries, or provide climatological (monthly mean) values at the domain boundaries from a global database consisting of simulation outputs.
The land-use interface (prepdomains) is used to construct necessary terrain files and is used only once per domain construction.

The model core organization is described below. Briefly, it consists of the call of a main FORTRAN program (src/main/chimere.f90) which initializes and reads all data, and runs an integration covering the time period specified. The output files are then available for postprocessing.

The whole system is piloted by a top calling script which:

- reads parameters of the simulation
- links all necessary files into a temporary directory where all programs are executed
- compiles all necessary code
- runs the 3 "dynamic" interfaces (meteo, emission and boundary conditions)
- runs the model itself

The top calling script is executed to carry out a simulation. It must be edited and the options/directories must be chosen and defined by changing the values of parameters inside the script. In the distribution proposed on the CHIMERE web site, the example top calling script is chimere.sh (see p.21).

3.1 The CHIMERE time integration

The CHIMERE model is splitted into two main parts: a general initialization phase and the time integrations of all chemistry-transport equations over the whole simulation duration (Figure 3.2).

- The initialization phase is dedicated to the read of all input parameters as well as the two first meteorological and emissions fields (in order to start the run with realistic meteorological and concentrations fields).
- The time integration phase is itself split into three parts:
A predefined **hourly time-step**. This time step is fixed and is due to the fact that meteorological and emissions data have to be, at least, hourly provided.

- **A user’s defined coarse time step ‘nphour’**, corresponding to the time interpolation of "physical" parameters, such as wind, temperature, reactions rates etc. **NEW:** A second time-step is dynamically estimated in the meteorological pre-processor. Using the horizontal wind field, the vertical velocity and the entrainment in the updraft (in case of deep convection), a recommended time step is estimated. During the run, if the user time step is too low, the model time integration is increase to reach the recommended value. The **Figure 3.3** presents preconised 'number of model integration per hor' for the CFL estimated only with the horizontal wind (u), the vertical wind (w) and the entainment in the updraft (up). For a test run over the CONT5 domain and 10 vertical levels (from surface to 500hPa), with a user time step of 10mn (i.e nphour=6), we can see than deep convection requires to increase the time resolution up to 6mn (i.e nphour=10). If the user’s defined time-step is lower that the recommended one, the user’s choice is applied (even if this is not the optimal choice).

- **A user’s defined "fine" time step 'ichemstep’**: this corresponds to the integration of the chemical mechanism, including delta concentrations due to all processes. This is achieved by the two-step scheme. Due to the stiffness of the chemical system to solve, this time step must be at least 30s (or less if possible).

![Figure 3.3: Calculation of the number of integration steps per hour to respect the CFL over a complete 120 hours simulation.](image)

### 3.2 Directories

The **chimere2008** top directory contains individual files, the test case top calling script and subdirectories. These subdirectories are :

- **chemprep/** Contains data files for gas, aerosols and MELCHIOR mechanisms.
- **domains/** A subdirectory where coordinate files COORD_[domain] and land-use files LANDUSE_[domain] must be placed for each CHIMERE model domain. This directory contains also the file **domainlist.nml** where each line corresponds to a model domain.
- **makefiles.hdr/** Contains examples of Makefile headers suited for several architectures and compilers.
• **meteo/**: This directory contains all the elements for the first step of meteo interfaces, i.e. the horizontal/time interpolation and reformatting of data. When the user starts from a new meteorological data set, a new shell script named `interf-[metname]` along with a read/interpolate fortran routine and the associated Makefile-[model] should be created by the user and placed in this directory. It is recommended to start from `interf-mm5`, `preppmm5.f90` and `Makefile-mm5` as templates to help creating these interface programs.

• **scripts/**: Contains families of scripts pointed to by the top level script. While the top level script mainly defines variables, execution scripts in this directory perform the low-level tasks.

• **src/**: Contains the source code of CHIMERE.

• **toolbox/**: This directory contains two simple graphical interfaces for post-processing. See the documentation below.

• **util/**: Contains some utility programs.

There are other directories created when running the model:

• **tmpdir/**: temporary directory, where all executable and temporary data files go. This allows in particular several simulations to be performed simultaneously by calls from a unique directory, provided the name of the temporary directory contains the unique process number ($$$ in Bourne Shell) or script execution date and time. Optionally this directory can be left after a model simulation (especially when debugging). By default it is removed at the end of top calling script execution. The address of this directory is given in the top calling script (environment variable `tmpdir`). It does not need to be created before the run.

**Hint for developers:**
You can `cd` to `tmpdir` to modify the model code and test your modifications. The model environment and parameters are defined in files `chimere.nml` (also known as INFOS in previous versions) and `chimere_params.f90` which remain until you erase `tmpdir`. To perform a model run, without the overhead of replaying all the meteo and boundary conditions stuff, just type:

```
make && time mpirun -np N -ssi rpi sysv ./chimere.e
```

with `N = number of subdomains + 1`

• **outdir/**: An output directory where simulation output files are placed during the run. The address of this directory is given in the top calling script (variable `simuldir`). It does not need to be created before the run.

### 3.3 Model Files

The **Figure 3.4** shows the input and output files together with the executable programs or script that are used at all stages of the calling script. See the color legend in the right bottom corner. The yellow files are constructed before the run. File paths below are given relative to `chimere2008`.

#### 3.3.1 Scripts

• `chimere.sh` is the top calling script to run the test case. It should be used as a template for user top calling scripts. When all data sets are ready for a simulation, the top calling script must be executed in order to run the simulation.

• `scripts/make-chemgeom.sh` is the first sub-script called by `chimere.sh`. This script prepares chemistry files (if not already done) and geometry (if not already done).

• `scripts/alloc-param.sh` and `check-dates.sh` are called by `chimere.sh` to initialize simulation parameters.

• `scripts/chimere-compil.sh` is called by `chimere.sh` to compile the model code.

• `scripts/chimere-init.sh` is called by `chimere.sh`. It inherits the user definitions stated in `chimere.sh` and initializes other simulation parameters used by other scripts.

• `scripts/chimere-meteo.sh` is called by `chimere.sh`. It inherits the user definitions stated in `chimere.sh`, performs some sanity checks, and launches the meteorological interfaces.

• `scripts/chimere-emis.sh` is called by `chimere.sh`. It inherits the user definitions stated in `chimere.sh`, performs some sanity checks, and launches the emissions interface.
3.3.2 Makefiles

The compilation process of most of the code in the distribution is driven by Makefiles. To ensure consistency, all Makefile call a unique top level header, `Makefile.hdr`, which deals with all the user installation details. `Makefile.hdr` is in fact a symbolic link to one of the predefined headers located in `makefiles.hdr` subdirectory.

3.3.3 Meteo interface - 1st stage

The interface described below is for the MM5 mesoscale modelling system.

1. **meteo/**
   - `meteo/interf-mm5`, `meteo/interf-wrf`: scripts to run the first-step meteorological interface (interpolation + formatting). Called by the top calling script.
   - `meteo/COORD_MM5[domain]_DOT`: contains the “dot” coordinates of the MM5 file.

---

Figure 3.4: Schematic general structure of the scripts, data files of CHIMERE

- `scripts/chimere-bound.sh` is called by `chimere.sh`. It inherits the user definitions stated in `chimere.sh`, performs some sanity checks, and launches the initial / boundary conditions interfaces.
- `scripts/chimere-run.sh` is called by `chimere.sh`. It inherits the user definitions stated in `chimere.sh`, performs some sanity checks, and launches the execution of the model core.
- `chemprep/chemprep.univ.sh` rebuilds a chemical mechanism (see explanations p.57)
- `util/define_geom` utility script used to define the geometric progressing layer vertical grid. It outputs a file with hybrid sigma-p coefficients defining the grid.
- **util/make_mm5_coord**: utility script allowing to extract coordinates and terrain altitude, in longitude/latitude format, of a MM5 TERRAIN file. Edit the script itself for giving information. The output files are `meteo/COORD_MM5[domain]` and `meteo/COORD_MM5[domain]_DOT` used by `interf-mm5`.
- **meteo/prepm5.f90, meteo/prepwrf.f90**: read MM5 and WRF standard output files and perform interpolations to the model grid. Compiled to `prepm5.e` and `prepwrf.e` and called by the scripts `interf-mm5` and `interf-wrf`.
- **meteo/mm5_consts.f90, meteo/wrf_consts.f90**: define uniquely the physical constants used by `prepxxx.f90`.
- **meteo/subs.f90**: contains various helper subroutines.
- **meteo/wcall1.f90**: subroutine calculating the weighting coefficients for horizontal interpolation.
- **meteo/io.f90**: I/O subroutines.
- **meteo/calendar.f90**: calendar subroutines.
- **meteo/Makefile-mm5, meteo/Makefile-wrf**: drives the compilation of the first stage of the meteo interface.

2. **tmpdir/**

   - **tmpdir/meteo/metargs-mm5, tmpdir/meteo/metargs-wrf**: ASCII files generated by the top calling script. Passed to `meteo/interf-mm5` and `meteo/interf-wrf`. Contain parameters to drive `prepm5.e` and `prepwrf.e`.
   - **tmpdir/meteo/prepm5.nml or tmpdir/meteo/prepwrf.nml**: namelist generated by `meteo/interf-mm5` or `meteo/interf-wrf` to drive `prepm5.e` or `prepwrf.e`.
   - **tmpdir/METINFOS.nml**: namelist generated by `meteo/interf-mm5` or `meteo/interf-wrf` to drive `diagmet.e` and `diagbio.e`.
   - **tmpdir/meteo/exdomout.nc**: output netCDF file generated by `meteo/interf-mm5` or `meteo/interf-wrf`. Contains meteo variables interpolated to the model grid.

### 3.3.4 Source directories

All the model sources are available in the `src/` directory. Some sub-directories are present as follows:

1. **src/** contains source subdirectories and template files
   - **chimere_params.f90.sed** module is a template used to build `chimere_params.f90`. It will hold compile-time parameters of the “master” namespace. Each parameter is defined by a macro, which is substituted before compile time by the top level script.
   - **worker_params.f90.sed** module is a template used to build `worker_params.f90`. It will hold compile-time parameters of the ‘worker’ namespace. Each parameter is defined by a macro, which is substituted before compile time by the top level script.

2. **src/main** contains top level programs
   - **chimere.f90** is the main program. Compiled to `chimere.e` and called by the execution script.
   - **integrun.f90** is the main subroutine called by `chimere.f90` for the “master” process
   - **master_locvalues.f90** is a subroutine used to calculate time interpolated values in the “master” side

3. **src/modules** contains master-side modules
   - **chimere_common.f90** module holds all the variables shared in the “master” namespace, type definitions and allocation subroutines
   - **chimere_consts.f90** module holds most of the physical and mathematical constants used throughout the model
• **master_message_subsf90** module contains a bunch of message passing subroutines, used by the “master” side

• **message_defs.f90** module holds constant indexes for message passing between master and workers

4. **src/initio** contains the initialization and I/O subroutines

• **ini*.f90** are initialization subroutines executed once by the “master” process

• **readhour.f90, renewhour.f90** provide the model with hourly meteo and boundary conditions, under “master” control

• **outprint.f90, write*.f90** periodically save the model results to files, under “master” control

• other..*.f90 are helpers for initialization subroutines

5. **src/model** contains the code for the “workers”

• **worker.f90** is the main subroutine called by **chimere.f90** for the “worker” processes

• **worker_common.f90** module holds all the variables shared in the “worker” namespace, type definitions and allocation subroutines

• **worker_message_subsf90** module contains a bunch of message passing subroutines, used by the “worker” side

• other..*.f90 are the heart of CHIMERE model

Despite the capability of Fortran 95 to perform dynamic memory allocation, CHIMERE still uses static allocation, and ‘common’ declarations, because it proved to be significantly faster at run time.

6. **src/iso** contains the ISORROPIA thermodynamic model. Unmodified ISORROPIA library. Called by workers, but has its own namespace.

7. **src/tools** contains helpers. Some helpers, used both by master and workers.

8. **src/diag** contains the diagnostic codes **diagmet** and **diagbio**

• **diagmet.f90** is the second step (diagnostic meteorological model) of the meteorological interface. Compiled to **diagmet.e** and called by the top calling script. It is not origin-dependent for meteo data. It ingests the output files from the first meteo interface.

• **diagbio.f90** generates the biogenic emissions file. Compiled to **diagbio.e** and called by the top calling script.

The code has been splitted into 3 files, for clarity. The main program **diagmet.f90** deals with all the I/O overhead, and calls subroutines in **diagmet_science.f90** which contains only the subroutines needed for meteorological diagnostic. The shared variables are held in **diagmet_common.f90**. In principle, a scientist wanting to develop a new diagnostic routine has mostly to deal with the **diagmet_science.f90** file, and has little to worry about the tedious netCDF I/O interface.

A similar approach is used in **diagbio.f90**, where emissions calculation is additionally separated into two modules: **diagbio_megan.f90** for biogenic emissions and **diagbio_science.f90** for salts and mineral dust.

9. **src/prep** contains the code to prepare emissions and boundary conditions before core model run

• **prepemis.f90** is the standard emission interface. Compiled to **prepemis.e** and called by the top calling script.

• **prep_bound.f90** is the boundary and initial conditions interface for the coarse (non-nested) runs. Compiled to **prep_bound.e** and called by the top calling script.

• **prep_chimere.f90** is the boundary and initial conditions interface for the nested runs. Compiled to **prep_chimere.e** and called by the top calling script.
3.3.5 Dynamic files generated at runtime

All these files are created and moved into the `tmpdir` directory.

1. **Dynamic parameters**

   - `tmpdir/METINFOS.nml`: namelist generated by `meteo/interf-[metname]` to drive `diagmet.e` and `diagbio.e`.
   - `tmpdir/diagmet.nml`: namelist generated by the top calling script to drive `diagmet.e`.
   - `tmpdir/diagbio.nml`: namelist generated by the top calling script to drive `diagbio.e`.
   - `tmpdir/prep_bound.nml`: namelist generated by the top calling script to drive `prep_bound.e`.
   - `tmpdir/prep_chimere.nml`: namelist generated by the top calling script to drive `prep_chimere.e`.
   - `tmpdir/prepemis.nml`: namelist generated by the top calling script to drive `prepemis.e`.
   - `tmpdir/chimere.nml`: namelist containing most of the runtime parameters for the model. This file was the file named INFOS in previous model versions.

2. **Dynamic data files**

   - `tmpdir/meteo/exdomout.nc`: netCDF file generated by the first stage of the meteo interface. Contains meteo parameters interpolated to the model grid.
   - `tmpdir/METEO.nc`: netCDF file generated by the meteo diagnostic stage. Contains meteo and diagnosed parameters.
   - `tmpdir/BOUN_CONCS.nc`: netCDF file generated by the boundary/initial conditions interface. Contains timely lateral and top concentrations.
   - `tmpdir/INI_CONCS.nc`: netCDF file generated by the boundary/initial conditions interface. Contains initial concentrations.

3.4 Routine calls sequence

The general schematic diagram of calls of the CHIMERE model `chimere.f90` is given in Figure 3.5. Only major routines are cited, not the utility ones.

1. The main program `chimere.f90` first calls all initialization routines (in `inichimere.f90` and further calls) where all "static data" data are read and many variables defined.

2. Then it calls the integration routines `integrun.f90` and `worker.f90` which then

   (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 model step, one needs hourly data for the start and the end of the hour in order to linearly interpolate, for meteo data, boundary conditions and biogenic emissions. The code keeps only these two instants in memory as explained in the Figure 3.6. Within the coarse time step, a call to routine `physics.f90` is made, in order to define all variables that will be taken as constants during the time step (linear interpolation of meteo data, calculation of reaction rates etc...), and a loop on the fine time step is carried out, with a call to the numerical solver `twostep.f90`. 
4. The `twostep.f90` routine calls the core routine `prodloss.f90` which calculates, for all processes, the loss and production fluxes and sums them. Thus `prodloss.f90` calls `emissions.f90` (emission fluxes), `transmix.f90` (transport and mixing fluxes), `chemistry.f90` (chemical fluxes) and `deposition.f90` (deposition fluxes). The routine `transmix.f90` calculates the PPM concentrations at the boundaries with routines contained in the `ppm.f90` package.

3.5 Parallelism

3.5.1 Domains division

Only the core of the model (chimere.e) has been parallelized. The meteo and preparation interfaces remain sequential tasks.

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 `nzdoms` and `nmdoms`, numbers of subdomains in each dimension. Then the main execution script launches `nzdoms*nmdoms+1` chimere.e processes in parallel, through the MPI command

```
mpirun -np nzdoms*nmdoms+1 ./chimere.e
```

One of these processes will be called the master, the others will be workers. There are as many workers as CHIMERE subdomains.

**Remark:** Only `nzdoms*nmdoms` processors are required to execute the code, because the master process is very light, and is generally working when workers are sleeping. Unless you are not limited in CPU number, don’t waste
In a first order analysis, we could consider that:

1. the master performs all initializations
2. the master sends to each worker its share of data, corresponding to its subdomain
3. the workers work ...
4. the workers send their results to the master
5. the master gathers results, and saves that to output files
6. the master sends new boundary conditions to workers

This would be a beautiful simple world, with birds and flowers ...

In a second order analysis, we realize that a worker process requires not only the share of data corresponding to its subdomains, but also a stripe of boundary data belonging to its neighbours. The reason is that to calculate a target value at point $i$, you need source values at point $i-3$, $i-2$, $i-1$, $i$, $i+1$, $i+2$, $i+3$. In two dimensions, of course.

So, the algorithm becomes:

1. the master performs all initializations
2. the master sends to each worker its share of data, corresponding to its subdomain, plus a surrounding “halo”
3. the workers work, and at each fine time step, update their halo by getting it from their neighbour
4. the workers send their results to the master
5. the master gathers results, and saves that to output files
6. the master sends new boundary conditions to workers

Not so complicated, isn’t it? Let us see where the Devil is hidden ...

In fact, to calculate a target value at point $i$ and time $t_{n+1}$, you need source values at point $i-3$, $i-2$, $i-1$, at time $t_{n+1}$, and source values $i$, $i+1$, $i+2$, $i+3$ at time $t_n$. You cannot simply calculate target values at time $t_{n+1}$ from a set of...
source values at time $t_n$. 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 realized 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.

Last, and hopefully least:
What we have just explained in the previous paragraph for the $i$ direction also applies to the $j$ direction.
The dependency 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 independent 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 reasonable 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.

3.5.2 Recommendations

The optimal number of subdomains $nzdoms \times nmdoms$ results from a tradeoff between the true computing time and the time wasted in message passing between processes. The message passing time is roughly proportional to the perimeters of the subdomains, and the true computation time to their area. For a 80x80 domain and the gas version, you will waste CPU time in message passing if you go beyond 4x4 subdomains. On the other hand, the same 80x80 simulation with the aerosols version, sea salts and 3 Gauss-Seidel iterations should still be efficient with 8x8 subdomains.

It is the user’s responsibility to try with an increasing number of subdomains (and an increasing number of processors, of course) on a typical forecast configuration, and to decide of the best choice.

Another technical point of detail, which may have severe impacts on performance, is the type of process locking used for collective communications. Suppose that you have divided your domain in 4 subdomains, and you run on 4 processors (defined in your MPI hostlist file). The main calling script will spawn 5 processes: one master, and 4 workers. One processor, generally the first in your hostfile, will be overbooked with the master and the first worker. This is generally not a problem, because master and worker do not work at the same time. But suppose the following case: the master has called the 4 workers, and they have a rendez-vous (synchronization) somewhere in the code. The workers work, and the master sleeps until the program reaches the rendez-vous. No problem, if the master process is really sleeping: the load of the first processor will be 1, produced either by the master, or by the first worker. But if your MPI library has chosen what they call spin-lock process locking, the master process will not sleep waiting. It will poll continuously to check if the workers have reached the rendez-vous. In that case, the load of the first processor will be 2: master (polling) and worker1(working) and the whole computing time will be roughly multiplied by 2. With LAM/MPI, this situation is avoided by selecting the sysv System Services Interface. This is the rationale of the command

```
mpirun -np \$((\$nzdoms*\$nmdoms+1)) -ssi rpi sysv ./chimere.e
```

found at the end of the main execution script.
If you have the opportunity to use many processors for CHIMERE, you can "waste" a processor for the master process. That is, for the 4 subdomains case, you will list 5 processors in your MPI hostlist file. In that case, and only in that case, you can let MPI choose spin-locking for collective communications: it will be faster to have the master polling continuously than waking it from a real sleep. This is achieved, with LAM/MPI, with this command replacing the previous one:
3.6 Core model input and output files

3.6.1 Input files

As discussed above, a CHIMERE run consists in several actions which are performed in the top calling script:

- running the interfaces,
- building the namelists and
- executing ./chimere.e.

This section describes the necessary input files to the third action. All these files, listed in the chimere.nml, must be present in the temporary directory before running the executable chimere.e. The top calling script copies and constructs these files in the temporary directory automatically. We also list the corresponding internal file name used during the run. In the following list, we define:

- $modeldir: the model directory
- $tmpdir: the simulation directory
- $outputdir: the output results directory
- inputdata.211001011.6: an example of chemistry data directory automatically produced by the new chemprep script.
- CONT5: an example of simulation horizontal domain

The files are:

- fnoutspec = '/$modeldir/chemprep/inputdata.211001011.6/OUTPUT_SPECIES.full': File containing the name of output species
- fnspec = '/$modeldir/chemprep/inputdata.211001011.6/ACTIVE_SPECIES': File of all active species names
- fnchem = '/$modeldir/chemprep/inputdata.211001011.6/CHEMISTRY': File containing preprocessed reactions
- fnstoi = '/$modeldir/chemprep/inputdata.211001011.6/STOICHIOMETRY': File containing stoichiometric coefficients
- fnrates = '/$modeldir/chemprep/inputdata.211001011.6/REACTION_RATES': File containing tables of reaction rates
- fnfamilies = '/$modeldir/chemprep/inputdata.211001011.6/FAMILIES': Definition of families
- fnphot = '/$modeldir/chemprep/inputdata.211001011.6/PHOTO_PARAMETERS': Tabulated zenithal angles
- fnaerosol = '/$modeldir/chemprep/inputdata.211001011.6/AEROSOL': aerosols; number of bins, molar weights
- fnaeromin = '/$modeldir/chemprep/inputdata.211001011.6/AEROMIN.bin': Parameters for ISORROPIA scheme for aerosols
- fnaeroorg = '/$modeldir/chemprep/inputdata.211001011.6/AEROORG.bin': Parameters for the SOA scheme
- fnprim = '/$modeldir/chemprep/inputdata.211001011.6/PRIMARY': Size distribution of aerosols
- fnsemivol = '/$modeldir/chemprep/inputdata.211001011.6/SEMIVOL'
- fnanthro = '/$modeldir/chemprep/inputdata.211001011.6/ANTHROPIC': List of anthropic emitted species
3.6.2 Output files

The following output files are listed in file `chimere.nml` (Their format is more precisely described in §2.4.2,p.28)

- **fnout** Timely concentrations of selected output species and families
- **fnothers** Timely meteorological parameters
- **fnconcs** Concentrations of active species at the end of the model run. Generally used as a restart file for subsequent runs.
- **fnfdepos** Timely depositions

Notes: Longitude and latitudes are now included into the METEO.nc file.

3.7 Running real cases

Here are the following steps to achieve for running a simulation with the model:
1. First step is to install all the code and data, ensure that LAM/MPI is installed, configure `nodelist.example` and `Makefile.hdr`. We assume this step has been done since you were able to run the test case.

2. Second step to achieve is to find meteorological data. There is no meteorological model provided with CHIMERE. CHIMERE can use many meteorological models but users should provide their own interface, except for MM5, for which an interface is proposed in the code, which takes as input standard MM5 output files (MMOUT files). If MM5 (which is a free software, available on the NCAR\(^1\) web site, is chosen as the meteorological driver, it has to be initialized with other global meteorological data (analyses), which often are not public. However the National Centers for Environmental Prediction (NCEP) and the National Center for Atmospheric Research provide a wide variety of meteorological data, analyses etc... online. Our experience deals with real-time AVN/NCEP analyses and forecasts that can be obtained on a web server. When setting up a meteorological simulation using MM5, choose a meteorological domain entirely containing the CHIMERE domain, and a time period entirely containing the CHIMERE period to simulate. The PBL option "MRF" (Option #5), based on the Troen and Mahrt (1986) parameterization should be preferred because it is most consistent with the model mixing formulation. The Schultz (Option #8) microphysics parameterization has also been tested with CHIMERE and is recommended.

Users of MM5-3.6.x should pay attention to the fact that the convective velocity is really badly estimated in the MRF PBL scheme, leading to too strong u* in the case of strong heat fluxes, and therefore a too strong dust emission. Users of MM5-3.7 should not worry about that, because the Beljaars parametrization has been included in the mainstream code.

Also in case of the use of MM5, the MM5 coordinate files (`COORD_MM5[dom]` and `COORD_MM5[dom]_DOT`) must be constructed from the MM5 TERRAIN file.

3. Prepare the land-parameters for the CHIMERE domain. There are 3 necessary land files: The coordinate file, the file containing the fractions in all of 9 land-use classes for all grid cells, and the file containing biogenic emission potentials. The directory provided online (`prepdomains`) only works for Europe. It works with the Global Land Cover Facility (GLCF) land-use data base (see documentation below). Other methodologies should be found for other parts of the world, in particular for constructing the emission potentials. Perhaps as a very first experiment for a simulation outside Europe it could be advised to remove them by setting a `BIOFACS_[domain]` file with 5 columns with zeros in them. `LANDUSE_[domains]` files are easier to construct from the GLCF data base.

4. Emission files have to be constructed and specified for the chosen mechanism. This is often a difficult task. If the standard emission procedure is followed (one emission file per calendar month containing hourly emissions for all MELCHIOR emitted species and for 3 typical days (week day, saturday and sunday), the interface program `prepmis.f90` proposed here (in the `src_[model]` directories can be used. Otherwise the interface program must be recoded.

5. The top calling script (`chimere-mm5` for instance) must be modified according to the conditions of the simulation. Edit this Bash-shell script and make changes in the values of the variables as required.

6. Run the model by invoking the top calling script.

   ```
   nice -10 chimere.sh | tee chimere.log
   ```

### 3.8 How to add a variable in the code?

#### 3.8.1 From the master to workers

Using previous versions of the model, to add a new parameter was an easy task. With the parallelized version, this is a little bit more complicated. If you need to know a value in the core model, but using some data estimated

\(^1\)http://www.ncar.ucar.edu
with the pre-processors, you need to transfer informations from the master to the workers. Basically, 5 subroutines have to be updated if you want a new variable (real or integer, scalar or array). We present an example based on the deep convection implementation: to know if we are in a convective cell or not, we use a flag named 'ideep(nzonal,nmerid)'. This array has to be hourly updated in the 'locvalues.f90' subroutine (for the estimation of variables at the chemical time-step).

1. src/modules/message_defs.f90:

   integer,parameter :: ias_ideep =__LINE__

2. src/modules/master_message_subs.f90:
   From the master to the worker in send_frac_hourly_int_arrays(ip,dom(ip))

   integer,dimension(:,,:),allocatable :: ibuf2
   allocate(ibuf2(nzcount,nmcount))
   ibuf2=ideep(izstart:izend,imstart:imend)
   call mpi_send(ibuf2, nzcount*nmcount, mpi_integer,ip, ias_ideep, mpi_comm_world,ierr)
   deallocate(ibuf2)

3. src/modules/chimere_common.f90:

   integer,dimension(nzonal,nmerid) :: ideep

4. src/model/worker_common.f90:

   integer,dimension(nzonalmax,nmeridmax) :: ideep

5. src/model/worker_message_subs.f90:
   The worker receives the information from the master in recv_frac_hourly_int_arrays

   integer,dimension(:,,:),allocatable :: ibuf2
   allocate(ibuf2(nzonal,nmerid))
   call mpi_recv(ibuf2, nzonal,nmerid, mpi_integer,0, ias_ideep, mpi_comm_world,mpi_status_ignore,ierr)
   ideep(1:nzonal,1:nmerid) = ibuf2(:,;)
   deallocate(ibuf2)

3.8.2 From workers to the master

If you need to know the values of a variable calculated in the core worker model (i.e., to write values to an output file), you need to transfer information from the worker to the master, as only the master includes the appropriate subroutines for writing output files. Four source files have to be modified if you want to pass a variable from workers to the master (real or integer, scalar or array). Consider an example of a photolysis rates variable (phrate worker array) that could be computed in a radiation scheme. Array values are updated hourly, based on clear-sky values per photolysis reaction included in the PHOTO_PARAMETERS file, the zenith angle calculated by the model, and the altitude. Therefore, a 4-dimensional array phrate(nphotmax,nzonal,nmerid,nverti) (nphotmax is the maximum number of photolysis reactions considered in CHIMERE, nzonal is the number of zonal cells for each worker core, nmerid is the number of meridian cells for each worker core, and nverti is the number of vertical layers) is calculated every hour. With this convention, source files that we need to modify are the following.
1. `src/modules/chimere_common.f90`:
   Define an array in the master COMMON, with the same type and dimensions as the phrate array,
   
   ```
   real(kind=8), dimension(nphotmax,nzonal,nmerid,nverti) :: phrate
   ```
   
The name of this array can be either the same as in the worker (as in this example, `phrate`) or a different one.

2. `src/modules/message_defs.f90`:
   Define an integer index for receiving the array by the master:
   
   ```
   integer,parameter :: iar_phrate =__LINE__
   ```

3. `src/modules/master_message_subs.f90`:
   We usually want to receive the new variable by the master every time it is updated. This updating frequency is essential in order to decide in which master subroutine we should receive the new variable. In this example, `phrate` is updated hourly so, following the master data flow (see `src/main/integrun.f90`), the most appropriate master subroutine is,
   
   ```
   subroutine recv_locvalues(ip,dom)
   ```
   
   Since we need a 4-dimensional array buffer, we have to include in this subroutine:
   
   ```
   real(kind=8), dimension(:,:,:,:), allocatable :: dbuf4
   ```
   
   Then we call `mpi_recv` as follows:
   
   ```
   allocate(dbuf4(nphotmax,nzcount,nmcount,nverti))
   call mpi_recv(dbuf4, nphotmax*nzcount*nmcount*nverti, 
                mpi_double_precision, ip, iar_phrate, 
                mpi_comm_world,status,ierr)
   phrate(1:nphotmax, izstart:izend, imstart:imend,1:nverti) = dbuf4
   deallocate(dbuf4)
   ```

4. `src/model/worker_message_subs.f90`:
   We have to send the original `phrate` array values from the worker. Again, following the worker data flow (see `src/model/worker.f90`), the most appropriate subroutine to do that in the worker in our example is:
   
   ```
   subroutine worker_send_locvalues
   ```
   
   We have to include in this subroutine:
   
   ```
   real(kind=8), allocatable, dimension(:,:,:,:) :: dbuf4
   ```
   
   Then we call `mpi_send` as follows:
   
   ```
   allocate(dbuf4(nphotmax,nzonal,nmerid,nverti))
   dbuf4(:,:,:,:) = phrate(1:nphotmax,1:nzonal,1:nmerid,1:nverti)
   call mpi_send(dbuf4, nphotmax*nzonal*nmerid*nverti, 
                mpi_double_precision, 0, iar_phrate, 
                mpi_comm_world,ierr)
   deallocate(dbuf4)
   ```

We highly recommend to understand both `integrun.f90` and `worker.f90` data flows before applying any of those changes.
Chapter 4

[domains]: Domains and landuse

The first step for a personal CHIMERE configuration is to define a new coordinate file and all associated landuse data. Thus, a grid has to be built and tools are proposed in the CHIMERE suite. The starting point is to build the COORD_<domain> file (where <domain> is the name of the simulation domain). This file enables to:

- Create all landuse data files needed for the deposition and the biogenic emissions,
- Create all anthropogenic emissions files with the own EMEP emissions interface

In this model version, the landuse data are automatically calculated. The user has only to fill in his new domain name and coordinates in the domains/domainlist.nml. During the first run over this domain, the landuse programs will be launched and will create all data files for CHIMERE.

4.1 The domains/domainlist.nml file

All landuse files need only to be constructed once per model domain. The interface is now automatically called by the top calling script. If the domain already exists, the run continues. If the user defines a new domain, the domain preparation interface is launched in the scripts/make-chemgeom.sh script. The only job to do for the user is to give minimum informations in the parameter file $modeldir/domains/domainlist.nml.

<table>
<thead>
<tr>
<th>domain</th>
<th>NX</th>
<th>NY</th>
<th>DX</th>
<th>DY</th>
<th>XMIN</th>
<th>YMIN</th>
</tr>
</thead>
<tbody>
<tr>
<td>CONT1</td>
<td>65</td>
<td>33</td>
<td>0.5</td>
<td>0.5</td>
<td>-10.0</td>
<td>40.5</td>
</tr>
<tr>
<td>CONT3</td>
<td>67</td>
<td>46</td>
<td>0.5</td>
<td>0.5</td>
<td>-10.5</td>
<td>35.0</td>
</tr>
<tr>
<td>CONT4</td>
<td>70</td>
<td>44</td>
<td>0.5</td>
<td>0.5</td>
<td>-10.5</td>
<td>36.0</td>
</tr>
<tr>
<td>CONT5</td>
<td>79</td>
<td>47</td>
<td>0.5</td>
<td>0.5</td>
<td>-14.0</td>
<td>35.0</td>
</tr>
<tr>
<td>FRA10</td>
<td>101</td>
<td>111</td>
<td>0.15</td>
<td>0.1</td>
<td>-5.0</td>
<td>41.0</td>
</tr>
</tbody>
</table>

For example, the line describing the CHIMERE ‘CONT3’ domain corresponds to the mesh displayed in the Figure 4.1. The first column, the <domain> is a name given by the user to the model horizontal grid/domain. The other columns are the number of cells in the x-direction, the number of cells in the y-direction, and the coordinates of the South-West cell center, as follows:

4.2 The main scripts and programs

The proposed domain interface is based on the Global Land Cover Facility (GLCF\(^1\)) 1km×1km resolution\(^2\) database from the University of Maryland, following the methodology of [Hansen and Reed, 2000]. This global land cover classification is based on the imagery from the AVHRR satellites analyzed to distinguish fourteen land cover classes (Table 4.1).

---

\(^1\)http://glcf.umbc.umd.edu/data/landcover
\(^2\)Actually, the resolution of the source files used is 0.08°
Figure 4.1: Examples of horizontal domains. The dots represent the MM5 EUR1 grid cells centers and the rectangle represents the boundaries of the CHIMERE CONT3 domain.

The scripts and programs are:

- `domains/prep_glcf.sh`. Runs `prep_glcf.e` program.
- `domains/src/prep_glcf.f90`. Extracts original GLCF land cover classes data for the specified Chimere domain. It takes `<cont>0500ag.asc` global data files (see below) as input and generates `LAND_GLCF.bin` for the chosen domain.
- `domains/prep_landuse.sh`: Uses the domain names as input argument and runs the program `landuse.f`.
- `domains/src/landuse.f`: Calculates the 9-category fractions on the CHIMERE grid from the GLCF classes. It takes the `LAND_GLCF.bin` file generated by `prep_glcf.e` and creates `LANDUSE_<domain>`.

The raw data files are now provided in a separate directory. This directory is called "BIGFILES" in the Chimere top-calling script and is the location of the big files shared by all users. All data are:

- `$BIGFILES/<cont>0500ag.asc`, where `<cont>` is 'af' for Africa, 'ea' for Eurasia, 'na' for North America, 'sa' for South America, and 'ap' for the rest of the World. Ascii files containing the original GLCF land cover type index [0-13], i.e., the land cover class value prevailing in each 0.0083° grid cell (see the GLCF documentation).
- `$tmpdir/domains/LAND_GLCF.bin`: GLCF land cover classes for the specified Chimere domain. Generated by `prep_glcf.e` when `domains/<domain>/LANDUSE_<domain>` is absent.
- `domains/<domain>/LANDUSE_<domain>`: The land cover classes aggregated to the nine Chimere categories on the Chimere grid. This is the file used by Chimere.
- `domains/src/LAND_AGGREGATION`: This is the matrix of correspondence between the 14 GLCF categories and the nine CHIMERE categories, used by `landuse.f`.

The interface can take 5-30 minutes (or more...). At the end, a directory with the domain name is created containing the necessary files for CHIMERE.
### Table 4.1: GLCF land cover types

<table>
<thead>
<tr>
<th>Number</th>
<th>Land Cover Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Water</td>
</tr>
<tr>
<td>1</td>
<td>Evergreen Needleleaf Forest</td>
</tr>
<tr>
<td>2</td>
<td>Evergreen Broadleaf Forest</td>
</tr>
<tr>
<td>3</td>
<td>Deciduous Needleleaf Forest</td>
</tr>
<tr>
<td>4</td>
<td>Deciduous Broadleaf Forest</td>
</tr>
<tr>
<td>5</td>
<td>Mixed Forest</td>
</tr>
<tr>
<td>6</td>
<td>Woodland</td>
</tr>
<tr>
<td>7</td>
<td>Wooded Grassland</td>
</tr>
<tr>
<td>8</td>
<td>Closed Shrubland</td>
</tr>
<tr>
<td>9</td>
<td>Open Shrubland</td>
</tr>
<tr>
<td>10</td>
<td>Grassland</td>
</tr>
<tr>
<td>11</td>
<td>Cropland</td>
</tr>
<tr>
<td>12</td>
<td>Bare Ground</td>
</tr>
<tr>
<td>13</td>
<td>Urban and Built</td>
</tr>
</tbody>
</table>

### 4.3 Details on output files

#### 4.3.1 Horizontal domain COORD

Model domains are entirely defined by their grid cell centres description. It is given in the \texttt{COORD_<domain>} file which must be placed in the \texttt{domains/} directory. The \texttt{COORD_<domain>} file contains lines with a longitude (in decimal degree) and a latitude (in decimal degree). Each pair of coordinates stands for a grid cell centre, described (from top to bottom of file) from West to East then from South to North.

In the definition of a new CHIMERE domain, the user must check carefully whether the domain is quasi-rectangular. Most projection systems work, even a regular grid in geographic coordinates (longitude-latitude) provided the resolution is not too coarse (say > 2 degree).

Model grids can be any quasi-rectangular grid, with weakly varying spatial step. It is assumed, in particular for the Parabolic Piecewise Method for transport that locally (over 5 consecutive cells) the grid size is constant in each direction, with sizes equal to that of the locally central cell. Sphericity effects are taken into account, but therefore linearized. It is important to note that the COORD file for CHIMERE is absolutely not the same that the one used for MM5 (or others meteorological inputs). An example is displayed in the Figure 4.1. If the user wants to define his own horizontal grid, we propose a script described in §B.2, p.122.

#### 4.3.2 Vertical grid VCOORD

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:

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

where $P_{surf}$ is the surface pressure and the coefficients $a_k$ and $b_k$ must be given. These hybrid coefficients $a_k$ and $b_k$ are provided by the user as an input file, \texttt{VCOORD.[vname]} in the \texttt{data/vgrid/} directory. [vname] is a user-defined name. As an option, the vertical grid can be defined automatically given the top pressure and sigma at the top of the surface layer. Then the resolution varies upward as a geometric progression. If the user wants to define his own vertical grid, we propose a script described in §B.3, p.123.
4.3.3 Land-use types

The LANDUSE_<domain> file contains the fraction of land use per grid cell (numbers between 0 and 1). In the chimere2008 model version, 9 land-use types are recognized. They are described in the Table 4.2, with the correspondence to the GLCF (University of maryland) data base used for the propoposed prepdomains interface.

<table>
<thead>
<tr>
<th>Category Number</th>
<th>Description</th>
<th>Corresponding GLCF</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Agricultural land / crops [11]:</td>
<td>Cropland</td>
</tr>
<tr>
<td>2</td>
<td>Grassland [10]+1/2[7]:</td>
<td>Grassland+1/2Wooded grassland</td>
</tr>
<tr>
<td>3</td>
<td>Barren land/bare ground [12]:</td>
<td>Bare ground</td>
</tr>
<tr>
<td>4</td>
<td>Inland Water</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Urban [13]:</td>
<td>Urban and built-up</td>
</tr>
<tr>
<td>6</td>
<td>Shrubs [8]+[9]:</td>
<td>Shrublands</td>
</tr>
<tr>
<td>7</td>
<td>Needleleaf forest [1]+[3]+1/2([5]+[6])+1/4[7]</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Ocean [0]:</td>
<td>Water</td>
</tr>
</tbody>
</table>

Table 4.2: Landuse categories used in CHIMERE
Chapter 5

[chemprep]: The chemical preprocessor

The chemprep preprocessor is completely new for this version chimere2008. The new procedure is as follows:

- 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 chemprep/chemprep.univ.sh is executed to create all prerequisite files.
- This script will create a new directory in the main directory chemprep and called something like input-data.2110011210.6 where the arguments following the string inputdata. correspond to the simulation options. In this new directory, all input files required by CHIMERE for a simulation are stored.
- If a new simulation with the same options is launched or in case of a restart, the files will be used as it. If the main run options change, the script will be automatically launched again to create new files.

5.1 Options available in the main script chimere.sh

These options are listed in the main script chimere.sh (p.21) as follows:

- mecachim option for the gas-chemistry mechanism.
  No gas phase chemistry [0];
  melchior complete [1];
  melchior reduced [2]

- outresol option for the output species accuracy for aerosols. Possible values are:
  low resolution [0],
  full resolution [1],
  tonest resolution [2].

- nbins option for the aerosols bins number.
  The cut off diameters 2.5 and 10 um are respected.
  You can change the parameter dmax in ./chemprep/chemprep.univ.sh in order to adapt the size distribution.
  This script is adapted for nbins=4 and nbins=6.

- aero options for simulation of aerosols or not.
  [0=no; 1=yes]
  (if yes : Minimal configuration with NO3, NH4, SO4, PPM, Sec. Org. Aer., WATER)

- seasalt option for the sea salts simulation.
  none=0,
  inert=1,
  active=2.
  "active" means sea salts react with chemistry
These dusts are considered as emitted within the domain (mainly resuspension). [0] if not, [1] if yes.

Persistent Organic Pollutants.

pops=1 only if meca<0 and aero=1

Dust considered only at the model domain boundaries.
dustout=0 is the default value
dustout=1 only if aero=1

Primary carbonaceous species
[0=no; 1=yes]
(if yes, OCAR and BCAR are used)

Specify the number of tracers:
trc=0, no tracer

SOA chemistry:
no SOA chemistry [0],
simple [1],
medium [2]
(or complex [3], not yet available!!).
Note that soatyp>0 only possible if aero = 1

With these options, the script scripts/make-chemgeom.sh is launched.

5.2 The first script make-chemgeom.sh

This script, displayed in p.119, checks if input data files are already existing or not. If not, new ones are created by using the chemprep/chemprep.univ.sh script. The "label" for the output directory is built in this script as follows:

labchem=mecachim $aero $seasalt $pops $dustout $carb $trc $soatyp $dust $outresol $nbins

For example and for a simulation with only gas-phase chemistry and no aerosols, but with a passive tracer, the directory created will be with the label "1000010000.0".

5.3 The second script chemprep.univ.sh

The main program is the script chemprep/chemprep.univ.sh, p.124. It uses the programs included in the directory chemprep/src:

- chemprep-distrib.f: Program for the calculation of aerosol size distribution. Note that the minimum and maximum value of the aerosols diameters are fixed with the values: $D_{min}=0.001\mu m$ and $D_{max}=40\mu m$. The user may easily change these values in the chemprep.univ.sh script.
- chemprep-families.f: Reads all species and builds the file containing the name and contents of families.

5.4 Manual changes of the mechanism

The chemical preprocessing scripts allows to modify reactions, add reactions or even completely change the chemical mechanism. It allows an easy writing of reactions and from two initial input files REACTIONS and PHOTORATES produces other files which are read by chimere. Once the two input files are prepared by the user, the preprocessor is invoked by running the make-chemgeom.sh script.

The conditions for the set of reactions to be used are listed below:
• Only reactions of the type $A_1(+B_1(+C_1,...)) \rightarrow (a^*)A_2+(b^*)B_2+...$ are accepted. Letters with index 1 denote the reactants and those with index 2 the products. $a$, $b$, (if any) are stoichiometric coefficients. If omitted they are considered to equal 1. Negative stoichiometric coefficients are accepted, but not recommended since it can lead to negative concentrations. Stoichiometry in the reactants is not allowed.

• Only species which are consumed at least once in the mechanism will be considered as active species in the Chimere 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 $A \rightarrow X$ ($A$ is the concerned product), with a negligible reaction rate, i.e. $k=1e^{-10}$ (s$^{-1}$).

• There are special species whose concentration is only prescribed in the model. These are H2O (water vapor), N2 (molecular nitrogen), O2 (molecular oxygen) and 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 Chimere code and are not an easy issue.

• Reactions have a maximum of 4 reactant species and as many products as desired (less than 1000!). To increase the number of reactants one needs just to change the parameter nreactamax in the header files src/chimere_params.f90.sed and src/worker_params.f90.sed.

• It may be desirable that stoichiometric coefficients vary with some parameters like temperature. This is however not possible here in this version.

• All reactions will be processed by the solver in an equivalent manner. There is no family grouping (like often used for the QSSA solver) available.

• Clear sky photolysis rates must be precalculated by the user, and depend only on the zenith angle Z. No dependence on height is allowed, even in the presence of clouds. The proposed tabulated values correspond to an average value between 0 and 1000m. This should be a good compromise for the boundary layer. Introducing a dependence with height will be considered in a future model version. The 12 Tabulated J values must correspond to zenith angles of (in degree) 90,88,86,83,80,75,70,60,45,30,15,0

• 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. See syntax in the next section.

• 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.

<table>
<thead>
<tr>
<th>Type number</th>
<th>Name</th>
<th>Math</th>
<th>syntax in REACTIONS file</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Constant K</td>
<td>$K$</td>
<td>$k=value$</td>
</tr>
<tr>
<td>2</td>
<td>Simplified Arrhenius</td>
<td>$K=Aexp(-B/T)$</td>
<td>$k(T)=Aexp(-B/T), A=value, B=value$</td>
</tr>
<tr>
<td>3</td>
<td>Arrhenius</td>
<td>$K=Aexp(-B/T)(300/T)^{**N}$</td>
<td>$k(T)=Aexp(-B/T)(300/T)^{**N}, A=value, B=value, N=value$</td>
</tr>
<tr>
<td>4</td>
<td>Troe or Falloff</td>
<td>$K=K_0^{<em>E/(1+M</em>(K_0/K_\infty))}$</td>
<td>$K(T,M)=troe(A0,B0,N0,A_\infty,B_\infty,N_\infty,J)$</td>
</tr>
<tr>
<td>5</td>
<td>Photolysis</td>
<td>$J=J(Z)$</td>
<td>$J(Z)=photorate(label)$</td>
</tr>
<tr>
<td>6</td>
<td>Special type</td>
<td>$K=K_1*K_2/(1+K_2)$</td>
<td>$k(T)=SPECIAL_1(A1,B1,A2,B2)$</td>
</tr>
</tbody>
</table>

2 $label$ is a label standing for recognition of chemprep in the PHOTORATES file. All photolysis rates must appear tabulated in this latter file, the tabulated values corresponding to specific zenith angles (see above).

3 * Special form for radical recombination (not used in this mechanism)
7 | Special type | $k(T) = k_1(T)/1 + k_2(T)$  
| | $k_1 = A_1 \exp(\frac{-B_1}{T})$  
| | $k_2 = A_2 \exp(\frac{-B_2}{T})$  

8 | Special type | $k(T) = 2\sqrt{f_1 f_2 f_3 f_4 / (1 + f_3 f_4)}$  
| | $f_1 = A_1 \exp(-B_1/T)$  
| | $f_2 = A_2 \exp(-B_2/T)$  
| | $f_3 = A_3 \exp(-B_3/T)$  
| | $f_4 = A_4 \exp(-B_4/T)$

9 | Special type | $k(T) = 2\sqrt{f_1 f_2 (1 - \sqrt{f_3 f_4})(1 - f_4)/(2 - f_3 - f_4)}$  
| | $f_1 = A_1 \exp(-B_1/T)$  
| | $f_2 = A_2 \exp(-B_2/T)$  
| | $f_3 = A_3 \exp(-B_3/T)$  
| | $f_4 = A_4 \exp(-B_4/T)$

10 to 12 | Not used

13 | Aggregated ozone photolysis | $J = J_0 \frac{H_2O}{H_2O + M(0.02909 \exp(70/T) + 0.06545 \exp(110d0/T))}$  
| | $H_2O =$ water vapor (molec/cm$^3$)  
| | $M =$ air density (molec/cm$^3$)

14 | Modified Troe | $K(T,M) = \text{troe}(A_0, B_0, N_0, A_{\infty}, B_{\infty}, N_{\infty}, f)$

15 to 20 | Not used

### 5.5 Syntax of the REACTIONS file

Following are the rules for writing the REACTIONS file:

- 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.
- A reaction is written as:  
  
  $A_1 (+ A_2 (+ A_3 (+ A_4)) \rightarrow (a_1*) B_1 + (a_2*) B_2 + (a_3*) B_3 ...$  
  
  Rate expression where $A_i$ are the reactants (max=4), $B_i$ the products and $a_i$ are possible stoichiometric (written if different from 1).
- Rate expression follows the syntax rules defined in the above table.
- Reactants can also be one (or several) of the prescribed species (H$_2$O, N$_2$, O$_2$ or M).
- A1+A1(+A1)...->... are allowed (repetition of the same reactant, like HO2+HO2...
- Syntax like A1+A2-A3 is NOT allowed
- Negative stoichiometric coefficients are allowed, but not recommended (correctness of results not guaranteed because never tested). In that case, however, use the syntax ±a1*B1 in the products.
- Families are defined using the syntax:  
  
  FAM=A1+A2+A3...  
  
  Ai must be active species, and can be repeated as many times as desired. There is no limitation to the number of family elements.

### 5.6 Syntax of the PHOTORATES file

The syntax is very simple: Each line contains information for one photolysis reaction. There can be more photolysis reactions than in the REACTIONS file. The unused ones are ignored. There must be an exact correspondence to:

- User can define a new type. Changes must be done in chemprep.awk (recognition of a new syntax), Binichem.awk (define the number of constant ltabrate, and in Brates.f (new math expression).
- Aggregation of the ozone photolysis with reactions with atomic oxygen. The photolysis rates must be those of ozone only, hence the budget reaction is O$_3$->2*OH.
- Fall-off specific to NO$_2$+OH+M->HNO3
- User can define a new type. Changes must be done in chemprep.awk (recognition of a new syntax), Binichem.awk (define the number of constant ltabrate, and in Brates.f (new math expression).
between the labels of the photolysis reactions defined in the REACTIONS file. This label is the first column. The other columns contain the 12 tabulated photolysis rate values for the zenith angles from 90 to 0 degrees (see above table).
Chapter 6

Emissions and depositions

6.1 Emissions

Two types of emissions are managed in the model:

- §6.1.1: the anthropogenic emissions: need some pre-processing calculations done by the user,
- §6.1.2: the biogenic and natural emissions: are automatically calculated by the CHIMERE preprocessor using landuse data.

6.1.1 Anthropogenic surface emissions

The Figure 6.1 shows the two main steps to provide surface anthropogenic emissions to the CHIMERE model.

- Step 1: The user can use any type of emissions data as inputs. But at the end of his own preparation, he must provide monthly files anthropogenic emissions (except soil NO) given in 12xNSPEC files corresponding to each month and to each of NSPEC emitted species EMIS.<domain>,<MM>,<SPEC>,[sp].nc. They are given for three typical days: Week day, Saturday, Sunday and for the 24 UT hour. MM is the calendar month, SPEC is emitted species, "s" stands for surface and "p" for point emissions.
- Step 1 bis: Another way, but only over Europe, is to use the emiSURF emissions interface provided in this model distribution.
- Step 2: The CHIMERE pre-processor called prepemis will make automatically emissions files specifically for the period to run.

Figure 6.1: Flowchart of the several steps to produce CHIMERE compliant anthropogenic surface emissions fluxes
6.1.1.1 Before the step 1

With this model version, we propose raw data as follows:

- **EMEP data:** Annual data of anthropogenic emissions for the four classes NOx, SO2, CO, PM10, PM2.5, NH3 and NMVOC are taken from the EMEP data base ([Vestreng, 2003](http://www.emep.int)) the EMEP-MSCWEST web site.

- **LA data:** PM2.5 can be replaced by organic and black carbon emission species (respectively OCAR and BCAR) given by the Laboratoire d’Aerologie / OMP (thanks to Cathy Lioussse, Bruno Guillaume and Robert Rosset, [http://www.aero.obs-mip.fr](http://www.aero.obs-mip.fr)).

- **POPs data:** Additional data for benzo(a)pyrene, benzo(k)fluoranthene and benzo(b)fluoranthene (respectively BaP, BbF and BbF) can be requested on the EMEP-MSCEAST web site ([http://www.msceast.org](http://www.msceast.org)).

These data are processed altogether using the emiSURF interface during the step 1.

6.1.1.2 Step 1: The emiSURF interface

**Surface emissions**

The main script is `emis-surf.sh`. Several steps are performed with this interface.

- The first one is to project raw EMEP data onto the CHIMERE grid. This is done using a 1km resolution landuse database in order to preserve accuracy on emissions spatialization. The data are an annual inventory of VOC, NOx, CO, SO2, NH3, OCAR, BCAR, BaP, BbF, BbF and PM per activity sectors.

- A time shift is estimated over Europe to taken into account the 'summer time’. It is done by country, from April to September a one hour shift is added.

- A spatial interpolation (ensuring mass conservation) is done for every sets of emissions providing mass over the chosen CHIMERE domain.

- Finally, all data are merged into a single file, after an aggregation and speciation steps step (from SNAP species to real species to model species). These emissions are now over the CHIMERE grid (nzonal × nmerid), by SNAP sectors (nsectors), for three day types (ntypedays) and for the 24 hours (nhourpdays). ntypedays is 3, nt=1 standing for weekday emissions, nt=2 for saturday emissions and nt=3 for sunday emissions. This does not allow, in particular, to have emissions depending on meteorology or other factors than the day of week, but this keeps databases relatively compact.

□ Then the VOC and NOx are disaggregated into real compounds using a speciation profile (depending on activity sector). In the data provided on CHIMERE server, the NOx speciation is 90% of the NO2-equivalent mass of NOx is NO, 9.2% is NO2 and 0.8% is HONO. The VOC speciations come from either the british PORG speciation for the continental model or from the University of Stuttgart (IER) for the Ile-de-France case. The british speciation is public.

□ The construction of model-species emissions then requires a [Middleton et al., 1990](http://example.com) type of aggregation procedure. Each real VOC emissions are aggregated into one or several model VOC(s), with a branching coefficient and a reactivity-dependent coefficient (see the Middleton procedure), for which reactivities and molar masses for both the real and the model VOCs have to be known. In the british speciation, 227 real VOCs are considered, and the following AGGREGATION file gives these informations for the MELCHIOR mechanism. Reactivities have been calculated on the basis of the Leeds University MCM Master Mechanism and the Kwok structure/reactivity approach. The SPECIATION files gives the VOCs list and the 9-activity speciation, and the way VOCs are preprocessed in CHIMERE for continental scale.

The resulting files `EMIS.<domain>,<MM>,<SPEC>.s.nc` must contain emissions for the model species specified in the file `ACTIVE_SPECIES`. In the complete version of the model, the model needs:
<table>
<thead>
<tr>
<th>Chemicals</th>
<th>Symbols</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nitrogen monoxide</td>
<td>NO</td>
</tr>
<tr>
<td>Nitrous acid</td>
<td>NNO</td>
</tr>
<tr>
<td>Ammoniac</td>
<td>NH3</td>
</tr>
<tr>
<td>Methane</td>
<td>CH4</td>
</tr>
<tr>
<td>n-Butane</td>
<td>NC4H10</td>
</tr>
<tr>
<td>Ethane</td>
<td>C2H6</td>
</tr>
<tr>
<td>Isoprene</td>
<td>C5H8</td>
</tr>
<tr>
<td>Formaldehyde</td>
<td>HCHO</td>
</tr>
<tr>
<td>Methyl ethyl Ketone</td>
<td>CH3COE</td>
</tr>
<tr>
<td>Primary particulate matter</td>
<td>PPM_fin</td>
</tr>
<tr>
<td>Primary particulate matter</td>
<td>PPM_coa</td>
</tr>
<tr>
<td>Benzo(a)pyrene</td>
<td>BaP_fin</td>
</tr>
<tr>
<td>Benzo(b)fluoranthene</td>
<td>BbF_fin</td>
</tr>
<tr>
<td>Primary organic carbon</td>
<td>OCAR_fin</td>
</tr>
<tr>
<td>Primary black carbon</td>
<td>BCAR_fin</td>
</tr>
</tbody>
</table>

Primary particle species (H2SO4, PPM, BaP, BbF, OCAR, BCAR) are split over three modes:

- XXX_fin : $\phi < 2.5 \mu m$
- XXX_coa : $2.5 < \phi < 10 \mu m$
- XXX_big : $\phi > 10 \mu m$

Some additional details:

- PPM_fin refers to PM2.5, PPM_coa to PM10-PM2.5. H2SO4, BaP, BbF, OCAR, BCAR are assumed to be in the fine mode.
- In the present version, emissions of Methanol and Ethanol are also included but not used. They are only used in the complete version of the chemical mechanism (MELCHIOR1).
- In case of the use of the EMEP data, the data are spatially interpolated from the EMEP grid onto the CHIMERE grid passing through a 1 km grid which gives the land use data over Europe at this resolution.
- The NMVOC emissions have to be split into 9 classes represented within the model chemical mechanism. To get this goal, they are first distributed for each country into different activity sectors (traffic, solvents, industrial and residential combustion, others), according to data prepared by IER (Institute for Energy Economics and Rational Use of Energy, University of Stuttgart; [Society, 1994]). For each sector, NMVOC emissions are then split into 32 classes with similar structure and reactivity, and using standard NMVOC profiles again from IER [Society, 1994]. Third, NMVOCs from these 32 classes are aggregated into the 9 classes represented within CHIMERE, by applying mass and reactivity weighting as proposed by [Middleton et al., 1990]. It should be noted that the uncertainty in the spatial distribution of individual VOC emissions is large, given that the VOC profiles are assumed to be the same all over Europe, with the exception of traffic, where national differences (e.g. the ratio gasoline / diesel) are taken into account. Monthly, daily and hourly variations of the emissions are modeled by imposing respective variations from the [Society, 1994] data base.
- In rural areas, NO emissions from microbial processes may be an important source. Since these emissions strongly depend on temperature, they are processed in the model as "biogenic" emissions and often called in this manual "biogenic NO emissions". In soils NO is produced in a reaction chain of oxidation and reduction from ammonium which is used in fertilizers. CHIMERE uses a European inventory of soil NO emissions from [Stohl et al., 1996]. This inventory estimates the soil emissions to be during the summer months at the order of about 20% of the emissions from combustion on a European average but with large differences between the countries. In the model, these NO emissions are only considered during the months of May to August.

### Point emissions

The program **emis-point.sh** creates emission files for the point sources required by CHIMERE if the flag "point_emissions" is set to 1 in the top calling script. You can treat all point emission sources anthropic (e.g.
Chimere

stacks) or biogenic (e.g. volcanoes). The script creates 12xNSPEC monthly files according to the time variation set in inputdata directory (SEASONAL-FACS.data and HOURLY-PROFS.data).

- You have to create a file that contains all emission source points in the sources directory named SOURCES-

  <year>: 
  
  | Flag for treatment in prepemis.f90: |  
  |---|---|
  | 0 | Put the emission at altitude "height" in the corresponding CHIMERE level |
  | 1 | Put the emission at altitude "height" + "additional height" due to thermal and dynamical processes, in the corresponding CHIMERE level. |
  | 2 | Use Plume in Grid treatment (not yet available in CHIMERE |
  | Long | Source point longitude |
  | Lat | Source point latitude |
  | Country | Country code number |
  | Snap | Activity sector code |
  | temp | Temperature (°C) |
  | Vel | Emission velocity (m s⁻¹) |
  | Diam | Diameter (m) |
  | Height | Stack height (m) above ground level |
  | BCAR, etc. | List of species emissions : in t y⁻¹ |

It is advisable to have the same species list as for the surface emissions. One line in the file corresponds to one source point. As an example the "SOURCES-2003" file provided in the sources directory corresponds to the volcanoes in Italy (Lipari, Stromboli and Etna). Be careful, for the moment no point-source emissions are available in the EMEP inventory. Altitudes instead of heights above the ground level are used for volcanoes.

- You need to change some parameters and paths in emis-point.sh: input and output directories, domain, year, etc.

- After this you are ready to start the emission preprocessing by launching emis-point.sh. If the program runs successfully, you will obtain 12xNSPEC resulting files EMIS.<domain>.<MM>.<SPEC>.p.nc for each month and for each species. Each file represents emissions in molecule cm⁻² s⁻¹ for each sources at each CHIMERE grid cell for a given day type and hour of the day.

At the end of Step 1, you need to have one file per month and per species called EMIS.<domain>.<MM>.<SPEC>.[sp].nc projected on the Chimere grid and for several day types. The format is presented in §E.1, p.147.

6.1.1.3 Step 2: the prepemis interface

The Step 2 consists in making the hourly, simulation adapted, AEMISSIONS.nc file from the database. If we forget the netCDF overhead, this is a fairly simple procedure, achieved by the src/prep/prepemis.f90 program. Take a look at this program. It calls a calendar subroutine idaytype that identifies day types and possibly extra fixed holidays taken as Sundays (Christmas, 1st January, ...). Beware that these days are for France. One may want to edit this subprogram (idaytype in src/tools/calendar.f90) in order to change some specific dates.

At this second stage there is a possibility to apply a flat reduction (or increase) on all primary pollutants, in order to account for time shift between the simulation year and the inventory year. Unfortunately the possibility is not given to distinguish factors for various pollutants, but this is not a difficult change in src/prep/prepemis.f90.

When the chemprep/inputdata directory is created for the simulation, and if the simulation requires anthropic emissions, the ANTHROPIC file is generated with all the species required for the simulation. If the carb flag is set to 1, the PPM_fin emissions are automatically replaced by OCAR_fin and BCAR_fin emissions. CHIMERE requires a file containing hourly anthropic emissions for the time period to simulate. This file must be placed in the
temporary directory and its name is AEMISSIONS.nc. This file is encoded in the netCDF format, and its structure is defined in Appendix E.4, p.151.

6.1.2 Biogenic emissions

6.1.2.1 The biogenic emitted species

In the complete version of the model, the model needs:

- NO — Nitrogen monoxide
- APIE — α-pinene
- LIMONE — Limonene
- OCIMEN — Ocinene (Lumped class)
- DUST_big — Mineral dust, resuspended matter, soil matter
- DUST_fin — Mineral dust, resuspended matter, soil matter
- NA_coa — Sodium fraction of sea salts
- H2SO4_coa — Sulfate fraction of sea salts
- SALT_coa — Total dry sea salts
- HCL_coa — Chloride fraction of sea salts
- WATER_coa — Water fraction of sea salts

Emissions of six Chimere species: isoprene, α-pinene, β-pinene, limonene, ocimene, and NO, are calculated using the MEGAN model data and parameterizations. The new MEGAN model ([Guenther et al., 2006], 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:

\[ ER_i = EF_i \times \gamma_i(T, PPFD, LAI) \times \rho_i \]  

(6.1)

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, \( \gamma_i \) (unitless) is an emission activity factor that accounts for deviations from canopy standard conditions, and \( \rho_i \) 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:

\[ ER_i = EF_i \times \gamma_{T,i} \times \gamma_{PPFD} \times \gamma_{LAI} \]  

(6.2)

The MEGAN model provides input EF and LAI data over a global grid, hereafter 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, δ3-carene, ocimene, and nitrogen oxide. EFs 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.

Terpene and humulene emissions are not calculated in this model version and are set to zero.

Sea salt emissions are calculated by [Monahan, 1986]:

\[ \frac{dF}{dr} = 1.373r^{-3.41} \left( 1 + 0.057r^{-1.05} \right)^{10^{1.19}e^{-n^2}} \]  

(6.3)
\[ B = \frac{0.38 - \log(r)}{0.65} \]  

(6.4)

\( F \) is the flux of sea salt particle number in particles m\(^{-2}\) s\(^{-1}\) µm\(^{-1}\), \( r \) the particle radius in µm and \( U_{10} \) is the wind speed at 10 m in m s\(^{-1}\).

Dust emissions inside the domain are calculated according [Vautard et al., 2005a].

### 6.1.2.2 Biogenic emission interface

The **BEMISSIONS.nc** required by the model file should contain 2D hourly biogenic emissions with Isoprene, Terpenes and Nitrogen Monoxide. It is created by the **src/diag/diagbio.f90** program. The format of the **BEMISSIONS.nc** netCDF file is given in **Appendix E.5, p.153**, with 2D records containing the biogenic emission rates in molec/cm\(^2\)/s.

- 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. Original emission factor data are netcdf files called `<spec>all200021.nc` corresponding to the period 2000–2001 with `<spec>` referring to a given species. LAI data are 12 netcdf files of the format `laivyyyyyymm.nc` each file corresponding to one month of the year. 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.

- Module **diagbio_megan** in **src/diag/diagbio_megan.f90** contains subroutine **megan_bioemis** called from **src/diag/diagbio.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 [Guenther et al., 2006]. The module also contains the data used for the calculations.

- Biogenic emissions for dust and sea salts are calculated in the **calculate_bioemissions** subroutine from **src/diag/diagbio_science.f90** module, which is called by **src/diag/diagbio.f90**. The file **LAN-DUSE_<domain>** that contains land use fractions (see **Table 4.2**) is required for sea salt calculations by **calculate_bioemissions**.

### 6.1.3 Tracer emissions

In the top calling script, the flag **trc** represents the number of tracer used in the simulations. A tracer is defined as an inert species advected, wet and dry deposited. In **chemprep/inputdata** directory the **DEPO_SPEC** and **WETD_SPEC** files can be changed if the user want to unactivate the deposition processes for the tracers (by removing the tracer line). A tracer is attached to a source location. When the **trc** flag is activated (>0), in the corresponding **chemprep/inputdata** directory two default files are generated:

- **TRACER**
- **TRACER_PROFILE**

A tracer family is automatically created as the sum of each tracer in the **chemprep/inputdata.<chemistry>.<nbin>/FAMILIES** file.

In the **TRACER** file, the main characteristics of the sources are given by default (one tracer per line), they have to be changed by the user:

- **Name** .............................................................. Tracer name: TRC1, TRC2, ... , TRC[trc]
- **Lat.** .............................................................. Longitude of the source (decimal)
- **Lon.** .............................................................. Latitude of the source (decimal)
- **Beg. date** ............................................................. Starting date of the emission (YYYYYMMDDHH)
• End date .................................................. End date of the emission (YYYYMMDDHH)
• Height .......................................................... Emission height (m)
• Sigma .......................................................... Sigma for a gaussian profile (m)
• Emitted mass .................................................... Total mass emitted between "Beg. date" and "End date" (ton)
• Molar mass ....................................................... Molar mass of the emitted species (g/mol)
• Level number ............................ If > 0, put all the mass at this level, if = 0 put the mass at the emission height

Figure 6.2: Injection height of the tracer

If Level number = 0, the emission vertical profile has a gaussian shape with Sigma and Height the two parameters (Figure 6.2).

In the TRACER_PROFILE file, each line represents the default hourly emission profile of the tracers (one tracer per line, one hour per column), they have to be changed by the user. The hourly emission profile is rescaled in the src/prep/preparemis.f90 program in order to be mass conservative.

6.1.4 Natural emissions

Text published in [Vautard et al., 2005b]

Modelling dust entrainment into the atmosphere has been the subject of many studies since the work of [Bagnold, 1941]. Several detailed models have been proposed, which take into account the soil composition, moisture and several subtle effects such as the Owen effect on friction velocity ([Gillette, 1977]; [Marticorena and Bergametti, 1995]; [Nickovic et al., 2001]; [Zender et al., 2003]). However most of the modelling work was motivated by the need, for climate studies, to represent aerosol concentrations and their feedback onto radiation at the global scale. Only a few studies address the problem of dust erosion in the perspective of air quality modelling, where soil other than arid or semiarid must be considered. Such is the case of the Columbia Plateau case study of [Clairborn et al., 1998]. A common characteristic of all these studies has been to focus on major dust entrainment events, like those associated to wind storms in desert areas or agricultural areas with highly erodible soils, where PM10 concentrations reach several hundreds or thousands of $\mu$g.m$^{-3}$. We are not aware of any model application for "background dust" typical concentrations like those observed in Europe (typically a few $\mu$g.m$^{-3}$). Here, we examine the impact of the introduction of local erosion on PM10 model skill. Since our goal is not to develop a sophisticated erosion scheme for temperate regions, we follow the classical approach of dust emission modelling in a simplified manner. The dust vertical flux (in g.m$^2$.s$^{-1}$) $F$ is modelled with a functional
approach similar to that of [Zender et al., 2003], which itself is based on the [Marticorena and Bergametti, 1995] approach and the [White, 1986] formula:

$$ F = \alpha C u_{ss} \left( u_{ss}^2 - u_{st}^2 \right) $$

(6.5)

where $u_{ss}$ denotes the saltation friction velocity, $u_{st}$ the threshold friction velocity and $C$ is a coefficient that may depend on several surface factors. The sandblasting efficiency, $\alpha$, describes the ratio of the vertical flux to the horizontal saltation flux. The saltation friction velocity $u_{ss}$ corresponds to that encountered on erodible parcels of the model grid cell, usually smoother than typical vegetated surfaces found in Europe. It is calculated by using the 10m wind field, a saltation roughness length of $5.10^{-4}$ m and the assumption of neutral stability as in most previous studies. In order to keep the formulation simple, the threshold friction velocity is assumed to depend only on gravimetric soil moisture $w$. As in [Zender et al., 2003] and [Nickovic et al., 2001], this dependence uses the [Fecan et al., 1999] formulation:

$$ u_{st} = f_w u_{s0} $$

(6.6)

where $u_{s0}$ is the minimal friction velocity for dust entrainment over dry soil, and is taken to be uniformly constant. The soil moisture factor $f_w$ is given by

$$
\begin{align*}
  w < w_t & \quad f_w = 1 \\
  w > w_t & \quad f_w = \sqrt{1 + 1.21(100(w - w_t))^{0.68}}
\end{align*}
$$

(6.7)

where $w_t$ is the gravimetric soil moisture content (in kg.kg$^{-1}$) above which entrainment is inhibited by soil moisture. This threshold is usually calculated as a function of the fraction of clay, in a rather empirical manner. In Europe during the summer season, the volumetric soil moisture provided by the NCEP GFS analyses and MM5 drops below 0.15 (gravimetric equivalent of about 0.10) only after several weeks without precipitation. We take here the uniform constant value $w_t=0.1$ kg.kg$^{-1}$ throughout the model domain, which corresponds to a large clay fraction ([Fecan et al., 1999]).

The independence of threshold friction velocity on soil type is a strong assumption, since our aim is only to test the impact of introducing erosion processes. [Iversen and White, 1982] and [Shao and Lu, 2000] have proposed formulations linking the threshold velocity to the soil particle distribution. In both cases a minimal value of 0.2 m.s$^{-1}$ is found, related to soil particles with a mean diameter of 0.1 mm. According to [Xuan, 2004], threshold friction velocities adjusted to wind tunnel observations are too large as they do not account for subgrid scale turbulence. Following this argument we divided the previous minimal threshold by 2 and take $u_{s0}=0.1$ m.s$^{-1}$.

The sandblasting efficiency is taken as $\alpha = 510^{-5}$ m$^{-1}$ an order of magnitude obtained from the measurements, over Northern Spain and Niger, of [Gomes et al., 2003], and it is again kept uniformly constant, although its dependence can be made explicit as a function of clay content ([Marticorena and Bergametti, 1995]).

In the ideal case of a bare, non crusted soil, the factor $C$ has been taken equal to ([Zender et al., 2003]):

$$ C_i = \frac{2.61 \rho_{air}}{g} $$

(6.8)

In practice, the soil is covered with vegetation, snow or water over most of Western Europe and is very often crusted. We, therefore, define:

$$ C = f'_c f_b C_i $$

(6.9)

where $f_c$ is the crustation factor, which has been found from field studies to lie in the 0.001-0.1 range ([Goossens, 2003]; [Gomes et al., 2003]), and $f_b$ is the fraction of bare ground where dust is available for entrainment. In practice, the product $f_c f_b$ is taken here uniformly constant over land and is tuned to optimize the model/observation correlations while keeping the average mineral dust fraction close to that found in the climatological observations of [Putaud et al., 2004]. This leads to $f_c f_b = 410^{-3}$: This strong assumption of a uniform factor leads to a large uncertainty. However, our results remain valid if other factors of the same order of magnitude
are taken. Due to this uncertainty, our results should be considered in a qualitative manner only, which is enough for a sensitivity study. The [Fecan et al., 1999] parameterization of the effect of soil moisture was unable to inhibit dust erosion in sporadic events of fall wind and rain storms over Western Europe under wet conditions. Dust emissions are therefore set to zero when gravimetric soil moisture exceeds 0.2, and kept as the scheme calculates them only when it is below 0.16, a linear interpolation being performed between these two values. The above emissions are assumed to be total emissions. Since emissions of mineral dust occurs primarily in the coarse mode, one assumes half of emissions in the 10-40 \( \mu m \) mode, 45% of emissions in the 2.510 \( \mu m \) mode and the remaining 5% are distributed in the fine mode using the same distribution as for anthropogenic.
6.2 Dry deposition and resuspension

6.2.1 For gases:

Dry deposition is considered for model gas species \(i\) and is parameterized as a downward flux \(F_{\text{d},i} = -v_{\text{d},i}c_i\) out of the lowest model layer with \(c_i\) being the concentration of species \(i\). The deposition velocity is, as commonly, described through a resistance analogy ([Wesely, 1989]):

\[
v_{\text{d},i} = \frac{1}{R_a + R_{b,i} + R_{c,i}}
\]

Stable case

\[
R_a = \frac{1}{ku_*} \left[ \ln \left( \frac{z}{z_0} \right) + 4.7(\zeta - \zeta_0) \right]
\]

Neutral case

\[
R_a = \frac{1}{ku_*} \left[ \ln \left( \frac{z}{z_0} \right) \right]
\]

Unstable case

\[
R_a = \frac{1}{ku_*} \left[ \ln \left( \frac{z}{z_0} \right) + \ln \left( \frac{(\eta_0^2 + 1)(\eta_0 + 1)^2}{(\eta_r^2 + 1)(\eta_r + 1)^2} \right) + 2 \left( \tan^{-1}\eta_r - \tan^{-1}\eta_0 \right) \right]
\]

where \(\eta_0 = (1 - 15\zeta_0)^{1/4}\), \(\eta_r = (1 - 15\zeta_r)^{1/4}\) and \(\zeta_0 = z_0/L\), \(L\) is the Obukov length, \(k\) the von Karman constant and \(z_0\) is the dynamical roughness length. \(z_0\) depends on the fraction of land-use for each category and on the season. See Section on land-use and on meteorology for more details.

According to [Hicks et al., 1987] the quasi laminar resistance \(R_{b,i}\) is calculated as:

\[
R_{b,i} = \frac{2}{ku_*} \left( \frac{S_{C_i}}{P_r} \right)^{2/3}
\]

where \(S_{C_i}\) and \(P_r\) are the Schmidt and Prandtl number, respectively, \(k\) the von Karman’s constant and \(u_*\) the friction velocity.

The formulation of the surface resistance \(R_c\) follows [Erisman et al., 1994]. 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]). All species-related parameters are gathered in the DEPO_SPEC file located in all chemprep/inputdata directories related to the simulation.

In the DEPO_SPEC parameter file:

<table>
<thead>
<tr>
<th>charspec</th>
<th>dMx</th>
<th>dHx</th>
<th>dLx</th>
<th>dRl</th>
</tr>
</thead>
<tbody>
<tr>
<td>O3</td>
<td>48</td>
<td>0.01</td>
<td>1</td>
<td>2000</td>
</tr>
<tr>
<td>SO2</td>
<td>64</td>
<td>1e5</td>
<td>0</td>
<td>1e-20</td>
</tr>
<tr>
<td>NO2</td>
<td>46</td>
<td>0.01</td>
<td>0.1</td>
<td>2000</td>
</tr>
<tr>
<td>NO</td>
<td>30</td>
<td>2e-3</td>
<td>0</td>
<td>2000</td>
</tr>
<tr>
<td>NH3</td>
<td>17</td>
<td>1e+5</td>
<td>0</td>
<td>1e-20</td>
</tr>
</tbody>
</table>

These columns are read and used in the \texttt{initio/inidepo.f90} subroutine. The parameters are:

1. charspec: species name
2. dMx: molar mass of the model species
3. dHx: effective Henry’s law constant (M atm\(^{-1}\)) for the gas
4. df0: a normalized (0 to 1) reactivity factor for the dissolved gas
5. rRwat: parameter is not used

In initio/inidepo.f90, dHx and df0 are used for the mesophyllic resistance value (definition in
[Seinfeld and Pandis, 1997], p.973) as:

- **Mesophylle resistance Rm factor**

  ```
  do nd=1,ndepo
    Rm(nd)=1d-2/(((dHx(nd)/3d3)+(1d2*df0(nd)))
  enddo
  ```

- **Quasi-laminar boundary layer resistance Rb factor**

  ```
  do nd=1,ndepo
    DH2O_Dx=sqrt(dMx(nd)/18d0)
    factRb(nd) = ((rNu/(DH2O*prandtl))
                   &* DH2O_Dx)**(2d0/3d0) &
                   &* 2d0/vkarm
  enddo
  ```

Here DH2O and DH2O_Dx are the molecular diffusivity of water and gaseous species, respectively.
The main land/seasonal parameters can be found in the file DEPO_LAND file which can be found in all
chemprep/inputdata directories. Seasonal variations of resistances are taken from 3 Wesely Seasons:

- #1 (summer with full vegetation), from may to october. In August, September and October, however, LAI,
stomatal resistance and deposition height are divided by two for the CROPLAND category in order to take
into account harvested fields.
- #3 (winter without frost) from november through march.
- #5 (transitional season with short annuals) for April.

Most land parameters are taken from [Wesely, 1989], but LAI are drawn from the NASA/EOSDIS Oak Ridge
National Laboratory using average LAI field measurements, for Summer.

Dry deposition (for gases) parameters are initialized in routine src/initio/inidepo.f90. At the coarse time step
frequency, deposition velocities are calculated in routine src/model/depvel.f90. Deposition fluxes are calculated
at the fine time step frequency in routine src/model/deposition.f90.

### 6.2.2 For aerosols:

Dry deposition for aerosols also makes use of a resistance scheme. The dry deposition velocity is expressed as:

\[
v_d = v_s + \frac{1}{r_a + r_b + r_ar_bv_s}
\]

where \(v_s\) is the sedimentation velocity expressed as in [Seinfeld and Pandis, 1997], \(R_a\) the aerodynamic resistance
and \(R_b\) the quasi-laminar resistance for particles.

The aerodynamic resistance \(R_a\) is calculated as the integral of the inverse of the diffusivity coefficient \(K(z)\) up to
the middle of the model surface layer, which can be estimated using the analytical formulae of the surface-layer
similarity profiles for \(K\) ([Seinfeld and Pandis, 1997]).

Over vegetal canopies, corrections have been implemented according [Zhang et al., 2001], [Giorgi, 1986] and
[Peters and Eiden, 1992]. A complete discussion on mecanistic processes of dry deposition is available in
[Petroff, 2005].
6.3 Wet scavenging

When the aerosol option is selected, scavenging for gas/aerosols in clouds or rain droplets is taken into account. Wet scavenging is formulated as follows:

- **For gases in clouds**: Nitric acid, ammonia in the gas phase are scavenged by cloud droplets. This process is assumed to be reversible. Moreover, for in-cloud scavenging, dissolved gases in a non precipitating cloud can reappear in the gas phase due to cloud dissipation. Equilibrium between dissolved gases concentration and gas-phase concentrations follow Seinfeld and Pandis (1998).

  \[ \Gamma = \frac{pD_g}{6.10^6 u_g D^2} (2 + 0.6 R e^{1/2} S c^{1/3}) \]  
  \( p \) being the precipitation rate (mm h\(^{-1}\)), \( D_g \) the molecular diffusion coefficient (\( m^2 s^{-1} \)), \( u_g \) the raindrop velocity (m s\(^{-1}\)), \( R e \) and \( S c \) respectively the Reynolds and Schmidt numbers of drops. [Mircea and Stefan, 1998] and references therein give relationships between \( u_g \) and hydrometeor diameter for various types of precipitation. In the model, sulfur dioxide and hydrogen peroxide are also scavenged by precipitation.

- **For gases in rain droplets below the clouds**: Dissolution of gases in precipitating drops is assumed to be irreversible, both for HNO\(_3\) and NH\(_3\). The scavenging coefficient is expressed as:

- **For particles in clouds**: Particles can be scavenged either by coagulation with cloud droplets or by precipitating drops. Particles also act as cloud condensation nuclei to form new droplets. This latter process of nucleation is the most efficient one in clouds. According to [Tsyro, 2002] and [Guelle et al., 1998], the deposition flux is written as:

  \[ \left[ \frac{dQ_r^k}{dt} \right] = -\frac{\epsilon_l P_r w_l h Q_r^k}{w_l h} \]  
  with \( P_r \) being the precipitation rate released in the grid cell (g cm\(^{-2}\) s\(^{-1}\)), \( h \) the cell thickness (cm) and \( \epsilon \) an empirical uptake coefficient (in the range 0 - 1) depending on particle composition. \( l \) and \( k \) are respectively the bin and composition subscripts.

- **For particles in rain droplets below the clouds**: Particles are scavenged by raining drops, the deposition flux of particles being:

  \[ \left[ \frac{dQ_l^k}{dt} \right] = -\frac{\alpha p E_l u_g Q_l^k}{u_g} \]  
  with \( \alpha \) being an empirical coefficient, \( p \) the precipitation rate in the grid cell (g cm\(^{-2}\) s\(^{-1}\)), \( E \) a collision efficiency coefficient between particles and raining drops ([Loosmore and Cederwall, 2004]) and \( u_g \) the falling drop velocity (cm s\(^{-1}\)).

6.3.1 Resuspension of particulate matter

Text published in [Vautard et al., 2005b]

Saltation, which is necessary to transfer small mineral particles from the soil into the atmosphere, is not the only natural aerosol upward entrainment process. Resuspension of freshly deposited small particles lying at the surface by turbulent wind is a distinct process, their extraction resulting from the imbalance between adhesive and lifting forces (see e.g. [Reeks and Hall, 2001]). Such particles can originate from the atmosphere or the biosphere, and are particularly easy to extract shortly after deposition ([Loosmore, 2003]). They can be made of vegetal debris obtained from the mechanical or biological degradation of canopy elements, spores, pollens, etc. ([Simoneit and Mazurek, 1982]). The evidence of the presence of biogenic non mineral compounds in the continental aerosols has been reported several times (see e.g. [Andreae and Crutzen, 1997], for instance in the form of "humic matter", consisting in carbonaceous polymers (see [Gelencser et al., 2002] and references therein), and
cellulose ([Kunit and Puxbaum, 1996]). As shown by [Gelencser et al., 2002], some of these aerosols could be of secondary nature. However, for the sake of simplicity, we assume that this biogenic aerosol component is of primary origin. In order to represent these processes, we use a bulk formulation based on the simple resuspension rate empirical formula of [Loosmore, 2003], which was shown to provide a very good fit to the available resuspension measurement data:

$$\lambda = 0.01 \frac{u_z^{1.43}}{\tau^{1.03}}$$

(6.19)

where $\lambda$ is the resuspension rate (in s$^{-1}$) and $\tau$ is the time after the start of resuspension. In the experiments used by the previous authors, the particles are first deposited then resuspended. In reality, deposition and resuspension are simultaneous, and the available dust concentration on the ground is governed by resuspension, washout by runoff and absorption by soil water, production by deposition and other biological or mechanical processes. The detail of all these processes is essentially unknown, and we assume here that the available concentration of dust does only depend on the wetness of the surface. In this empirical view, the resuspension flux is governed by:

$$F = Pf(w)u_z^{1.43}$$

(6.20)

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. The soil water modulation factor is vanishing for wet soil (gravimetric soil water content of 0.2 or more, as before, and is equal to 1 for dry soil, $w$ less than 0.1). A linear relation is assumed between these values. The resuspension flux is found to be $F=1800$ $\mu$g m$^{-2}$ h$^{-1}$ for a dry soil and for a friction velocity of 1 m.s$^{-1}$. In Northern Europe the typical MM5 value for volumetric soil water content is 25%, and 0.5 m.s$^{-1}$ for the friction velocity. This leads to a mass flux of about 170 $\mu$g m$^{-2}$ h$^{-1}$ and then about 1.4 ton.km$^{-2}$ and per year, which is much larger than the anthropogenic emissions in most remote rural areas, but an order of magnitude smaller than that in urban areas. Finally, in the absence of any information the reentrained PM10 particle mass is supposed to be distributed in a standard atmospheric size distribution: $2/3$ of the mass as PM2.5 and $1/3$ as coarse PM1-PM2.5. Within PM2.5, particles are distributed as for the anthropogenic emissions.
Chapter 7

Initial and boundary conditions

7.1 Chemical boundary conditions

7.1.1 Internal boundary conditions

Boundary conditions can be either "external" (see next sections) or given by a coarse-resolution CHIMERE simulation. When a coarse-resolution simulation is available, i.e., when the `out.[coarse-label].nc` file exists, the `prep_chimere.f90` interface in the `src/prep` directory is used to generate the lateral and top boundary conditions to a nested run.

7.1.2 Gas boundary conditions from global models

In the first case and in the present release, a choice of monthly climatologies is offered between the MOZART second-generation global chemistry-transport model (MOZART2) [Horowitz et al., 2003], or the LMDz-INCA2 chemistry-transport model, [Hauglustaine et al., 2004]. The concentration of a few species are considered as constant (but depending on month) throughout the simulation. The MOZART climatologies have been calculated and are kindly provided here by the Max-Planck Institut, Hamburg, by M. Schultz, G. Brasseur, C. Granier and D. Niehl. In the second case, a first coarse run must have been carried out.

The user can also provide a different, independent set of boundary conditions to the model, which can be time-varying or constant during the simulation. A new interface to build data for CHIMERE has to be constructed. The current interface in `src/prep` directory is `prep_bound.f90` (and module `prep_helpers.f90` used by it). See technical documentation for the format of boundary conditions files.

For the moment, global scale model concentrations are given in ASCII format, therefore, several other files are needed, for describing the species considered at the boundaries. For MOZART2 lateral or top boundaries, only the following species are present in the available files:

- O3 .............................. Ozone
- NO2 .......................... Nitrogen Dioxide
- CO ............................ Carbon Monoxide
- PAN ............................ Peroxy-Acetyl-Nitrate
- CH4 ............................ Methane
- C2H6 ............................ Ethane
- HCHO ............................ Formaldehyde
- HNO3 ............................. Nitric Acid

For LMDz-INCA2 lateral or top boundaries, only the following species are available:

- O3 .............................. Ozone
- NO .............................. Nitrogen Monoxide
- NO2 .............................. Nitrogen Dioxide
- HNO3 ............................. Nitric Acid
- PAN ............................ Peroxy-Acetyl-Nitrate
- H2O2 ........................... Hydrogen peroxide
- CO .............................. Carbon Monoxide
- CH4 ............................. Methane
- HCHO ............................ Formaldehyde
- C2H6 ............................ Ethane
The file containing the MOZART2 and LMDz-INCA2 "boundary" species names are respectively `MOZART2_SPEC` and `LMDZINCA2_SPEC`. If other interfaces had to be built, another file like this one would have to be present in the `chemprep/boundaries_spec` directory.

<table>
<thead>
<tr>
<th>Species</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>C4H10</td>
<td>Butane (lumped class)</td>
</tr>
<tr>
<td>C2H4</td>
<td>Ethene</td>
</tr>
<tr>
<td>C3H6</td>
<td>Propene (lumped class)</td>
</tr>
<tr>
<td>OXYL</td>
<td>Xylene (lumped class)</td>
</tr>
<tr>
<td>C2H6</td>
<td>Ethane</td>
</tr>
<tr>
<td>NC4H10</td>
<td>Butane and higher alkanes</td>
</tr>
<tr>
<td>O3</td>
<td>Ozone</td>
</tr>
<tr>
<td>NO2</td>
<td>Nitrogen Dioxide</td>
</tr>
<tr>
<td>PAN</td>
<td>Peroxy-Acetyl-Nitrate</td>
</tr>
<tr>
<td>HNO3</td>
<td>Nitric Acid</td>
</tr>
<tr>
<td>SO2</td>
<td>Sulphur Dioxide</td>
</tr>
<tr>
<td>CO</td>
<td>Carbon Monoxide</td>
</tr>
<tr>
<td>CH4</td>
<td>Methane</td>
</tr>
<tr>
<td>HCHO</td>
<td>Formaldehyde</td>
</tr>
<tr>
<td>CH3CHO</td>
<td>Acetaldehyde</td>
</tr>
<tr>
<td>GLYOX</td>
<td>Glyoxal</td>
</tr>
<tr>
<td>MGLYOX</td>
<td>Methyl-Glyoxal</td>
</tr>
<tr>
<td>CH3COE</td>
<td>Methyl-Ethyl Ketone</td>
</tr>
<tr>
<td>ASPOMM</td>
<td>Accumulation mode soluble particulate organic matter</td>
</tr>
<tr>
<td>AIPOMM</td>
<td>Accumulation mode insoluble particulate organic matter</td>
</tr>
<tr>
<td>CIDUSTM</td>
<td>Coarse mode insoluble dust</td>
</tr>
<tr>
<td>ASSO4M</td>
<td>Accumulation mode soluble sulphate</td>
</tr>
<tr>
<td>CSSSSM</td>
<td>Coarse mode soluble sea salt</td>
</tr>
<tr>
<td>AIBCMM</td>
<td>Accumulation mode insoluble black carbon</td>
</tr>
<tr>
<td>ASBCCM</td>
<td>Accumulation mode soluble black carbon</td>
</tr>
</tbody>
</table>

The boundary species names and species are read in routine `src/initio/iniboun.f90`. Then in practice the boundary concentrations values are written to the edges outside the `conc` array containing all concentrations in routines `src/main/master_locvalues.f90` and `src/model/locvalues.f90`.

### 7.1.3 Aerosols boundary conditions from global models

For the moment, aerosol boundary concentrations on several bins provided with Chimere distribution are from two sources: the Global model GOCART (NASA,[Ginoux et al., 2001]) and LMDz-INCA ([Schulz et al., 2006, Textor et al., 2006, Dentener et al., 2006], we use the dataset label 'LMDZAERO'). GOCART provides the following species:

- DUST: Desert dusts
- HBBC: Hydrophobic black carbon
- HBOC: Hydrophobic organic carbon
- HLBC: Hydrophilic black carbon
- HLOC: Hydrophilic organic carbon
- SULF: Sulfates
- SALT: Sea salts

- ASPOMM: Accumulation mode soluble particulate organic matter
- AIPOMM: Accumulation mode insoluble particulate organic matter
- CIDUSTM: Coarse mode insoluble dust
- ASSO4M: Accumulation mode soluble sulphate
- CSSSSM: Coarse mode soluble sea salt
- AIBCMM: Accumulation mode insoluble black carbon
- ASBCCM: Accumulation mode soluble black carbon

Nitrate and Ammonium are not accounted for in this version. These data are monthly climatologies of two years (2000 and 2001).

LMDz-INCA's species are:

- ASPOMM: Accumulation mode soluble particulate organic matter
- AIPOMM: Accumulation mode insoluble particulate organic matter
- CIDUSTM: Coarse mode insoluble dust
- ASSO4M: Accumulation mode soluble sulphate
- CSSSSM: Coarse mode soluble sea salt
- AIBCMM: Accumulation mode insoluble black carbon
- ASBCCM: Accumulation mode soluble black carbon

The data are monthly climatologies of five years (1998-2002).

The file `XXX_builder` (XXX is the name of a global concentrations dataset, GOCART or LMDZAERO at the moment) is used to build the interface between XXX and CHIMERE species. The `prep_bound.f90` interface in `src/prep` directory is used for both lateral and top boundary concentrations, for both aerosol datasets. This interface allows to interpolate the data from GOCART or LMDZAERO bin sizes to the CHIMERE ones.
7.1.4 Boundary conditions interface

Boundary conditions must be provided to CHIMERE in the netCDF file BOUN_CONCS.nc constructed dynamically from the top calling script. Its structure is given in Appendix E.6, p.154. The file contains two main variables:

- **lat_conc**: variable containing hourly lateral mixing ratios in ppbv.
- **top_conc**: variable containing hourly top mixing ratios in ppbv.

The list of top boundary species is read by CHIMERE from the file itself. There are basically three options for building this file:

- Either boundary conditions are provided by the user from external data, in which case she/he should only follow the guidelines on the format of files.
- This is a "nest run" and a coarse-resolution run is used for boundary conditions (the CHIMERE option).
- This is a "coarse run" and climatological or monthly gas boundary conditions from MOZART2 or LMDz-INCA2 are used.

In the case of a coarse run, the top calling script launches prep_bound.f90 that prepares both gas and aerosol boundary and initial conditions from global datasets. The user can choose which species (s)he wants to be included in a simulation, and which dataset will be used as a source for each species. This information is provided by the user in the file chemprep/ boundaries_spec/SOURCE_SELECTOR (see comments in this file about the format). The script chemprep/boun_filter.sh converts this file into the file source_selector in the chemprep/inputdata directory, which specifies the source dataset for each used species and is read by prep_bound.f90.

In the case of a nested run, the top calling script launches prep_chimere.f90 that prepares both gas and aerosol boundary and initial conditions from Chimere output of a previous course run (file out.[label].nc file). The address of the coarse-run simulation file is given in the top calling script, as well as the option for nesting and other necessary parameters of the coarse run.

7.2 Initial concentrations

7.2.1 Initial concentrations species

There are two possible initialisation modes:

- Initialisation by reading initial concentrations in a restart file. This file must contain the "end" concentrations of a previous simulation, as provided in the end.[label].nc (and exe.[label].nc if the pops flag is activated). The species list is contained in the end.[label].nc file itself and is read by CHIMERE during initialisation.
- Initialisation by interpolating global concentration fields. This is the most common option used for a first initialisation.
- Initialisation by interpolating boundary conditions. Because of technical limitations, it is used only when the pops flag is activated (pops=1)

The top calling script checks the existence and nonemptiness of a file called ini.[label].nc (and iex.[label].nc if the pops flag is activated) in the simulation directory, which can be a copy of an "end" file. If it exists and is nonempty, then the initial conditions are used. In the other case, initial conditions are interpolated from boundary conditions. Initial conditions are read and put into arrays conc and conco in routine src/initio/iniconc.f90.

7.2.2 Initial concentrations interface

There are two types of initial conditions that can be used for CHIMERE: Initialized concentrations and noninitialized concentrations. In the first case a file named ini.[label].nc and placed in the simulation directory (the
same as where the CHIMERE output files `out.[label].nc`, `par.[label]` and `end.[label]` are. In the second case there is no need to provide a file; initial conditions will be taken as interpolated from the global field, layer by layer. The initial concentration file `INI_CONCS.nc` generated by `prep_bound.e` is copied into the default initialisation file `inicoarse.nc` in the output directory. The latter file is then used to initialise the next simulations, when no restart files are available.

It is clear that initialisation is important for large-scale simulations (domains larger than about 500 Km × 500 km). Under extremely stagnant conditions, spin-up of one or two days may also be necessary at local (100 Km) scale. At regional scale, the spin-up time is in the order of 4-5 days (for ozone and precursors).

The format of the `ini.[label].nc` file is exactly the same as that of the `end.[label].nc` file. It contains the concentrations encoded in the netCDF format, described in Appendix E.7, p.154.

Concatenating short simulations in order to perform a long-term simulation is really easy: At the end of each simulation the `end.[previous_label].nc` can be copied or linked to the `ini.[new_label].nc` file, this is done in the main calling scripts.
Chapter 8

Meteorology

8.1 Meteorological input data

CHIMERE does not contain any meteorological driver. It is only a chemical-transport model, meaning that meteorological data have to be provided in order to perform a simulation. The provided code only contains a meteorological interface which allows to start from standard meteorological variables and transforms it to CHIMERE inputs. In practice the meteorological interface consists in the construction of a single file `METEO.nc` to be made available as input to the executable `chimere.e` in the temporary directory `tempdir`. It contains all hourly meteorological variables in a single netCDF file.

The construction of this files is made in two stages:

- The first stage is an interpolation stage: it takes meteorological variables from a meteorological database, simulation or forecast file and transforms it into standard variables given on the horizontal CHIMERE grid as hourly values. **This stage has to be constructed by the user.** Specific files with specific formats and filenames must be built. We provide two examples of meteorological interfaces in the current version: for WRF and MM5 models.
- The second stage is a diagnostic meteorological interface (the `src/diag/diagmet.f90`). It is a generic program allowing to transform standard meteo variables into CHIMERE inputs. It also allows construction of possibly non-available variables from standard variables and some parameterizations.

8.1.1 First stage: the "interf" programs

The top calling script calls a program named `interf-[metname]` in the `meteo` directory. This program (usually a bourne shell script) has to be written by the user. One example is given: `interf-mm5` which is a generic interface to mm5. If other meteorological data is to be used, the user should give it a name, `[metname]`, and write the `interf-[metname]` bash program. She/he should also change the "`meteo=" line in the top calling script.

In turn, `interf-[metname]` calls a fortran routine, which was named in the case of MM5 `prepmm5.f90`. This `prep[metname].f90` routine has several roles to play:

- extract hourly meteorological fields and time information from the Meteorological Model files, starting on the first CHIMERE hour simulation and ending on the last simulation hour.
- Horizontally interpolate variables to CHIMERE grid. However the original vertical levels are kept.
- Write these variables and metadata to the `exdomout.nc` netCDF file with a given **mandatory** structure.

`interf-[metname]` generates also two parameter files:

- `METINFOS.nml` contains parameters for the 2nd meteo stage `diagmet.f90`
- `prep[metname].nml` contains parameters for `prep[metname].f90`

The Table 8.1 lists the mandatory and optional variables which must or can be output by the interpolation stage.
### Mandatory Variables

<table>
<thead>
<tr>
<th>File Name</th>
<th>Variable</th>
<th>Dimension</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>lon</td>
<td>longitude of gridpoints</td>
<td>2D</td>
<td>degrees_east</td>
</tr>
<tr>
<td>lat</td>
<td>latitude of gridpoints</td>
<td>2D</td>
<td>degrees_north</td>
</tr>
<tr>
<td>winz</td>
<td>Zonal component of the wind</td>
<td>3D</td>
<td>m/s</td>
</tr>
<tr>
<td>winm</td>
<td>Meridional component of the wind</td>
<td>3D</td>
<td>m/s</td>
</tr>
<tr>
<td>temp</td>
<td>Temperature</td>
<td>3D</td>
<td>K</td>
</tr>
<tr>
<td>sphu</td>
<td>Specific humidity</td>
<td>3D</td>
<td>kg/kg</td>
</tr>
<tr>
<td>alti</td>
<td>Altitude of half layers on which all 3D meteo variables should be given</td>
<td>3D</td>
<td>m</td>
</tr>
<tr>
<td>pres</td>
<td>Pressure at half layers on which all 3D meteo variables should be given</td>
<td>3D</td>
<td>Pa</td>
</tr>
<tr>
<td>cliq</td>
<td>3D cloud liquid water content for radiation (excluding rain water)</td>
<td>3D</td>
<td>Kg/Kg</td>
</tr>
<tr>
<td>tem2</td>
<td>2m Temperature</td>
<td>2D</td>
<td>K</td>
</tr>
</tbody>
</table>

### Optional Variables

<table>
<thead>
<tr>
<th>File Name</th>
<th>Variable</th>
<th>Dimension</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>rain</td>
<td>3D rain water (to be added to cloud water for radiation)</td>
<td>3D</td>
<td>Kg/Kg</td>
</tr>
<tr>
<td>cice</td>
<td>3D ice (for radiation)</td>
<td>3D</td>
<td>Kg/Kg</td>
</tr>
<tr>
<td>sshf</td>
<td>Surface sensible heat flux, optionally read or recalculated</td>
<td>2D</td>
<td>W/m2</td>
</tr>
<tr>
<td>slhf</td>
<td>Surface latent heat flux, optionally read</td>
<td>2D</td>
<td>W/m2</td>
</tr>
<tr>
<td>usta</td>
<td>Friction velocity, optionally read or recalculated from Louis [1982]</td>
<td>2D</td>
<td>m/s</td>
</tr>
<tr>
<td>hght</td>
<td>Boundary layer height, optionally read or recalculated</td>
<td>2D</td>
<td>m</td>
</tr>
<tr>
<td>lowc</td>
<td>Low cloud fraction, optional for radiation</td>
<td>2D</td>
<td>m</td>
</tr>
<tr>
<td>medc</td>
<td>Medium cloud fraction, optional for radiation</td>
<td>2D</td>
<td>m</td>
</tr>
<tr>
<td>higc</td>
<td>High cloud fraction, optional for radiation</td>
<td>2D</td>
<td>m</td>
</tr>
</tbody>
</table>

Table 8.1: Mandatory and optional variables in exdomout.nc meteo file. The parameters quoted with ¹ are mandatory only for the aerosol option

#### 8.1.1.1 Recommendations

There are a number of recommendations that can be given about the choices of the optional variables. Some are listed below.

- **If boundary layer height** is not available from meteo data, it will be calculated using the maximum between a formula for stable air (using a critical Richardson number=0.5), and a parameterization for unstable boundary layer based on a simplification of the Cheinet [2002, PhD thesis] work. These calculations assume a dry boundary layer. The user should use meteorological model’s boundary layer height if more elaborate and whenever possible, especially if it accounts for clouds.

- **The friction velocity** $u_*$ is used for deposition and calculation of diffusivities. It is a particularly sensitive parameter for ozone in Summer through the calculation of aerodynamic resistance $R_a$. It should strongly depend on land use type, which are critical to deposition. In meteorological models, especially large-scale, roughness lengths are sometimes estimated too roughly for the application to high-resolution deposition. The recalculation of $u_*$ which is proposed in the diagmet.f90 interface uses the Louis [1982] formulation which is particularly robust and gave satisfactory results. We advise to use it anyway, in particular to have deposition consistent with a high-resolution land use.
- **Heat fluxes** are also used for the calculation of $W^*$ and therefore mixing, and the height of the boundary layer. In fact only the virtual heat flux is required, which can be recalculated from an empirical formula [Priestley, 1948] using temperatures in the first meteorological model layers. This formula is not very accurate. It is strongly advised to use heat fluxes from the meteorological model.

- For **radiation** several calculation options are offered. All are based on a statistical model of the clouds optical thickness.

  - The first option uses liquid/ice water (whenever available), a variable which can be very model-dependent and which is rarely verified against observations. Moreover it does not take into account fractionary cloudiness. However it may be suited for high/medium clouds which are often stratiform clouds in the midlatitudes.

  - The second option uses cloud fractions (whenever available), which is less meaningful to radiation but could be used when cloud/ice water is not available.

  - The third option uses relative (to liquid water) humidity, and has been tuned using J(NO2) measurements taken over a long period near Paris (manuscript in preparation). The formulation, modifying radiation for RH>0.75 should only be used for low clouds. It has the advantage of being a more robust parameterization since it depends on a highly verified variable.

In summary, our preference is: RH formulation (option 2 in METINFOS.nml) for low clouds, and for liquid/ice water formulation (whenever available) for medium and high clouds, and cloud fractions otherwise.

### 8.1.1.2 The METINFOS.nml file

METINFOS.nml is a parameter file that is passed to diagmet.f90 to provide information about the chosen options. In the MM5 example it is constructed from a template file meteo/METINFOS-mm5.nml.sed where macros are replaced by values when running interf-[metname]. METINFOS.nml should be placed in tmpdir temporary directory. It has the following structure (example drawn from MM5):

```
&metoptions
  opt%idstart = 2003073000 ! Start date of the run
  opt%nsho = 1904 ! grid cell for screen display

  !
  opt%w10m = 0 ! 10m wind option (0=read from file, 1=recalculate)
  opt%usta = 0 ! Ustar option (0=read from file, 1=recalculate from Louis 1982)
  opt%flux = 0 ! Flux option (0=read, 1=recalculate from Priestley 1948)
  opt%pblh = 0 ! Boundary layer height (0= read from file,1=recalculate)
  opt%cice = 0 ! Cloud Ice Wat (0=Use in addition to cloud water, 1=don't)
  opt%rain = 0 ! Rain Wat (0=Use in addition to cloud water, 1=don't)

  !
  opt%upm = 0 ! Urban corrections for BLH
  opt%ufx = 0 ! Urban corrections for flux
  opt%uwc = 1 ! Urban corrections for wind

  !
  opt%clol = 2 ! Low cloud option (0=read cloudiness, 1=Use Liq. Wat., 2=Use RH)
  opt%crhl = 0.85

  !
  opt%clom = 1 ! Medium cloud option for attenuation ...
  opt%crhm = 0.95

  !
  opt%cloh = 1 ! High cloud option for attenuation ...
  opt%crhh = 0.95

  !
  opt%soim = 0 ! Erosion/Resusp.: (0=read soil moist. [m3/m3],1=don't=>no erosion)
  opt%eros = 1 ! Erosion option
  opt%resu = 0 ! Resuspension option
```

8.1.1.3 The prep[metname].nml file

prep[metname].nml is a parameter file that is passed to prep[metname].f90 to provide information about options and file paths. It is constructed by interf-[metname]. prep[metname].nml file should be placed in the tmpdir/metoe directory. It has the following structure (example drawn from MM5):

```plaintext
&args
  ids = 2003073000
    ! start date of the run
  nhours = 12
    ! length of the run in hours
  domain = 'CONT3'
    ! CHIMERE domain
  fn_mdom = 'COORD_MM5EUR1'
    ! MM5 ‘crs’ coordinates file
  fn_mdom_dot = 'COORD_MM5EUR1_DOT'
    ! MM5 ‘dot’ coordinates file
  fn_coo = '/ws3/monge/CHIMERE-AERO-NEW/V200501Hjlm/domains/COORD_CONT3'
    ! CHIMERE domain coordinates file
  nzonal = 67
    ! number of model grid cells in the zonal direction
  nmerid = 46
    ! number of model grid cells in the meridian direction
  fn_mm5 = '/COMPERES/pub/MM5/AVN/2003/EUR1/MMOUT_EUR1_20030730_20030803_S'
    ! MM5 meteo file
  fn_out = '/ws3/monge/CHIMERE-AERO-NEW/V200501Hjlm/28089/exdomout.nc'
    ! prepmm5.e output file. Do not modify.
/
```

8.1.1.4 Examples: the MM5 and WRF interfaces

The interf-mm5 interface starts from a unique MM5 file, MMOUT_[mm5-domain], specified exactly as in the V3.6 MM5 format, see the NCAR web site for a full documentation of MM5 and output format (http://ncar.ucar.edu). The MMOUT file is assumed to be written in big-endian order, i.e., with word byte order reversed relative to standard x86 PC words. In this interface it is assumed that the output frequency for MM5 simulations is 1 hour, so there is no time interpolation for input to CHIMERE.

![Figure 8.1](image-url)

Figure 8.1: How to prepare MM5 coordinates files for CHIMERE, from the original MM5 terrain file

Before using the MM5 interface, three coordinate files for the MM5 grid must be constructed manually Figure 8.1. There is a file for the CRS coordinates (used for all variables but wind) and a file for the DOT coordinates (shifted grid for the wind fields). A utility for making these file coordinates is provided in the CHIMERE util directory. Edit the make_mm5_coord utility script for more help. These files (COORD_MM5[domain], COORD_MM5[domain]_DOT) and HEIGHT_MM5[domain]) must be placed in the meteo directory once they are constructed.

There are no prior assumptions on the choice of physical options in MM5. In our applications, however, the following options have been used and can therefore be recommended:
- Schultz microphysics,
- MRF (Troen-Mahrt) PBL
- Grell cumulus scheme
- RRTM radiation scheme
- In all applications over Europe where AVN/NCEP is used as forcing data, we found that nudging was necessary (grid FDDA option), with nudging coefficients varying from 0.00001 to 0.0001, depending on the spatial scale, but no meteorological observations were used.

The `prepwrf` interface takes a WRF output netcdf file `wrfout...` given by the `meteo_file` variable in `chimere.sh` and then by `fn_wrf` in `prepwrf.nml`. The list of WRF variables to process and their parameters are specified in the structure variable `varmeta` and are initialized in subroutine `inivars`. You might need to adjust this initialization (and the corresponding definitions of the parameters between `IFIRST` and `ILAST` in the beginning of the program), if your WRF output has variables additional to those specified in `varmeta`. The domain coordinate files `COORD_WRF`, `COORD_WRF_U`, and `COORD_WRF_V` are created by `prepwrf` from the WRF output netcdf files, so in contrast to MM5 you do not have to use an additional program for it. The `metdom` variable specified in `chimere.sh` is not used by the WRF interface.

### 8.1.1.5 Building your own "interf" interface

There are two steps to perform this task:

- create a new `interf-[metname]` script
- create a new `prep[metname].f90` routine

The first step is trivial, since it consists in reading arguments from a file `metargs-[metname]` generated by the top calling script, copying/translating them to two namelists `prep[metname].nml` and `METINFOS.nml`, and finally compiling and launching the executable `prep[metname].e`. It is recommended to use `interf-mm5` as a template.

The second step might be a painful job.
The user may start from scratch. In that case, the only prerequisites are:

- The format of input files
- The list of output variables, together with their mandatory name
- The structure of the netCDF output file named `exdomout.nc`. This structure is entirely defined by the header listing in Appendix E.2, p.148.

However, the recommended procedure is to start from `prepm5.f90` as a template. The interpolation stage and the output to the netCDF file are already written and should not be so much modified. The real work for the developer of a new interface would be:

- to understand the data structures of the program. In short, a table contains all metadata of input and output files. This metadata structure is linked to a meteo data array by the index of the current variable. The fields in the metadata structure are used to instruct a rather general program what to do with the current variable. The macro `_LINE_` is used to generate unique and monotonic indexes to variables.
- to create the read interface for their meteorological data, paying attention to synchronisation: read and write the right data at the right time slot.

### 8.1.2 Second stage: the diagmet diagnostic model

The second stage of the meteorological interface consists in the production of the meteo input file for CHIMERE, `METEO.nc`, from the output meteo file of the first stage `exdomout.nc`. This is achieved by the program `src/diagmet.f90`. It is a generic program, designed to be independent on the origin of the meteorological data. `diagmet.f90` requires the following set of files:
• **exdomout.nc** : Contains the hourly 2D and 3D meteorological fields constructed by the first stage of the meteo interface, interpolated to the CHIMERE grid. It is encoded in the netCDF format. For 3D fields, the vertical grid is still that of the meteorological forcing model. There are mandatory and optional fields (see first-stage section above).

• **METINFOS.nml** : Namelist containing the meteorological option flags and parameters.

• **diagmet.nml** : Namelist containing file paths.

• **LANDUSE_domain** : File containing the fraction of land use per grid cell. There are 9 land use categories in this model version, see the Land-use interface below for the description of these classes. The file therefore contains 9 columns with numbers between 0 and 1. This file must be constructed once for each model domain using the Land-use (or an equivalent) interface. This file is generally located in the **domains/** directory and is pointed to by an address in **diagmet.nml**.

• **VCOORD.[vgname]** : This file contains the vertical grid hybrid sigma-p coefficients for the top of all CHIMERE layers. This file is generally located in the **data/vgrid/** directory and is pointed to by an address in **diagmet.nml**.

• **LANDPAR** : This file gives some land-use/season dependent surface parameters: roughness height Zo, extra fluxes, wind reduction or extra urban heat-island temperature increment. It is generally located in **data/** directory, and is pointed to by an address in **diagmet.nml**.

Model coordinates are read from the netCDF file **exdomout.nc**.

The diagmet program calculates the following parameters by using the following parameterizations:

- Radiation attenuation
- Boundary layer height
- Friction velocity U* (From the [Louis, 1979], [Louis et al., 1982] parameterization)
- Aerodynamic resistance (Seinfeld and Pandis, 1997))
- Sensible heat flux (read or recalculated using the [Priestley and Taylor, 1967] formula.
- Obukov length
- Convective velocity W*

The output file **METEO.nc** contains one record per hour and its structure is documented in Appendix E.3, p.150.

The main variables contained in this file are:

- **3D data**:
  - Horizontal wind (transport)
  - Temperature (chemistry, thermodynamics)
  - Density (chemistry and transport)
  - Specific humidity (chemistry/thermodynamics)
  - Height of model layers (model geometry)
  - Cloud liquid water content (for the aerosol option, aqueous-phase chemistry, scavenging)

- **2D data**:
  - Temperature at 2m (deposition, biogenic emissions)
  - Photolysis attenuation in the column due to clouds (chemistry/photolysis)
  - Boundary layer height for mixing
  - Friction velocity u* for mixing and deposition
  - Obukov length for mixing
  - Convective velocity scale w* for mixing
  - Aerodynamic resistance for deposition
  - Total Precipitation rate for aerosol scavenging
8.2 Transport and mixing

8.2.1 Horizontal transport

Three schemes for horizontal transport of chemical concentrations in the model are now available. The choice between the three may be done species by species by changing the second column flag in the ACTIVE_SPECIES file. The schemes are:

- A simple first order upwind scheme named **upwind** in the model: low computational cost but not very accurate and very diffusive.
- The first order Van Leer scheme: low computational cost and good accuracy for transport of high concentrations in plumes.
- The three order scheme **PPM** (for 'Parabolic Piecewise Method'), proposed by [Colella and Woodward, 1984]): high computational cost but the more accurate for long-live species, naturally diffusive in a chemistry-transport model.

These three schemes are all defined and used in the **src/model/transmix.f90** program. We consider the following conservation equation:

\[
\frac{\partial u}{\partial t} + \frac{\partial F_i(u)}{\partial x_i} = 0
\]  

(8.1)

where \( u = (\rho, \rho v_x, \rho v_y, \rho v_z, e) \) the physical conserved quantities (spatially) and \( F(u) \) the corresponding flux. Each values and associated fluxes are represented at the center of each cell, \( u_n = u(x_n); F_n \). Of course, for a more realistic approach, we would represent these fluxes at each cells interface (\( F_{n+1/2} \)): thus, the problem of the advection schemes is to represent these interfaces fluxes by using only the known mean values fluxes. Formally, we can estimate the next time step \( u \) with the following equation:

\[
u_n^{t+\Delta t} = u_n^{t+\Delta t} - \left( \frac{F_{n+1/2}^t - F_{n-1/2}^t}{\Delta x} \right) \Delta t
\]  

(8.2)

This equation is exact because it ensures the mass conservation between inside and outside fluxes for each directions. To estimate the value of \( F_{n+1/2}^t \), we have to use the known values of \( F_n \) et \( F_{n+1} \). The way we use all these adjacent fluxes defines the "centered" or "advanced" scheme type and induces its numerical stability or not.

The following scheme is named "upwind" because it follows the advected flux. By the way, we choose the flux value we use as a function of the place we are and the direction we want to go, as:

\[
F_{n+1/2}^t = \begin{cases} 
F_n^t & \text{if } v > 0 \\
F_{n+1}^t & \text{if } v < 0
\end{cases}
\]  

(8.3)

This scheme is clearly asymmetrical. In addition, numerical experiments showed it is very diffusive. To avoid these effects, more complex schemes were developed: they are using more cells and thus are able to limit numerical errors due to too crude interpolations between cells. In the CHIMERE model, we implemented two advection schemes recognized for their good numerical accuracy: the VanLeer scheme in the **src/model/vanleer.f90** program and the PPM scheme in the **src/model/ppm.f90**.

Horizontal mass fluxes (independent of species, first order) are calculated first at each coarse time step in routine **src/model/htransport.f90**, then the complete calculation using the species concentration and PPM is performed at each fine time step in routine **src/model/transmix.f90**.

8.2.2 Optional correction of surface wind speed to account for urban effects

Since most large-scale weather models do not contain any "urban parametrization", the possibility of correcting the wind speed in the surface layer (due to increased roughness) in cities is offered. This will automatically be balanced by a vertical wind component calculated in the mass balance (see vertical transport below). This correction
Chimere

has however no effect at continental scale where anyway the fraction of urban area in model grid cells remains very small. It has effects on urban versions of the model, and mostly on primary pollutants. The urban parameterization option switch lies in the chimere-[metname] top calling script, and is relayed by the METINFOS-[metname].nml file to be given as input to src/diag/diagmet.f90. However it is not advised to use the switch for MM5 data.

8.2.3 Vertical transport

Vertical transport is assumed to balance horizontal mass divergence/convergence. The scheme used is the first-order UPWIND scheme. More precisely, the algorithm is as follows: In the lowest layer, the mass imbalance is calculated by summing up all incoming mass fluxes and subtracting all outgoing mass fluxes, and a vertical wind through the interface between the lowest and the second layer is calculated to compensate this imbalance. Then the same operation is performed on the second layer, including now in the mass imbalance the fluxes at the cell bottom interface. Then all layers are scanned upward to the top of the model. Vertical transport mass fluxes (independent of species, first order) are calculated first at each coarse time step in routine src/model/vtransport.f90, then the complete calculation using the species concentration is performed at each fine time step in routine src/model/transmix.f90.

8.2.4 Turbulent mixing

Horizontal turbulent fluxes are not considered. Vertical turbulent mixing takes place only in the boundary-layer. The formulation uses K-diffusion following the parameterization of [Troen and Mahrt, 1986], without counter-gradient term. In each model column, diffusivity $K(z)$ is calculated as:

$$K_z = kw_{z} z \left(1 - \frac{z}{h}\right)^{1/3} \quad (8.4)$$

where $w_z$ is a vertical scale given by similarity formulae.

- In the stable case (surface sensible heat flux < 0): $w_z = u_s/(1 + 4.7z/L)$
- in the unstable case: $w_z = (u_{3s}^3 + 2.8ew_{3s}^3)^{1/3}$

where $e = \max(0.1, z/h)$, $L$ is the Obukhov Length, $w_z$ is the convective velocity scale, $u_s$ the friction velocity and $h$ the boundary layer height. A minimal $K(z)$ is assumed, with a value of 0.01 m²/s.

The boundary layer height $h$ can be either directly provided by the meteorological model or recalculated in the src/diag/diagmet.f90 meteo diagnostic model. In the latter case the boundary layer height is considered as the maximum of the [Troen and Mahrt, 1986] boundary layer height calculated from the Richardson number profile, as the lowest altitude where $R_i = 0.5$, and a more convectively-based boundary layer height calculation. The latter is based on a simplified and diagnostic version of the approach of [Cheinet, 2002]. It 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 is taken as the height where vertical velocity stops. Thermals are initiated with a non-vanishing vertical velocity and potential temperature departure, depending on the turbulence similarity parameters in the surface layer.

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:

Net Flux = $V_k (C_{k+1} D_{k}/D_{k+1} - C_k)/H_k$ ,

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:

$$V_k = \frac{K_z}{\frac{1}{2} (H_k + H_{k+1})} \quad (8.5)$$
When the top of the boundary layer is above the model top, $H_{k+1}$ is replaced by $h - \text{<model-top>}$ in the previous equation.

The $K(z)$ profile is calculated in routine `src/model/mixing.f90`, as well as the turbulent velocity profile. Vertical mixing mass fluxes (independent of species, first order) are calculated first at each coarse time step in routine `src/model/vtransport.f90`, then the complete calculation using the species concentration is performed at each fine time step in routine `src/model/transmix.f90`.

### 8.2.5 Deep convection

The deep convection scheme describes effects of subgrid scale clouds on tropospheric convection, after [Tiedtke, 1989]. The hourly fluxes of entrainment and detrainement in the updrafts and the downdrafts are estimated during the diagmet stage. Profiles are written in the METEO.nc file and read, hour by hour, by CHIMERE. These fluxes are used during the calculation in the `locvalues.f90` and `transmix.f90` routine.

#### 8.2.5.1 Convective due to clouds in the troposphere

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.

![Figure 8.2: Temperatures changes due to shallow and deep convection and Schematic view of a convective cloud](image)

#### 8.2.5.2 Updrafts and downdrafts under convective clouds

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.

Under the cloud, updrafts and downdrafts are observed, Figure 8.2. 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.

### 8.2.6 Parameterization

In CHIMERE, the Tiedtke scheme is implemented. The main goal of this scheme is to estimated a convection mass flux:

\[
M(z) = \rho(z)(a_{up}w_{up}(z) + a_{dw}w_{dw}(z)) = -\rho(z)a_{env}w_{env}(z)
\]  

(8.6)

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

\[
\frac{\partial M(z)}{\partial z}(z) = E(z) - D(z)
\]  

(8.7)

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:

\[
E(z) = E_{up}(z) + E_{dw}(z) \quad \text{and} \quad D(z) = D_{up}(z) + D_{dw}(z)
\]  

(8.8)

### 8.2.7 Pre-processing in diagmet

These four fluxes are estimated in the `diagmet_science.f90` pre-processor routine. An example of estimated fluxes are displayed in the [Figure 8.3](#).

![Figure 8.3: Entrainment and detrainment fluxes in the updrafts and downdrafts.](image)

In order to use the convection scheme, some new calculations were added in the meteorological pre-processor: estimate of vertical wind speed (independently of possibly calculated with the meso-scale meteorological model such as MM5 or WRF), conversion of \(w\) from m/s into Pa/s with the following relation:

\[
w \left[ \frac{Pa}{s} \right] = \frac{\partial z}{\partial t} \left[ \frac{m}{s} \right] \frac{\partial p}{\partial z} \left[ \frac{Pa}{s} \right]
\]  

(8.9)

using the hydrostatic relation:

\[
\frac{\partial p}{\partial z} = -\rho g
\]  

(8.10)
with $g=9.81 \text{ m.s}^{-2}$, $p = \rho RT$ with $R=2.8703 \text{ J.kg}^{-1}$ and the pressure $p$ in mb. We have:

$$\frac{\partial p}{\partial z} = \frac{-gp}{100RT} \quad (8.11)$$

The additional factor 1/100 is only to convert pressure from hPa to Pa. We thus have the direct relation:

$$w \left[ \frac{Pa}{s} \right] = w \left[ \frac{m}{s} \right] \times \frac{-9.8 \times p}{100 \times 2.8703 \times T} \quad (8.12)$$

The output of `diagmet.f90` for deep convection fluxes are written into the `METEO.nc` file and are in kg.m$^{-2}.s^{-1}$.

### 8.2.8 Implementation in CHIMERE

In the CHIMERE model itself, these fluxes are used to estimates chemical concentrations fluxes as:

\[
\frac{\partial M_{up}(z)c_{up}(z)}{\partial z} = E_{up}(z)c_{env} - D_{up}(z)c_{up} \quad (8.13)
\]

\[
\frac{\partial M_{dw}(z)c_{dw}(z)}{\partial z} = E_{dw}(z)c_{env} - D_{dw}(z)c_{dw} \quad (8.14)
\]

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 8.4.

![Figure 8.4: Convective atmospheric column and its environment](image)

The net flux is first estimated in the `locvalues.f90` routine. At this stage, the fluxes are in molecules.cm$^{-2}.s^{-1}$ (after conversion in `iniphys.f90`). For each grid cell, the Equation 8.7 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:
\[ FLXU(z) = \frac{\partial M_{up}(z)}{\partial z}(z) = E_{up}(z) - D_{up}(z) + FLXU(z - 1) \]
\[ FLXD(z) = \frac{\partial M_{dw}(z)}{\partial z}(z) = E_{dw}(z) - D_{dw}(z) + FLXD(z - 1) \] (8.15)

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 `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:

\[ FLXU(nv) \times \text{concu}(nv) = FLXU(nv - 1) \times \text{concu}(nv - 1) + dpeu(nv) \times \text{conc}(nv) - dpdu(nv) \times \text{concu}(nv) \] (8.16)

and we have:

\[ \text{concu}(nv) = \frac{FLXU(nv - 1) \times \text{concu}(nv - 1) + dpeu(nv) \times \text{conc}(nv)}{dpdu(nv) + FLXU(nv)} \] (8.17)
Chapter 9

Chemistry

9.1 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. It is based on the application of a Gauss-Seidel iteration scheme to the 2-step implicit backward differentiation (BDF2) formula:

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

(9.1)

with \( c^n \) being the vector of chemical concentrations at time \( t_n \), \( \Delta t \) 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

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

(9.2)

The implicit nonlinear system obtained in this scheme can be solved pertinently with a Gauss-Seidel method (Verwer, 1994).

Figure 9.1: Principle of 'operator-splitting' versus Chimere integration
In CHIMERE the production and loss terms $P$ and $L$ in equation 9.2 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. The "operator splitting" technique (see e.g. [McRae et al., 1982], which is still a standard way of solving the 3D transport-chemistry-problem, and the associated "splitting error" are therefore avoided. 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.

In practice, time stepping has two time steps: A coarse time step, where all physical variables are calculated, and a fine step (refined step in order to gain accuracy on chemistry. During loops over fine steps, physical variables are maintained constant.

In the provided version for the continental models, we chose a "very quick formulation", with a 10-minute physical step, no sub-chemical steps, i.e. all processes are stepped to 10 minutes, with only 1 Gauss-Seidel iteration. It is STRONGLY recommended, in order to have more accurate results, to use 2 Gauss-Seidel iterations, which increases by 2 the computer time. 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.

The TWOSTEP formulae and loops are performed in module `src/model/twostep_mod.f90`.

### 9.2 Gas-phase chemistry

#### 9.2.1 Chemical mechanisms (gas phase)

CHIMERE offers the option to include different gas phase chemical mechanisms. The original, complete scheme ([Lattuati, 1997]), hereafter called MELCHIOR1, 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 NOx conditions and NOx-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]. In order to reduce the computing time a reduced mechanism with less than species and about 120 reactions is derived from MELCHIOR ([Derognat et al., 2003]), following the concept of "chemical operators" ([Carter, 1990]). This reduced mechanism is called MELCHIOR2 hereafter.

The base chemical mechanisms files are placed in `chemprep/data`, the mechanisms are automatically generated and placed in the `chemprep/inputdata` directories. New mechanisms can be implemented, in this case a work on the `chemprep/chemprep.awk` is required in adequation with the `REACTIONS` file for a new set of reactions. Not all forms of reactions are allowed. Arrhenius, Troe, photolysis and few other classical reaction rate types are only allowed. Introducing other types implies some recoding. See the technical documentation (the chemprep preprocessor). A file with altitude-dependent photolysis rates must also be provided, named `PHOTO_Parameters[meca]` in `chemprep/data`.

Photolysis rates are calculated under clear sky conditions as a function of height using the TUV model ([Madronich et al., 1998]). Then clouds are taken into account in a highly parameterized fashion. **Important notice:** Photolysis rates are estimated only up to 9000m. If you want to use the model with higher vertical levels, you have to remember CHIMERE results would be correct for transport (with $z_{max}=14000m$) but NOT for gas chemistry.

Gas-phase chemistry constants is read in initialization routine `src/initio/inichem.f90`. Then rates are calculated at the coarse time step in routine `src/model/rates.f90`, but chemical fluxes are calculated for all species at the fine time step in routine `src/model/chemistry.f90`. 
### 9.2.2 Species list of the reduced MELCHIOR2 gas-phase chemical mechanism

<table>
<thead>
<tr>
<th>Species</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Inorganic compounds</strong></td>
<td></td>
</tr>
<tr>
<td>O3</td>
<td>ozone</td>
</tr>
<tr>
<td>H2O2</td>
<td>hydrogen peroxide</td>
</tr>
<tr>
<td>OH</td>
<td>hydroxy radical</td>
</tr>
<tr>
<td>HO2</td>
<td>hydroperoxy radical</td>
</tr>
<tr>
<td>NO</td>
<td>nitric oxide</td>
</tr>
<tr>
<td>NO2</td>
<td>nitrogen dioxide</td>
</tr>
<tr>
<td>NO3</td>
<td>nitrogen trioxide</td>
</tr>
<tr>
<td>N2O5</td>
<td>dinitrogen pentoxide</td>
</tr>
<tr>
<td>HONO</td>
<td>nitrous acid</td>
</tr>
<tr>
<td>CO</td>
<td>carbon monoxide</td>
</tr>
<tr>
<td>SO2</td>
<td>sulfur dioxide</td>
</tr>
<tr>
<td><strong>Hydrocarbon species</strong></td>
<td></td>
</tr>
<tr>
<td>CH4</td>
<td>methane</td>
</tr>
<tr>
<td>C2H6</td>
<td>ethane</td>
</tr>
<tr>
<td>C3H8</td>
<td>n-butane</td>
</tr>
<tr>
<td>C2H4</td>
<td>ethene</td>
</tr>
<tr>
<td>C3H6</td>
<td>propene</td>
</tr>
<tr>
<td>C3H8</td>
<td>o-xylene</td>
</tr>
<tr>
<td>APINEN</td>
<td>α-pinene</td>
</tr>
<tr>
<td>BPINEN</td>
<td>β-pinene</td>
</tr>
<tr>
<td>LIMONE</td>
<td>limonene</td>
</tr>
<tr>
<td>TERPEN</td>
<td>terpenes (lumped class)</td>
</tr>
<tr>
<td>HUMULE</td>
<td>Humulene (lumped class)</td>
</tr>
<tr>
<td>OCIMEN</td>
<td>Ocimene (lumped class)</td>
</tr>
<tr>
<td><strong>Carbonyls</strong></td>
<td></td>
</tr>
<tr>
<td>HCHO</td>
<td>formaldehyde</td>
</tr>
<tr>
<td>CH3CHO</td>
<td>acetaldehyde</td>
</tr>
<tr>
<td>CH3COE</td>
<td>methyl ethyl ketone</td>
</tr>
<tr>
<td>GLYOX</td>
<td>glyoxal</td>
</tr>
<tr>
<td>MGLYOX</td>
<td>dimethyl glyoxal</td>
</tr>
<tr>
<td>CH3COY</td>
<td>dimethyl vinyl ketone</td>
</tr>
<tr>
<td>MEMALD</td>
<td>unsaturated dicarboxyls, reacting like 4-oxo-2-pentenal</td>
</tr>
<tr>
<td>MVK</td>
<td>methyl vinyl ketone</td>
</tr>
<tr>
<td>MAC</td>
<td>methacroleine</td>
</tr>
<tr>
<td><strong>Organic nitrates</strong></td>
<td></td>
</tr>
<tr>
<td>PAN</td>
<td>peroxyacetyl nitrate</td>
</tr>
<tr>
<td>CARNIT</td>
<td>nitrate carbonyl taken as α-nitrooxy acetone</td>
</tr>
<tr>
<td>ISNI</td>
<td>unsaturated nitrate from isoprene degradation</td>
</tr>
<tr>
<td><strong>Organic peroxides</strong></td>
<td></td>
</tr>
<tr>
<td>CH3O2H</td>
<td>methyl hydroperoxide</td>
</tr>
<tr>
<td>PPA</td>
<td>peroxy acetyl radical</td>
</tr>
<tr>
<td><strong>(Per)oxy radicals</strong></td>
<td></td>
</tr>
<tr>
<td>CH3O2</td>
<td>methyl peroxy radical</td>
</tr>
<tr>
<td>CH3COO</td>
<td>peroxy acetyl radical</td>
</tr>
</tbody>
</table>

### Reaction list of the reduced MELCHIOR2 chemical mechanism

<table>
<thead>
<tr>
<th>Reactions</th>
<th>Kinetic constants</th>
<th>Ref.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Inorganic chemistry</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>O3+NO → NO2</td>
<td>Aexp, B/T, A=1.8e-12, B=1370</td>
<td>[1]</td>
</tr>
<tr>
<td>O3+NO2 → NO3</td>
<td>Aexp, B/T, A=1.2e-13, B=2450</td>
<td>[1]</td>
</tr>
<tr>
<td>O3+OH → HO2</td>
<td>Aexp, B/T, A=1.9e-12, B=1000</td>
<td>[1]</td>
</tr>
<tr>
<td>O3+HO2 → OH</td>
<td>Aexp, B/T, A=1.4e-14, B=600</td>
<td>[1]</td>
</tr>
<tr>
<td>NO+HO2 → OH+NO2</td>
<td>Aexp, B/T, A=3.7e-12, B=240</td>
<td>[1]</td>
</tr>
<tr>
<td>NO2+OH+M→HNO3</td>
<td>troe(3.4e-30, 3.2, 2.4, 7.7e-11, 0, 1, 4, 0.30)</td>
<td>[2], n9</td>
</tr>
<tr>
<td>HO2+OH → H2O</td>
<td>Aexp, B/T, A=4.8e-11, B=250</td>
<td>[1]</td>
</tr>
<tr>
<td>H2O2+OH → HO2</td>
<td>Aexp, B/T, A=2.9e-12, B=160</td>
<td>[1]</td>
</tr>
<tr>
<td>HNO3+OH → NO3</td>
<td>Aexp, B/T, A=5.5e-15, B=985</td>
<td>[3], n1</td>
</tr>
<tr>
<td>CO+OH → HO2+CO2</td>
<td>Aexp, B/T(300/T)³, A=2e-13, B=0, N=1</td>
<td>[3], n1</td>
</tr>
<tr>
<td>HO2+HO2 → H2O2</td>
<td>Aexp, B/T, A=2.2e-13, B=740</td>
<td>[1], n1</td>
</tr>
<tr>
<td>HO2+HO2+H2O → H2O2</td>
<td>Aexp, B/T, A=4.52e-34, B=2827</td>
<td>[1], n1</td>
</tr>
</tbody>
</table>

*For the concept of “chemical operators” see Carter (1990) and Aumont et al. (1997)*
<table>
<thead>
<tr>
<th>Reactions</th>
<th>Kinetic constants</th>
<th>Ref.</th>
</tr>
</thead>
<tbody>
<tr>
<td>NO$_2$+H$_2$O $\rightarrow$ NO$_2$+OH</td>
<td>$4e^{-12}$</td>
<td>[1]</td>
</tr>
<tr>
<td>NO$_2$+H$_2$O $\rightarrow$ HNO$_3$+H$_2$O</td>
<td>2e-15</td>
<td>[3]</td>
</tr>
<tr>
<td>NO$_3$+NO $\rightarrow$ 2*N$_2$O$_5$</td>
<td>$Ae^{-B/T}, A=1.8e-11, B=110$</td>
<td>[1]</td>
</tr>
<tr>
<td>NO$_3$+NO$_2$ $\rightarrow$ NO$_5$+NO$_2$</td>
<td>$Ae^{-B/T}, A=4.5e-14, B=1260$</td>
<td>[3]</td>
</tr>
<tr>
<td>NO$_2$+NO$_3$+M $\rightarrow$ N$_2$O$_5$</td>
<td>$troe(2.7e-30,0.3,4.2e-12,0.2,0.33)$</td>
<td>[1], n9</td>
</tr>
<tr>
<td>N$_2$O$_5$+M $\rightarrow$ NO$_3$+NO$_2$</td>
<td>$troe(1e-3,1,10000,5,5,9,7e14,110800,-0.1,0.33)$</td>
<td>[1], n9</td>
</tr>
<tr>
<td>N$_2$O$_5$+H$_2$O $\rightarrow$ 2*N$_2$O$_3$</td>
<td>2.6e-22</td>
<td>[4]</td>
</tr>
<tr>
<td>N$_2$O$_5$+H$_2$O $\rightarrow$ 2*N$_2$H$_4$</td>
<td>2e-39</td>
<td>[4]</td>
</tr>
<tr>
<td>NO$_2$+OH+M $\rightarrow$ HONO</td>
<td>$troe(7e-31,0,2.6,1.5e-11,0.5,0.6)$</td>
<td>[1], n9</td>
</tr>
<tr>
<td>HONO+OH $\rightarrow$ NO$_2$</td>
<td>$Ae^{-B/T}, A=1.8e-11, B=390$</td>
<td>[1]</td>
</tr>
<tr>
<td>NO$_2$+OH+O$_2$ $\rightarrow$ 2*N$_2$O$_5$</td>
<td>$Ae^{-B/T}, A=3.3e-39, B=-530.0$</td>
<td>[1]</td>
</tr>
<tr>
<td>NO$_2$ $\rightarrow$ HONO</td>
<td>0.5*depo(NO$_2$)</td>
<td>[18]</td>
</tr>
</tbody>
</table>

**SOx chemistry**

<table>
<thead>
<tr>
<th>Reactions</th>
<th>Kinetic constants</th>
<th>Ref.</th>
</tr>
</thead>
<tbody>
<tr>
<td>SO$_2$+CH$_3$O$_3$ $\rightarrow$ H$_2$SO$_4$+HCHO+H$_2$O</td>
<td>4e-17</td>
<td>[3]</td>
</tr>
<tr>
<td>SO$_2$+OH+M $\rightarrow$ H$_2$SO$_4$+H$_2$O</td>
<td>$troe(4e-31,0,3,3,2e-12,0,0,0.45)$</td>
<td>[3], n9</td>
</tr>
</tbody>
</table>

**OH attack to hydrocarbons**

<table>
<thead>
<tr>
<th>Reactions</th>
<th>Kinetic constants</th>
<th>Ref.</th>
</tr>
</thead>
<tbody>
<tr>
<td>CH$_3$+OH $\rightarrow$ CH$_3$O</td>
<td>$Ae^{-B/T}, A=2.3e-12, B=1765$</td>
<td>[1]</td>
</tr>
<tr>
<td>CH$_2$=CH=OH $\rightarrow$ CH$_3$CHO+oRO$_2$</td>
<td>$Ae^{-B/T}, A=7.9e-12, B=1030$</td>
<td>[1]</td>
</tr>
<tr>
<td>CH$_3$H$_2$+OH $\rightarrow$ 0.9<em>CH$_3$CH=COE+0.1</em>CH$_3$CHO+0.1<em>CH$_3$COO +0.9</em>oRO$_2$</td>
<td>$Ae^{-B/T}(300/T)^N, A=1.3e-12, B=-190, N=-2$</td>
<td>[5]</td>
</tr>
<tr>
<td>C$_2$H$_4$+OH+M $\rightarrow$ 2*HCHO+oRO$_2$</td>
<td>$troe(7e-29,0,3,1,9e-12,0,0,0.7)$</td>
<td>[1], n9</td>
</tr>
<tr>
<td>C$_3$H$_6$=OH+M $\rightarrow$ HCHO+CH$_3$CHO+oRO$_2$</td>
<td>$troe(8e-27,0,3,5,3e-11,0,0,0.5)$</td>
<td>[1], n9</td>
</tr>
<tr>
<td>OXYL+OH $\rightarrow$ MEMALD+MGLYOX+oRO$_2$</td>
<td>1.37e-11</td>
<td>[5]</td>
</tr>
<tr>
<td>CH$_5$=O $\rightarrow$ 0.32<em>MAC+0.42</em>MVK +0.74*HCHO +obio</td>
<td>$Ae^{-B/T}, A=2.55e-11, B=-410$</td>
<td>[1]</td>
</tr>
<tr>
<td>APINE+OH $\rightarrow$ 0.8<em>CH$_3$CHO+0.8</em>CH$_3$COE+obio</td>
<td>$Ae^{-B/T}, A=1.21e-11, B=-444$</td>
<td>[17]</td>
</tr>
<tr>
<td>BPINE+OH $\rightarrow$ 0.8<em>CH$_3$CHO+0.8</em>CH$_3$COE+obio</td>
<td>$Ae^{-B/T}, A=1.21e-11, B=-444$</td>
<td>[17]</td>
</tr>
<tr>
<td>LIMONE+OH $\rightarrow$ 0.8<em>CH$_3$CHO+0.8</em>CH$_3$COE+obio</td>
<td>$Ae^{-B/T}, A=1.21e-11, B=-444$</td>
<td>[17]</td>
</tr>
<tr>
<td>HUMULE+OH $\rightarrow$ 0.8<em>CH$_3$CHO+0.8</em>CH$_3$COE+obio</td>
<td>$Ae^{-B/T}, A=1.21e-11, B=-444$</td>
<td>[17]</td>
</tr>
<tr>
<td>OCIMEN+OH $\rightarrow$ 0.8<em>CH$_3$CHO+0.8</em>CH$_3$COE+obio</td>
<td>$Ae^{-B/T}, A=1.21e-11, B=-444$</td>
<td>[17]</td>
</tr>
<tr>
<td>TERPEN+OH $\rightarrow$ 0.8<em>CH$_3$CHO+0.8</em>CH$_3$COE+obio</td>
<td>$Ae^{-B/T}, A=1.21e-11, B=-444$</td>
<td>[17]</td>
</tr>
</tbody>
</table>

**OH attack to carboxyls and peroxides**

<table>
<thead>
<tr>
<th>Reactions</th>
<th>Kinetic constants</th>
<th>Ref.</th>
</tr>
</thead>
<tbody>
<tr>
<td>HCHO+OH $\rightarrow$ CO+H$_2$O</td>
<td>$Ae^{-B/T}, A=8.6e-12, B=-20$</td>
<td>[1]</td>
</tr>
<tr>
<td>CH$_3$CHO+OH $\rightarrow$ CH$_3$COO</td>
<td>$Ae^{-B/T}, A=5.6e-12, B=-310$</td>
<td>[1]</td>
</tr>
<tr>
<td>MEMALD+OH $\rightarrow$ GLYOX+MGLYOX+oRO$_2$</td>
<td>5.6e-11</td>
<td>[1]</td>
</tr>
<tr>
<td>CH$_3$OE+OH $\rightarrow$ CH$_3$CO+oRO$_2$</td>
<td>$Ae^{-B/T}(300/T)^N, A=2.92e-13, B=-414, N=-2$</td>
<td>[1]</td>
</tr>
<tr>
<td>GLYOX+OH $\rightarrow$ 2*CO+H$_2$O</td>
<td>1.1e-11</td>
<td>[1]</td>
</tr>
<tr>
<td>MGROXY+OH $\rightarrow$ CH$_3$COO+CO</td>
<td>1.5e-11</td>
<td>[1]</td>
</tr>
<tr>
<td>MVK+OH $\rightarrow$ 0.266<em>MGLYOX +0.266</em>HCHO +0.684<em>CH$_3$CHO +0.684</em>CH$_3$COO +0.05<em>ISNI +0.95</em>oRO$_2$</td>
<td>$Ae^{-B/T}, A=4.1e-12, B=-453$</td>
<td>[6]</td>
</tr>
<tr>
<td>MAC+OH $\rightarrow$ 0.5<em>CH$_3$OE+0.5</em>CO$_2$+0.5*oPAN</td>
<td>$Ae^{-B/T}, A=1.8e-11, B=-175$</td>
<td>[6]</td>
</tr>
<tr>
<td>CH$_3$OH+H$_2$O $\rightarrow$ CH$_3$OH</td>
<td>$Ae^{-B/T}, A=1.9e-12, B=-190$</td>
<td>[1]</td>
</tr>
<tr>
<td>PPA+OH $\rightarrow$ CH$_3$COO</td>
<td>$Ae^{-B/T}, A=1.9e-12, B=-190$</td>
<td>n2</td>
</tr>
<tr>
<td>CH$_3$OH+H$_2$O $\rightarrow$ HCHO+OH</td>
<td>$Ae^{-B/T}, A=1.1e-12, B=-190$</td>
<td>[1]</td>
</tr>
<tr>
<td>oROOH+OH $\rightarrow$ 0.8<em>OH+0.2</em>oRO$_2$</td>
<td>$Ae^{-B/T}, A=4.35e-12, B=-455$</td>
<td>n2</td>
</tr>
<tr>
<td>oBIOH+OH $\rightarrow$ OH</td>
<td>8e-11</td>
<td>n2</td>
</tr>
<tr>
<td>PANH+OH $\rightarrow$ 0.2*oPAN</td>
<td>1.64e-11</td>
<td>n2</td>
</tr>
<tr>
<td>CARNIT+OH $\rightarrow$ CH$_2$CHO+CO+NO$_2$</td>
<td>$k(T)=Ae^{-B/T}, A=5.6e-12, B=-310$</td>
<td>[1]</td>
</tr>
</tbody>
</table>

**OH attack to hydrocarbons and aldehydes**

<table>
<thead>
<tr>
<th>Reactions</th>
<th>Kinetic constants</th>
<th>Ref.</th>
</tr>
</thead>
<tbody>
<tr>
<td>C$_2$H$_4$+NO$_3$ $\rightarrow$ 0.5*CARNIT+HCHO+oRO$_1$</td>
<td>2e-16</td>
<td>[1]</td>
</tr>
<tr>
<td>C$_5$H$_8$+NO$_3$ $\rightarrow$ 0.5<em>CARNIT +1.5</em>HCHO +0.5<em>CH$_3$CHO +0.5</em>H$_2$O +oRO$_1$</td>
<td>9.45e-15</td>
<td>[1]</td>
</tr>
</tbody>
</table>
### Reactions

<table>
<thead>
<tr>
<th>Reactions</th>
<th>Kinetic constants</th>
<th>Ref.</th>
</tr>
</thead>
<tbody>
<tr>
<td>APINEN+NO₃ → CH₂CHO+CH₃COE+oRN1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BPINEN+NO₃ → CH₂CHO+CH₃COE+oRN1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LIMONE+NO₃ → CH₂CHO+CH₃COE+oRN1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OCIMENE+NO₃ → CH₂CHO+CH₃COE+oRN1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HUMULE+NO₃ → CH₂CHO+CH₃COE+oRN1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TERPEN+NO₃ → CH₂CHO+CH₃COE+oRN1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C5H₆+NO₃ → 0.85<em>ISNI +0.1</em>MAC +0.05*MVK</td>
<td>7.8e-13</td>
<td>[6]</td>
</tr>
<tr>
<td>+0.15<em>HCHO +0.8</em>HO₂ +oRN1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HCHO+NO₃ → CO+HNO₂+HO₂</td>
<td>5.8e-16</td>
<td>[1]</td>
</tr>
<tr>
<td>CH₂CHO+NO₃ → CH₃COO+HNO₃</td>
<td>2.8e-15</td>
<td>[9]</td>
</tr>
<tr>
<td>CH₂O₂+NO₃ → HCHO+HO₂+NO₂</td>
<td>1.2e-12</td>
<td>[10]</td>
</tr>
<tr>
<td>CH₂COO+NO₃ → CH₃O₂+NO₂+CO₂</td>
<td>4e-12</td>
<td>[10]</td>
</tr>
<tr>
<td>oRO₂+NO₃ → NO₂+HO₂</td>
<td>1.2e-12</td>
<td>[10]</td>
</tr>
<tr>
<td>obio+NO₃ → NO₂+HO₂</td>
<td>1.2e-12</td>
<td>[10]</td>
</tr>
<tr>
<td>oPAN+NO₃ → NO₂+HO₂</td>
<td>4e-12</td>
<td>[10]</td>
</tr>
<tr>
<td>oRN₁+NO₃ → 1.5*NO₂</td>
<td>1.2e-12</td>
<td>[10]</td>
</tr>
</tbody>
</table>

#### Organic radical conversion

<table>
<thead>
<tr>
<th>Reactions</th>
<th>Kinetic constants</th>
<th>Ref.</th>
</tr>
</thead>
<tbody>
<tr>
<td>CH₃O₂+NO → HCHO+NO₂+HO₂</td>
<td></td>
<td>[1]</td>
</tr>
<tr>
<td>CH₂CO+NO → CH₂O₂+NO₂+CO₂</td>
<td>2e-11</td>
<td></td>
</tr>
<tr>
<td>oRO₂+NO → NO₂+HO₂</td>
<td>4e-12</td>
<td>n3</td>
</tr>
<tr>
<td>obio+NO → 0.86<em>NO₂+0.78</em>HO₂+0.14*ISNI</td>
<td></td>
<td>[6]</td>
</tr>
<tr>
<td>oPAN+NO → NO₂+HO₂</td>
<td>1.4e-11</td>
<td>n3</td>
</tr>
<tr>
<td>oRN₁+NO → 1.5*NO₂</td>
<td>4e-11</td>
<td>n3</td>
</tr>
</tbody>
</table>

#### Organic radical recombination

<table>
<thead>
<tr>
<th>Reactions</th>
<th>Kinetic constants</th>
<th>Ref.</th>
</tr>
</thead>
<tbody>
<tr>
<td>CH₃O₂+HO₂ → CH₂O₂H</td>
<td></td>
<td>[1]</td>
</tr>
<tr>
<td>CH₂CO+HO₂ → 0.67<em>PPA+0.33</em>O₃</td>
<td></td>
<td>[1]</td>
</tr>
<tr>
<td>oRO₂+HO₂ → oROOH</td>
<td></td>
<td>n4</td>
</tr>
<tr>
<td>obio+HO₂ → obioH</td>
<td></td>
<td>n4</td>
</tr>
<tr>
<td>oPAN+HO₂ → PANH</td>
<td></td>
<td>n4</td>
</tr>
<tr>
<td>oRN₁+HO₂ → X</td>
<td></td>
<td>n4</td>
</tr>
<tr>
<td>CH₂O₂+CH₂O₂ → 1.35<em>HCHO+0.7</em>HO₂</td>
<td></td>
<td>[1], n5</td>
</tr>
<tr>
<td>Reactions</td>
<td>Kinetic constants</td>
<td>Ref.</td>
</tr>
<tr>
<td>-----------</td>
<td>------------------</td>
<td>------</td>
</tr>
<tr>
<td>CH$_3$COO$+$CH$_3$O$_2$ $\rightarrow$ 0.5$\times$CH$_3$O$_2$+0.5$\times$CO$_2$+HCHO+0.5$\times$HO$_2$</td>
<td>Ae$^{-B/T}$,A=3.34e-12,B=-400</td>
<td>[1], n5</td>
</tr>
<tr>
<td>oRO$_2$+CH$_3$O$_2$ $\rightarrow$ 0.65$\times$HCHO+0.8$\times$HO$_2$+0.35$\times$CH$_3$OH</td>
<td>Ae$^{-B/T}$,A=1.5e-13,B=220</td>
<td>n5</td>
</tr>
<tr>
<td>obio$+$CH$_3$O$_2$ $\rightarrow$ 0.8$\times$HO$_2$+0.5$\times$HCHO</td>
<td>Ae$^{-B/T}$,A=2.44e-11,B=223</td>
<td>n5</td>
</tr>
<tr>
<td>oPA$+$CH$_3$O$_2$ $\rightarrow$ HCHO+0.5$\times$HO$_2$</td>
<td>Ae$^{-B/T}$,A=7.9e-12,B=140</td>
<td>n5</td>
</tr>
<tr>
<td>CH$_3$COO$+$CH$_5$COO $\rightarrow$ 2.$\times$CH$_3$O$_2$+2.$\times$CO$_2$</td>
<td>Ae$^{-B/T}$,A=2.8e-12,B=530</td>
<td>[1], n5</td>
</tr>
<tr>
<td>oRO$_2$+CH$_3$COO $\rightarrow$ 0.8$\times$CH$_3$O$_2$+0.8$\times$CO$_2$+0.8$\times$HO$_2$</td>
<td>Ae$^{-B/T}$,A=8.6e-13,B=260</td>
<td>n5</td>
</tr>
<tr>
<td>obio$+$CH$_3$COO $\rightarrow$ 0.5$\times$HCHO+1.5$\times$HO$_2$+0.7$\times$CO$_2$</td>
<td>Ae$^{-B/T}$,A=1.18e-11,B=127</td>
<td>n5</td>
</tr>
<tr>
<td>oPA$+$CH$_3$COO $\rightarrow$ CH$_3$O$_2$+CO$_2$+HO$_2$</td>
<td>Ae$^{-B/T}$,A=3.34e-12,B=400</td>
<td>n5</td>
</tr>
<tr>
<td>oRO$_2$+oRO$_2$ $\rightarrow$ 1.3$\times$HO$_2$</td>
<td>6.4e-14</td>
<td>n5</td>
</tr>
</tbody>
</table>

**organic nitrates and pernitrates**

<table>
<thead>
<tr>
<th>Reactions</th>
<th>Kinetic constants</th>
<th>Ref.</th>
</tr>
</thead>
<tbody>
<tr>
<td>CH$_3$COO$+$NO$_2$+M $\rightarrow$ PAN</td>
<td>troe(2.7e-28,0.71,1.12e-11,0,0.9,0.3)</td>
<td>[1], n9</td>
</tr>
<tr>
<td>oPA$+$NO$_2$+M $\rightarrow$ toPAN</td>
<td>troe(2.7e-28,0.71,1.12e-11,0,0.9,0.3)</td>
<td>n6</td>
</tr>
<tr>
<td>PAN$+$M $\rightarrow$ CH$_3$COO+NO$_2$</td>
<td>troe(4.9e-3,121000,0.5,4e16,13830,0,0,3)</td>
<td>[1], n9</td>
</tr>
<tr>
<td>toPA$+$M $\rightarrow$ oPA$+$NO$_2$</td>
<td>troe(4.9e-3,121000,0.5,4e16,13830,0,0,3)</td>
<td>n6</td>
</tr>
<tr>
<td>PAN$+$OH $\rightarrow$ HCHO+NO$_2$+CO$_2$</td>
<td>Ae$^{-B/T}$,A=9.5e-13,B=650</td>
<td>[1]</td>
</tr>
<tr>
<td>toPA$+$OH $\rightarrow$ NO$_2$+CO$_2$</td>
<td>Ae$^{-B/T}$,A=3.25e-13,B=500</td>
<td>n6</td>
</tr>
</tbody>
</table>

**photolytic reactions**

<table>
<thead>
<tr>
<th>Reactions</th>
<th>Kinetic constants</th>
<th>Ref.</th>
</tr>
</thead>
<tbody>
<tr>
<td>O$_3$ $\rightarrow$ 2$\times$OH</td>
<td>photorate</td>
<td>[3], n7</td>
</tr>
<tr>
<td>NO$_3$ $\rightarrow$ NO+O$_3$</td>
<td>photorate</td>
<td>[3]</td>
</tr>
<tr>
<td>NO$_3$ $\rightarrow$ NO$_2$+O$_3$</td>
<td>photorate</td>
<td>[3], [11]</td>
</tr>
<tr>
<td>NO$_3$ $\rightarrow$ NO</td>
<td>photorate</td>
<td>[3], [11]</td>
</tr>
<tr>
<td>N$_2$O$_5$ $\rightarrow$ NO$_2$+NO$_3$</td>
<td>photorate</td>
<td>[3]</td>
</tr>
<tr>
<td>H$_2$O$_2$ $\rightarrow$ 2$\times$OH</td>
<td>photorate</td>
<td>[3]</td>
</tr>
<tr>
<td>HNO$_2$ $\rightarrow$ NO$_2$+OH</td>
<td>photorate</td>
<td>[3]</td>
</tr>
<tr>
<td>HONO $\rightarrow$ NO+OH</td>
<td>photorate</td>
<td>[3]</td>
</tr>
<tr>
<td>HCHO $\rightarrow$ CO$_2$+HO$_2$</td>
<td>photorate</td>
<td>[3]</td>
</tr>
<tr>
<td>HCHO $\rightarrow$ CO$_2$H$_2$</td>
<td>photorate</td>
<td>[3]</td>
</tr>
<tr>
<td>CH$_3$CHO $\rightarrow$ CH$_3$O$_2$+HO$_2$+CO</td>
<td>photorate</td>
<td>[1]</td>
</tr>
<tr>
<td>CH$_3$CO$-$CH$_3$COO$+$CH$_3$CHO+oRO$_2$</td>
<td>photorate</td>
<td>[11], [12]</td>
</tr>
<tr>
<td>CH$_3$COY $\rightarrow$ 2$\times$CH$_3$COO</td>
<td>photorate</td>
<td>[13], [14]</td>
</tr>
<tr>
<td>MGLYOX $\rightarrow$ CH$_3$COO+HO$_2$+CO</td>
<td>photorate</td>
<td>[1]</td>
</tr>
<tr>
<td>GLYOX $\rightarrow$ 0.6$\times$HO$_2$+2$\times$CO+0.7$\times$H$_2$</td>
<td>photorate</td>
<td>[13], [14]</td>
</tr>
<tr>
<td>MEMALD $\rightarrow$ 0.5$\times$MVK +0.5$\times$MALEIC +0.5$\times$oPA$+$0.5$\times$HCHO+0.5$\times$HO$_2$</td>
<td>photorate</td>
<td>[15]</td>
</tr>
<tr>
<td>CH$_3$O$_2$H $\rightarrow$ HCHO+OH+HO$_2$</td>
<td>photorate</td>
<td>[3]</td>
</tr>
<tr>
<td>PPA$-$CH$_3$O$_2$+CO$_2$+OH</td>
<td>photorate</td>
<td>n8</td>
</tr>
<tr>
<td>oROOH $\rightarrow$ OH+HO$_2$</td>
<td>photorate</td>
<td>n8</td>
</tr>
<tr>
<td>obio$+$OH $\rightarrow$ OH+HO$_2$</td>
<td>photorate</td>
<td>n8</td>
</tr>
<tr>
<td>PANH $\rightarrow$ OH+HO$_2$</td>
<td>photorate</td>
<td>n8</td>
</tr>
<tr>
<td>PAN$-$CH$_3$COO+NO$_2$</td>
<td>photorate</td>
<td>[3]</td>
</tr>
</tbody>
</table>

---

[9] DeMore et al., 1990  
[10] Canosa-Mas et al., 1996  
• Rate constants containing a density dependent term in a minor reaction channel have been simplified by putting an average boundary layer molecular density of $2.5 \times 10^{19}$ molecules/cm$^3$.

• 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).

• The rate constant of the operator reaction oRO$_2$+NO has been set to that of the i-C$_3$H$_7$O$_2$+NO reaction given in [16], that of oPAN+NO to that of CH$_3$COO$_2$+NO [1], the rate constant for oRO$_2$NO+NO is taken from that of ISNIR (nitrate peroxy radicals from isoprene degradation) + NO [6].

• The rate constants for the operator reactions oRO, obio, RO$_2$NI+HO$_2$ have been set to that of the reaction C$_2$H$_5$O$_2$+HO$_2$ [1], that of oPAN+HO$_2$ has been taken from CH$_3$COO$_2$+HO$_2$ [1].

• Radical conserving and terminating pathways are combined to single reactions; rate constants in operator reactions are affected by interpreting oRO$_2$ as 2-butyl peroxy radical [16], obio as a peroxy radical from isoprene degradation [6] and oPAN as CH$_3$COO$_2$.

• Reaction rates of the operators oPAN and OPAN have been set to those of the peroxy acetyl radical and PAN, respectively.

• The reactions O$_3$+h$\nu$ → O(1D), O(1D)+H$_2$O → 2OH, O(1D)+M → O(3P)+M and O(3P)+O$_2$+M → O$_3$+M have been combined to reaction O$_3$+h$\nu$ → O(1D) by taking into account H$_2$O 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$_3$OOH.

• Three body Troe reactions [1], given in the form:

$$k = \frac{k_0[M]}{1 + \frac{k_0[M]}{k_\infty} f^p}$$

with

$$p = (1 + (\log_{10}(k_0[M]/k_\infty))^2)^{-1},$$

$$k_0 = A_0 e^{-\frac{B_0}{T}} \left(\frac{T}{300}\right)^{-n}, \text{ and } k_\infty = A_\infty e^{-\frac{B_\infty}{T}} \left(\frac{T}{300}\right)^{-n},$$

The parameters are given in the order $A_0, B_0, n, A_\infty, B_\infty, m, f$.

9.3 Aerosol processes

9.3.1 Overview

CHIMERE consists in the baseline gas-phase version with MELCHIOR2 chemistry, together with a sectional aerosol module. This module accounts for 7 species (primary particle material, nitrate, sulfate, ammonium, biogenic secondary organic aerosol (SOA), anthropogenic SOA and water). Potentially, Chloride et Sodium can be included (high computing time). In its initial version the module uses 6 bins from 10 nm to 40 µm. Physical processes into account are described below. See also sections on chemistry (aqueous-phase chemistry, heterogeneous chemistry and SOA formation) below. Dry and wet scavenging/deposition are also described in separate sections.
The physical processes accounted for are coagulation ([Gelbard and Seinfeld, 1980]), absorption ([Nenes et al., 1998]) and nucleation ([Kulmala et al., 1998],[Kavouras et al., 1998]). See following subsections for details.

- **Coagulation**: Classical theory already used by [Gelbard and Seinfeld, 1980].
- **Absorption**: Absorption is taken into account for both inorganic and organic species but their processing is different. For inorganic species, equilibrium concentrations are computed with the thermodynamic module ISORROPIA (or a tabulated version) presented in [Nenes et al., 1998]. For secondary organic species, equilibrium concentrations are calculated through a temperature dependent partitioning coefficient ([Pankow, 1994]. Please note that the ISORROPIA code is included in CHIMERE, please check the ISORROPIA web site for downloading the latest version.
- **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. Although 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.

### 9.3.2 Composition and mathematical representation of aerosols

*This section was published in [Bessagnet et al., 2005].*

Atmospheric aerosols are represented by their size distributions and compositions. The sectional representation described by [Gelbard and Seinfeld, 1980] has been used for the density distribution function. The sectional approach is quite useful to solve the governing equation for multicomponent aerosols. It discretizes the density distribution in a finite number of bins ([Warren, 1986]). Thus, all particles in section \( l \) have the same composition and are characterized by their mean diameter \( d_l \). The aerosol module actually uses six bins from 10 nm to 40 \( \mu \)m, following a geometrical progression. For a given \( x \) as \( x = \ln(m) \), with \( m \) the particle mass, \( q(x) \) is the density distribution as defined in **Equation 9.3**, \( Q \) being the mass concentration function. \( Q_l^k (\mu g m^{-3}) \) is the mass concentration of component \( k \) in section \( l \) and \( Q_l (\mu g m^{-3}) \) is the total mass concentration in section \( l \) (**Equation 9.4**).

\[
q(x) = \frac{dQ}{dx} \tag{9.3}
\]

\[
Q_l = \int_{x_{l-1}}^{x_l} q(x) dx = \sum_k Q_l^k \tag{9.4}
\]

<table>
<thead>
<tr>
<th>Model species</th>
<th>Species</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>pPPM</td>
<td>Anthropogenic primary species EC, OCp, and other industrial dusts</td>
<td>primary</td>
</tr>
<tr>
<td>pSOA</td>
<td>Anthropogenic and Biogenic secondary organic aerosol (ASOA+BSOA)</td>
<td>secondary</td>
</tr>
<tr>
<td>pH2SO4</td>
<td>Equivalent Sulfate(*)</td>
<td>secondary</td>
</tr>
<tr>
<td>pHNO3</td>
<td>Equivalent Nitrate(*)</td>
<td>secondary</td>
</tr>
<tr>
<td>pNH3</td>
<td>Equivalent Ammonium(*)</td>
<td>primary emitted, secondary transferred</td>
</tr>
<tr>
<td>pWATER</td>
<td>Water</td>
<td></td>
</tr>
</tbody>
</table>

Table 9.2: List of aerosol species. (*): ions, molecules, crystals

In the model, particles are composed of species listed in Table 9.2. Sulfate is formed through gaseous and aqueous oxidation of SO2. 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 pNH₃ represents an equivalent ammonium as the sum of NH₄⁺ ion, NH₃ liquid, NH₄NO₃ solid, etc.

### 9.3.3 Coagulation

Since \( Q_{kl}^k \) is the mass concentration of component \( k \) in section \( l \), the mass balance equation for coagulation follows Equation 9.5:

\[
\frac{dQ_{kl}^k}{dt}_{coag} = \frac{1}{2} \sum_{i=1}^{l-1} \sum_{j=1}^{l-1} \left[ 1a_{i,j,l}Q_{j}^kQ_{i}^k + 1b_{i,j,l}Q_{i}^kQ_{j}^k \right] - \frac{1}{2} \sum_{i=1}^{l-1} \left[ 2a_{i,l}Q_{l}^kQ_{i}^k - 2b_{i,l}Q_{l}^kQ_{i}^k \right] - \frac{1}{2} \sum_{i=1}^{m} 4\beta_{i,l}Q_{i}^k - Q_{k}^l \sum_{i=1}^{m} 4\beta_{i,l}Q_{i}^k
\]

(9.5)

The sectional coagulation coefficients \( 1a_{i,j,l}, 1b_{i,j,l}, 2a_{i,l}, 2b_{i,l}, 3\beta, 4\beta \) (Fuch, 1964) depend on particle characteristics and meteorological data such as temperature, pressure and turbulence parameters. For submicronic particles, coagulation is essentially driven by Brownian motions. For coarse particles sedimentation is an efficient process.

### 9.3.4 Absorption

The absorption flux \( J (\mu g m^{-3}s^{-1}) \) of a semi-volatile inorganic or organic species onto a monodisperse aerosol is:

\[
J = \frac{1}{\tau}(G - G_{eq})
\]

(9.6)

with \( G \) and \( G_{eq} (\mu g m^{-3}) \), respectively, are the gas phase and equilibrium concentrations. The characteristic time \( \tau \) is:

\[
\tau = \frac{1 + \frac{8\lambda}{\alpha d N}}{2\pi cdN}
\]

(9.7)

with \( \lambda \) (m) is the mean free path of air molecules, \( d \) (m) the particle diameter, \( N \) (particle \( m^{-3} \)) the particle number concentration, \( \alpha \) the accommodation coefficient of the transferred species and \( c \) (m s⁻¹), its mean molecular velocity. For a semi-volatile species \( k \), a mean absorption coefficient \( H_{l}^k \) (s⁻¹) is defined at section \( l \) as:

\[
\left[ \frac{dQ_{l}^k}{dt} \right]_{abs} = H_{l}^kQ_{l}^k
\]

(9.8)

\[
H_{l}^k = \frac{12\lambda c_{0}}{\rho_{p}d_{l}^2 \left( 1 + \frac{8\lambda}{\alpha_{k}d_{l}} \right)} (G_{l}^k - G_{l,eq}^k)
\]

(9.9)

where \( \rho_{p} \) is the particle density (fixed at 1500 kg m⁻³ here). 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]):

\[
G_{l,eq}^k = \frac{Q_{l}^k}{\rho_{p}d_{l}^2 K_{k}^p}
\]

(9.10)
with OM (µgm⁻³) is the absorbent organic material concentration. Considering the thermodynamic equilibrium between the gas and particulate phases, this coefficient is given by

\[ K_p^k = \frac{10^{-6}}{MW_{om} \zeta_k p_i^0} \]  

(9.11)

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), \( p_i^0 \) the vapor pressure of product \( i \) as a pure liquid (atm) and \( \zeta \) the activity coefficient of species in the bulk aerosol phase. The coefficient \( \zeta \), difficult to calculate, is assumed constant and equal to unity. Moreover, an empirical formulae can be used to estimate \( K_p^k \):

\[ \log(K_p^k) = -0.61 \log(p_i^0) - 4.74 \]  

(9.12)

according to [Kaupp and Umlauf, 1992] for organic species.

### 9.3.5 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.

### 9.3.6 ISORROPIA

*This section is extracted from the PHD manuscript of Alma Hodzic, 2005.*

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 calculate the thermodynamical equilibrium of the system: sulfate-nitrate-ammonium-sodium-chloride-water at a given temperature and relative humidity. The possible species are the following:

- **Gas phase**: NH₃, HNO₃, HCl, H₂O.
- **Liquid phase**: NH₄⁺, Na⁺, H⁺, Cl⁻, NO₃⁻, SO₄²⁻, HSO₄⁻, OH⁻, H₂O, HNO₃(aq), HCl(aq), NH₄NO₃(aq), H₂SO₄(aq).
- **Solid phase**: (NH₄)₂SO₄, NH₄HSO₄, (NH₄)₃H(SO₄)₂, NH₄NO₃, NH₄Cl, NaCl, NaNO₃, NaHSO₄, Na₂SO₄.

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). In the CHIMERE model, the calculation of the thermodynamic equilibrium can be done by interpolating a look-up table (see characteristics in Table 9.3). This look-up table has been pre-calculated and is delivered with the model in the file AEROMIN.bin. 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 µ.g.m⁻³. Because of numerical limits, Sodium and Chloride are not accounted for in this table. The active seassalt version needs the to use the on-line coupling.
### Table 9.3: Characteristics of the look-up table used for the calculation of the thermodynamic equilibrium with ISORROPIA.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Value</th>
<th>Increment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Temperature (K)</td>
<td>260</td>
<td>312</td>
</tr>
<tr>
<td>Relative Humidity</td>
<td>0.3</td>
<td>0.99</td>
</tr>
<tr>
<td>( \text{H}_2\text{SO}_4, \text{HNO}_3, \text{NH}_3 (\mu\text{g.m}^{-3}) )</td>
<td>( 10^{-2} )</td>
<td>65</td>
</tr>
</tbody>
</table>

The use of the look-up table allows to run the model faster, some errors can occur around each deliquescent point. This approach has been compared to the on-line coupling and tested on an entire year (2001) at the regional scale (0.5°x0.5°) over Europe.

### Table 9.4: Relative humidity at deliquescent points for the species used by ISORROPIA in the system sulfate-nitrate-ammonium-water.

<table>
<thead>
<tr>
<th>Species</th>
<th>DRH (298.15K)</th>
<th>Mixing</th>
<th>MDRH (298.15K)</th>
</tr>
</thead>
<tbody>
<tr>
<td>( (\text{NH}_4)_2\text{SO}_4 )</td>
<td>0.799</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( \text{NH}_4\text{NO}_3 )</td>
<td>0.618</td>
<td></td>
<td>0.600</td>
</tr>
<tr>
<td>( \text{NH}_4\text{HSO}_4 )</td>
<td>0.400</td>
<td></td>
<td>0.360</td>
</tr>
<tr>
<td>( (\text{NH}_4)_3\text{H(SO}_4)_2 )</td>
<td>0.690</td>
<td></td>
<td>0.675</td>
</tr>
</tbody>
</table>

The comparison shows the use of an on-line coupling leads to a weak decrease of the mean concentrations (13% for the nitrates, and no more than 2% for the sulfates) on the EMEP site. Ammonium concentrations are slightly increase (1 to 5%). In absolute values, the differences never exceed 0.5 \( \mu\text{g.m}^{-3} \) in average for the nitrates and 0.1 \( \mu\text{g.m}^{-3} \) for the sulfates and ammonium. The aerosol water content is slightly decreased (3 to 13% ; 0.1 to 2.8 \( \mu\text{g.m}^{-3} \)).

The sensitivity tests showed very weak differences, less than ± 20%. The differences are higher in winter compare to summer. The highest differences are observed above 50-60% for the nitrates and ammonium corresponding to the main deliquescent points. Indeed, the look-up table is not enough refined in this zone to account for the strong variations of the concentrations. For the water, the highest differences are observed in the range 95-100% where the absorption of water can increase by a factor of 1000. The differences do not depend on the temperature (not shown here).

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 \( \text{NH}_3/\text{NH}_3 \) and \( \text{HNO}_3/\text{NO}_3 \).

### 9.3.7 Multiphase chemistry

#### 9.3.7.1 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]; [Hoffman and Calvert, 1985]; [Lee and Schwartz, 1983]):

- \( \text{SO}_2^{aq} + \text{O}_3^{aq} \rightarrow \text{SO}_4^{2-} \)
- \( \text{HSO}_4^2 + \text{O}_3^{aq} \rightarrow \text{SO}_4^{2-} \)
- \( \text{SO}_3^{2-} + \text{O}_3^{aq} \rightarrow \text{SO}_4^{2-} \)
- \( \text{SO}_2^{aq} + \text{H}_2\text{O}_2^{aq} \rightarrow \text{SO}_4^{2-} \)
- \( \text{SO}_2^{aq} + \text{NO}_2^{aq} \rightarrow \text{SO}_4^{2-} \)
- \( \text{SO}_3^{2-}(\text{Fe}^{3+}) \rightarrow \text{SO}_4^{2-} \)
$\text{SO}_2$, $\text{H}_2\text{O}_2$ and $\text{O}_3$ in the aqueous phase are in equilibrium with the concentrations in the gas phase. Moreover, aqueous $\text{SO}_2$ is dissociated into $\text{HSO}_3^-$ and $\text{SO}_2^{2-}$. Catalyzed oxidation reactions of sulfur dioxide in aqueous droplets with iron and manganese are considered, following [Hoffman and Calvert, 1985] among others. Henry’s law coefficient and other aqueous equilibrium constants are used ([Seinfeld and Pandis, 1997]). Sulfur chemistry is very pH sensitive. pH is kept bound between 4.5 and 6.0.

### 9.3.7.2 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]:

- $\text{HO}_2 \rightarrow 0.5\text{H}_2\text{O}_2 \quad \gamma = 0.2$
- $\text{NO}_3 \rightarrow \text{HNO}_3 \quad \gamma = 0.001$
- $\text{NO}_2 \rightarrow 0.5\text{HNO}_3 + 0.5\text{HONO} \quad \gamma = 0.0001$
- $\text{N}_2\text{O}_5 \rightarrow 2\text{HNO}_3 \quad \gamma = 0.01 - 1$

The first-order rate constant $k$ for gas heterogeneous loss onto particles is given by:

$$k = \sum_l \left( \frac{d_l}{2D_g} + \frac{4}{V\gamma} \right)^{-1} A_l$$

(9.13)

with $d_l$, the particle diameter (m), $D_g$, the reacting gas molecular diffusivity (m$^2$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 $\text{N}_2\text{O}_5$ is assumed to be temperature-dependent in the range 0.01 - 1 ([De Moore et al., 1994]) with increasing values for decreasing temperatures. A recent study ([Aumont et al., 2003]) suggests that NO2 reactions on wet surfaces could be an important source for HONO production during wintertime smog episodes, so, a new reaction is added (also present in the gas-phase mechanism, see above).

### 9.3.8 Secondary organic aerosol chemistry

The complete chemical scheme implemented in CHIMERE includes biogenic and anthropogenic precursors (Table 9.5). Biogenic precursors include API ($\alpha$-pinene and sabinene), BPI ($\beta$-pinene and $\delta^3$-carene), LIM (limonene), OCI (myrcene and ocimene) and ISO (isoprene). Anthropogenic precursors include TOL (benzene, toluene and other mono-substituted aromatics), TMB (Trimethylbenzene and other poly-substituted aromatics), and NC4H10 (higher alkanes). SOA formation is represented according to a single-step oxidation of the relevant precursors and gas-particle partitioning of the condensable oxidation products. The gas-particle partitioning formulation has been described in detail by [Pun, 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 Raoults’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. Enthalpy of vaporization are given in brackets ($kJ.mol^{-1}$) for each SOA compounds: AnA0D (88), AnA1D(88), AnA2D(88), BiA0D(88), BiA1D(88), BiA2D(109), AnBmP(88), AnBmP(88), BiBmP(175). The full name of compounds are explicited in Table 9.5 caption. The absorption process in CHIMERE is implemented as in [Bowman et al., 1997]. A dynamical approach is adopted to describe the gas/particle conversion since the model time-step is about 5 min. and using the approach by [Bowman et al., 1997], the characteristic time for mass transfer can exceed 20 min. for coarse particles.

$$ J_i = \frac{1}{\tau_i} (G_i - G_{i}^{eq}) $$

(9.14)

$J_i (\mu g.m^{-3}.s^{-1})$ is the absorption or desorption flux of species $i$, $\tau_i (s)$ 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, 2006].

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, 2006]. Higher alkanes and isoprene were added to the original chemical mechanism of [Pun, 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].

9.4 Impact of clouds

9.4.1 Impact on photolysis

In this version clouds are assumed to lie above the model top, so that there is no cloud albedo effect within the model domain depth. For all photolysed species, clear sky photolysis rates $J_c(z)$ are multiplied throughout model columns by an attenuation coefficient $A(d)$ depending on the total cloud optical depth (COD) $d$. Using the TUV model, and a large set of CODs for clouds at various altitudes, the attenuation relative to the clear-sky case has been fitted as a function of COD with the formula:

$$ A(d) = e^{-0.11d^{2/3}} $$

(9.15)

Several options are offered in order to calculate the COD. Total COD, $d$, is the sum of partial CODs from 3 cloud layers, low clouds $d_l$, medium clouds $d_m$ and high clouds $d_h$. The limits between these cloud is user-chosen, but depend on the meteorological model. For MM5, limits of 2000m and 6000m are proposed. For each cloud layer three options are possible for the calculation of the partial cloud optical depth:

- Calculation as a function of Liquid/Ice Water Content in the column; using an assumption of sphericity and equivalent droplet size of 6 microns, an assumption of hexagonal shape for ice particles, the formula for cloud optical depth is $180.C_w + 67.C_i$, where $C_w$ and $C_i$ are respectively the liquid water column (in Kg/m$^2$) and ice column for this cloud layer. This option is chosen by setting the cloud option to ”1” in the METINFOS.nml file.
- Liquid/Ice water are generally unverified and unstable parameters in atmospheric models. Basing a cloud parameterization on such parameters can be risky. A more robust parameterization consists in using relative humidity only. The proposed parameterization (option ”2” in METINFOS.nml file) consists in parameterizing the COD as a function of the integral $\mathbf{R}$, over the cloud depth, of the relative humidity above 75%. It is
An even simpler parameterization (Option "0" in METINFOS.nml) can be achieved by making the COD

\[ \text{COD} = \text{sample COD} \times \text{RH} \]

>COD src/initio/iniphys.f90

The formulation, modifying radiation for RH

and has been tuned using J(NO2) measurements taken over a long period near Paris (manuscript in preparation).

but could be used when cloud/ice water is not available. The third option uses relative (to liquid water) humidity,

midlatitudes. The second option uses cloud fractions (whenever available), which is less meaningful to radiation

fractionary cloudiness. However it may be suited for high/medium clouds which are often stratiform clouds in the

be very model-dependent and which is rarely verified against observations. Moreover it does not take into account

Recommendations

The first option using liquid/ice water (whenever available), is based on a variable which can

assumed that small cloud formation (in particular cumulus clouds) starts at 75% relative humidity. Normalization

of R leads to the formulation (for instance for low clouds): \( d_l = aR/dz \), where \( a=0.02 \) is chosen such that a 1000m-thick layer has an optical cloud depth of 20.

- An even simpler parameterization (Option "0" in METINFOS.nml) can be achieved by making the COD

simply proportional to the cloud fraction (if available) for each cloud layer. Coefficient tuning led to proportionality coefficients of: 50 for low-clouds, 10 for medium clouds and 2 for high clouds. This means, for instance, that a sky covered with 100% of high clouds has an optical depth of 2. Tuning was performed with ECMWF cloud fraction data and should change with the meteorological model.

Table 9.5: 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 nondissociative species (BiA0D), a biogenic once-dissociative species (BiA1D) and a biogenic twice-dissociative species (BiA2D), three hydrophobic species that include an anthropogenic nondissociative species (AnA0D), an anthropogenic twice-dissociative species (AnA2D), a biogenic once-dissociative species (BiA1D) and a biogenic twice-dissociative species (BiA2D), and two surrogate compounds for the isoprene oxidation products.

<table>
<thead>
<tr>
<th>Reactions</th>
<th>kinetic rates (molec.cm(^{-3}).s(^{-1}))</th>
</tr>
</thead>
<tbody>
<tr>
<td>TOL+OH → 0.004×AnA0D + 0.001×AnA1D + 0.084×AnBmP + 0.013×AnBIP</td>
<td>1.81×10(^{-12})exp(355/T)</td>
</tr>
<tr>
<td>TMB+OH → 0.002×AnA0D + 0.002×AnA1D + 0.001×AnA2D + 0.088×AnBmP + 0.006×AnBIP</td>
<td>9.80×10(^{-9})/T</td>
</tr>
<tr>
<td>NC4H10+OH → 0.07×AnBmP</td>
<td>3.60×10(^{-12})exp(190/T)</td>
</tr>
<tr>
<td>API+OH → 0.30×BiA0D + 0.17×BiA1D + 0.10×BiA2D</td>
<td>1.21×10(^{-11})exp(444/T)</td>
</tr>
<tr>
<td>API+O3 → 0.18×BiA0D + 0.16×BiA1D + 0.05×BiA2D</td>
<td>1.01×10(^{-15})exp(-732/T)</td>
</tr>
<tr>
<td>API+NO3 → 0.80×BiBmP</td>
<td>1.19×10(^{-12})exp(490/T)</td>
</tr>
<tr>
<td>BPI+OH → 0.07×BiA0D + 0.08×BiA1D + 0.06×BiA2D</td>
<td>2.38×10(^{-11})exp(357/T)</td>
</tr>
<tr>
<td>BPI+O3 → 0.09×BiA0D + 0.13×BiA1D + 0.04×BiA2D</td>
<td>1.50×10(^{-17})</td>
</tr>
<tr>
<td>BPI+NO3 → 0.80×BiBmP</td>
<td>2.51×10(^{-12})</td>
</tr>
<tr>
<td>LIM+OH → 0.20×BiA0D + 0.25×BiA1D + 0.005×BiA2D</td>
<td>1.71×10(^{-10})</td>
</tr>
<tr>
<td>LIM+O3 → 0.09×BiA0D + 0.10×BiA1D</td>
<td>2×10(^{-16})</td>
</tr>
<tr>
<td>OCI+OH → 0.70×BiA0D + 0.075×BiA1D</td>
<td>5.10×10(^{-8})/T</td>
</tr>
<tr>
<td>OCI+O3 → 0.50×BiA0D + 0.055×BiA1D</td>
<td>7.50×10(^{-14})/T</td>
</tr>
<tr>
<td>OCI+NO3 → 0.70×BiA0D + 0.075×BiA1D</td>
<td>4.30×10(^{-9})/T</td>
</tr>
<tr>
<td>ISO+OH → 0.232×ISOPA1 + 0.0288×ISOPA2</td>
<td>2.55×10(^{-11})exp(410/T)</td>
</tr>
</tbody>
</table>

The attenuation coefficient is calculated in the src/diag/diagmet.f90 diagnostic interface model. It is read in src/initio/iniphys.f90, photolysis rates are calculated in routine src/model/photorates.f90 every coarse time step.
9.4.2 Wet scavenging

9.4.2.1 In cloud scavenging

In cloud, particle scavenging is difficult to model. Particles can be scavenged either by coagulation with cloud droplets or by precipitating drops. Particles also act as cloud condensation nuclei to form new droplets. This latter process of nucleation is the most efficient one in clouds. According to [Tsyro, 2002] and [Guelle et al., 1998], the deposition flux is written as:

\[
\frac{dQ^k_{\text{incl}}}{dt} = \frac{\zeta_1 p_r}{w_1 h} Q^k_l
\]

(9.16)

with \( p_r \) is the precipitation rate released in the grid cell (g cm\(^{-2} \) s\(^{-1} \)), \( w_l \) the liquid water content (g cm\(^{-3} \)), \( h \) the grid height (cm) and \( \zeta \) an empirical uptake coefficient (in the range 0-1) depending on particle composition.

9.4.2.2 Sub-cloud scavenging

Particles are scavenged by raining drops, the deposition flux of particles being given by:

\[
\frac{dQ^k_{\text{subcl}}}{dt} = -\frac{\alpha p E_1}{u_g} Q^k_l
\]

(9.17)

with \( \alpha \) is an empirical coefficient, \( p \) the precipitation rate in the grid cell (g cm\(^{-2} \) s\(^{-1} \)), \( E \) a collision efficiency coefficient between particles and raining drops ([Slinn, 1983] and \( u_g \) the falling drop velocity (cm s\(^{-1} \)). Assuming a constant drop diameter (2 mm), this parameterization is an approximation of equations described in [Seinfeld and Pandis, 1997] and [Jung et al., 2002]. In the next developments, this equation will be improved.
Bibliography


109


Vestreng, V. (2003). Review and revision of emission data reported to CLRTAP. *EMEP Status report*.


Appendix A

References using CHIMERE


18. Menut L., I. Coll and S. Cautenet, 2005, Impact of meteorological data resolution on the forecasted ozone concentrations during the ESCOMPTE IOP 2a and 2b, Atmospheric Research - ESCOMPTE Special Issue, 74, 139-159, 2005


24. Menut L., I. Coll and S. Cautenet, Impact of meteorological data resolution on the forecasted ozone concentrations during the ESCOMPTE IOP 2a and 2b, Atmospheric Research - ESCOMPTE Special Issue, 74, 139-159, 2005.


45. Honoré, C., Vautard, R., Beekmann, M., Low and High Nox chemical regimes in an urban environment, Environmental Modelling and Software, 15, 559-564, 2000


Appendix B

Some scripts used in the CHIMERE suite

B.1 The scripts/make-chemgeom.sh for preprocessing of domain and chemistry input data

# This script is called by the main chimere script and prepares the inputdata related to the simulation.
# 1. chemical files
# 2. domain definition

# Vertical coordinates:
# The VCOORD file is automatically generated and the result is put in domains/VCOORD
# Horizontal coordinates:
# Make your own COORD_${dom} file and put it into /domains/HCOORD directory
# or
# Add lines in the domains/domainlist.nml file to create a new one

#-------------------------------------
# Creating temporary directory
function checkexist { 
  [ -s $1 ]|| { echo "non existent $1 ; exiting." ; exit 1; }
}

echo Temporary Directory : ${chimere_tmp} 
[ -z ${chimere_tmp} ]&& { echo "${0}: TMP directory not defined" ; exit 1 ; }
rm -rf ${chimere_tmp} 
mkdir -p ${chimere_tmp} 
[ -d ${chimere_tmp} ]|| { echo "$0: Cannot create ${chimere_tmp}" ; exit 1 ; }

# Other directories
mkdir -p ${simuldir}

#-------------------------------------
# CHEMISTRY PRE-PROCESSING
bins=0
reactive=0
if [ "${aero} = 1 -o ${dustout} = 1 -o ${seasalt} = 1 -o ${carb} = 1 -o ${dust} = 1" ]; then
  bins=1
fi
if [ "${bins} = 0" ]; then
  nbins=0
fi
if [ $(mecachim) -gt 0 ] ; then
  reactive=1
fi

labchem=${mecachim}${aero}${seasalt}${pops}${dustout}${carb}${trc}${soatyp}${dust}.${nbins}
chempremdir=${chimere_root}/chemprep
domaindir=${chimere_root}/domains
export fnboundir=${chempremdir}/boundaries_spec
export fnbounsel=${fnboundir}/SOURCE_SELECTOR
checkexist ${fnbounsel}
export chemprepfic=${chempremdir}/inputdata.${labchem}
export DATADIR=${chemprepfic}

echo "Directory to use for chemistry: "
echo "--> " $chemprepfic
if [ -d $chemprepfic ] ; then
  echo "--> directory already existing."
  echo "....Go to the domain pre-processor"
fi
if [ ! -d $chemprepfic ] ; then
  echo "--> directory not existing. Running pre-processor..."
  cd $chemprepfic
  \rm -rf $(chimere_tmp)/chemprep
  mkdir -p $(chimere_tmp)/chemprep
  \cp -f $(chempremdir)/src/*.e $(chimere_tmp)/chemprep
  \cp -f $(chempremdir)/chemprep.* $(chimere_tmp)/chemprep
  cd $(chimere_tmp)/chemprep
  . chemprep.univ.sh
  . boun_filter.sh > $(chemprepfic)/source_selector
  cd $(chimere_root)
  script=`echo $0 | gawk '{print $1}'`
cat $script > $(chemprepfic)/$script.save

echo "+++=====================================================================================+++"
echo "+++ FIRST STEP COMPLETED : A NEW CHEMISTRY DATA DIRECTORY HAS BEEN CREATED +++
echo "+++ Go in : " $chemprepfic
echo ""
echo """
echo "+++ If necessary you can change files as OUTPUT_SPECIES, OUTPUT_PARAMETERS, FAMILIES"
echo "+++ Check the following files before running your own simulation : 
  "
echo "+++ AEROSOL, REACTIONS, PRIMARY"
if [ $(trc) -ge 1 ] ; then
  echo "+++ YOU ARE USING TRACERS...."
  echo "+++ Did you check the TRACER and TRACER_PROFILE files?"
fi

exit 0
fi

#------------------------------- GEOMETRY

\rm -rf $(chimere_tmp)/domains
mkdir -p $(chimere_tmp)/domains
### Chimere

```bash
###
echo " o create VCOORD file for vertical characteristics: 
"$first_layer_pressure" 
echo " o surface level: " $first_layer_pressure 
echo " o top level: " $top_chimere_pressure 
\cp ${domaindir}/define_geom.sh ${chimere_tmp}/domains 
\cp ${domaindir}/*.awk ${chimere_tmp}/domains

if [ ! -e ${domaindir}/VCOORD ] ; then 
  echo " o create directory: 
  "mkdir ${domaindir}/VCOORD 
fi 
cd ${chimere_tmp}/domains 
echo "now in: " ${chimere_tmp}/domains . ${chimere_tmp}/domains/define_geom.sh

# HORIZONTAL 
# check existenz of COORD file
export coordfile=${domaindir}/HCOORD/COORD_${dom}

if [ -e $coordfile ] ; then 
  echo " o horizontal domain file already exists"
  echo " >>> " $coordfile 
  echo " >>> continue the input data preprocessing...
else 
  echo " o create file "$coordfile
  \cp ${domaindir}/makeCOORDdomains.sh ${chimere_tmp}/domains 
  \cp ${domaindir}/domainlist.nml ${chimere_tmp}/domains 
  . ${chimere_tmp}/domains/makeCOORDdomains.sh ${dom}
  \cp COORD_${dom} ${domaindir}/HCOORD 
  echo " *** NEW coordinates file: "$coordfile
fi 

# check needs of deep convection parameterization or not (only if Dx > 0.25 degrees)
if [ $ideepconv -eq 1 ] ; then
  nxx="gawk 'BEGIN{stop=0}{x[FNR]=$1}END{for(k=2;k<=FNR;k++) {if(x[k-1]>x[k]&&stop==0){print k-1;stop=1}}}' ${coordfile}"
  nyy="expr ${ntotal} / ${nxx}"
  export ideepconv="gawk -f coord-resol.awk -v nxx=${nxx} -v nyy=${nyy} -v coordfile=${coordfile} ${coordfile}"
  echo " o deep convection parameterization automatically selected"
  if [ $ideepconv -eq 0 ] ; then 
    echo " o deep convection parameterization automatically not selected 
fi 
if [ $ideepconv -eq 0 ] ; then 
  echo " o deep convection parameterization not used 
fi 

# making corresponding landuse files
if [ -s ${domaindir}/${dom}/LANDUSE_${dom} ] ; then 
  echo " o Landuse file is already generated"
  echo " >>> continue the input data preprocessing...
else 
  echo " o create Landuse files..."
  echo " !!!may take more than 10 min... please wait...."
  \cp ${domaindir}/prep_glcf.sh ${chimere_tmp}/domains
```
\cp $(domaindir)/prep_landuse.sh $(chimere_tmp)/domains
\cp $(coordfile) $(chimere_tmp)/domains
\cp $(domaindir)/src/* $(chimere_tmp)/domains

cd $(chimere_tmp)/domains
echo "1. Extract raw data from GLCF global data base"
   $(chimere_tmp)/domains/prep_glcf.sh $(dom) $(coordfile) $(bigfilesdir)
   ln -sf LAND_GLCF.bin $(chimere_tmp)/domains/LAND_GLCF_${dom}.bin
echo "2. Format landuse data for chimere"
   $(chimere_tmp)/domains/prep_landuse.sh $(dom) $(coordfile)

   if [-e $(chimere_tmp)/domains/end.landuse ]; then
      \mv $(chimere_tmp)/domains/$${dom} $(domaindir)
   else
      echo "Preprocessing for LANDUSE failed. Check files in $(chimere_tmp)/domains"
      exit 0
   fi

fi

# making MEGAN input
if [-s $(domaindir)/$${dom}/EFMAP_LAI_${dom}.nc ]; then
   echo " o Biogenic MEGAN files are already generated"
echo "   \!\!\!may take more than 180 min... please wait...."
   \cp $(domaindir)/prep_megan.sh $(chimere_tmp)/domains
   \cp $(coordfile) $(chimere_tmp)/domains
   \cp $(domaindir)/src/* $(chimere_tmp)/domains
cd $(chimere_tmp)/domains
   . $(chimere_tmp)/domains/prep_megan.sh

   if [-e $(chimere_tmp)/domains/end.megan ]; then
      \mv $(chimere_tmp)/domains/$${dom}/EFMAP_LAI_${dom}.nc $(domaindir)/$${dom}
   else
      echo "Preprocessing for MEGAN failed. Check files in $(chimere_tmp)/domains"
      exit 0
   fi

fi

**B.2 The domains/makeCOORDdomains script for horizontal grid definition**

To create a new COORD file, you first have to add a new line in the domainlist.nml file describing your new domain. After that, run manually the **makeCOORDdomains** script (in the domains/ directory or here below) with the required domain name:

```
makeCOORDdomains CONT3
```

with the following **makeCOORDdomains** script:

```
#!/bin/sh

if [ $# -eq 0 ]; then
   echo "define the domain name"
   exit 0
else
   dom=$1
fi
```
B.3 The `util/define_geom` script for vertical grid definition

To make your own VCOORD file, go to the directory `util/`, edit the `define_geom` script and adapt the top pressure level to the one you need.

N.B.: It is recommended to have a maximum ptop of 200 hPa since the global model used for boundary conditions delivered data only up to this value.

The `define_geom` script:

```bash
#!/bin/sh

# Defines a geometrical-progression vertical layering
# For the CHIMERE model. It actually produces sets of
# Hybrid sigma-p coefficients Ak and Bk such that the
# pressure at top of layer k is Pk = Ak*1e5 + Bk*Ps in Pa
# where Ps is Surface pressure

ptop=700 # Pressure at model top
nlay=8 # Number of layers
prel=995 # Pressure (hPa) of the first layer top when Ps = 1000 hPa
fout=../data/vgrid/VCOORD.8GEOM700

echo ${nlay} ${prel} ${ptop} | gawk -f geom.awk > ${fout}
```
Recommendations

- When defining manually the hybrid coefficients the user shall verify that under all possible meteorological conditions these layers do not cross one another or get mixed up.
- It is highly recommended that top level be a pressure level ($b_k = 0$), since it has been found to minimize mass loss through top boundary.
- Using the MOZART and LMDz-INCA boundary conditions, one should not use a model top above 200 hPa.
- In the present formulation for cloud/radiation/photolysis, it is assumed that the model domain is below the clouds, i.e. no cloud albedo is taken into account. This puts constraints on model top to be not too high. This will be changed in future releases.

B.4 The chemprep/chemprep.univ.sh for chemistry preprocessing

```bash
#!/bin/bash
#
# This script allows to prepare a simulation case in terms of species used, # specified chemical schemes, number of aerosol bins, bin sizes, boundary conditions.
#
cut1=2.5
cut2=10.
#
# dmin and dmax are used as tuner parameter to adjust the aerosol # distribution (they are not the min and max cut off diameter !!)
dmin=0.001
if [ ${nbins} -le 12 ] ; then
dmax=12.
else
dmax=40.
fi
rm -rf ${chemprepfic}
mkdir -p ${chemprepfic}
if [ ! -s chemprep-distrib.e ] ; then
  echo "*** ERROR: compile programs before use by typing make in chemprep directory"
  rm -rf ${chemprepfic}
  exit
fi
if [ ${mecachim} = 0 -a ${aero} = 0 -a ${seasalt} = 0 -a ${pops} = 0 -a ${dustout} = 0 -a ${carb} = 0 -a ${trc} = 0 -a ${dust} = 0 ] ; then
  echo "*** ERROR : Inconsistent..."
  echo "All chemical parameters set to zero..."
  echo "Nothing to do !!"
  rm -rf ${chemprepfic}
  exit
fi
if [ ${mecachim} = 0 -a ${aero} = 1 ] ; then
  echo "*** ERROR : Inconsistent..."
  echo "If you use _aero_=1 ---> _mecachim_>0"
  rm -rf ${chemprepfic}
  exit
fi
if [ ${mecachim} = 0 -a ${soatyp} != 0 ] ; then
  echo "*** ERROR : Inconsistent..."
  echo "If you use _mecachim_=0 ---> _soatyp_=0"
  rm -rf ${chemprepfic}
  exit
fi
```
if [ ${seasalt} = 2 -a ${aero} = 0 ] ; then
  echo "*** ERROR : Inconsistent...
  echo "If you use _aero_=0 ---> _seasalt_=0 or 1"
  \rm -rf ${chemprepfic}
  exit
fi

if [ ${nbins} = 0 ] ; then
  if [ ${aero} = 1 -o ${seasalt} -gt 0 -o ${dust} = 1 -o ${dustout} = 1 -o ${pops} = 1 -o ${carb} = 1 ] ; then
    echo "*** ERROR : Inconsistent...
    echo "_nbins_ must be > 1 !! 
    \rm -rf ${chemprepfic}
    exit
  fi
fi

if [ ${mecachim} = "0" ] ; then
  meca=none
  fi
if [ ${mecachim} = "1" ] ; then
  meca=melchior1
fi
if [ ${mecachim} = "2" ] ; then
  meca=melchior2
fi

if [ ${soatyp} = "1" ] ; then
  soa="spl"
fi
if [ ${soatyp} = "2" ] ; then
  soa="med"
fi
if [ ${soatyp} = "3" ] ; then
  soa="cpx"
fi

# Check dependencies
if [ ${aero} = 1 -a ${soatyp} = 0 ] ; then
  echo "Inconsistent...
  echo "In that case, _soatyp_ must be 1 or 2"
  \rm -rf ${chemprepfic}
  exit
fi

# Check dependencies
if [ ${pops} = 1 ] ; then
  if [ ${aero} = 0 -o ${mecachim} = 0 ] ; then
    echo "Inconsistent...
    echo "In that case, _pops_ must be 0"
    \rm -rf ${chemprepfic}
    exit
  fi
fi

# rm -f .trc*
if [ ${trc} -ge 1 ] ; then
  touch .trc_REACTIONS
  touch .trc_OUTPUT_SPECIES
  datei='date -d "$di $hours" +%Y%m%d%H'
  datef='date -d "$di $hours $nhours hours" +%Y%m%d%H'
  sed -e s,_DATEI_,${datei}, \
    -e s,_DATEF_,${datef}, \
$(chempremdir)/data/TRACER.default.sed > TRACER.default

gawk '{if(FNR==1){print $0};if(FNR==2){for(i=1;i<="$trc";i++){print "TRC"i,$0}}}' TRACER.default > ${chemprepfic}/TRACER

gawk '{print $0}END{for(j=1;j<"${nhours}";j++) {printf("%s ", 1)}}' ${chemprepdir}/data/TRACER_PROFILE.default > ${chemprepfic}/TRACER_PROFILE

tracer_list='gawk '{if(FNR>1){print $1}}' ${chemprepfic}/TRACER'

gawk '{if(FNR>1){print $1}}' ${chemprepfic}/TRACER > .trc_ANTHROPIC

gawk '{if(FNR>1){print $1, 100., 0.01, 0.1, 2000}}' ${chemprepfic}/TRACER > .trc_DEPO_SPEC

gawk '{if(FNR>1){print $1, 2, 1., 1.51e-2, 1.}}' ${chemprepfic}/TRACER > .trc_WETD_SPEC

echo "$trc" "TRCF" "$tracer_list" > .trc_FAMILIES

for trc_spec in ${tracer_list}
  do
      echo $trc_spec"->X k=1e-18" > .tmp2
      cat .tmp2 .trc_REACTIONS > .tmp3
      mv .tmp3 .trc_REACTIONS
  done

for trc_spec in ${tracer_list}
  do
      echo $trc_spec" 0." > .tmp2
      cat .tmp2 .trc_OUTPUT_SPECIES > .tmp3
      mv .tmp3 .trc_OUTPUT_SPECIES
  done

echo "TRCF 0." > .tmp2
  cat .tmp2 .trc_OUTPUT_SPECIES > .tmp3
  mv .tmp3 .trc_OUTPUT_SPECIES

fi

echo "Main options"
s
eph "$gas phase" : "$mecachim"
eph "$aerosols" : "$aero"
eph "$sea salt" : "$seasalt"
eph "$carbon" : "$carb"
eph "$pops" : "$pops"
eph "$dust" : "$dust"
eph "$dust...out!" : "$dustout"
eph "$tracer" : "$trc"

# create families at the end of REACTIONS
# and separate GASP / AERO models
ln -sf ${bigfilesdir}/AEROMIN.bin ${chemprepfic}/AEROMIN.bin
ln -sf ${bigfilesdir}/AEROORG.bin ${chemprepfic}/AEROORG.bin

# make FAMILIES and AEROSOL

touch AEROSOL

touch MAKEPRI

touch ANTHROPIC

if [ $(mecachim) -gt 0 ]; then
touch ANTHROPIC
     mv ANTHROPIC tmp3
     cat tmp3 $(chempremdir)/data/ANTHROPIC.univ >> ANTHROPIC
fi

if [ $(aero) -eq 1 ]; then
     mv ANTHROPIC tmp3
     cat tmp3 $(chempremdir)/data/ANTHROPIC.aero >> ANTHROPIC
fi

if [ $(carb) = "0" ]; then
     mv ANTHROPIC tmp3
     cat tmp3 $(chempremdir)/data/ANTHROPIC.ppm >> ANTHROPIC
fi

if [ $(carb) = "1" ]; then
     mv ANTHROPIC tmp3

cat tmp3 $(chemprepdir)/data/ANTHROPIC.carb >> ANTHROPIC
fi
if [ ${soatyp} -gt 0 ]; then
  mv ANTHROPIC tmp3
  cat tmp3 $(chemprepdir)/data/ANTHROPIC.med >> ANTHROPIC
fi
if [ ${pops} -eq 1 ]; then
  mv ANTHROPIC tmp3
  cat tmp3 $(chemprepdir)/data/ANTHROPIC.pops >> ANTHROPIC
fi
if [ ${trc} -ge 1 ]; then
  mv ANTHROPIC tmp3
  cat tmp3 .trc_ANTHROPIC >> ANTHROPIC
fi
touch REACTIONS
#
make REACTIONS unique file
#rm REACTIONS
if [ ${trc} -ge 1 ]; then
  cat .trc_REACTIONS >> REACTIONS
fi
if [ ${mecachim} -gt 0 ]; then
  cat $(chemprepdir)/data/REACTIONS.univ.${meca} >> REACTIONS
  if [ ${aero} = "0" ]; then
    cat $(chemprepdir)/data/REACTIONS.univ.other.${meca} >> REACTIONS
  fi
fi
if [ ${aero} = "1" ]; then
  mv AEROSOL tmp
  mv MAKEPRI tmp2
  cat tmp2 $(chemprepdir)/data/MAKEPRI.aero >> MAKEPRI
  cat $(chemprepdir)/data/REACTIONS.univ.aero >> REACTIONS
  cat $(chemprepdir)/data/REACTIONS.univ.${meca}.soa-${soa} >> REACTIONS
  cat tmp $(chemprepdir)/data/AEROSOL.aero.${soa} >> AEROSOL
fi
if [ ${seasalt} = "1" ]; then
  cat $(chemprepdir)/data/REACTIONS.univ.isal >> REACTIONS
  mv AEROSOL tmp
  mv MAKEPRI tmp2
  cat tmp $(chemprepdir)/data/AEROSOL.isal > AEROSOL
  mv MAKEPRI tmp2
  cat tmp2 $(chemprepdir)/data/MAKEPRI.isal >> MAKEPRI
fi
if [ ${dust} = "1" ]; then
  cat $(chemprepdir)/data/REACTIONS.univ.dust >> REACTIONS
  mv AEROSOL tmp
  mv MAKEPRI tmp2
  cat tmp $(chemprepdir)/data/AEROSOL.dust > AEROSOL
  mv MAKEPRI tmp2
  cat tmp2 $(chemprepdir)/data/MAKEPRI.dust >> MAKEPRI
fi
if [ ${seasalt} = "2" ]; then
  cat $(chemprepdir)/data/REACTIONS.univ.nacl >> REACTIONS
  mv AEROSOL tmp
  mv MAKEPRI tmp2
  cat tmp $(chemprepdir)/data/AEROSOL.nacl > AEROSOL
  mv MAKEPRI tmp2
  cat tmp2 $(chemprepdir)/data/MAKEPRI.nacl >> MAKEPRI
fi
if [ ${pops} = "1" ]; then
  cat $(chemprepdir)/data/REACTIONS.univ.pops >> REACTIONS
  mv AEROSOL tmp
  cat tmp $(chemprepdir)/data/AEROSOL.pops > AEROSOL
mv MAKEPRI tmp2
cat tmp2 ${chemprepdir}/data/MAKEPRI.pops >> MAKEPRI
fi
if [ ${dustout} = "1" ] ; then
cat ${chemprepdir}/data/REACTIONS.univ.dustout >> REACTIONS
mv AEROSOL tmp
cat tmp ${chemprepdir}/data/AEROSOL.dustout > AEROSOL
mv MAKEPRI tmp2
cat tmp2 ${chemprepdir}/data/MAKEPRI.dustout >> MAKEPRI
fi
if [ ${carb} = "1" ] ; then
cat ${chemprepdir}/data/REACTIONS.univ.carb >> REACTIONS
mv AEROSOL tmp
cat tmp ${chemprepdir}/data/AEROSOL.carb > AEROSOL
mv MAKEPRI tmp2
cat tmp2 ${chemprepdir}/data/MAKEPRI.carb >> MAKEPRI
fi

touch FAMILIES.auto
touch ACTIVE_SPECIES.aero
if [ ${aero} = "1" -o ${seasalt} -gt 0 -o ${dustout} = "1" -o ${carb} = "1" -o ${dust} = "1" ] ; then
  echo "======= Building aerosols size distributions"
  echo $nbins $cut1 $cut2 $dmin $dmax > tmpdistrib
  ./chemprep-distrib.e
  sort -g -k 2,2 AEROSOL -o tmp
  echo "# Number of sections D(2.5) D(10.0)" > aerobin.univ
  cat .fambound >> aerobin.univ
  \rm .fambound
  echo "# Name of aerosol components, properties (no blank line at the end!)
  >> aerobin.univ
  echo "NAME TYPE Molar mass Origin [0=bio,1=anth]" >> aerobin.univ
  cat aerobin.univ tmp > AEROSOL
  \rm aerobin.univ
  echo "---------------------------------------------"
  echo "AEROSOL file for this simulation:"
  echo "---------------------------------------------"
  cat AEROSOL
  ./chemprep-families.e
  mv PRIMARY ${chemprepfic}/PRIMARY
  mv AEROSOL ${chemprepfic}/AEROSOL
  mv MAKEPRI ${chemprepfic}/MAKEPRI
else
  \rm -rf PRIMARY AEROSOL MAKEPRI
fi

\rm tmp
touch tmp
if [ ${mecachim} -gt 0 ] ; then
  \cp ${chemprepdir}/data/PHOTO_PARAMETERS.$(meca) ${chemprepfic}/PHOTO_PARAMETERS
  cat ${chemprepdir}/data/FAMILIES.gasp.$(meca) > tmp
fi
\cp ${chemprepdir}/data/BIOGENIC.univ ${chemprepfic}/BIOGENIC
\mv ANTHROPIC ${chemprepfic}/ANTHROPIC
cat tmp FAMILIES.auto ${chemprepdir}/data/FAMILIES.users > FAMILIES
sed -e s/=/" /g FAMILIES > .tmp
sed -e s/+/" /g .tmp > .tmp1
gawk '{print NF-1,$0}' .tmp1 > FAMILIES
if [ ${trc} -ge 1 ]; then
cat FAMILIES .trc_FAMILIES > tmp
mv tmp FAMILIES
fi
mv FAMILIES ${chemprepfic}/FAMILIES
sed -e s/=/" / FAMILIES.auto | gawk '{print $1,"100."}' > .OUTPUT_FAMILIES
rm FAMILIES.auto

#echo "=====================================
#echo "FAMILIES file for this simulation:"
#echo "-------------------------------------"
#cat ${chemprepfic}/FAMILIES
#echo "=====================================
#echo "Preparation of chemistry CHIMERE files:"
#echo "-------------------------------------"
#rm .species
touch ${chemprepfic}/data/PHOTO_PARAMETERS.$(meca)
cp ${chemprepfic}/data/PHOTO_PARAMETERS.$(meca) PHOTO_PARAMETERS
gawk -f chemprep.awk PHOTO_PARAMETERS REACTIONS
rm PHOTO_PARAMETERS

sort .species | uniq | gawk '{print $0}' > ACTIVE_SPECIES.gas
# full active species file
cat ACTIVE_SPECIES.gas > tmp
sort tmp | uniq | gawk '{print NR,$0}' > ACTIVE_SPECIES.gas
cat ACTIVE_SPECIES.aero > tmp
ngasspec=`wc -l ACTIVE_SPECIES.gas | gawk '{print $1}'`
gawk '{print $1,"0."}' ACTIVE_SPECIES.aero > .OUTPUT_SPECIES.aero
gawk '{print NR","$ngasspec","$0}' tmp > ACTIVE_SPECIES.aero
cat ACTIVE_SPECIES.gas ACTIVE_SPECIES.aero > .ACTIVE_SPECIES
# Put default value (=1) for transport scheme
gawk '{print $0,"$iadv"}' .ACTIVE_SPECIES > ${chemprepfic}/ACTIVE_SPECIES
rm ACTIVE_SPECIES.gas ACTIVE_SPECIES.aero .ACTIVE_SPECIES

cp ${chemprepfic}/data/DEPO_SPEC.univ DEPO_SPEC
cp ${chemprepfic}/data/WETD_SPEC.univ WETD_SPEC
if [ ${trc} -ge 1 ]; then
cat DEPO_SPEC .trc_DEPO_SPEC > .tmpl1
mv .tmpl1 DEPO_SPEC
fi
mv DEPO_SPEC ${chemprepfic}/DEPO_SPEC
mv DEPO_SPEC ${chemprepfic}/DEPO_SPEC

#echo "-------------------------------------"
#echo "Other parameters: vegetation, deposition etc."

cp ${chemprepfic}/data/VEGETATION.univ ${chemprepfic}/VEGETATION
cp ${chemprepfic}/data/SEA.univ ${chemprepfic}/SEA
cp ${chemprepfic}/data/DEPO_PARS.univ ${chemprepfic}/DEPO_PARS
cp ${chemprepfic}/data/SEMIVOL.univ ${chemprepfic}/SEMIVOL
cp ${chemprepfic}/data/SOIL.univ ${chemprepfic}/SOIL

cp ${chemprepfic}/data/OUTPUT_PARAMETERS ${chemprepfic}/OUTPUT_PARAMETERS

cp ${chemprepfic}/data/LANDPAR ${chemprepfic}/LANDPAR
#echo "-------------------------------------"
#echo "Boundary conditions parameters"
#echo " Building GOCART parameters file"
for boundmodel in GOCART LMDZAERO
do
touch ${boundmodel}_builder
if [ ${aero} = "1" ] ; then
  mv ${boundmodel}_builder tmp3
cat tmp3 ${chempreddir}/boundaries_spec/${boundmodel}_builder.aero.${soa} >> ${boundmodel}_builder
fi
if [ ${carb} = "0" ] ; then
  mv ${boundmodel}_builder tmp3
cat tmp3 ${chempreddir}/boundaries_spec/${boundmodel}_builder.ppmcarb >> ${boundmodel}_builder
fi
if [ ${seasalt} = "0" ] ; then
  mv ${boundmodel}_builder tmp3
cat tmp3 ${chempreddir}/boundaries_spec/${boundmodel}_builder.aero.so4 >> ${boundmodel}_builder
fi
if [ ${seasalt} = "2" ] ; then
  mv ${boundmodel}_builder tmp3
cat tmp3 ${chempreddir}/boundaries_spec/${boundmodel}_builder.nacl >> ${boundmodel}_builder
fi
if [ ${dustout} = "0" ] ; then
  if [ ${dust} = "1" ] ; then
    mv ${boundmodel}_builder tmp3
cat tmp3 ${chempreddir}/boundaries_spec/${boundmodel}_builder.dust >> ${boundmodel}_builder
  fi
fi
if [ ${seasalt} = "1" ] ; then
  if [ ${aero} = "1" ] ; then
    mv ${boundmodel}_builder tmp3
cat tmp3 ${chempreddir}/boundaries_spec/${boundmodel}_builder.aero.isal >> ${boundmodel}_builder
  fi
  if [ ${aero} = "0" ] ; then
    mv ${boundmodel}_builder tmp3
cat tmp3 ${chempreddir}/boundaries_spec/${boundmodel}_builder.isal >> ${boundmodel}_builder
  fi
fi
if [ ${dustout} = "1" ] ; then
  mv ${boundmodel}_builder tmp3
cat tmp3 ${chempreddir}/boundaries_spec/${boundmodel}_builder.aero.dustout >> ${boundmodel}_builder
fi
if [ ${carb} = "1" ] ; then
  mv ${boundmodel}_builder tmp3
cat tmp3 ${chempreddir}/boundaries_spec/${boundmodel}_builder.carb >> ${boundmodel}_builder
fi
if [ ${dust} = "1" -a ${aero} = "0" -a ${dustout} = "0" ] ; then
  mv ${boundmodel}_builder tmp3
cat tmp3 ${chempreddir}/boundaries_spec/${boundmodel}_builder.dust >> ${boundmodel}_builder
fi

mv ${boundmodel}_builder ${chemprepfic}/${boundmodel}_builder
done
if [ ${mecachim} != "0" ] ; then
  # echo " Building MOZART parameters file"
  \cp ${chempreddir}/boundaries_spec/MOZART2_SPEC.${meca} ${chemprepfic}/MOZART2_SPEC
  # echo " Building LMDZINCA parameters file"
  \cp ${chempreddir}/boundaries_spec/LMDZINCA2_SPEC.${meca} ${chemprepfic}/LMDZINCA2_SPEC
fi
#echo "----------------------------------------"
#echo "Output species list"
\rm tmp2 tmp1
touch tmpl
touch tmp2
\rm tmp
if [ ${mecachim} != "0" ] ; then
  mv tmp1 tmp
  cat tmp $(chempremdir)/output_species/OUTPUT_SPECIES.gas.low > tmp1
  mv tmp2 tmp
  cat tmp $(chempremdir)/output_species/OUTPUT_SPECIES.gas.full > tmp2
fi

# low resolution output file
cp tmp1 OUTPUT_SPECIES.low
cp tmp2 OUTPUT_SPECIES.full
if [ ${aero} = "1" ] ; then
  mv tmp2 tmp
  cat tmp $(chempremdir)/output_species/OUTPUT_SPECIES.aero.$(soa) > tmp2
fi
if [ ${seasalt} = "1" ] ; then
  mv tmp2 tmp
  cat tmp $(chempremdir)/output_species/OUTPUT_SPECIES.isal > tmp2
fi
if [ ${seasalt} = "2" ] ; then
  mv tmp2 tmp
  cat tmp $(chempremdir)/output_species/OUTPUT_SPECIES.nacl > tmp2
fi
if [ ${dust} = "1" ] ; then
  mv tmp2 tmp
  cat tmp $(chempremdir)/output_species/OUTPUT_SPECIES.dust > tmp2
fi
if [ ${carb} = "1" ] ; then
  mv tmp2 tmp
  cat tmp $(chempremdir)/output_species/OUTPUT_SPECIES.carb > tmp2
fi
if [ ${pops} = "1" ] ; then
  mv tmp2 tmp
  cat tmp $(chempremdir)/output_species/OUTPUT_SPECIES.pops > tmp2
fi
if [ ${dustout} = "1" ] ; then
  mv tmp2 tmp
  cat tmp $(chempremdir)/output_species/OUTPUT_SPECIES.dustout > tmp2
fi
mv tmp2 tmp
  cat tmp .OUTPUT_SPECIES.aero > tmp2

if [ ${trc} -ge 1 ] ; then
  mv OUTPUT_SPECIES.full tmp1
  cat tmp1 .trc_OUTPUT_SPECIES > OUTPUT_SPECIES.full
  mv OUTPUT_SPECIES.low tmp2
  cat tmp2 .trc_OUTPUT_SPECIES > OUTPUT_SPECIES.low
fi

# families are always included in the output species file
mv OUTPUT_SPECIES.full tmp1
  cat tmp1 .OUTPUT_FAMILIES > OUTPUT_SPECIES.full
mv OUTPUT_SPECIES.low tmp2
  cat tmp2 .OUTPUT_FAMILIES > OUTPUT_SPECIES.low
mv OUTPUT_SPECIES.low ${chemprepfic}
mv OUTPUT_SPECIES.full ${chemprepfic}
touch $(chemprepfic)/STOICHIOMETRY

mv CHEMISTRY $(chemprepfic)/CHEMISTRY
mv REACTIONS $(chemprepfic)/REACTIONS
mv STOICHIOMETRY $(chemprepfic)/STOICHIOMETRY
mv REACTION_RATES $(chemprepfic)/REACTION_RATES
rm tmp* .tmp*

B.5 The ./install.sh for Chimere download and installation

#!/bin/bash

# CHIMERE installation script
# Run this script from the directory where you want Chimere to be installed
# Usage: ./install.sh [bigfiles_root_dir]

# Chimere version
version=2008b
# EmiSURF name
emis=emi2008b
# Test case archive file names
tstemis=emi2003forHW
tstmeteo=MMOUT_EUR2_20030730_20030803
# Boundary conditions file names
bc_inca=INCA-200501.tar.gz
bc_goc=GOCART-200410.tar.gz
bc_aero=LMDZAERO.tar.bz2
# GLCLF archive
glcfn=data_glcf
# AEROxx.bin files
bigf=bigfiles
# MEGAN data
megan=MEGAN
#-----------------------
# Archive types
#-----------------------
verf=chimere${version}
emisf=${emis}.tar.bz2
modf=${verf}.tgz
tstmeteof=${tstmeteo}.gz
tstemisf=${tstemis}.tar.gz
GLCF=${glcf}.tar.bz2
bigf=${bigf}.tgz
meganf=${megan}.tar.bz2
#-----------------------
gawk --version 2>/dev/null >/dev/null || 
    { echo "You need gawk to install CHIMERE. Bye!"; exit 1; }

doinst=0
domkfile=1
dir='pwd'

echo "-------------------------------------------------------------"
echo " CHIMERE v.$version and emiSURF v.$emis Installation    "
echo "-------------------------------------------------------------"
echo "This script can be used to install Chimere and/or to generate Makefile headers" 
echo -n "Do you want to install Chimere? (y/[n]) : "
read stat
if [ $# -eq 0 ]; then
    echo -n "What is your root directory for big files? ["$dir"]: 
    read bigdir
else
    bigdir=$1
fi

if [ $# -eq 0 ]; then
    echo -n "What is your root directory for big files? ["$dir"]: 
    read bigdir
else
    bigdir=$1
fi

[ [ "${stat}" = "y" || "${stat}" = "Y" ] ] && doinst=1

if [ ${doinst} -eq 1 ]; then
    if [ $# -eq 0 ]; then
        echo -n "What is your root directory for big files? ["$dir"]: 
        read bigdir
    else
        bigdir=$1
    fi

    [ [ $# -eq 0 ] ] && bigdir=${dir}

    echo "Your root big files directory is "$bigdir"

    # Check if we can write to the bigfiles dir
    mkdir -p "$bigdir" 2>/dev/null
    [ [ $? -eq 0 && -w "$bigdir" ] ] || { echo "The "$bigdir" directory is not writable. Bye!" ; exit 1 ; }

    mkdir -p "$bigdir"/METEO
    mkdir -p "$bigdir"/EMISSIONS
    mkdir -p "$bigdir"/OUTPUTS
    mkdir -p "$bigdir"/TEST/EMIS
    mkdir -p "$bigdir"/TEST/METEO
    mkdir -p "$bigdir"/ARCH

    downl=0 # whether to download data

    if [ [ [ -f "$modf" || -f "$tstemisf" || -f "$tstmeteof" || -f "$emisf" || -f "$bc_inca" || -f "$bc_goc" || -f "$bc_aero" || -f "$glcf" ] ] ]; then
        echo -n "One or more Chimere archives are found in your current directory. Still want to download? (y/[n]): 
        read stat
        [ [ "${stat}" = "y" || "${stat}" = "Y" ] ] && downl=1
    fi

    # 1. Download CHIMERE source code and data
    # 1.0. Chimere sources
        echo "Can't download "$modf". Bye!"; exit 1; }

    # 1.1. bigfiles: AEROMIN.bin and AEROORG.bin
        echo "Can't download "$bigf". Bye!"; exit 1; }

    # 1.2. Data for the Heat Wave test case: emissions
        echo "Can't download "$tstemisf". Bye!"; exit 1; }

    # 1.3. Data for the Heat Wave test case: MM5 meteorology
        echo "Can't download "$tstmeteof". Bye!"; exit 1; }

    # 1.4. Emissions interface: the complete emiSURF preprocessor archive
        echo "Can't download "$emisf". Bye!"; exit 1; }

    # 1.5. Boundary conditions
        echo "Can't download "$bc_inca". Bye!"; exit 1; }

        echo "Can't download "$bc_goc". Bye!"; exit 1; }

        echo "Can't download "$bc_aero". Bye!"; exit 1; }

        echo "Can't download "$glcf". Bye!"; exit 1; }
# 1.6. GLCF land use classes database
[[ ! -f ${glcf} || ${downl} -eq 1 ]]
{ wget http://www.lmd.polytechnique.fr/chimere/downloads/${glcf} ||
  { echo "Can't download ${glcf}. Bye!"; exit 1; } }
# 1.7. MEGAN data
[[ ! -f ${meganf} || ${downl} -eq 1 ]]
{ wget http://www.lmd.polytechnique.fr/chimere/downloads/${meganf} ||
  { echo "Can't download ${meganf}. Bye!"; exit 1; } }
# 2. Uncompress the archives

```
# Uncompressing the archives...
{ echo "Uncompressing the archives..." } ||
tar xzvf ${modf} > tar_uncompress.lst
ls ${bigdir}/*.bin >/dev/null 2>/dev/null || { tar xzvf ${bigf} -C ${bigdir} > tar_bigf.lst ||
  { echo "Can't extract ${bigf} to ${bigdir}. Bye!"; exit 1; } }
ls ${bigdir}/TEST/EMIS/*.nc >/dev/null 2>/dev/null || { tar xzvf ${tstemisf} -C ${bigdir}/TEST/EMIS > tar_emisf.lst ||
  { echo "Can't extract ${tstemisf} to ${bigdir}/TEST/EMIS. Bye!"; exit 1; } }
ls ${bigdir}/INCA/INCA* >/dev/null 2>/dev/null || { tar xzvf ${bc_inca} -C ${bigdir} > tar_bicf.lst ||
  { echo "Can't untar ${bc_inca} to ${bigdir}. Bye!"; exit 1; } }
ls ${bigdir}/GOCART/GOCART* >/dev/null 2>/dev/null || { tar xzvf ${bc_goc} -C ${bigdir} > tar_bicg.lst ||
  { echo "Can't untar ${bc_goc} to ${bigdir}. Bye!"; exit 1; } }
ls ${bigdir}/LMDZAERO/INCA-AEROSOL-clim* >/dev/null 2>/dev/null || { tar xzvf ${bc_aero} -C ${bigdir} > tar_bica.lst ||
  { echo "Can't untar ${bc_aero} to ${bigdir}. Bye!"; exit 1; } }
```
# 4. Configure the Makefiles

if [ ${domkfile} -eq 1 ]; then 

cd ${verf} 2>/dev/null || { echo "Can’t find Chimere source directory ${verf}. Bye!" ; exit 1; }

# 4.1. Try to determine whether the system is 32 or 64 bit
syst='arch 2>/dev/null' ||
{ echo "arch command is not available. Is your system 64-bit? (y/[n])" ;
read stat ;
[[ "${stat}" = "y" || "${stat}" = "Y" ]] && syst="64"; }
suf="" 
echo $syst | grep "64" >/dev/null && suf="-64"

# 4.2. Choose Fortran compiler
let i=0
echo "Available Fortran compilers:"
for cmp in g95 ifort pgf90 ; do
cmpp='which ${cmp} 2>/dev/null'
if [ ${#cmpp} -ne 0 ]; then
CMP[$i]=$cmpp
    echo "[$i] ${CMP[$i]}"
(( i++ ))
fi
done
[ $i -eq 0 ] && { echo "You need to have a Fortran compiler installed!"; exit 1; }
let n=0
if [ $i -gt 1 ]; then
    echo -n "Choose your compiler, please: 
    read n
    [ ${#CMP[$n]} -eq 0 ] && { echo "Wrong choice!" ; exit 1 ; }
fi
REALFC=${CMP[$n]}

# 4.3. Netcdf library
echo "-----------------------------"
netcdflst='locate libnetcdf.a'
[ ${#netcdflst} -eq 0 ] && { echo "You need to install a netcdf library! Bye." ; exit 1; }
let i=0
echo "Available NetCDF library versions:"
for f in ${netcdflst}; do
    netcdfa[$i]="dirname $(dirname ${f})"
    echo "[$i] ${netcdfa[$i]}"
(( i++ ))
done
let n=0
if [ $i -gt 1 ]; then
    echo -n "Choose your version, please: 
    read n
    [ ${#netcdfa[$n]} -eq 0 ] && { echo "Wrong choice!" ; exit 1 ; }
fi
NETCDFLIB=${netcdfa[$n]}/lib
NETCDFINC=${netcdfa[$n]}/include

# 4.4. MPI
MF77='which mpif77 2>/dev/null' || { echo "You need to have MPI installed!"; exit 1; }
which lamboot 2>/dev/null >/dev/null && suflam="-lam"

# 4.5. Resume
echo "----------------------------"
echo "Your chosen options:"
echo "NETCDFLIB = "${NETCDFLIB}"" 
echo "NETCDFINC = "${NETCDFINC}"" 
echo "REALFC = "${REALFC}"" 
echo "MF77 = "${MF77}""
echo -n "Your system is 
if [ "$(suf)" = "-64" ]; then 
  echo "64 bit"
else 
  echo "32 bit"
fi
if [ "$(suflam)" = "-lam" ]; then 
  echo "Your MPI version is LAM"
else 
  if which opal_wrapper 2>/dev/null >/dev/null ; then 
    echo "Your MPI version is Open MPI"
  else 
    echo "Your MPI version is neither OpenMPI nor LAM!"
    echo "Attention! You might have to manually adjust your Makefile.hdr"
    echo "for your MPI realization!"
  fi
fi
echo -n "Is it OK? (y/[n]) : 
read stat
[[ "$stat" = "y" || "$stat" = "Y" ]] || exit 0

# 4.6. Generate the new Makefile.hdr
rm -f Makefile.hdr
gawk '{ str = "$0;" if ($1 ~~ "NETCDFLIB" && $2 == "=") sub($3, "'${NETCDFLIB}'", str); if ($1 ~~ "NETCDFINC" && $2 == "=") sub($3, "'${NETCDFINC}'", str); if ($1 == "REALFC" && $2 == "=") sub($3, "'${REALFC}'", str); if ($1 == "MF77" && $2 == "=") sub($3, "'${MF77}'", str); print str; }' makefiles.hdr/Makefile.hdr.'basename $(REALFC)"${suf}"$(suflam) > Makefile.hdr
fi
Appendix C

How To install netCDF under GNU/Linux

C.1 Background

Since version ch2008b CHIMERE requires the netCDF package to work. This note describes how to install netCDF for the supported compilers.

NetCDF installation has been tested with:

- ifort 32 bits (version 8.1.028)
- ifort 64 bits (version 8.1.027)
- g95 32 bits (version 4.0.0 20050129)
- g95 64 bits (version 4.0.0 20050124)
- pgf90 32 bits (version 5.2.4)
- pgf90 64 bits (version pgf90 6.0-5)
- netcdf (version 3.6.4)

C.2 Download

- Download netCDF (source package) from the URL:

  cd /tmp
  wget ftp://ftp.unidata.ucar.edu/pub/netcdf/netcdf.tar.gz

- Decompress and detar

  tar xvfz netcdf.tar.gz

C.3 Configure netCDF

At the time of writing of this How-To, the current version of netCDF is 3.6.4. If you downloaded a newer version, replace 3.6.4 by the new version number in the following text.

  cd netcdf-3.6.4/src

The configuration process depends on several environment variables that you shall set according to your compiler. Although it may be possible to link CHIMERE executables compiled with compiler A with the netCDF library compiled with compiler B, we strongly recommend to use the same compiler. The time you will spend to install a new netCDF library is negligible compared to the time you could loose trying to match inconsistent libraries.
We suppose you want to install netcdf somewhere under /opt. If you choose to install under /usr/local, or under your home directory, replace /opt your installation path in the following text. We also suppose your shell is like bash or ksh. If you use csh, modify the export statements according to your shell syntax.

C.3.1 ifort

As root:

    mkdir /opt/netcdf-3.6.4-ifort

As an ordinary user:

    export FC=ifort
    export F90=ifort
    export FFLAGS="-fltconsistency"
    export CPPFLAGS="-DNDEBUG -DpgiFortran"
    ./configure --prefix=/opt/netcdf-3.6.4-ifort
    make
    make test

As root:

    make install

C.3.2 g95

As root:

    mkdir /opt/netcdf-3.6.4-g95

As an ordinary user:

    export CC='gcc'
    export CXX='g++'
    export CPPFLAGS=''-DNDEBUG -DgFortran'
    export CFLAGS=''-O -g'
    export FC='g95'
    export F77='g95'
    export FFLAGS=''-O -g -fno-second-underscore'
    export FCFLAGS=''-O -g -fno-second-underscore'
    ./configure --prefix=/opt/netcdf-3.6.4-g95
    make
    make test

As root:

    make install
C.3.3 pgf90

Some versions of pgf90 prior to 5.2 have experienced problems with the F90 interface of netCDF. Do not try to install a library for which "make test" has failed.

As root:

```bash
mkdir /opt/netcdf-3.6.4-pgf90
```

As an ordinary user:

```bash
export CC=/usr/bin/gcc
export CPPFLAGS='-DNDEBUG -DpgiFortran'
export CFLAGS=-O
export FC=pgf90
export FFLAGS="-O -w"
export CXX=g++
./configure --prefix=/opt/netcdf-3.6.4-pgf90
make
make test
```

As root:

```bash
make install
```

C.3.4 ifort 64 bit on Intel EMT64 or AMD Opteron

As root:

```bash
mkdir /opt/netcdf-3.6.4-ifort
```

As an ordinary user:

```bash
export FC=ifort
export F90=ifort
export CFLAGS="-O -m64"
export FFLAGS="-mp"
export CPPFLAGS="-DNDEBUG -DpgiFortran"
./configure --prefix=/opt/netcdf-3.6.4-ifort
make
make test
```

As root:

```bash
make install
```

C.3.5 g95 64 bit on Intel EMT64 or AMD Opteron

As root:

```bash
mkdir /opt/netcdf-3.6.4-g95-64
```

As an ordinary user:
export CC='gcc'
export CXX='g++'
export CPPFLAGS=''-DNDEBUG -DgFortran' 
export CFLAGS=''-O -m64' 
export FC='g95'
export F77='g95'
export FFLAGS=''-O -g -fno-second-underscore' 
export FCFLAGS=''-O -g -fno-second-underscore' 
./configure --prefix=/opt/netcdf-3.6.4-g95-64 
make 
make test

As root :

make install

C.3.6 pgf90 64 bit on Intel EMT64 or AMD Opteron

Do not try to install a library for which "make test" has failed.

As root :

    # make install

As root :

    # mkdir /opt/netcdf-3.6.4-pgf90-64

As an ordinary user :

    export CC=/usr/bin/gcc
    export CPPFLAGS=''-DNDEBUG -DpgiFortran' 
    export CFLAGS=''-O 
    export FC=pgf90 
    export F90=pgf90 
    export FFLAGS=''-O -w'' 
    export CXX=g++
    ./configure --prefix=/opt/netcdf-3.6.4-pgf90-64
    make
    make test

As root :

    # make install

C.4 Configure CHIMERE

In the directory makefiles.hdr/, you will find 6 files :

    Makefile.hdr.ifort
    Makefile.hdr.ifort-64
    Makefile.hdr.g95
    Makefile.hdr.g95-64
    Makefile.hdr.pgf90
    Makefile.hdr.pgf90-64
In the file corresponding to your compiler, modify the line `NETCDF =` to point to your netCDF directory. For instance:

```
NETCDF = /opt/netcdf-3.6.4-g95
```

Follow the comments in `Makefile.hdr.xxxx` for the other options. Do not forget to link your compiler-specific `Makefile.hdr.xxxx` to `Makefile.hdr`. For instance:

```
ln -sf makefiles.hdr/Makefile.hdr.ifort Makefile.hdr
```
Appendix D

How To install MPI under GNU/Linux

D.1 Background

CHIMERE is now parallelized, using the distributed memory scheme, and requires an MPI compatible Message Passing Library to be installed and configured on the host or cluster it will run on.  
*MPI is required even if CHIMERE is planned to run on a single processor machine!*

Several free MPI libraries are available for download. MPICH and LAM/MPI are well known. The newer Open-MPI is claimed to be the continuation of LAM/MPI.

CHIMERE has been tested with LAM/MPI and Open MPI. This HOW-TO describes briefly the installation and basic configuration process for LAM/MPI and Open MPI on a RedHat-like GNU-Linux distribution. We recommend the precompiled package installation for RedHat-like distributions. However, some 64 bit Linux distributions have a precompiled LAM-MPI binary which is not compatible with other libraries. In that case, you will have to install the LAM-MPI package from source, as described at the end of the document.

Tested with:

- Fedora Core 4
- LAM-7.1.1

D.2 LAM/MPI Installation

RedHat-like distributions include a RPM for LAM/MPI. It is generally called "lam". Here is the installation process for a Fedora Core 4 system. You must have administrative privileges to perform it.

- If you use YUM as a package tool:

  ```
  yum install lam
  ```

- If you use RPM:

  ```
  □ download lam-7.1.1-7.FC4.i386.rpm from your Fedora repository
  □ you may also want libaio-devel-0.3.104-2.i386.rpm
  □ install :

  rpm -Uvh libaio-devel-0.3.104-2.i386.rpm lam-7.1.1-7.FC4.i386.rpm
  ```

- Edit `/etc/lam/lam-bhost.def` to set the list of hosts you will include in your LAM cluster. For instance, for a cluster of 3 bi-processors, lam-bhost.def would contain:
For a uniprocessor machine, lam-bhost.def would only contain:

localhost

- Instead of using the system wide lam-bhost.def, you can create and use your own hosts file, and boot LAM/MPI using this file. A file named “nodelist.example” is provided as a template in this distribution. Edit it according to your environment.

D.3 Testing

- type:
  lamboot

  to boot your LAM/MPI environment using system-wide hosts list, or
  lamboot <my_host_list>

  to boot your own host list defined in file <my_host_list>

- type:
  lamnodes

  to verify that all nodes declared in the hosts file are up.

If cas of insuccess, get LAM/MPI documentation from their web site.

Most problems occur because the user is not allowed to perform rsh or ssh connexions to the hosts listed in hostfile. Check with your system administrator that you can ssh to the target hosts. If the hosts are connected to a private network, ask her/him to allow rsh on these hosts. The advantage of using rsh is that you can schedule unattended jobs using cron or at. Otherwise you need to change your ssh passphrase for an empty one, which can be a security issue.

D.4 Uniprocessor users

Parallelism is mainly useful for Air Pollution Forecast Agencies, or for scientists running long periods of reanalysis. For the users who just want to run CHIMERE as they were accustomated to, on a single processor machine, here is a simple summary of what they shall do with LAM/MPI:

- install LAM/MPI as described above
  yum install lam

- check that ssh is installed on your machine
Most modern Linux distributions come with ssh installed and configured.

- edit /etc/lam/lam-bhost.def to keep only the line
  
  localhost

- boot LAM/MPI
  
  lamboot

- Enjoy CHIMERE!

### D.5 Installation from source

Thanks to Gabriele Curci at aquila.infn.it

Some 64 bit Linux vendors have compiled the lam package with the "second-underscore" option. This leads to an incompatibility between lam and netcdff libraries. If you get error messages like this in your make.log file, you are probably in that case:

```
twostep_mod.f90:(.text+0xbdb6): undefined reference to `mpi_irecv_
  `
  twostep_mod.f90:(.text+0xc2cb): undefined reference to `mpi_issend_
  `
  twostep_mod.f90:(.text+0xc33a): undefined reference to `mpi_irecv_
  `
  twostep_mod.f90:(.text+0xc7f0): undefined reference to `mpi_issend_
```

To overcome this problem, you have to install LAM-MPI from source. Here is the process, copied from Gabriele’s message:

- login as root
- download LAM source, I installed latest version 7.1.2 (wget http://www.lam-mpi.org/download/files/lam-7.1.2.tar.bz2)
- unzip the source (tar jxvf lam-7.1.2.tar.bz2)
- cd lam-7.1.2
- ./configure "FC=g77 -fno-second-underscore" --prefix=/usr/local
  
  make
  make install

### D.6 Open MPI installation

This section briefly explains how to install Open MPI with g95 and ifort compilers from sources. The sources can be downloaded from the Open MPI home page [http://www.open-mpi.org].

**With g95 on a 32 bit system**

```
export COMPILO=g95
./configure CC=gcc CXX=g++ F77=$COMPILO FC=$COMPILO \
  --prefix=/opt/openmpi-1.2.5-$COMPILO \
  --with-mpi-f90-size=medium
```
make all
make install
eredit /etc/profile to set the PATH variable
eredit /etc/ld.so.conf.d/openmpi.conf
ldconfig

**With g95 on a 64 bit system**

export COMPILO=g95
./configure CC=gcc CXX=g++ F77=$COMPILO FC=$COMPILO \  
CFLAGS=-m64 CXXFLAGS=-m64 FFLAGS=-m64 FCFLAGS=-m64 \  
--prefix=/opt/openmpi-1.2.5-$COMPILO \  
--with-mpi-f90-size=medium
make all
make install
eredit /etc/profile to set the PATH variable
eredit /etc/ld.so.conf.d/openmpi.conf
ldconfig

**Build options with ifort for 64 and a 32 bit systems**

For a 64 bit system:

export CC=gcc
export CXX=g++
export CFLAGS='--02 -m64'
export CXXFLAGS='--02 -m64'
export LDFLAGS='--02'
export FC=ifort
export FCFLAGS='--02 -m64'
export F77=ifort
export FFLAGS='--02 -m64'
./configure --prefix=/opt/openmpi-1.2.5-ifort-64 --with-mpi-f90-size=medium

For a 32 bit system you need to remove the -m64 flag from the options above.
Appendix E

Structure of the CHIMERE netCDF files

E.1  EMIS.[domain].[MM].[SPEC].s.nc and EMIS.[domain].[MM].[SPEC].p.nc

```plaintext
netcdf EMIS.CONT5.01.NO2.s {
  dimensions:
  Time = 24 ;
  west_east = 79 ;
  south_north = 47 ;
  bottom_top = 6 ;
  type_day = 3 ;

  variables:
  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 EMEP_levels(bottom_top) ;
  EMEP_levels:units = "meters" ;
  EMEP_levels:long_name = "Cut_off_heights_for_redistribution" ;
  float NO2(Time, type_day, bottom_top, south_north, west_east) ;
  NO2:units = "molecule/cm2/s" ;
  NO2:long_name = "NO2 Emission" ;

  // global attributes:
  :Title = "CHIMERE SUITE" ;
  :Sub-title = "Hourly surfacic emission" ;
  :Chimere_type = "EMISSIONS" ;
  :Generating_process = "Generated by sectoremis" ;
  :Conventions = "None" ;
  :Domain = "CONT5" ;
  :Emission_version = "2008b" ;
  :history = "" ;
}
}
```

```plaintext
netcdf EMIS.CONT5.01.NO2.p {
  dimensions:
  Time = 24 ;
  west_east = 79 ;
  south_north = 47 ;
  Sources = 8 ;
  type_day = 3 ;

  variables:
  float lon(south_north, west_east) ;
```
E.2  exdomout.nc

```c
netcdf exdomout {

dimensions: 
Time = UNLIMITED ; // (13 currently)
DateStrLen = 19 ;
west_east = 67 ;
south_north = 46 ;
bottom_top = 32 ;
variables:
char Times(Time, DateStrLen) ;
float lon(south_north, west_east) ;
lon:units = "degrees_east" ;
lon:long_name = "Longitude" ;
float lat(south_north, west_east) ;
l lat:units = "degrees_north" ;
lat:long_name = "Latitude" ;
long Sources_PIG(Sources) ;
Sources_PI G:units = "flag-0-1" ;
Sources_PI G:long_name = "Pig_treatment" ;
float Sources_lat(Sources) ;
Sources_lat:units = "Degrees" ;
Sources_lat:long_name = "Y_coordinate_of_source_point" ;
long Sources_ilat(Sources) ;
Sources_ilat:units = "cell_number" ;
Sources_ilat:long_name = "Y_coordinate_of_source_point" ;
float Sources_lon(Sources) ;
Sources_lon:units = "Degrees" ;
Sources_lon:long_name = "X_coordinate_of_source_point" ;
long Sourcesilon(Sources) ;
Sourcesilon:units = "cell_number" ;
Sourcesilon:long_name = "X_coordinate_of_source_point" ;
float Sources_Temp(Sources) ;
Sources_Temp:units = "Celcius_degree" ;
Sources_Temp:long_name = "Temperature_of_source_point" ;
float Sources_Height(Sources) ;
Sources_Height:units = "Meter" ;
Sources_Height:long_name = "Height_of_source_point" ;
float Sources_Diameter(Sources) ;
Sources_Diameter:units = "Meter" ;
Sources_Diameter:long_name = "Diameter_of_source_point" ;
float Sources_Velocity(Sources) ;
Sources_Velocity:units = "Meter/s" ;
Sources_Velocity:long_name = "Velocity_of_source_point_emission" ;
float NO2(Time, type_day, Sources) ;
NO2:units = "molecule/cm2/s" ;
NO2:long_name = "NO2 Emission" ;
}
```
lon:long_name = "Longitude";
float lat(south_north, west_east);
lat:units = "degrees_north";
lat:long_name = "Latitude";
float sshf(Time, south_north, west_east);
sshf:units = "W/m^2";
sshf:long_name = "Surface sensible heat flux";
float slhf(Time, south_north, west_east);
slhf:units = "W/m^2";
slhf:long_name = "Surface latent heat flux";
float usta(Time, south_north, west_east);
usta:units = "m/s";
usta:long_name = "Frictional velocity";
float tem2(Time, south_north, west_east);
tem2:units = "K";
tem2:long_name = "2m air temperature";
float soim(Time, south_north, west_east);
soim:units = "m^3/m^3";
soim:long_name = "Soil Moisture level 1";
float hght(Time, south_north, west_east);
hght:units = "m";
hght:long_name = "PBL height from MM5";
float rh2m(Time, south_north, west_east);
rh2m:units = "fraction";
rh2m:long_name = "Relative Humidity at 2m";
float lspc(Time, south_north, west_east);
lspc:units = "kg/m^2";
lspc:long_name = "Large scale precipitation";
float copc(Time, south_north, west_east);
copc:units = "kg/m^2";
copc:long_name = "convective precipitation";
float u10m(Time, south_north, west_east);
u10m:units = "m/s";
u10m:long_name = "10 m U wind";
float v10m(Time, south_north, west_east);
v10m:units = "m/s";
v10m:long_name = "10 m V wind";
float temp(Time, bottom_top, south_north, west_east);
temp:units = "K";
temp:long_name = "Temperature";
float cliq(Time, bottom_top, south_north, west_east);
cliq:units = "kg/kg";
cliq:long_name = "Cloud liquid water mixing ratio";
float rain(Time, bottom_top, south_north, west_east);
rain:units = "kg/kg";
rain:long_name = "Rain water mixing ratio";
float sphu(Time, bottom_top, south_north, west_east);
sphu:units = "kg/kg";
sphu:long_name = "Specific humidity";
float cice(Time, bottom_top, south_north, west_east);
cice:units = "kg/kg";
cice:long_name = "Ice mixing ratio";
float pres(Time, bottom_top, south_north, west_east);
pres:units = "Pa";
pres:long_name = "Pressure";
float alti(Time, bottom_top, south_north, west_east);
alti:units = "m";
alti:long_name = "Altitude of half-sigma level";
float winz(Time, bottom_top, south_north, west_east);
winz:units = "m/s";
winz:long_name = "Zonal wind";
float winm(Time, bottom_top, south_north, west_east);
winm:units = "m/s";
winm:long_name = "Meridional wind"

// global attributes:
:Title = "CHIMERE SUITE";
:Sub-title = "Interpolated Meteo file";
:Generating_process = "Generated by prepexmm5";
:Conventions = "";
:Domain = "CONT3";
:history = "File exdomout.nc was generated on",
"Tue Nov 15 14:25:27 2005 by <user> on <host>",
"from input file EUR1/MMOUT_EUR1_20030730_20030803_S"
:mm5_nxx = 85;
:mm5_nxx_DOT = 86;
:mm5_nyy = 75;
:mm5_nyy_DOT = 76;
:mm5_nlev = 32;
}

E.3 METEO.nc

netcdf METEO {
dimensions:
Time = UNLIMITED ; // (13 currently)
DateStrLen = 19 ;
west_east = 67 ;
south_north = 46 ;
bottom_top = 8 ;
variables:
char Times(Time, DateStrLen);
float lon(south_north, west_east);
lon:units = "degrees_east";
lon:long_name = "Longitude";
float lat(south_north, west_east);
l lat:units = "degrees_north";
l lat:long_name = "Latitude";
float tem2(Time, south_north, west_east);
tem2:units = "K";
 tem2:long_name = "2m air temperature";
float atte(Time, south_north, west_east);
atte:units = "0-1";
atte:long_name = "Cloud attenuation";
float hght(Time, south_north, west_east);
hght:units = "m";
hght:long_name = "PBL height";
float usta(Time, south_north, west_east);
usta:units = "m/s";
usta:long_name = "Frictional velocity";
float aerr(Time, south_north, west_east);
aerr:units = "*";
aerr:long_name = "*";
float obuk(Time, south_north, west_east);
obuk:units = "m";
obuk:long_name = "Obukov length";
float wsta(Time, south_north, west_east);
wsta:units = "m/s";
wsta:long_name = "Wstar";
float sreh(Time, south_north, west_east);
sreh:units = "0-1";
sreh:long_name = "Sfc rel. humidity";
float topc(Time, south_north, west_east);
topc:units = "Kg/m^2";
topc:long_name = "Total precip";
float w10m(Time, south_north, west_east);
w10m:units = "m/s";
w10m:long_name = "10m wind";
float w10s(Time, south_north, west_east);
w10s:units = "m/s";
w10s:long_name = "Saltation wind";
float soim(Time, south_north, west_east);
soim:units = "m^3/m^3";
soim:long_name = "Soil moisture";
float alti(Time, bottom_top, south_north, west_east);
alti:units = "m";
alti:long_name = "Altitude of layer";
float winz(Time, bottom_top, south_north, west_east);
winz:units = "m/s";
winz:long_name = "Zonal wind";
float winm(Time, bottom_top, south_north, west_east);
winm:units = "m/s";
winm:long_name = "Meridional wind";
float temp(Time, bottom_top, south_north, west_east);
temp:units = "K";
temp:long_name = "Temperature";
float sphu(Time, bottom_top, south_north, west_east);
sphu:units = "Kg/kg";
sphu:long_name = "Specific humidity";
float airm(Time, bottom_top, south_north, west_east);
airm:units = "molec/cm3";
airm:long_name = "Air density";
float kzzz(Time, bottom_top, south_north, west_east);
kzzz:units = "*";
kzzz:long_name = "Kz";
float clwc(Time, bottom_top, south_north, west_east);
clwc:units = "Kg/Kg";
clwc:long_name = "Cloud wat. content";

// global attributes:
:Title = "CHIMERE SUITE";
:Sub-title = "Meteo file";
:Generating_process = "Generated by diagmet";
:Conventions = "";
:Domain = "CONT3";
:history = "File METEO.nc was generated on",
"Tue Nov 15 14:25:39 2005 by <user> on <host>",
"from input file exdomout.nc"
"File exdomout.nc was generated on",
"Tue Nov 15 14:25:27 2005 by <user> on <host>",
"from input file MMOUT_EUR1_20030730_20030803_S"

E.4 AEMISSIONS.nc

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 APINEN(Time, bottom_top, south_north, west_east);  
APINEN:units = "molecule/cm2/s";  
APINEN:long_name = "APINEN Emission";  
float C2H4(Time, bottom_top, south_north, west_east);  
C2H4:units = "molecule/cm2/s";  
C2H4:long_name = "C2H4 Emission";  
float C2H6(Time, bottom_top, south_north, west_east);  
C2H6:units = "molecule/cm2/s";  
C2H6:long_name = "C2H6 Emission";  
float C3H6(Time, bottom_top, south_north, west_east);  
C3H6:units = "molecule/cm2/s";  
C3H6:long_name = "C3H6 Emission";  
float C5H8(Time, bottom_top, south_north, west_east);  
C5H8:units = "molecule/cm2/s";  
C5H8:long_name = "C5H8 Emission";  
float CH3CHO(Time, bottom_top, south_north, west_east);  
CH3CHO:units = "molecule/cm2/s";  
CH3CHO:long_name = "CH3CHO Emission";  
float CH3COE(Time, bottom_top, south_north, west_east);  
CH3COE:units = "molecule/cm2/s";  
CH3COE:long_name = "CH3COE Emission";  
float CH4(Time, bottom_top, south_north, west_east);  
CH4:units = "molecule/cm2/s";  
CH4:long_name = "CH4 Emission";  
float CO(Time, bottom_top, south_north, west_east);  
CO:units = "molecule/cm2/s";  
CO:long_name = "CO Emission";  
float HCHO(Time, bottom_top, south_north, west_east);  
HCHO:units = "molecule/cm2/s";  
HCHO:long_name = "HCHO Emission";  
float HONO(Time, bottom_top, south_north, west_east);  
HONO:units = "molecule/cm2/s";  
HONO:long_name = "HONO Emission";  
float NC4H10(Time, bottom_top, south_north, west_east);  
NC4H10:units = "molecule/cm2/s";  
NC4H10:long_name = "NC4H10 Emission";  
float NH3(Time, bottom_top, south_north, west_east);  
NH3:units = "molecule/cm2/s";  
NH3:long_name = "NH3 Emission";  
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 OXYL(Time, bottom_top, south_north, west_east);  
OXYL:units = "molecule/cm2/s";  
OXYL:long_name = "OXYL Emission";
float SO2(Time, bottom_top, south_north, west_east);
SO2:units = "molecule/cm2/s";
SO2:long_name = "SO2 Emission";
float H2SO4_fin(Time, bottom_top, south_north, west_east);
H2SO4_fin:units = "molecule/cm2/s";
H2SO4_fin:long_name = "H2SO4_fin 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";
float OCAR_fin(Time, bottom_top, south_north, west_east);
OCAR_fin:units = "molecule/cm2/s";
OCAR_fin:long_name = "OCAR_fin Emission";
float BCAR_fin(Time, bottom_top, south_north, west_east);
BCAR_fin:units = "molecule/cm2/s";
BCAR_fin:long_name = "BCAR_fin Emission";
float TOL(Time, bottom_top, south_north, west_east);
TOL:units = "molecule/cm2/s";
TOL:long_name = "TOL Emission";
float TMB(Time, bottom_top, south_north, west_east);
TMB:units = "molecule/cm2/s";
TMB:long_name = "TMB Emission";

// global attributes:
:Title = "CHIMERE SUITE";
:Sub-title = "Emissions file - Surfacic + Point sources + Fires";
:Generating_process = "Generated by prepemis";
:Conventions = "";
:Domain = "CONT5";
:history = "";
}

E.5 BEMISSIONS.nc
	netcdf BEMISSIONS {
	dimensions:
	  Time = UNLIMITED ; // (13 currently)
	  DateStrLen = 19 ;
	  west_east = 67 ;
	  south_north = 46 ;
	  biospecies = 11 ;
	n variables:
	  char Times(Time, DateStrLen) ;
	  float lon(south_north, west_east) ;
	  float lat(south_north, west_east) ;
	  float emisb(Time, south_north, west_east, biospecies) ;

	// global attributes:
	:Title = "CHIMERE SUITE";
	:Sub-title = "Bioemissions file";
	:Generating_process = "Generated by calbio";
	:Conventions = "";
	:Domain = "CONT3";
	:history = "File BEMISSIONS.nc",
	  "was generated on Tue Nov 15 14:25:40 2005 ",
	  "by <user> on <host> from input file",
	  "METEO.nc\n",
	}
E.6 BOUN_CONCS.nc

```netcdf
BOUN_CONCS {

dimensions:
  Time = UNLIMITED ; // (121 currently)
  DateStrLen = 19 ;
  SpStrLen = 23 ;
  west_east = 101 ;
  south_north = 111 ;
  bottom_top = 20 ;
  h_boundary = 424 ;
  Species = 169 ;

variables:
  char Times(Time, DateStrLen) ;
  char species(Species, SpStrLen) ;
  float lon(south_north, west_east) ;
  float lat(south_north, west_east) ;
  float top_conc(Time, south_north, west_east, Species) ;
  float lat_conc(Time, bottom_top, h_boundary, Species) ;

// global attributes:
  :Title = "CHIMERE SUITE" ;
  :Sub-title = "Fine boundary concentrations file" ;
  :Generating_process = "Generated by prep_CHIMERE" ;
}
```

E.7 end.nc

```netcdf
end.20030730_20030803_heat-wave {

dimensions:
  Time = UNLIMITED ; // (1 currently)
  DateStrLen = 19 ;
  SpecStrLen = 16 ;
  Species = 119 ;
  west_east = 67 ;
  south_north = 46 ;
  bottom_top = 8 ;

variables:
  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" ;
  char species_name(Species, SpecStrLen) ;
  char Times(Time, DateStrLen) ;
  float APINEN(Time, bottom_top, south_north, west_east) ;
  APINEN:units = "molecules/cm3" ;
  float C2H4(Time, bottom_top, south_north, west_east) ;
  C2H4:units = "molecules/cm3" ;
  float C2H6(Time, bottom_top, south_north, west_east) ;
  C2H6:units = "molecules/cm3" ;
  float C3H6(Time, bottom_top, south_north, west_east) ;
  C3H6:units = "molecules/cm3" ;
  float C5H8(Time, bottom_top, south_north, west_east) ;
  C5H8:units = "molecules/cm3" ;
  float CARNIT(Time, bottom_top, south_north, west_east) ;
```
CARNIT:units = "molecules/cm³";  
float CH3CHO(Time, bottom_top, south_north, west_east);  
CH3CHO:units = "molecules/cm³";  
float CH3COO(Time, bottom_top, south_north, west_east);  
CH3COO:units = "molecules/cm³";  
float CH3COO(Time, bottom_top, south_north, west_east);  
CH3COO:units = "molecules/cm³";  
float CH3COY(Time, bottom_top, south_north, west_east);  
CH3COY:units = "molecules/cm³";  
float CH3O2(Time, bottom_top, south_north, west_east);  
CH3O2:units = "molecules/cm³";  
float CH4(Time, bottom_top, south_north, west_east);  
CH4:units = "molecules/cm³";  
float CO(Time, bottom_top, south_north, west_east);  
CO:units = "molecules/cm³";  
float DUSTAQ(Time, bottom_top, south_north, west_east);  
DUSTAQ:units = "molecules/cm³";  
float GLYOX(Time, bottom_top, south_north, west_east);  
GLYOX:units = "molecules/cm³";  
float H2O2(Time, bottom_top, south_north, west_east);  
H2O2:units = "molecules/cm³";  
float H2SO4(Time, bottom_top, south_north, west_east);  
H2SO4:units = "molecules/cm³";  
float H2SO4AQ(Time, bottom_top, south_north, west_east);  
H2SO4AQ:units = "molecules/cm³";  
float HCHO(Time, bottom_top, south_north, west_east);  
HCHO:units = "molecules/cm³";  
float HNO3(Time, bottom_top, south_north, west_east);  
HNO3:units = "molecules/cm³";  
float HNO3AQ(Time, bottom_top, south_north, west_east);  
HNO3AQ:units = "molecules/cm³";  
float HO2(Time, bottom_top, south_north, west_east);  
HO2:units = "molecules/cm³";  
float HONO(Time, bottom_top, south_north, west_east);  
HONO:units = "molecules/cm³";  
float ISNI(Time, bottom_top, south_north, west_east);  
ISNI:units = "molecules/cm³";  
float MAC(Time, bottom_top, south_north, west_east);  
MAC:units = "molecules/cm³";  
float MEMALD(Time, bottom_top, south_north, west_east);  
MEMALD:units = "molecules/cm³";  
float MGLYOX(Time, bottom_top, south_north, west_east);  
MGLYOX:units = "molecules/cm³";  
float MVK(Time, bottom_top, south_north, west_east);  
MVK:units = "molecules/cm³";  
float N2O5(Time, bottom_top, south_north, west_east);  
N2O5:units = "molecules/cm³";  
float NC4H10(Time, bottom_top, south_north, west_east);  
NC4H10:units = "molecules/cm³";  
float NH3(Time, bottom_top, south_north, west_east);  
NH3:units = "molecules/cm³";  
float NH3AQ(Time, bottom_top, south_north, west_east);  
NH3AQ:units = "molecules/cm³";  
float NO(Time, bottom_top, south_north, west_east);  
NO:units = "molecules/cm³";  
float NO2(Time, bottom_top, south_north, west_east);  
NO2:units = "molecules/cm³";  
float NO3(Time, bottom_top, south_north, west_east);  
NO3:units = "molecules/cm³";
float O3(Time, bottom_top, south_north, west_east); O3:units = "molecules/cm3";
float obio(Time, bottom_top, south_north, west_east); obio:units = "molecules/cm3";
float obioH(Time, bottom_top, south_north, west_east); obioH:units = "molecules/cm3";
float OH(Time, bottom_top, south_north, west_east); OH:units = "molecules/cm3";
float oPAN(Time, bottom_top, south_north, west_east); oPAN:units = "molecules/cm3";
float oRN1(Time, bottom_top, south_north, west_east); oRN1:units = "molecules/cm3";
float OR02(Time, bottom_top, south_north, west_east); OR02:units = "molecules/cm3";
float OR0OH(Time, bottom_top, south_north, west_east); OR0OH:units = "molecules/cm3";
float OXYL(Time, bottom_top, south_north, west_east); OXYL:units = "molecules/cm3";
float PAN(Time, bottom_top, south_north, west_east); PAN:units = "molecules/cm3";
float PANH(Time, bottom_top, south_north, west_east); PANH:units = "molecules/cm3";
float PPA(Time, bottom_top, south_north, west_east); PPA:units = "molecules/cm3";
float PPMAQ(Time, bottom_top, south_north, west_east); PPMAQ:units = "molecules/cm3";
float p1PPM(Time, bottom_top, south_north, west_east); p1PPM:units = "molecules/cm3";
float p1DUST(Time, bottom_top, south_north, west_east); p1DUST:units = "molecules/cm3";
float p1SOA(Time, bottom_top, south_north, west_east); p1SOA:units = "molecules/cm3";
float p1H2SO4(Time, bottom_top, south_north, west_east); p1H2SO4:units = "molecules/cm3";
float p1HNO3(Time, bottom_top, south_north, west_east); p1HNO3:units = "molecules/cm3";
float p1NH3(Time, bottom_top, south_north, west_east); p1NH3:units = "molecules/cm3";
float p1WATER(Time, bottom_top, south_north, west_east); p1WATER:units = "molecules/cm3";
float p2PPM(Time, bottom_top, south_north, west_east); p2PPM:units = "molecules/cm3";
float p2DUST(Time, bottom_top, south_north, west_east); p2DUST:units = "molecules/cm3";
float p2SOA(Time, bottom_top, south_north, west_east); p2SOA:units = "molecules/cm3";
float p2H2SO4(Time, bottom_top, south_north, west_east); p2H2SO4:units = "molecules/cm3";
float p2HNO3(Time, bottom_top, south_north, west_east); p2HNO3:units = "molecules/cm3";
float p2NH3(Time, bottom_top, south_north, west_east); p2NH3:units = "molecules/cm3";
float p2WATER(Time, bottom_top, south_north, west_east);
p2WATER:units = "molecules/cm3";
float p3PPM(Time, bottom_top, south_north, west_east);
p3PPM:units = "molecules/cm3";
float p3DUST(Time, bottom_top, south_north, west_east);
p3DUST:units = "molecules/cm3";
float p3SOA(Time, bottom_top, south_north, west_east);
p3SOA:units = "molecules/cm3";
float p3H2SO4(Time, bottom_top, south_north, west_east);
p3H2SO4:units = "molecules/cm3";
float p3HNO3(Time, bottom_top, south_north, west_east);
p3HNO3:units = "molecules/cm3";
float p3NH3(Time, bottom_top, south_north, west_east);
p3NH3:units = "molecules/cm3";
float p3WATER(Time, bottom_top, south_north, west_east);
p3WATER:units = "molecules/cm3";
float p4PPM(Time, bottom_top, south_north, west_east);
p4PPM:units = "molecules/cm3";
float p4DUST(Time, bottom_top, south_north, west_east);
p4DUST:units = "molecules/cm3";
float p4SOA(Time, bottom_top, south_north, west_east);
p4SOA:units = "molecules/cm3";
float p4H2SO4(Time, bottom_top, south_north, west_east);
p4H2SO4:units = "molecules/cm3";
float p4HNO3(Time, bottom_top, south_north, west_east);
p4HNO3:units = "molecules/cm3";
float p4NH3(Time, bottom_top, south_north, west_east);
p4NH3:units = "molecules/cm3";
float p4WATER(Time, bottom_top, south_north, west_east);
p4WATER:units = "molecules/cm3";
float p5PPM(Time, bottom_top, south_north, west_east);
p5PPM:units = "molecules/cm3";
float p5DUST(Time, bottom_top, south_north, west_east);
p5DUST:units = "molecules/cm3";
float p5SOA(Time, bottom_top, south_north, west_east);
p5SOA:units = "molecules/cm3";
float p5H2SO4(Time, bottom_top, south_north, west_east);
p5H2SO4:units = "molecules/cm3";
float p5HNO3(Time, bottom_top, south_north, west_east);
p5HNO3:units = "molecules/cm3";
float p5NH3(Time, bottom_top, south_north, west_east);
p5NH3:units = "molecules/cm3";
float p5WATER(Time, bottom_top, south_north, west_east);
p5WATER:units = "molecules/cm3";
float p6PPM(Time, bottom_top, south_north, west_east);
p6PPM:units = "molecules/cm3";
float p6DUST(Time, bottom_top, south_north, west_east);
p6DUST:units = "molecules/cm3";
float p6SOA(Time, bottom_top, south_north, west_east);
p6SOA:units = "molecules/cm3";
float p6H2SO4(Time, bottom_top, south_north, west_east);
p6H2SO4:units = "molecules/cm3";
float p6HNO3(Time, bottom_top, south_north, west_east);
p6HNO3:units = "molecules/cm3";
float p6NH3(Time, bottom_top, south_north, west_east);
p6NH3:units = "molecules/cm3";
float p6WATER(Time, bottom_top, south_north, west_east);
p6WATER:units = "molecules/cm3";
float M(Time, bottom_top, south_north, west_east);
M:units = "molecules/cm3";
float O2(Time, bottom_top, south_north, west_east);
O2:units = "molecules/cm3";
float N2(Time, bottom_top, south_north, west_east);  
N2:units = "molecules/cm³";
float H2O(Time, bottom_top, south_north, west_east);  
H2O:units = "molecules/cm³";
float TOTPAN(Time, bottom_top, south_north, west_east);  
TOTPAN:units = "molecules/cm³";
float NOX(Time, bottom_top, south_north, west_east);  
NOX:units = "molecules/cm³";
float OX(Time, bottom_top, south_north, west_east);  
OX:units = "molecules/cm³";
float NOY(Time, bottom_top, south_north, west_east);  
NOY:units = "molecules/cm³";
float ROOH(Time, bottom_top, south_north, west_east);  
ROOH:units = "molecules/cm³";
float HCNM(Time, bottom_top, south_north, west_east);  
HCNM:units = "molecules/cm³";
float PM10(Time, bottom_top, south_north, west_east);  
PM10:units = "molecules/cm³";
float PM25(Time, bottom_top, south_north, west_east);  
PM25:units = "molecules/cm³";
float PM10_anth(Time, bottom_top, south_north, west_east);  
PM10_anth:units = "molecules/cm³";
float PM25_anth(Time, bottom_top, south_north, west_east);  
PM25_anth:units = "molecules/cm³";
float PPM10_anth(Time, bottom_top, south_north, west_east);  
PPM10_anth:units = "molecules/cm³";
float PPM25_anth(Time, bottom_top, south_north, west_east);  
PPM25_anth:units = "molecules/cm³";
float pPPM(Time, bottom_top, south_north, west_east);  
pPPM:units = "molecules/cm³";
float pDUST(Time, bottom_top, south_north, west_east);  
pDUST:units = "molecules/cm³";
float pSOA(Time, bottom_top, south_north, west_east);  
pSOA:units = "molecules/cm³";
float pH2SO4(Time, bottom_top, south_north, west_east);  
pH2SO4:units = "molecules/cm³";
float pHNO3(Time, bottom_top, south_north, west_east);  
pHNO3:units = "molecules/cm³";
float pHN3(Time, bottom_top, south_north, west_east);  
pHN3:units = "molecules/cm³";
float pWATER(Time, bottom_top, south_north, west_east);  
pWATER:units = "molecules/cm³";

// global attributes:
>Title = "CHIMERE SUITE";
>Sub-title = "Final concentrations file";
>Chimere_type = "end";
>Generating_process = "Generated by chimere";
>Conventions = "None";
>Domain = "CONT3";
>history = "File end.20030730_20030803_heat-wave.nc",
"was generated on Tue Nov  8 12:27:58 2005",
"by <user> on <host>\n";