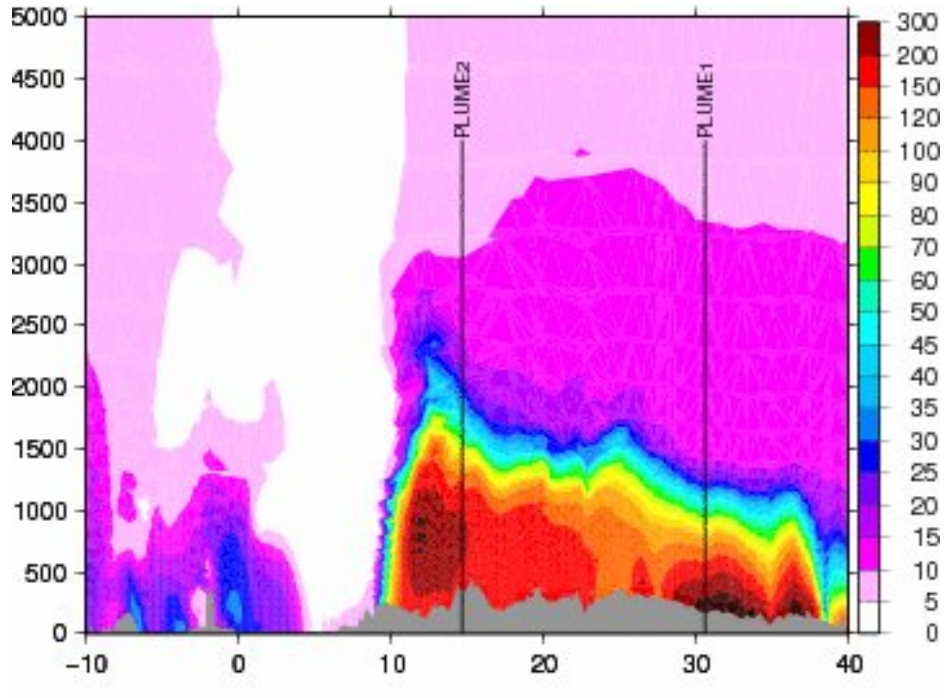
Documentation of the chemistry-transport model

Chimere

[version chimere2008]



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 envrionment . . . 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

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

C.3.3 pgf90 . . . 139
C.3.4 ifort 64 bit on Intel EMT64 or AMD Opteron . . . 139
C.3.5 g95 64 bit on Intel EMT64 or AMD Opteron . . . 139
C.3.6 pgf90 64 bit on Intel EMT64 or AMD Opteron . . . 140
C.4 Configure CHIMERE . . . 140

D How To install MPI under GNU/Linux **143**
D.1 Background . . . 143
D.2 LAM/MPI Installation . . . 143
D.3 Testing . . . 144
D.4 Uniprocessor users . . . 144
D.5 Installation from source . . . 145
D.6 Open MPI installation . . . 145

E Structure of the CHIMERE netCDF files **147**
E.1 EMIS.[domain].[MM].[SPEC].s.nc and EMIS.[domain].[MM].[SPEC].p.nc . . . 147
E.2 exdomout.nc . . . 148
E.3 METEO.nc . . . 150
E.4 AEMISSIONS.nc . . . 151
E.5 BEMISSIONS.nc . . . 153
E.6 BOUN_CONCS.nc . . . 154
E.7 end.nc . . . 154

List of Figures

1.1 *The simulation of the 2003 heat wave: surface ozone concentrations ($\mu g.m^{-3}$)* 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 atmopsheric 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,
- Cathy Liousse, Bruno Guillaume and Robert Rosset (LA / OMP, http://www.aero.obs-mip.fr) for organic and black carbon emissions.
- 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[1]

[1]Bertrand 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 μg.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

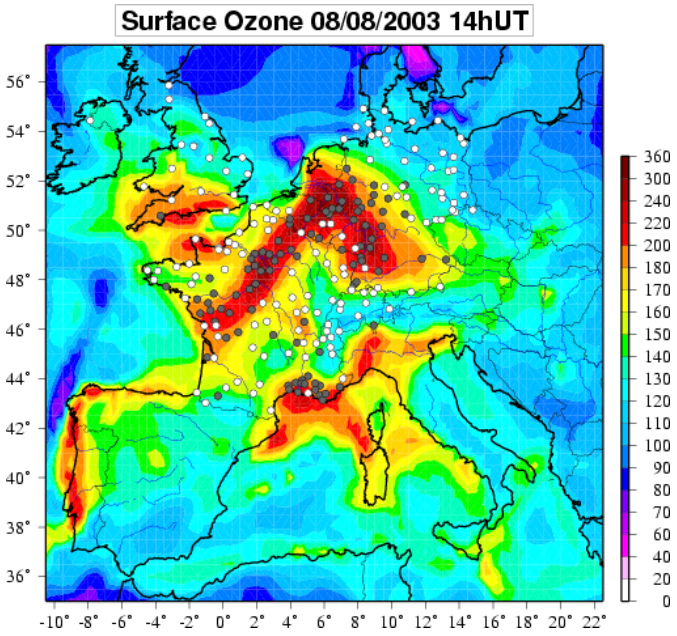


Figure 1.1: *The simulation of the 2003 heat wave: surface ozone concentrations ($\mu g.m^{-3}$)*

- **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 **exachim** : 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 foreseenable 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. **ncmapit**, **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

Models	
WRF model (free)	http://www.wrf-model.org/index.php
MM5 model (free)	http://box.mmm.ucar.edu/mm5
NCAR	http://ncar.ucar.edu
ISORROPIA	http://nenes.eas.gatech.edu/ISORROPIA
Librairies	
Unidata netCDF library (free)	http://www.unidata.ucar.edu/
Indiana university LAM/MPI library (free)	http://www.lam-mpi.org/
Graphics softwares	
GrADS graphical software (free)	http://grads.iges.org/grads/grads.html
GMT software (free)	http://gmt.soest.hawaii.edu/
Fortran compilers	
g95 compiler (free)	http://g95.sourceforge.net/
ifort compiler	http://www.intel.com/cd/software/products/asmo-na/eng/compilers/index.htm
pgf90 compiler	www.pgroup.com
Databases	
EMEP	http://www.emep.int
Global Land Cover Facility (GLCF)	http://glcf.umiacs.umd.edu/data/landcover
LAI from NASA/EOSDIS	http://www.daac.ornl.gov
Operational air quality forecasts with CHIMERE	
Western Europe	http://prevair.ineris.fr
The Paris Region	http://www.airparif.asso.fr
The Alsace Region	http://www.atmo-alsace.net
The Lyon Region	http://www.atmo-rhonealpes.org
The Toulouse Region	http://www.oramip.org
The Marseille area	http://www.atmopaca.org/

Table 2.1: *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`]. 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.

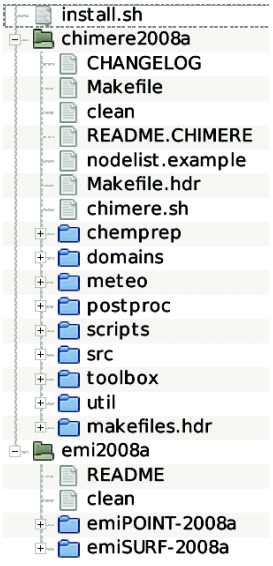
[1]For the use of these emissions data, please acknowledge EMEP (Yearly totals), IER (Time variations), TNO (Aerosol emissions), UK Dept of Environment (VOC speciation, Passant, 2002).

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 dor 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.

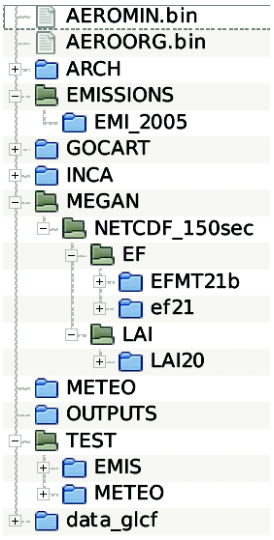
The listing of the installation script **install.sh** is given in **section B.5, p.132**.
You can also generate a new **Makefile.hdr** at any time after Chimere installation running **install.sh** and choosing not to download/install the model.

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.



(a) The current directory (where the user launched the installation script)



(b) The BIGFILES directory

Figure 2.1: *The directory tree after Chimere installation using* **install.sh** *script*

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 preformated 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 envrionment

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 course 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 :

Flag	Action	Script in the **scripts/** directory	Chimere component(s)
imakecompil	Model compilation	chimere-compil.sh	
imakemeteo	Meteorological interfaces	chimere-meteo.sh	interf-<model> and diagmet.e
imakeemis	Emissions interface	chimere-emis.sh	prepemis.e
imakebound	Inititial and boundary conditions	chimere-bound.sh	prep_bound.e/prep_chimere.e
imakerun	Running Chimere	chimere-run.sh	chimere.e

If some of the Chimere components are not to be launched, their resutls (e.g., the file METEO.nc for the meteorological interfaces) must be stored in the directory specified by the datadir variable in **chimere.sh**.

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

```
./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**:

```
#!/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:
imakecompil=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 )
```

```
_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 )
_accur=( 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 a
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....)
nsconcs=24
#     Save all cumulated deposition every ... houres (file dep...)
nsdepos=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
```

```
uwinfac=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=`pwd`/../../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=${bigfilesdir}/OUTPUTS
#     If this run is a continuation one, initial conditions will be obtained from the "end" file in
psimuldir=${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

#=======================================================
```

```
#               End of user changes
#=========================================================

# some checks
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

echo "---------------------------------------------------------"
echo "o   Make chemical input data and geometry"

. ${chimere_root}/scripts/make-chemgeom.sh
[ $? -eq 0 ] || { echo "Abnormal termination of make-chemgeom.sh"; exit 1; }

# Initialization of common parameters
. ${chimere_root}/scripts/chimere-init.sh

# Compilation
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
if [ ${imakemeteo} = "1" ] ; then
   .  ${chimere_root}/scripts/chimere-meteo.sh
   [ $? -eq 0 ] || { echo "Abnormal termination of chimere-meteo.sh"; exit 1; }
elif [[ ${imakeemis} = "1" || ${imakerun} = "1" ]] ; then
   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
if [ ${imakebound} = "1" ] ; then
   .  ${chimere_root}/scripts/chimere-bound.sh
   [ $? -eq 0 ] || { echo "Abnormal termination of chimere-bound.sh"; exit 1; }
elif [ ${imakerun} = "1" ] ; then
   ln -s ${datadir}/*_CONCS.nc  ${chimere_tmp} || \
         { echo "Cannot link ${datadir}/*_CONCS.nc to ${chimere_tmp}"; exit 1; }
fi

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

# Chimere run
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 is 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 recommanded)
# 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 _dust_ = 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
# metdom        Name of the meteorological domain
# 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
# phys          Frequency of physics calculation (number per hour)
# 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 `yyyymmddhh`
- 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 mesocscale 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, ony the first vertical cell is taken into account. On the other hand, for the wet deposition, fluxes are sumed up the whole column since, in the model, each vertical level may contribute to a net sink. Units of these two integrated fluxes are g/cm^2.

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**.

Model Parameter	**Variable**
wind	
iwinhs	Wind speed $\|U\|$ (m/s 1st vertical level)1
iwinh2	Wind speed $\|U\|$ (m/s 2nd vertical level)1
iwinvs	Vertical wind speed (m/s 1st vertical level)
iwinvt	Vertical wind speed (m/s top domain vertical level)
specific humidity, air density and temperature	
isphu	Specific humidity
iairm	Air density
itemp	Temperature (K)
iwinxs	Velocity-equivalent of mixing exchange rate (surface)
iwinxt	Velocity-equivalent of mixing exchange rate (top domain)
ithlay	Thickness of the model layer (in cm)
mixing height, radiation	
ihght	Mixing height (in meters)
iatte	Radiation attenuation [0-1]
izeni	zenithal angle (degrees)
item2	2m temperature (K)
diagnostic turbulent parameters	
iusta	Friction velocity u_* (m/s)
iaerr	Aerodynamical resistance
iobuk	Monin-Oblukov lengtth
iwsta	Convective velocity w_*
idepo	Dry deposition velocity

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 developping 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 succesfully 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. 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 ncmapit

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 AEMISSONS, BEMISSIONS, and Boundary conditions files due to the very specific format of these files.

Usage :

- 1st : build chimere2gmt.e
  `make`
- 2nd : run ncmapit.sh
  `./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:

Filename	Min	Max	Step	Filename	Min	Max	Step
CH4.cpt	1700	1900	20	pASOA.cpt	0	200	5/10/50
CO.cpt	0	1000	50	pH2SO4.cpt	0	200	5/10/50
HNO3.cpt	0	20	1	pHNO3.cpt	0	200	5/10/50
NO.cpt	0	50	2.5	PM10.cpt	0	200	5/10/50
NO2.cpt	0	150	10	PM25.cpt	0	200	5/10/50
NOY.cpt	0	150	10	pNH3.cpt	0	200	5/10/50
O3.cpt	0	150	10/50	pNH3-reg.cpt	0	20	2
PAN.cpt	0	6	0.3	pPPM.cpt	0	200	5/10/50
pHNO3-reg.cpt	0	50	5	SO2.cpt	0	100	10

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 adaptated 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**

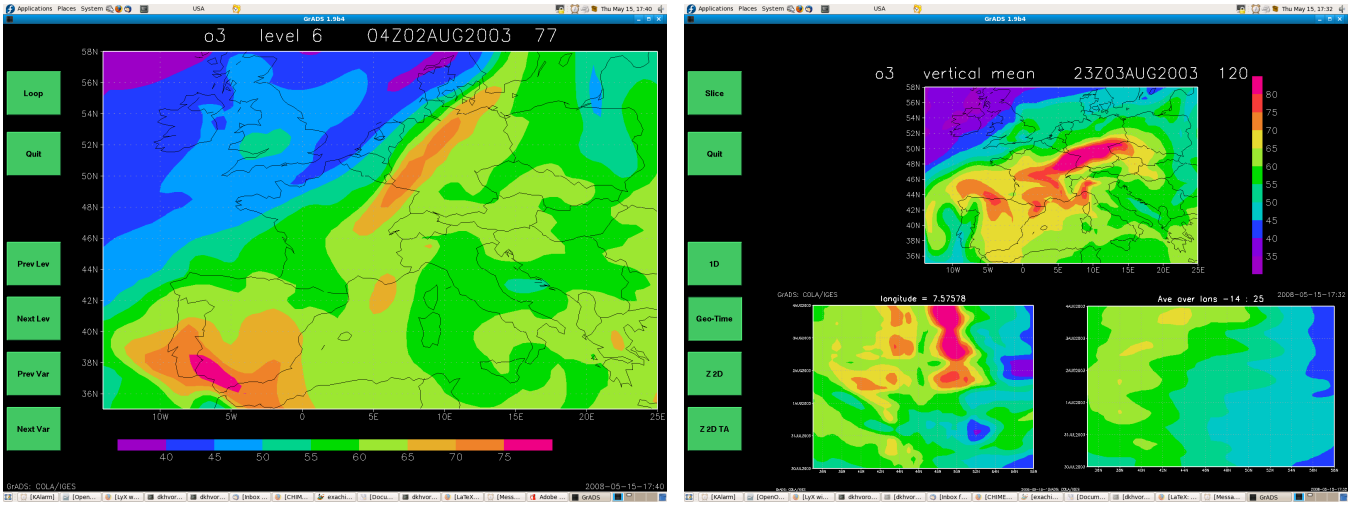
with:

- **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



(a) In the "Loop" Mode (b) In the "Slice" Mode

Figure 2.2: *Exachim screenshot*

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

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

Table 2.4: Contents of the "Slice" mode additional windows depending on the mode and mouse button pressed

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 2.2**a).

- 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 2.2**b). 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

  - ☐ its value at the selected grid point
  - ☐ the coordinates of the grid point
- clicks on the Level or Var buttons are ineffective.
- a click on "Data" resets the display in "Loop" mode.

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 dwcrd_x and dwcrd_y in the beginning of the GrADS script **exachim.gs**. They are otimized 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.

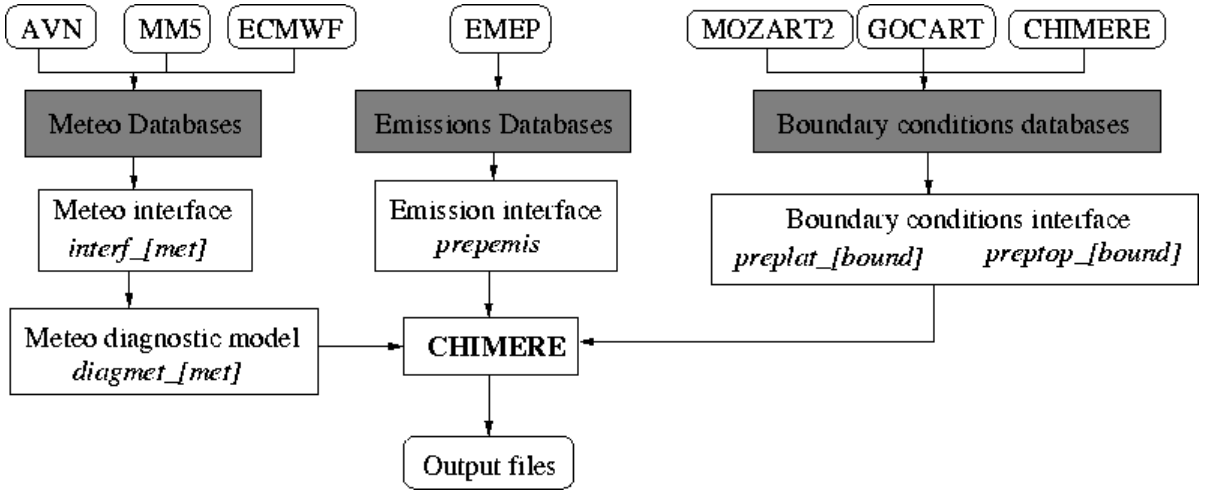


Figure 3.1: *Schematic structure of the CHIMERE input data*

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 convering 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**).

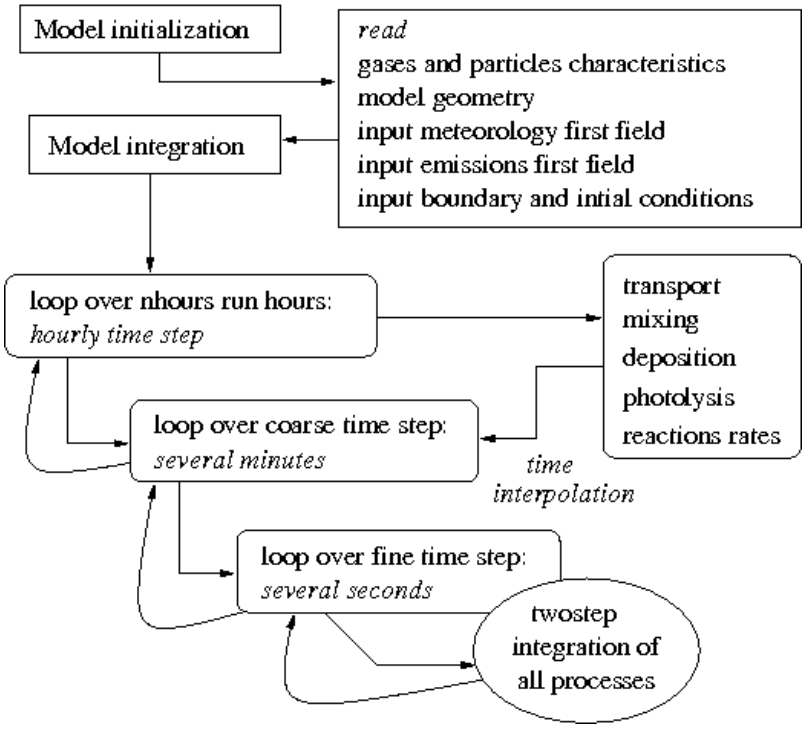


Figure 3.2: *General CHIMERE structure for time integration of all processes*

- The **initialization phase** is dedicated to the read of all inpout 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 splitted 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 entraiment 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).

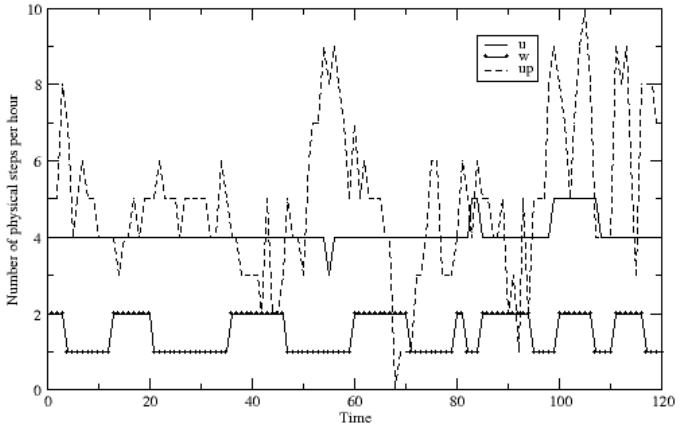


Figure 3.3: *Calculation of the number of integration steps per hour to respect the CFL over a complete 120 hours simulation.*

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**, **prepmm5.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 developpers :
  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 scripts 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 inititalize 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.

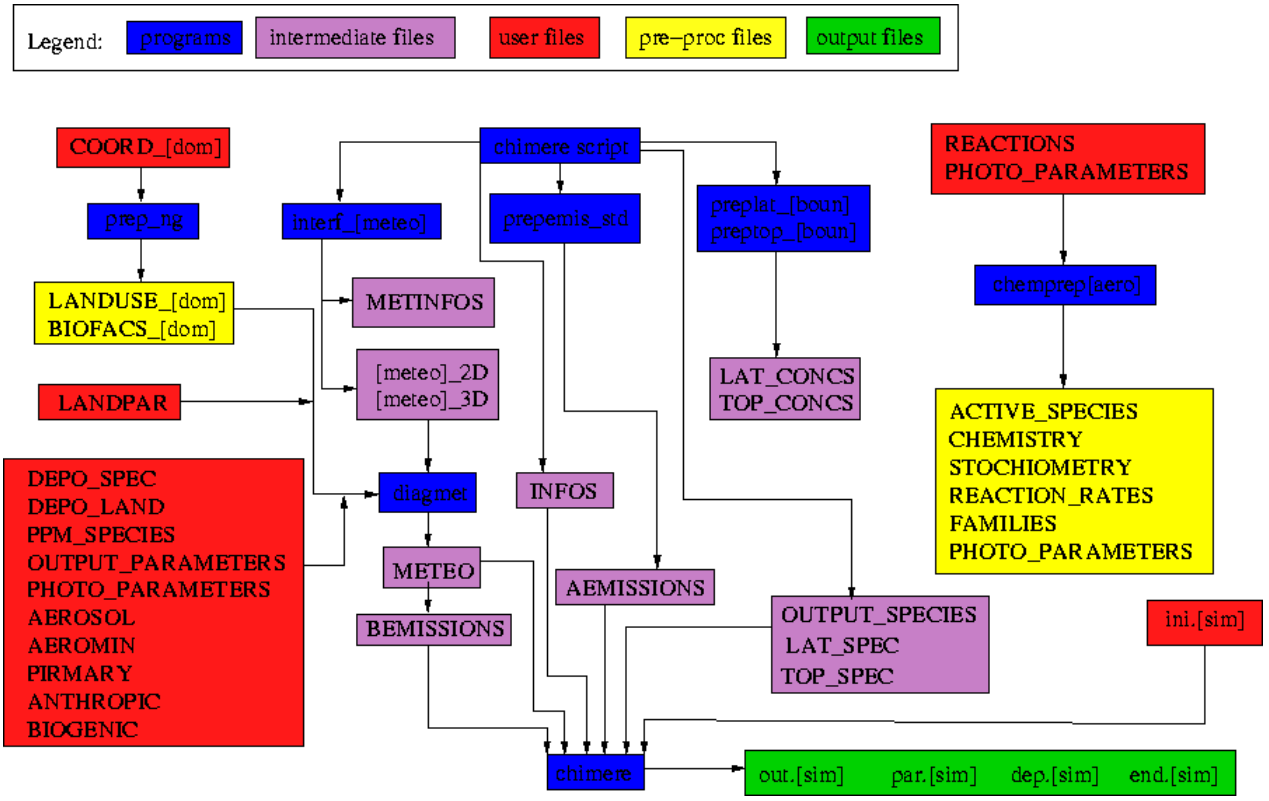


Figure 3.4: *Schematic general structure of the scripts, data files of CHIMERE*

- 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.

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 discribed 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]: contains the “cross” coordinates of the MM5 file.
   - meteo/COORD_MM5[domain]_DOT: contains the “dot” coordinates of the MM5 file.

- 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/prepmm5.f90, meteo/prepwrf.f90: read MM5 and WRF standard output files and perform interpolations to the model grid. Compiled to **prepmm5.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/wcal1.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 **prepmm5.e** and **prepwrf.e**.
- tmpdir/meteo/prepmm5.nml or tmpdir/meteo/prepwrf.nml: namelist generated by **meteo/interf-mm5** or **meteo/interf-wrf** to drive **prepmm5.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 subsituted 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 subsituted 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_subs.f90 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_subs 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/AEMISSIONS.nc: netCDF file generated by the anthropogenic emissions interface. Contains timely anthropogenic emissions.
   - tmpdir/BEMISSIONS.nc: netCDF file generated by **diagbio.e**. Contains timely biogenic emissions.
   - 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**.

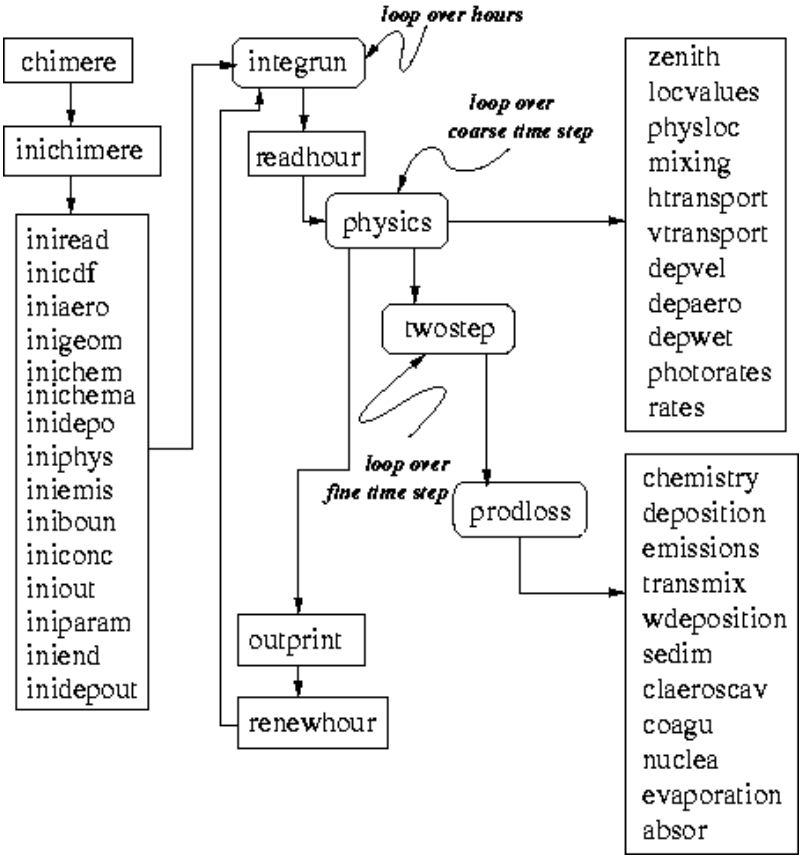


Figure 3.5: *Detailed list of CHIMERE routines and time loops*

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

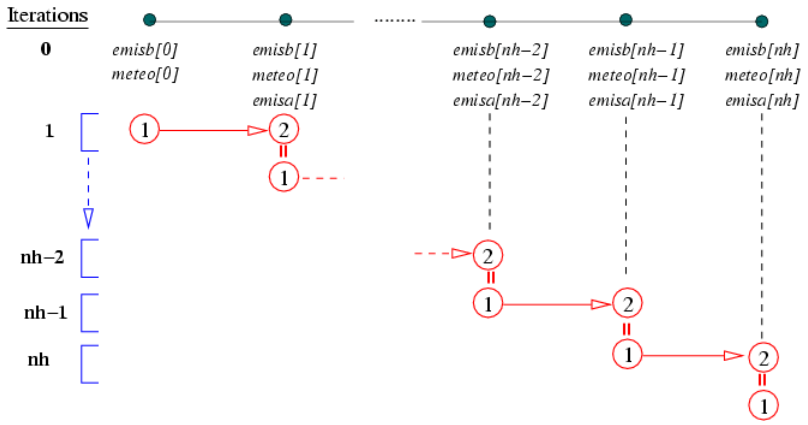


Figure 3.6: *Time processing in CHIMERE*

a CPU for the master and declare only **nzdoms*nmdoms** in your hostlist file.

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 **tn+1**, you need source values at point **i-3, i-2, i-1**, at time **tn+1**, and source values **i, i+1, i+2, i+3** at time **tn**. You cannot simply calculate target values at time **tn+1** from a set of

source values at time **tn**. That is, to start the calculation of the 5th line of a subdomain, you need to know the new values of the 3 last pixels of the 5th line of its left neighbour. This is realized by setting a dependancy 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 dependancy relationship in both directions leads to a straightforward conclusion : *it is impossible to parallelize CHIMERE.*
We have practically solved the problem by starting the computation of each row of subdomains as if it were an independant domain, with no upper and lower neighbour, exactly as we do for the main domain in the sequential scheme. But at the end of each fine time step, we update their upper and lower halos. This provides a reasonnable smoothing between subdomains, albeit not beeing 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*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 80*80 domain and the gas version, you will waste CPU time in message passing if you go beyond 4*4 subdomains. On the other hand, the same 80*80 simulation with the aerosols version, sea salts and 3 Gauss-Seidel iterations should still be efficient with 8*8 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 anf 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 fisrt processor will be 1, produced either by the master, or by the first worker. But if your MPI library has choosen 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 awaking it from a real sleep. This is achieved, with LAM/MPI, with this command replacing the previous one:

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

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

- fnbiogen = '/$modeldir/chemprep/inputdata.211001011.6/BIOGENIC':
  List of biogenic emitted species
- fndepoespe = '/$modeldir/chemprep/inputdata.211001011.6/DEPO_SPEC':
  coefficients for calculation of resistances / species
- fndepopars = '/$modeldir/chemprep/inputdata.211001011.6/DEPO_PARS':
  landuse for the grid
- fnwetd = '/$modeldir/chemprep/inputdata.211001011.6/WETD_SPEC':
  wet deposition species and parameters
- fnveget = '/$modeldir/chemprep/inputdata.211001011.6/VEGETATION':
  vegetation coefficients for the grid
- fnsea = '/$modeldir/chemprep/inputdata.211001011.6/SEA':
  sea over the grid
- fnsoil = '/$modeldir/chemprep/inputdata.211001011.6/SOIL':
  soil coefficients for the grid
- fnemisa = '/$modeldir/$tmpdir/AEMISSIONS.nc'
- fnemisb = '/$modeldir/$tmpdir/BEMISSIONS.nc'
- fnlatconc = '/$modeldir/$tmpdir/LAT_CONCS.nc'
- fntopconc = '/$modeldir/$tmpdir/TOP_CONCS.nc'
- fnmeteo = '/$modeldir/$tmpdir/METEO.nc':
  All meteo input fields in a single file
- fnlanduse = '/$modeldir/domains/CONT5/LANDUSE_CONT5'
- fninit = '$outputdir/ini.$startdate_$enddate_CONT5.nc'
- fniniex = '$outputdir/iex.$startdate_$enddate_CONT5.nc'
- fnout = '$outputdir/out.$startdate_$enddate_CONT5.nc':
  Timely concentrations of selected output species and families
- fnexc = '$outputdir/exc.$startdate_$enddate_CONT5.nc'
- fnothers = '$outputdir/par.$startdate_$enddate_CONT5.nc':
  Timely meteorological parameters
- fnconcs = '$outputdir/end.$startdate_$enddate_CONT5.nc':
  Concentrations of active species at the end of the model run. Generally used as a restart file for subsequent runs.
- fnendex = '$outputdir/exe.$startdate_$enddate_CONT5.nc'
- fndepos = '$outputdir/dep.$startdate_$enddate_CONT5.nc':
  Timely depositions

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

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.
- fndepos Timely depositions

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 prefered 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 **prepemis.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 anthropogniec emissions files with the ewn 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 gives minimum informations in the parameter file $modeldir/domains/domainlist.nml.

```
domain      NX  NY  DX    DY   XMIN    YMIN
CONT1       65  33  0.5   0.5  -10.0   40.5
CONT3       67  46  0.5   0.5  -10.5   35.0
CONT4       70  44  0.5   0.5  -10.5   36.0
CONT5       79  47  0.5   0.5  -14.0   35.0
FRA10      101 111 0.15   0.1   -5.0   41.0
```

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.umiacs.umd.edu/data/landcover

[2] Actually, the resolution of the source files used is 0.08°

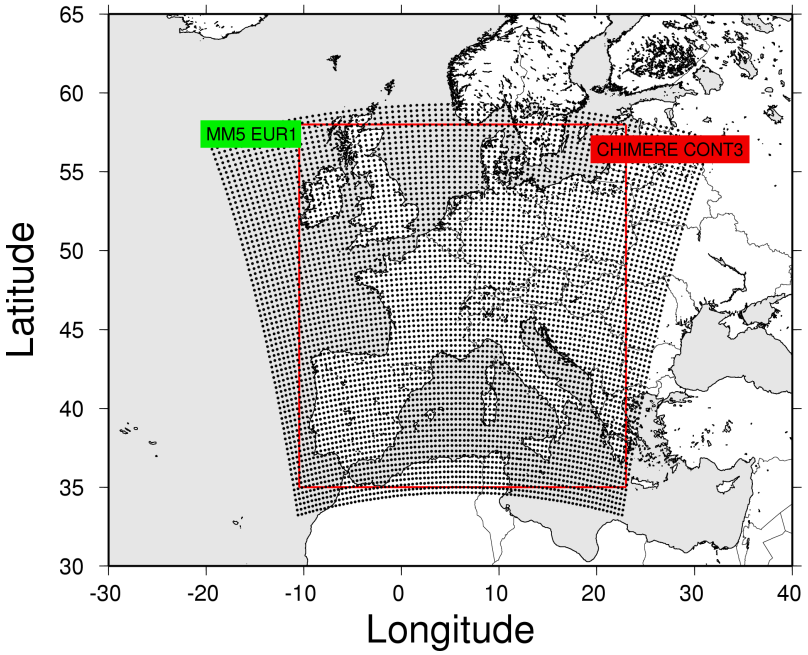


Figure 4.1: *Examples of horizontal domains. The dots represent the MM5 EUR1 grid cells centers and the rectangle represents the boundaries of the CHIMERE CONT3 domain.*

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.

0	Water
1	Evergreen Needleleaf Forest
2	Evergreen Broadleaf Forest
3	Deciduous Needleleaf Forest
4	Deciduous Broadleaf Forest
5	Mixed Forest
6	Woodland
7	Wooded Grassland
8	Closed Shrubland
9	Open Shrubland
10	Grassland
11	Cropland
12	Bare Ground
13	Urban and Built

Table 4.1: *GLCF land cover types*

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 **COORD_<domain>** file which must be placed in the **domains/** directory. The **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 tranpsort 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 absolutly 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, **VCOORD.[vgname]**, in the **data/vgrid/** directory. [vgname] 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 correspondance to the GLCF (University of maryland) data base used for the propoposed **prepdomains** interface.

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

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 coresponding 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 prerequesite files.
- This script will create a new directory in the main directory chemprep and called something like input-data.2110011210.6 where the arguments follwing 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[1]. 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

dust	These dusts are considered as emitted within the domain (mainly resuspension). [0] if not, [1] if yes.
pops	Persistent Organic Pollutants. pops=1 only if mecachim>0 and aero=1
dustout	Dust considered only at the model domain boundaries. dustout=0 is the default value dustout=1 only if aero=1
carb	Primary carbonaceous species [0=no; 1=yes] (if yes, OCAR and BCAR are used)
trc	Specify the number of tracers: trc=0 , no tracer
soatyp	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 **"1000001000.0".**

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_{pmin}=0.001μm and D_{pmax}=40μm. 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 A1(+B1(+C1....)->(a*)A2+(b*)B2+... 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->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.

Type number	Name	Math	syntax in REACTIONS file
1	Constant K	K	k=*value*
2	Simplified Arrhenius	K=Aexp(-B/T)	k(T)=Aexp(-B/T), A=*value*, B=*value*
3	Arrhenius	K=Aexp(-B/T)(300/T)**N	k(T)=Aexp(-B/T)(300/T)**N, A=*value*, B=*value*, N=*value*
4	Troe or Falloff	K=K0*f**E/(1+M*(K0/K_∞)) K0=A0exp(-B0/T)(300/T)**N0 K_∞=A_∞exp(-B_∞/T)(300/T)**N_∞ E=1/(1+(Log10(M*K0/K_∞))**2) M=air-density	K(T,M)=troe(A0,B0,N0,A_∞,B_∞,N_∞,f)
5	Photolysis	J=J(Z)	J(Z)=photorate(*label*)[2].
6	Special type	K=K1*K2/(1+K2) K1=A1exp(-B1/T) K2=A2exp(-B2/T)	k(T)=SPECIAL_1(A1,B1,A2,B2)[3]

[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=K1/(1+K2) K1=A1exp(-B1/T) K2=A2exp(-B2/T)	k(T)=SPECIAL_2(A1,B1,A2,B2)
8	Special type	K=2*sqrt(f1*f2*f3*f4/((1+f3)*(1+f4))) f1=A1exp(-B1/T) f2=A2exp(-B2/T) f3=A3exp(-B3/T) f4=A4exp(-B4/T)	k=SPECIAL_3(A1,B1,A2,B2,A3,B3,A4,B4)
9	Special type	K=2*sqrt(f1*f2)*(1-sqrt(f3*f4))*(1-f4)/(2-f3-f4) f1=A1exp(-B1/T) f2=A2exp(-B2/T) f3=A3exp(-B3/T) f4=A4exp(-B4/T)	k=SPECIAL_3(A1,B1,A2,B2,A3,B3,A4,B4)
10 to 12	Not used [4]		
13	Aggregated ozone photolysis	J=J0*H2O/(H2O+M*(0.02909*exp(70/T) +0.06545*exp(110d0/T))) H2O=water vapor (molec/cm3) M=air density (molec/cm3)	J(T,Z,H2O)=photorate(O3) [5]
14	Modified Troe [6]	Like Type #4 (Troe) but: E=1/(1+((Log10(M*K0/K_∞)-0.12)/1.2)**2)	K(T,M)=troe(A0,B0,N0,A_∞,B_∞,N_∞,f)
15 to 20	Not used [7]		

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: **A1(+A2(+A3(+A4)))->(a1*)B1+(a2*)B2+(a3*)B3...** *Rate expression* where **Ai** are the reactants (max=4), **Bi** the products and **ai** 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 (H2O, N2, O2 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 garanteed because never tested). In that case, however, use the syntax $\pm$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 correpsondence

[4] 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)

[5] Aggregation of the ozone photolysis with reactions with atomic oxygen. The photolysis rates must be those of ozone only, hence the budget reaction is O3->2*OH.

[6] Fall-off specific to NO2+OH+M->HNO3

[7] 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.

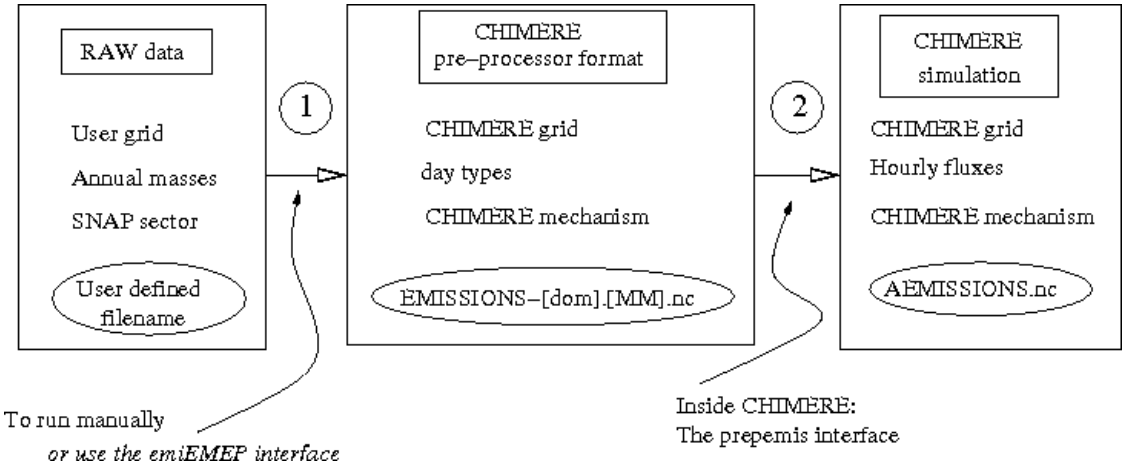


Figure 6.1: *Flowchart of the several steps to produce CHIMERE compliant anthropogenic surface emissions fluxes*

- 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.

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]) the EMEP-MSCWEST web site (http://www.emep.int).
- 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 Liousse, Bruno Guillaume and Robert Rosset, http://www.aero.obs-mip.fr).
- POPs data: Additional data for benzo(a)pyrene, benzo(k)fluoranthene and benzo(b)fluoranthene (respectively BaP, BkF and BbF) can be requested on the EMEP-MSCEAST web site (http://www.msceast.org).

These data are processed alltogether 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, BkF 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** $\times$ **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] 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 :

- NO ..... Nitrogen monoxide
- HONO ..... Nitrous acid
- NH3 ..... Ammoniac
- CH4[1] ..... Methane
- NC4H10 ..... n-Butane
- C2H6 ..... Ethane
- C5H8 ..... Isoprene
- HCHO ..... Formaldehyde
- CH3COE ..... Methyl ethyl Ketone
- PPM_fin ..... Primary particulate matter
- PPM_big ..... Primary particulate matter
- BaP_fin ..... Benzo(a)pyrene
- BkF_fin ..... Benzo(b)fluoranthene
- BCAR_fin ..... Primary black carbon
- NO2 ..... Nitrogen dioxide
- SO2 ..... Sulphur dioxide
- CO ..... Carbon monoxide
- C2H6 ..... Ethane
- C2H4 ..... Ethene
- C3H6 ..... Propene
- OXYL ..... o-Xylene
- CH3CHO ..... Acetaldehyde
- APINEN ..... α-pinene
- PPM_coa ..... Primary particulate matter
- H2SO4_fin ..... Primary sulfuric acid
- BbF_fin ..... Benzo(b)fluoranthene
- OCAR_fin ..... Primary organic carbon

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

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

Some additionnal details:

- PPM_fin reffers to PM2.5, PPM_coa to PM10-PM2.5. H2SO4, BaP, BbF, BkF, OCAR, BCAR are assumed to be in the fine mode.
- In the present version, emissions of Methanol and Ethanol are also included but are 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, following a classification of [Middleton et al., 1990], 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.

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>**:

PIG		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 (^{o}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 adaptated, **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
- APINEN ..... α-pinene
- LIMONE ..... Limonene
- OCIMEN ..... Ocimene (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
- C5H8 ..... Isoprene
- BPINEN ..... β-pinene
- TERPEN ..... Terpene (Lumped class)
- HUMULE ..... Humulene (Lumped class)
- DUST_coa Mineral dust, resuspended matter, soil matter
- 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 :

$$ER_i = EF_i \times \gamma_i(T, PPFD, LAI) \times \rho_i \tag{6.1}$$

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, γ_i (unitless) is an emission activity factor that accounts for deviations from canopy standard conditions, and ρ_i is a factor that accounts for production/loss within canopy. The canopy standard conditions relevant for this study are defined as: air temperature (T) of 303 K, photosynthetic photon flux density (PPFD) of 1500 $\mu mol.m^{-2}.s^{-1}$ at the top of the canopy, leaf area index (LAI) of 5 $m^2.m^{-2}$ and a canopy with 80% mature, 10% growing and 10% old foliage. The MEGAN model parameterizes the bulk effect of changing environmental conditions using three time-dependent input variables specified at top of the canopy: temperature (T), radiation (PPFD), and foliage density (LAI). The production/loss term within canopy is assumed to be unity ($\rho = 1$). The equation can then be expanded as :

$$ER_i = EF_i \times \gamma_{T,i} \times \gamma_{PPFD} \times \gamma_{LAI} \tag{6.2}$$

The MEGAN model provides input EF and LAI data over a global grid, herefater projected on the CHIMERE model grid. The current available choice for EFs is restricted to following species: isoprene, α-pinene, β-pinene, myrcene, sabinene, limonene, δ^3-carene, ocimene, and nitrogen oxide. EF's are static and refer to years 2000-2001. They are obtained summing up over several plant functional types (e.g. broadleaf and needle trees, shrubs, etc...). LAI database is given as a monthly mean product derived from MODIS observations, referred to base year 2000. Hourly emissions are calculated using 2-m temperature and short-wave radiation from a meteorological model output. The optimal choice for this work is the 150" resolution products proposed in the MEGAN inventory.
Terpene and humulene emissions are not calculated in this model version and are set to zero.
Sea salt emissions are calculated by [Monahan, 1986] :

$$\frac{dF}{dr} = 1.373 U_{10}^{3.41} r^{-3} (1 + 0.057 r^{1.05}) 10^{1.19 e^{-B^2}} \tag{6.3}$$

$$B = \frac{0.38 - log(r)}{0.65} \tag{6.4}$$

F is the flux of sea salt particle number in particles m^{-2} s^{-1} μm^{-1}, **r** the particle radius in μm and $\mathbf{U_{10}}$ 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/cm2/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 **makechemgeom.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 **laivyyyymm.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 **LANDUSE_<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 depostion 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 caracteristics 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. ..... Latitude of the source (decimal)
- Lon. ..... Longitude of the source (decimal)
- Beg. date ..... Starting date of the emission (YYYYMMDDHH)

- 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

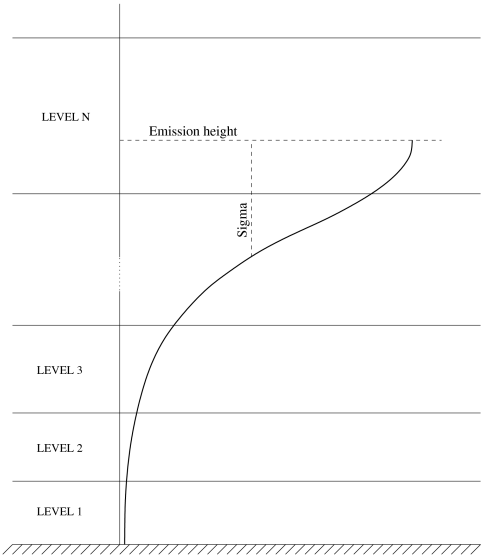


Figure 6.2: *Injection height of the tracer*

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/prepemis.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 modellingwork 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 μg.m^{-3}. We are not aware of any model application for "background dust" typical concentrations like those observed in Europe (typically a few μg.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:

$$F = \alpha C u_{*s} \left(u_{*s}^2 - u_{*t}^2\right) \tag{6.5}$$

where u_{*s} denotes the saltation friction velocity, u_{*t} the threshold friction velocity and C is a coefficient that may depend on several surface factors. The sandblasting efficiency, α; describes the ratio of the vertical flux to the horizontal saltation flux. The saltation friction velocity us 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:

$$u_{*t} = f_w u_{*0} \tag{6.6}$$

where u_{*0} 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

$$\begin{cases} w < w_t & f_w = 1 \\ w > w_t & f_w = \sqrt{1 + 1.21(100(w - w_t))^{0.68}} \end{cases} \tag{6.7}$$

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_{*0}=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]):

$$C_i = \frac{2.61 \rho_{air}}{g} \tag{6.8}$$

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

$$C = f_b f_c C_i \tag{6.9}$$

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 μm mode, 45% of emissions in the 2.510 μm 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 $F_{d,i} = -v_{d,i}c_i$ 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]):

$$v_{d,i} = \frac{1}{R_a + R_{b,i} + R_{c,i}} \tag{6.10}$$

Stable case

$$R_a = \frac{1}{ku_*}\left[ln\left(\frac{z}{z_0}\right) + 4.7(\zeta - \zeta_0)\right] \tag{6.11}$$

Neutral case

$$R_a = \frac{1}{ku_*}\left[ln\left(\frac{z}{z_0}\right)\right] \tag{6.12}$$

Unstable case

$$R_a = \frac{1}{ku_*}\left[ln\left(\frac{z}{z_0}\right) + ln\left(\frac{(\eta_0^2+1)(\eta_0+1)^2}{(\eta_r^2+1)(\eta_r+1)^2}\right) + 2\left(tan^{-1}\eta_r - tan^{-1}\eta_0\right)\right] \tag{6.13}$$

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:

$$R_{b,i} = \frac{2}{ku_*}\left(\frac{Sc_i}{Pr}\right)^{2/3} \tag{6.14}$$

where Sc_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:

```
O3      48      0.01    1       2000
SO2     64      1e5     0       1e-20
NO2     46      0.01    0.1     2000
NO      30      2e-3    0       2000
NH3     17      1e+5    0       1e-20
.....................................
```

These columns are read and used in the **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-laminary 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 :

$$v_d = v_s + \frac{1}{r_a + r_b + r_a r_b v_s} \tag{6.15}$$

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 revertible. 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).
- For **gases** in **rain droplets below the clouds**: Dissolution of gases in precipitating drops is assumed to be irrevertible, both for HNO_3 and NH_3. The scavenging coefficient is expressed as:

$$\Gamma = \frac{pD_g}{6.10^5 u_g D^2}(2 + 0.6R_e^{1/2}S_c^{1/3}) \tag{6.16}$$

  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}), Re and Sc 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 **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:

$$\left[\frac{dQ_l^k}{dt}\right] = -\frac{\epsilon_l P_r}{w_l h} Q_l^k \tag{6.17}$$

  with p_r being the precipitation rate released in the grid cell (g cm^{-2} s^{-1}), w_l the liquid water content (g cm^{-3}), h the cell thickness (cm) and eipsilon 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:

$$\left[\frac{dQ_l^k}{dt}\right] = -\frac{\alpha p E_l}{u_g} Q_l^k \tag{6.18}$$

  with a 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:

$$\lambda = 0.01 \frac{u_*^{1.43}}{\tau^{1.03}} \tag{6.19}$$

where λ is the resuspension rate (in s^{-1}) and τ 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:

$$F = P f(w) u_*^{1.43} \tag{6.20}$$

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 peroxyde
- CO ..... Carbon Monoxide
- CH4 ..... Methane
- HCHO ..... Formaldehyde
- C2H6 ..... Ethane

- NC4H10 . . . . . Butane and higher alkanes (lumped class)
- C2H4 . . . . . Ethene
- C3H6 . . . . . Propene
- OXYL . . . . . Xylene (lumped class)

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.

- O3 . . . . . Ozone
- NO2 . . . . . Nitrogen Dioxide
- PAN . . . Peroxy-Acetyl-Nitrate
- HNO3 . . . . . Nitric Acid
- SO2 . . . . . Sulphur Dioxide
- CO . . . . . Carbon Monoxide
- CH4 . . . . . Methane
- C2H6 . . . . . Ethane
- NC4H10 . . . . . Butane
- C2H4 . . . . . Ethene
- C3H6 . . . . . Propene
- OXYL . . . . . o-Xylene
- C5H8 . . . . . Isoprene
- APINEN . . . . . Alpha-Pinene
- HCHO . . . . . Formaldehyde
- CH3CHO . . . . . Acetaldehyde
- GLYOX . . . . . Glyoxal
- MGLYOX . . . Methyl-Glyoxal
- CH3COE . . . . . Methyl-Ethyl Ketone

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

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
- CSSSM . . . . . Coarse mode soluble sea salt
- AIBCM . . . . . Accumulation mode insoluble black carbon
- ASBCM . . . . . 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 non initialized 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 $\times$ 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.

File Name	Variable	Dimension	Units
Mandatory Variables			
lon	longitude of gridpoints	2D	degrees_east
lat	latitude of gridpoints	2D	degrees_north
winz	Zonal component of the wind	3D	m/s
winm	Meridional component of the wind	3D	m/s
temp	Temperature	3D	K
sphu	Specific humidity	3D	kg/kg
alti	Altitude of half layers on which all 3D meteo variables should be given	3D	m
pres	Pressure at half layers on which all 3D meteo variables should be given	3D	Pa
cliq	3D cloud liquid water content for radiation (excluding rain water)[1]	3D	Kg/Kg
tem2	2m Temperature	2D	K
copc	Convective Precipitation[1]	2D	kg/m^2/hour
lspc	Large-scale Precipitation[1]	2D	kg/m^2/hour
Optional Variables			
rain	3D rain water (to be added to cloud water for radiation)	3D	Kg/Kg
cice	3D ice (for radiation)	3D	Kg/Kg
sshf	Surface sensible heat flux, optionally read or recalculated	2D	W/m2
slhf	Surface latent heat flux, optionally read	2D	W/m2
usta	Friction velocity, optionally read or recalculated from Louis [1982]	2D	m/s
hght	Boundary layer height, optionally read or recalculated	2D	m
lowc	Low cloud fraction, optional for radiation	2D	m
medc	Medium cloud fraction, optional for radiation	2D	m
higc	High cloud fraction, optional for radiation	2D	m

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 Ra. 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/meteo** directory. It has the following structure (example drawn from MM5) :

```
&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.

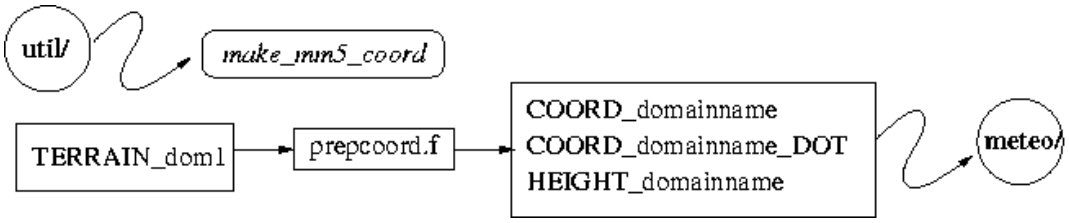


Figure 8.1: *How to prepare MM5 coordinates files for CHIMERE, from the original MM5 terrain file*

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
- RRTF 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 *initvars*. 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' varaible 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 **prepmm5.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 developper 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 stucture 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:

$$\frac{\partial u}{\partial t} + \frac{\partial F_i(u)}{\partial x_i} = 0 \tag{8.1}$$

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:

$$u_n^{t+\Delta t} = u_n^{t+\Delta t} - \left(\frac{F^t_{n+1/2} - F^t_{n-1/2}}{\Delta x} \right) \Delta t \tag{8.2}$$

This equation is exact because it ensures the mass conservation between inside and outside fluxes for each directions. To estimate the value of $F^t_{n+1/2}$, 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:

$$F^t_{n+1/2} = \begin{cases} F^t_n & \text{if } v > 0 \\ F^t_{n+1} & \text{if } v < 0 \end{cases} \tag{8.3}$$

This scheme is clearly asymmetrical. In addition, numerical experiments showed it is very diffusive. To avoid these effects, more complex schemes were developped: 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

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 substracting 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:

$$K_z = k w_s z \left(1 - \frac{z}{h}\right)^{1/3} \tag{8.4}$$

where w_s is a vertical scale given by similarity formulae.

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

where e = max(0.1,z/h), **L** is the Obukhov Length, w_* is the convective velocity scale, u_* the friction velocity and h the boundary layer height. A minimal K(z) is assumed, with a value of 0.01 m^2/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 = $\mathbf{V}_k(\mathbf{C}_{k+1}.\mathbf{D}_k/\mathbf{D}_{k+1} - \mathbf{C}_k)/\mathbf{H}_k$,
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:

$$V_k = \frac{K_z}{(\frac{1}{2}(H_k + H_{k+1}))} \tag{8.5}$$

When the top of the boundary layer is above the model top, H_{k+1} is replaced by h - <model-top> in the previous equation.
The K(z) profile is calculated in routine **src/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 Convection 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.

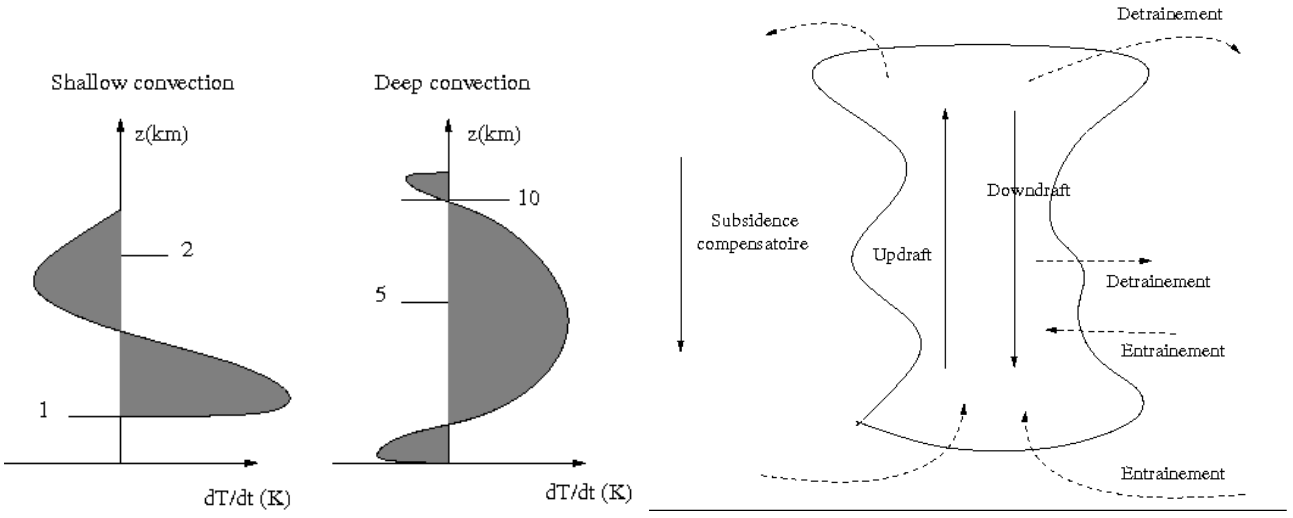


Figure 8.2: *[left] Temperatures changes due to shallow and deep convection and [right] Schematic view of a convective cloud*

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:

$$M(z) = \rho(z)(a_{up}\overline{w_{up}(z)} + a_{dw}\overline{w_{dw}(z)}) = -\rho(z)a_{env}\overline{w_{env}(z)} \tag{8.6}$$

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

$$\frac{\partial M(z)}{\partial z}(z) = E(z) - D(z) \tag{8.7}$$

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:

$$E(z) = E_{up}(z) + E_{dw}(z) \text{ and } D(z) = D_{up}(z) + D_{dw}(z) \tag{8.8}$$

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**.

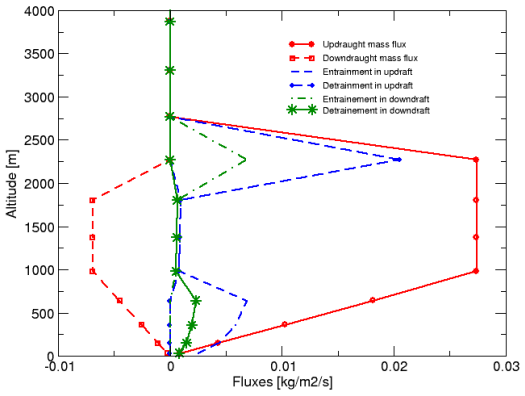


Figure 8.3: *Entrainment and detrainment fluxes in the updrafts and downdrafts.*

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:

$$w\left[\frac{Pa}{s}\right] = \frac{\partial z}{\partial t}\left[\frac{m}{s}\right]\frac{\partial p}{\partial z}\left[\frac{Pa}{s}\right] \tag{8.9}$$

using the hydrostatic relation:

$$\frac{\partial p}{\partial z} = -\rho g \tag{8.10}$$

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

$$\frac{\partial p}{\partial z} = \frac{-gp}{100RT} \tag{8.11}$$

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

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

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:

$$\frac{\partial M_{up}(z)c_{up}(z)}{\partial z} = E_{up}(z)c_{env} - D_{up}(z)c_{up} \tag{8.13}$$

$$\frac{\partial M_{dw}(z)c_{dw}(z)}{\partial z} = E_{dw}(z)c_{env} - D_{dw}(z)c_{dw} \tag{8.14}$$

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**.

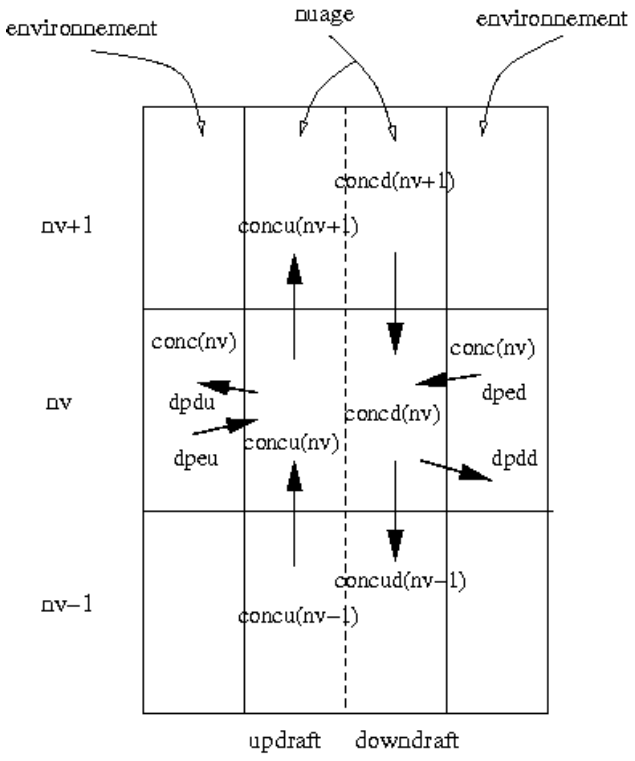


Figure 8.4: *Convective atmopsheric column and its environment*

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 calcualtion is done from surface to top domain in order to ensure mass conservation following the relation:

$$\begin{aligned} FLXU(z) = & \quad \frac{\partial M_{up}(z)}{\partial z}(z) = E_{up}(z) - D_{up}(z) + FLXU(z-1) \\ FLXD(z) = & \quad \frac{\partial M_{dw}(z)}{\partial z}(z) = E_{dw}(z) - D_{dw}(z) + FLXD(z-1) \end{aligned} \tag{8.15}$$

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:

$$\begin{aligned} FLXU(nv) \times concu(nv) \quad = & \; FLXU(nv-1) \times concu(nv-1) \\ & + dpeu(nv) \times conc(nv) \\ & - dpdu(nv) \times concu(nv) \end{aligned} \tag{8.16}$$

and we have:

$$concu(nv) \quad = \frac{FLXU(nv-1) \times concu(nv-1) + dpeu(nv) \times conc(nv)}{dpdu(nv) + FLXU(nv)} \tag{8.17}$$

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:

$$c^{n+1} = \frac{4}{3}c^n - \frac{1}{3}c^{n-1} + \frac{2}{3}\Delta t R(c^{n+1}), \tag{9.1}$$

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

$$c^{n+1} = \left(I + \frac{2}{3}\Delta t L(c^{n+1})\right)^{-1} \left(\frac{4}{3}c^n - \frac{1}{3}c^{n-1} + \frac{2}{3}\Delta t P(c^{n+1})\right). \tag{9.2}$$

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

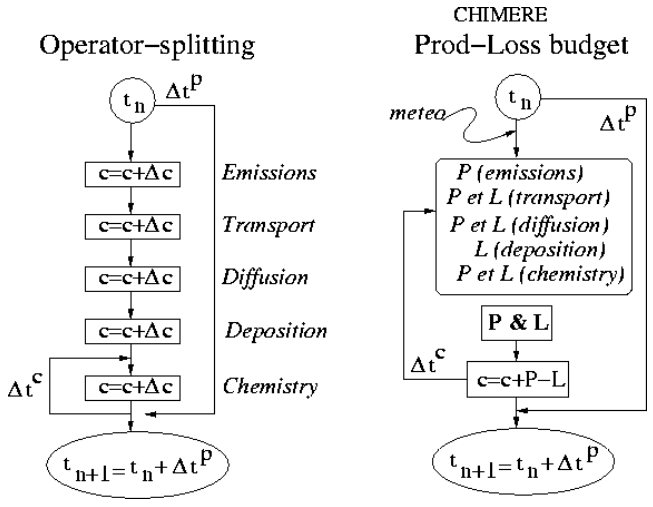


Figure 9.1: *Principle of 'operator-splitting' versus Chimere integration*

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 NO_x conditions and NO_x-nitrate chemistry. All rate constants are updated according to [Atkinson et al., 1997] and [De Moore et al., 1994].

Heterogeneous formation of HONO from deposition of NO2 on wet surfaces is now considered, using the formulation of [Aumont et al., 2003]. 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 mecanisms 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}=14 000m) 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

Species	Definition
Inorganic compounds	
O3	ozone
H2O2	hydrogen peroxide
OH	hydroxy radical
HO2	hydroperoxy radical
NO	nitric oxide
NO2	nitrogen dioxide
NO3	nitrogen trioxide
N2O5	dinitrogen pentoxide
HONO	nitrous acide
HNO3	nitric acide
CO	carbon monoxide
SO2	sulfur dioxide
Hydrocarbon species[1]	
CH4	methane
C2H6	ethane
NC4H10	n-butane
C2H4	ethene
C3H6	propene
OXYL	o-xylene
C5H8	isoprene
APINEN	α-pinene
BPINEN	β-pinene
LIMONE	limonene
TERPEN	terpenes (lumped class)
HUMULE	Humulene (lumped class)
OCIMEN	Ocimene (lumped class)
Carbonyls	
HCHO	formaldehyde
CH3CHO	acetaldehyde
CH3COE	methyl ethyl ketone
GLYOX	glyoxal
MGLYOX	methyl glyoxal
CH3COY	dimethyl glyoxal
MEMALD	unsaturated dicarbonyls, reacting like 4-oxo-2-pentenal
MVK	methyl vinyl ketone
MAC	methacroleine
Organic nitrates	
PAN	peroxyacetyl nitrate
CARNIT	nitrate carbonyl taken as α-nitrooxy acetone
ISNI	unsaturated nitrate from isoprene degradation
Organic peroxides	
CH3O2H	methyl hydroperoxide
PPA	peroxy acetyl acid
(Per)oxy radicals	
CH3O2	methyl peroxy radical
CH3COO	peroxy acetyl radical
Operators [2]	
oRO2	representing peroxy radicals from OH attack to C2H5, NCHH10, C2H4, C3H6, OXYL, CH3COE, MEMALD, and MVK
oROOH	representing organic peroxides from oRO2+HO2 reactions
obio	representing peroxy radicals produced by C5H8 and APINEN + OH reaction
obioH	representing biogenic organic peroxides from obio+HO2 and obio+obio reactions
oPAN	representing PAN homologue compounds (except PAN)
PANH	representing results from oPAN+HO2 reaction
toPAN	representing results from oPAN+NO2 reaction
oRN1	representing organic nitrate peroxy radicals from NO3 attack to C2H4, C3H6, C5H8, APINEN, BPINEN, LIMONE, TERPEN, OCIMEN, HUMULE and OH attack to ISNI

Reaction list of the reduced MELCHIOR2 chemical mechanism

Reactions	Kinetic constants	Ref.	
Inorganic chemistry			
O_3+NO $\rightarrow$ NO_2	$Ae^{-B/T}$,A=1.8e-12,B=1370	[1]	
O_3+NO_2 $\rightarrow$ NO_3	$Ae^{-B/T}$,A=1.2e-13,B=2450	[1]	
O_3+OH $\rightarrow$ HO_2	$Ae^{-B/T}$,A=1.9e-12,B=1000	[1]	
O_3+HO_2 $\rightarrow$ OH	$Ae^{-B/T}$,A=1.4e-14,B=600	[1]	
NO+HO_2 $\rightarrow$ OH+NO_2	$Ae^{-B/T}$,A=3.7e-12,B=-240	[1]	
NO_2+OH+M$\rightarrow$ HNO_3	troe(3.4e-30,0,3.2,4.77e-11,0,1.4,0.30)	[2],	n9
HO_2+OH$\rightarrow$ H_2O	$Ae^{-B/T}$,A=4.8e-11,B=-250	[1]	
H_2O_2+OH$\rightarrow$ HO_2	$Ae^{-B/T}$,A=2.9e-12,B=160	[1]	
HNO_3+OH$\rightarrow$ NO_3	$Ae^{-B/T}$,A=5.5e-15,B=-985	[3],	n1
CO+OH$\rightarrow$ HO_2+CO_2	$Ae^{-B/T}(300/T)^N$, A=2e-13, B=0, N=1	[3],	n1
HO_2+HO_2 $\rightarrow$ H_2O_2	$Ae^{-B/T}$,A=2.2e-13,B=-740	[1],	n1
HO_2+HO_2+$H_2O$$\rightarrow$ H_2O_2	$Ae^{-B/T}$,A=4.52e-34,B=-2827	[1],	n1

[2]For the concept of "chemical operators" see Carter (1990) and Aumont et al. (1997)

Reactions	Kinetic constants	Ref.
NO_3+HO_2→ NO_2+OH	4e-12	[1]
NO_3+H_2O_2 → HNO_3+HO_2	2e-15	[3]
NO_3+NO→ 2*NO_2	$Ae^{-B/T}$,A=1.8e-11,B=-110	[1]
NO_2+NO_3→ NO+NO_2	$Ae^{-B/T}$,A=4.5e-14,B=1260	[3]
NO_2+NO_3+M→ N_2O_5	troe(2.7e-30,0,3.4,2e-12,0,-0.2,0.33)	[1], n9
N_2O_5+M→ NO_3+NO_2	troe(1e-3,11000,3.5,9.7e14,11080,-0.1,0.33)	[1], n9
N_2O_5+H_2O→ 2*HNO_3	2.6e-22	[4]
N_2O_5+H_2O+H_2O→ 2*HNO_3	2e-39	[4]
NO+OH+M→ HONO	troe(7.e-31,0,2.6,1.5e-11,0,0.5,0.6)	[1], n9
HONO+OH→ NO_2	$Ae^{-B/T}$,A=1.8e-11,B=390	[1]
NO+NO+O_2→ 2*NO_2	$Ae^{-B/T}$,A=3.30E-39,B=-530.0	[1]
NO_2→ HONO	0.5*depo(NO_2)	[18]
SOx chemistry		
SO_2+CH_3O_2→ H_2SO_4+HCHO+HO_2	4e-17	[3]
SO_2+OH+M→ H_2SO_4+HO_2	troe(4e-31,0,3.3,2e-12,0,0,0.45)	[3], n9
OH attack to hydrocarbons		
CH_4+OH→ CH_3O_2	$Ae^{-B/T}$,A=2.3e-12,B=1765	[1]
C_2H_6+OH→ CH_3CHO+oRO_2	$Ae^{-B/T}$,A=7.9e-12,B=1030	[1]
NC_4H_{10}+OH→ 0.9*CH_3COE +0.1*CH_3CHO +0.1*CH_3COO +0.9*oRO_2	$Ae^{-B/T}(300/T)^N$,A=1.36e-12, B=-190, N=-2	[5]
C_2H_4+OH+M→ 2*HCHO+oRO_2	troe(7e-29,0,3.1,9e-12,0,0,0.7)	[1], n9
C_3H_6+OH+M→ HCHO+CH_3CHO+oRO_2	troe(8e-27,0,3.5,3e-11,0,0,0.5)	[1], n9
OXYL+OH→ MEMALD+MGLYOX+oRO_2	1.37e-11	[5]
$C5H_8$+OH→ 0.32*MAC +0.42*MVK +0.74*HCHO +obio	$Ae^{-B/T}$, A=2.55e-11, B=-410	[1]
APINEN+OH→ 0.8*CH_3CHO+0.8*CH_3COE+obio	$Ae^{-B/T}$, A=1.21e-11, B=-444	[17]
BPINEN+OH→ 0.8*CH_3CHO+0.8*CH_3COE+obio	$Ae^{-B/T}$, A=1.21e-11, B=-444	[17]
LIMONE+OH→ 0.8*CH_3CHO+0.8*CH_3COE+obio	$Ae^{-B/T}$, A=1.21e-11, B=-444	[17]
HUMULE+OH→ 0.8*CH_3CHO+0.8*CH_3COE+obio	$Ae^{-B/T}$, A=1.21e-11, B=-444	[17]
OCIMEN+OH→ 0.8*CH_3CHO+0.8*CH_3COE+obio	$Ae^{-B/T}$, A=1.21e-11, B=-444	[17]
TERPEN+OH→ 0.8*CH_3CHO+0.8*CH_3COE+obio	$Ae^{-B/T}$, A=1.21e-11, B=-444	[17]
OH attack to carbonyls and peroxides		
HCHO+OH → CO+HO_2	$Ae^{-B/T}$,A=8.6e-12,B=-20	[1]
CH_3CHO+OH →CH_3COO	$Ae^{-B/T}$,A=5.6e-12,B=-310	[1]
MEMALD+OH →GLYOX+MGLYOX+oRO_2	5.6e-11	[1]
CH_3COE+OH →CH_3COY+oRO_2	$Ae^{-B/T}(300/T)^N$, A=2.92e-13, B=-414, N=-2	[1]
GLYOX+OH → 2*CO+HO_2	1.1e-11	[1]
MGLYOX+OH → CH_3COO+CO	1.5e-11	[1]
MVK+OH → 0.266*MGLYOX +0.266*HCHO +0.684*CH_3CHO +0.684*CH_3COO +0.05*ISNI +0.95*oRO_2	$Ae^{-B/T}$,A=4.1e-12,B=-453	[6]
MAC+OH → 0.5*CH_3COE+0.5*CO_2+0.5*oPAN	$Ae^{-B/T}$,A=1.86e-11,B=-175	[6]
CH_3O_2H+OH → CH_3O_2	$Ae^{-B/T}$,A=1.9e-12,B=-190	[1]
PPA+OH → CH_3COO	$Ae^{-B/T}$,A=1.9e-12,B=-190	n2
CH_3O_2H+OH → HCHO+OH	$Ae^{-B/T}$,A=1.e-12,B=-190	[1]
oROOH+OH → 0.8*OH+0.2*oRO_2	$Ae^{-B/T}$,A=4.35e-12,B=-455	n2
obioH+OH → OH	8e-11	n2
PANH+OH → 0.2*oPAN	1.64e-11	n2
CARNIT+OH → CH_3CHO+CO+NO_2	k(T)=$Ae^{-B/T}$,A=5.6e-12,B=-310	
NO_3 attack to hydrocarbons and aldehydes		
C_2H_4+NO_3 → 0.5*CARNIT+HCHO+oRN1	2e-16 [1]	
C_3H_6+NO_3 → 0.5*CARNIT +1.5*HCHO +0.5*CH_3CHO +0.5*HO_2 +oRN1	9.45e-15	[1]

Reactions	Kinetic constants	Ref.
APINEN+NO_3 → CH_3CHO+CH_3COE+oRN1	$Ae^{-B/T}$,A=1.19e-12,B=490	[8]
BPINEN+NO_3 → CH_3CHO+CH_3COE+oRN1	$Ae^{-B/T}$,A=1.19e-12,B=490	[8]
LIMONE+NO_3 → CH_3CHO+CH_3COE+oRN1	$Ae^{-B/T}$,A=1.19e-12,B=490	[8]
OCIMEN+NO_3 → CH_3CHO+CH_3COE+oRN1	$Ae^{-B/T}$,A=1.19e-12,B=490	[8]
HUMULE+NO_3 → CH_3CHO+CH_3COE+oRN1	$Ae^{-B/T}$,A=1.19e-12,B=490	[8]
TERPEN+NO_3 → CH_3CHO+CH_3COE+oRN1	$Ae^{-B/T}$,A=1.19e-12,B=490	[8]
C_5H_8+NO_3 → 0.85*ISNI +0.1*MAC +0.05*MVK +0.15*HCHO +0.8*HO_2 +oRN1	7.8e-13	[6]
HCHO+NO_3 → CO+HNO_3+HO_2	5.8e-16	[1]
CH_3CHO+NO_3 → CH_3COO+HNO_3	2.8e-15	[9]
CH_3O_2+NO_3 → HCHO+HO_2+NO_2	1.2e-12	[10]
CH_3COO+NO_3 → CH_3O_2+NO_2+CO_2	4e-12	[10]
oRO_2+NO_3 → NO_2+HO_2	1.2e-12	[10]
obio+NO_3 → NO_2+HO_2	1.2e-12	[10]
oPAN+NO_3 → NO_2+HO_2	4e-12	[10]
oRN1+NO_3→ 1.5*NO_2	1.2e-12	[10]
O_3 attack to unsaturated hydrocarbons		
C_2H_4+O_3 → HCHO+0.12*HO_2+0.13*H_2+0.44*CO	$Ae^{-B/T}$,A=9.1e-15,B=2580	[1]
C_3H_6+O_3 → 0.53*HCHO +0.5*CH_3CHO +0.31*CH_3O_2 +0.28*HO_2 +0.15*OH +0.065*H_2 +0.4*CO +0.7*CH_4	$Ae^{-B/T}$,A=5.5e-15,B=1880	[1]
C_5H_8+O_3 → 0.67*MAC +0.26*MVK +0.55*OH +0.07*C_3H_6 +0.8*HCHO +0.06*HO_2 +0.05*CO +0.3*O_3	$Ae^{-B/T}$,A=1.2e-14,B=2013	[6]
MAC+O_3 → 0.8*MGLYOX +0.7*HCHO +0.215*OH +0.275*HO_2 +0.2*CO +0.2*O_3	$Ae^{-B/T}$,A=5.3e-15,B=2520	[6]
MVK+O_3 → 0.82*MGLYOX +0.8*HCHO +0.04*CH_3CHO +0.08*OH +0.06*HO_2 +0.05*CO +0.2*O_3	$Ae^{-B/T}$,A=4.3e-15,B=2016	[6]
APINEN+O_3 → 1.27*CH_3CHO +0.53*CH_3COE +0.14*CO +0.62*oRO_2 +0.42*HCHO +0.85*OH +0.1*HO_2	$Ae^{-B/T}$,A=1.0e-15,B=736	[8]
BPINEN+O_3 → 1.27*CH_3CHO +0.53*CH_3COE +0.14*CO +0.62*oRO_2 +0.42*HCHO +0.85*OH +0.1*HO_2	$Ae^{-B/T}$,A=1.0e-15,B=736	[8]
LIMONE+O_3 → 1.27*CH_3CHO +0.53*CH_3COE +0.14*CO +0.62*oRO_2 +0.42*HCHO +0.85*OH +0.1*HO_2	$Ae^{-B/T}$,A=1.0e-15,B=736	[8]
TERPEN+O_3 → 1.27*CH_3CHO +0.53*CH_3COE +0.14*CO +0.62*oRO_2 +0.42*HCHO +0.85*OH +0.1*HO_2	$Ae^{-B/T}$,A=1.0e-15,B=736	[8]
OCIMEN+O_3 → 1.27*CH_3CHO +0.53*CH_3COE +0.14*CO +0.62*oRO_2 +0.42*HCHO +0.85*OH +0.1*HO_2	$Ae^{-B/T}$,A=1.0e-15,B=736	[8]
HUMULE+O_3 → 1.27*CH_3CHO +0.53*CH_3COE +0.14*CO +0.62*oRO_2 +0.42*HCHO +0.85*OH +0.1*HO_2	$Ae^{-B/T}$,A=1.0e-15,B=736	[8]
organic radical conversion		
CH_3O_2+NO → HCHO+NO_2+HO_2	$Ae^{-B/T}$,A=4.2e-12,B=-180	[1]
CH_3COO+NO → CH_3O_2+NO_2+CO_2	2e-11	[1]
oRO_2+NO → NO_2+HO_2	4e-12	n3
obio+NO → 0.86*NO_2+0.78*HO_2+0.14*ISNI	$Ae^{-B/T}$,A=1.4e-11,B=180	[6]
oPAN+NO → NO_2+HO_2	1.4e-11	n3
oRN1+NO → 1.5*NO_2	4e-11	n3
organic radical recombination		
CH_3O_2+HO_2→ CH_3O_2H	$Ae^{-B/T}$,A=4.1e-13,B=-790	[1]
CH_3COO+HO_2→ 0.67*PPA+0.33*O_3	$Ae^{-B/T}$,A=4.3e-13,B=-1040	[1]
oRO_2+HO_2→ oROOH	$Ae^{-B/T}$,A=2.7e-13,B=-1000	n4
obio+HO_2→ obioH	$Ae^{-B/T}$,A=2.7e-13,B=-1000	n4
oPAN+HO_2→ PANH	$Ae^{-B/T}$,A=2.7e-13,B=-1000	n4
oRN1+HO_2→ X	$Ae^{-B/T}$,A=2.7e-13,B=-1000	n4
CH_3O_2+CH_3O_2→ 1.35*HCHO+0.7*HO_2	$Ae^{-B/T}$,A=1.13e-13,B=-356	[1], n5

Reactions	Kinetic constants	Ref.
$CH_3COO+CH_3O_2 \rightarrow 0.5*CH_3O_2+0.5*CO_2+HCHO+0.5*HO_2$	$Ae^{-B/T}$,A=3.34e-12,B=-400	[1], n5
$oRO_2+CH_3O_2 \rightarrow 0.65*HCHO+0.8*HO_2+0.35*CH_3OH$	$Ae^{-B/T}$,A=1.5e-13,B=-220	n5
$obio+CH_3O_2 \rightarrow 0.8*HO_2+0.5*HCHO$	$Ae^{-B/T}$,A=2.44e-11,B=223	n5
$oPAN+CH_3O_2 \rightarrow HCHO+0.5*HO_2$	$Ae^{-B/T}$,A=7.9e-12,B=-140	n5
$CH_3COO+CH_3COO \rightarrow 2.*CH_3O_2+2.*CO_2$	$Ae^{-B/T}$,A=2.8e-12,B=-530	[1], n5
$oRO_2+CH_3COO \rightarrow 0.8*CH_3O_2+0.8*CO_2+0.8*HO_2$	$Ae^{-B/T}$,A=8.6e-13,B=-260	n5
$obio+CH_3COO \rightarrow 0.5*HCHO+1.5*HO_2+0.7*CO_2$	$Ae^{-B/T}$,A=1.18e-11,B=127	n5
$oPAN+CH_3COO \rightarrow CH_3O_2+CO_2+HO_2$	$Ae^{-B/T}$,A=3.34e-12,B=-400	n5
$oRO_2+oRO_2 \rightarrow 1.3*HO_2$	6.4e-14	n5
organic nitrates and pernitrates		
$CH_3COO+NO_2+M \rightarrow PAN$	troe(2.7e-28,0,7.1,1.2e-11,0,0.9,0.3)	[1], n9
$oPAN+NO_2+M \rightarrow toPAN$	troe(2.7e-28,0,7.1,1.2e-11,0,0.9,0.3)	n6, n9
$PAN+M \rightarrow CH_3COO+NO_2$	troe(4.9e-3,12100,0,5.4e16,13830,0,0.3)	[1], n9
$toPAN+M \rightarrow oPAN+NO_2$	troe(4.9e-3,12100,0,5.4e16,13830,0,0.3)	n6, n9
$PAN+OH \rightarrow HCHO+NO_3+CO_2$	$Ae^{-B/T}$,A=9.5e-13,B=650	[1]
$toPAN+OH \rightarrow NO_3+CO_2$	$Ae^{-B/T}$,A=3.25e-13,B=-500	n6
$ISNI+OH \rightarrow 0.95*CH_3CHO+0.475*CH_3COE+0.475*MGLYOX+0.05*ISNI+0.05*HO_2$ + oRN1	3.4e-11	[6]
photolytic reactions		
$O_3 \rightarrow 2*OH$	photorate	[3], n7
$NO_2 \rightarrow NO+O_3$	photorate	[3]
$NO_3 \rightarrow NO_2+O_3$	photorate	[3], [11]
$NO_3 \rightarrow NO$	photorate	[3], [11]
$N_2O_5 \rightarrow NO_2+NO_3$	photorate	[3]
$H_2O_2 \rightarrow 2*OH$	photorate	[3]
$HNO_3 \rightarrow NO_2+OH$	photorate	[3]
$HONO \rightarrow NO+OH$	photorate	[3]
$HCHO \rightarrow CO+2*HO_2$	photorate	[3]
$HCHO \rightarrow CO+H_2$	photorate	[3]
$CH_3CHO \rightarrow CH_3O_2+HO_2+CO$	photorate	[1]
$CH_3COE \rightarrow CH_3COO+CH_3CHO+oRO_2$	photorate	[11], [12]
$CH_3COY \rightarrow 2*CH_3COO$	photorate	[13], [14]
$MGLYOX \rightarrow CH_3COO+HO_2+CO$	photorate	[1]
$GLYOX \rightarrow 0.6*HO_2+2*CO+0.7*H_2$	photorate	[13], [14]
$MEMALD \rightarrow 0.5*MVK+0.5*MALEIC+0.5*oPAN+0.5*HCHO+0.5*HO_2$	photorate	[15]
$CH_3O_2H \rightarrow HCHO+OH+HO_2$	photorate	[3]
$PPA \rightarrow CH_3O_2+CO_2+OH$	photorate	n8
$oROOH \rightarrow OH+HO_2$	photorate	n8
$obioH \rightarrow OH+HO_2$	photorate	n8
$PANH \rightarrow OH+HO_2$	photorate	n8
$PAN \rightarrow CH_3COO+NO_2$	photorate	[3]

- [1] Atkinson et al., 1997
- [2] Donahue et al., 1997
- [3] DeMore et al., 1997
- [4] Mentel et al., 1996
- [5] Atkinson, 1990
- [6] Paulson and Seinfeld, 1992
- [7] Le Bras, 1997
- [8] Atkinson et al., 1994
- [9] DeMore et al., 1990
- [10] Canosa-Mas et al., 1996
- [11] Johnston et al., 1996
- [12] Martinez et al., 1992

- [13] Jenkins et al., 1997
- [14] Plum et al., 1983
- [15] Bierbach et al., 1994
- [16] Lightfoot et al., 1996
- [17] Kirchner and Stockwell, 1996
- [18] Aumont, 2002

- n1 : Rate constants containing a density dependent term in a minor reaction channel have been simplified by putting an average boundary layer molecular density of 2.5*10^{19} molecules/cm^3.
- n2 : For operator reactions with OH, oROOH is interpreted as 2-butyl hydrogen peroxide, obioH as a peroxide from isoprene degradation, PANH represents unsaturated organic peroxides; due to lack of data, the rate constants of these compounds and of peroxy acetyl acid have been estimated using structure reactivity relationships in Kwok and Atkinson (1995).
- n3 : The rate constant of the operator reaction oRO_2+NO has been set to that of the i-C_3H7O_2+NO reaction given in [16], that of oPAN+NO to that of CH_3COO_2+NO [1], the rate constant for oRO_2NO+NO is taken from that of ISNIR (nitrate peroxy radicals from isoprene degradation) + NO [6].
- n4 : The rate constants for the operator reactions oRO_2, obio, RO_2NI+HO_2 have been set to that of the reaction C_2H5O_2+HO_2 [1], that of oPAN+HO_2 has been taken from CH_3COO_2+HO_2 [1].
- n5 : 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_3COO_2.
- n6 : Reaction rates of the operators oPAN and OPAN have been set to those of the peroxy acetyl radical and PAN, respectively.
- n7 : The reactions O_3+hν $\rightarrow$ O(1D), O(1D)+H_2O $\rightarrow$ 2OH, O(1D)+M $\rightarrow$ O(3P)+M and O(3P)+O_2+M $\rightarrow$ O_3+M have been combined to reaction O_3+h$\nu\rightarrow$ O(1D) by taking into account H_2O and M concentrations in order to adjust the photolysis frequency.
- n8 : Due to lack of data, photolysis frequencies of higher organic peroxides are taken as that of CH_3OOH.
- n9 : Three body Troe reactions [1], given in the form:

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

with

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

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

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, heterogenous 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 dl. The aerosol module actually uses six bins from 10 nm to 40 μm, 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 (μg m-3) is the mass concentration of component k in section l and Q_l (μg m-3) is the total mass concentration in section l (Equation 9.4).

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

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

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

Table 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 SO_2. Nitric acid is produced in the gas phase by NOx oxidation. N_2O_5 is converted into nitric acid via heterogeneous pathways by oxidation on aqueous aerosols. Ammonia is a primary emitted base converted in the aerosol phase by neutralization with nitric and sulfuric acids. Ammonia, nitrate and sulfate exist in aqueous,

gaseous and particulate phases in the model. As an example, in the particulate phase the model species pNH_3 represents an equivalent ammonium as the sum of NH_4^+ ion, NH_3 liquid, NH_4NO_3 solid, etc.

9.3.3 Coagulation

Since Q_l^k is the mass concentration of component k in section l, the mass balance equation for coagulation [Gelbard and Seinfeld, 1980] follows Equation 9.5 :

$$\begin{aligned}\left[\frac{dQ_l^k}{dt}\right]_{coag} &= \frac{1}{2}\sum_{i=1}^{l-1}\sum_{j=1}^{l-1}\left[{}^{1a}\beta_{i,j,l}Q_j^kQ_i + {}^{1b}\beta_{i,j,l}Q_i^kQ_j\right] \\ &- \sum_{i=1}^{l-1}\left[{}^{2a}\beta_{i,l}Q_iQ_l^k - {}^{2b}\beta_{i,l}Q_lQ_i^k\right] \\ &- \frac{1}{2}{}^{3}\beta_{l,l}Q_lQ_l^k - Q_l^k\sum_{i=l+1}^{m}{}^{4}\beta_{i,l}Q_i\end{aligned} \tag{9.5}$$

The sectional coagulation coefficients ${}^{1a}\beta, {}^{1b}\beta, {}^{2a}\beta, {}^{2b}\beta, {}^{3}\beta and {}^{4}\beta$ (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:

$$J = \frac{1}{\tau}(G - G_{eq}) \tag{9.6}$$

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

$$\tau = \frac{1 + \frac{8\lambda}{\alpha d}}{2\pi\lambda c d N} \tag{9.7}$$

with λ (m) is the mean free path of air molecules, d (m) the particle diameter, N (particle m^{-3}) the particle number concentration, α 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:

$$\left[\frac{dQ_l^k}{dl}\right]_{abso} = H_l^k Q_l \tag{9.8}$$

$$H_l^k = \frac{12\lambda c_k}{\rho_p d_l^2(1 + (8\lambda/\alpha_k d_l))}(G^k - G_{l,eq}^k) \tag{9.9}$$

where ρ_p is the particle density (fixed at 1500 kgm^{-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]):

$$G_{l,eq}^k = \frac{Q_l^k}{OM_l K_k^p} \tag{9.10}$$

with OM ($\mu g m^{-3}$) is the absorbent organic material concentration. Considering the thermodynamic equilibrium between the gas and particulate phases, this coefficient is given by

$$K_k^p = \frac{10^{-6}}{MW_{om}\zeta_k p_k^0} \tag{9.11}$$

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 ζ the activity coefficient of species in the bulk aerosol phase. The coefficient ζ, difficult to calculate, is assumed constant and equal to unity. Moreover, an empirical formulae can be used to estimate K$_p$:

$$log(K^p) = -0.61 log(p^0) - 4.74 \tag{9.12}$$

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 :	NH_3, HNO_3, HCl, H_2O.
Liquid phase :	NH_4^+, Na^+, H^+, Cl^-, NO_3^-,
	SO_4^{2-}, HSO_4^-, OH^-, H_2O, $HNO_{3(aq)}$, $HCl_{(aq)}$, $NH_{3(aq)}$, $H_2SO^2_{4(aq)}$.
Solid phase :	$(NH_4)_2SO_4$, NH_4HSO_4, $(NH_4)_3H(SO_4)_2$,
	NH_4NO_3, NH_4Cl, NaCl, $NaNO_3$, $NaHSO_4$, Na_2SO_4

Due to its low vapor pressure, sulfuric acid is entirely transfered in the condensed phase. Also, the Sodium is located in the particulate phase. Inorganic aerosols can be a mix of solid and liquid species depending on temperature and humidity. The solid/liquid phase transition is solved by ISORROPIA by computing the deliquescent relative humidities (transition relative humidity between the phases). 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 $\mu.g.m^{-3}$. Because of numerical limits, Sodium and Chloride are not accounted for in this table. The active seasalt version needs the to use the on-line coupling.

Variable	Value		Increment
	Min.	Max.	
Temperature (K)	260	312	2.5
Relative Humidity	0.3	0.99	0.05
H_2SO_4, HNO_3, NH_3 ($\mu.g.m^{-3}$)	10^{-2}	65	x 1,5

Table 9.3: *Characteristics of the look-up table used for the calculation of the thermodynamic equilibrium with ISORROPIA.*

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^{o}x0,5^o) over Europe.

Species	DRH (298.15K)	Mixing	MDRH (298.15K)
$(NH_4)_2SO_4$	0.799		
NH_4NO_3	0.618	NH_4NO_3, $(NH_4)_2SO_4$	0.600
NH_4HSO_4	0.400	$(NH_4)_3H(SO_4)_2$, NH_4HSO_4	0.360
$(NH_4)_3H(SO_4)_2$	0.690	$(NH_4)_3H(SO_4)_2$, $(NH_4)_2SO_4$	0.675

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

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 $\mu.g.m^{-3}$ in average for the nitrates and 0.1 $\mu.g.m^{-3}$ for the sulfates and ammonium. The aerosol water content is slightly decreased (3 to 13% ; 0,1 to 2,8 $\mu.g.m^{-3}$.
The sensitivity tests showed very weak differences, less than $\pm$ 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 NH_4/NH_3 and HNO_3/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]):

- $SO_2^{aq} + O_3^{aq} \rightarrow SO_4^{2-}$
- $HSO_3^- + O_3^{aq} \rightarrow SO_4^{2-}$
- $SO_3^{2-} + O_3^{aq} \rightarrow SO_4^{2-}$
- $SO_2^{aq} + H_2O_2^{aq} \rightarrow SO_4^{2-}$
- $SO_2^{aq} + NO_2^{aq} \rightarrow SO_4^{2-}$
- $SO_3^{2-}(Fe^{3+}) \rightarrow SO_4^{2-}$

- $HSO_3^-(Mn^{2+}) \rightarrow SO_4^{2-}$

SO_2, H_2O_2 and O_3 in the aqueous phase are in equilibrium with the concentrations in the gas phase. Moreover, aqueous SO2 is dissociated into HSO_3^- and SO_3^{2-}. Catalyzed oxidation reactions of sulfur dioxide in aqueous droplets with iron and manganese are considered, following [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]:

- $HO_2 \rightarrow 0.5H_2O_2$ ..... $\gamma = 0.2$
- $NO_3 \rightarrow HNO_3$ ..... $\gamma = 0.001$
- $NO_2 \rightarrow 0.5HNO_3 + 0.5\text{HONO}$ ..... $\gamma = 0.0001$
- $N_2O_5 \rightarrow 2HNO_3$ ..... $\gamma = 0.01$ - 1

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

$$k = \sum_l \left(\frac{d_l}{2D_g} + \frac{4}{V\gamma} \right)^{-1} A_l \tag{9.13}$$

with d_l, the particle diameter (m), D_g, the reacting gas molecular diffusivity (m2 s^{-1}), n, the mean molecular velocity (m s-1), A_l, the total surface area in the particle bin l and gamma the uptake coefficient of reactive species. The uptake coefficient for N_2O_5 is assumed to be temperature-dependent in the range 0.01 - 1 ([De Moore et al., 1994]) with increasing values for decreasing temperatures. 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 (α-pinene and sabinene), BPI (β-pinene and δ^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 Raoult's law. The large number of condensable organic compounds is represented by a set of surrogate compounds that cover the range of physico-chemical properties relevant for aerosol formation, i.e., water solubility and acid dissociation for hydrophilic compounds and saturation vapor pressure for hydrophobic compounds. These surrogate compounds were selected by grouping identified particulate-phase molecular products with similar properties. The molecular weight of each surrogate compound is determined based on its structure and functional groups. The Henry's law constant or the saturation vapor pressure

of the surrogate species is derived from the average properties of the group. Other properties are estimated using the structure of each surrogate compound. Enthalpy of vaporization are given in brackets ($kJ.mol^{-1}$) for each SOA coumpounds : AnA0D (88), AnA1D(88), AnA2D(88), BiA0D(88), BiA1D(88), BiA2D(109), AnBmP(88), AnBlP(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 decribe 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.

$$J_i = \frac{1}{\tau_i}(G_i - G_i^{eq}) \tag{9.14}$$

J_i ($\mu g.m^{-3}.s^{-1}$) is the absorption or desorption flux of species i, τ_i (s) is a characteristic time of the mass transfer that depends on particle size and the chemical properties of species i, G_i is the bulk gas-phase concentration of species i and G_i^{eq} is the gas-phase concentration of species i at thermodynamic equilibrium (i.e., at the surface of the particle). The equilibrium gas-phase concentrations are functions of the particle chemical composition, temperature and, for hydrophilic species, relative humidity, as described by [Pun, 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:

$$A(d) = e^{-0.11d^{2/3}} \tag{9.15}$$

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 **R**, over the cloud depth, of the relative humidity above 75%. It is

Reactions	kinetic rates ($molec.cm^{-3}.s^{-1}$)
TOL+OH → 0.004×AnA0D + 0.001×AnA1D + 0.084×AnBmP + 0.013×AnBlP	1.81×10^{-12}exp(355/T)
TMB+OH → 0.002×AnA0D + 0.002× AnA1D + 0.001×AnA2D + 0.088×AnBmP + 0.006×AnBlP	9.80×10^{-9}/T
NC4H10+OH → 0.07×AnBmP	1.36×10^{-12}exp(190/T)$^{-2}$
API+OH → 0.30×BiA0D + 0.17×BiA1D + 0.10×BiA2D	1.21×10^{-11}exp(444/T)
API+O3 → 0.18×BiA0D + 0.16×BiA1D + 0.05×BiA2D	1.01×10^{-15}exp (-732/T)
API+NO3 → 0.80×BiBmP	1.19×10^{-12}exp(490/T)
BPI+OH → 0.07×BiA0D + 0.08×BiA1D + 0.06×BiA2D	2.38×10^{-11}exp(357/T)
BPI+O3 → 0.09×BiA0D + 0.13×BiA1D + 0.04×BiA2D	1.50×10^{-17}
BPI+NO3 → 0.80×BiBmP	2.51×10^{-12}
LIM+OH → 0.20×BiA0D + 0.25×BiA1D + 0.005×BiA2D	1.71×10^{-10}
LIM+O3 → 0.09×BiA0D + 0.10×BiA1D	2×10^{-16}
OCI+OH → 0.70×BiA0D + 0.075×BiA1D	$5.10\times^{-8}$/T
OCI+O3 → 0.50×BiA0D + 0.055×BiA1D	7.50×10^{-14}/T
OCI+NO3 → 0.70×BiA0D + 0.075×BiA1D	4.30×10^{-9}/T
ISO+OH → 0.232×ISOPA1 + 0.0288×ISOPA2	2.55×10^{-11}exp(410/T)

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 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.*

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.

Recommendations The first option using liquid/ice water (whenever available),is based on 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.

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:

$$\left[\frac{dQ_l^k}{dt}_{incl} = \frac{\zeta_1 p_r}{w_1 h} Q_l^k\right] \tag{9.16}$$

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 ζ 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:

$$\left[\frac{dQ_l^k}{dt}_{subcl} = -\frac{apE_1}{u_g} Q_l^k\right] \tag{9.17}$$

with a 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

[Andreae and Crutzen, 1997] Andreae, M. and Crutzen, P. (1997). Atmospheric aerosols: biogeochemical sources and role in atmospheric chemistry. *Science*, 276:1052–1058.

[Atkinson et al., 1997] Atkinson, R., Baulsch, D. L., Cox, R. A., Hampton, R. F., Kerr, J. A., Rossi, M. J., and Troe, J. (1997). Evaluated kinetics, photochemical and heterogeneous data. *J. Phys. Chem.*, 26(3):521–1012.

[Aumont et al., 2003] Aumont, B., Chervier, F., and Laval, S. (2003). Contribution of HONO to the NOx/HOx/O_3 chemistry in the polluted boundary layer. *Atmos. Environ.*, 37:487–498.

[Baer and Nester, 1992] Baer, M. and Nester, K. (1992). Parameterization of trace gas dry deposition velocities for a regional mesoscale diffusion model. *Ann. Geophys.*, 10:912–923.

[Bagnold, 1941] Bagnold, R. (1941). The physics of blown sand and desert dunes. *Methuen and Co. Ltd. London*.

[Berge, 1993] Berge, E. (1993). Coupling of wet scavenging of sulphur to clouds in a numerical weather prediction model. *Tellus*, 45B:1–22.

[Bessagnet et al., 2005] Bessagnet, B., Hodzic, A., Blanchard, O., Lattuati, M., Le Bihan, O., and Marfaing, H. (2005). Origin of particulate matter pollution episodes in wintertime over the Paris Basin. *Atmos. Environ.*, in press.

[Bowman et al., 1997] Bowman, F. m., Odum, J. R., Seinfeld, J. H., and Pandis, S. N. (1997). Mathematical model for gas-particle parttioning of secondary organic aerosols. *Atmos. Environ.*, 31:3921–3931.

[Carter, 1990] Carter, W. P. L. (1990). A detail mechanism for the gas-phase atmospheric reactions of organic compounds. *Atmos. Environ.*, 24:481–518.

[Cheinet, 2002] Cheinet, S. (2002). *The parameterization of clear and cloudy convective boundary layers*. Ph.D. Thesis, Ecole Polytechnique, Paris, France.

[Clairborn et al., 1998] Clairborn, C., Lamb, B., Miller, A., Beseda, J., Clode, B., Vaughan, J., Kang, L., and Newvine, C. (1998). Regional measurements and modelling of windblown agricultural dust: the columbia plateau pm10 program. *Journal of Geophysical Research*, 103:753–767.

[Colella and Woodward, 1984] Colella, P. and Woodward, P. R. (1984). The piecewise parabolic method (PPM) for gas-dynamical simulations. *Journal of Computational Physics*, 11:38–39.

[De Moore et al., 1994] De Moore, W. B., Sandetr, S. P., Golden, D. M., Hampton, R. F., Kurylo, M. J., Howard, C. J., Ravishankara, A. R., Kolb, C. E., and Molina, M. J. (1994). Chemical kinetics and photochimical data for use in stratospheric modelling evaluation. *JPL publication*, 94, 26, JPL, Pasadena, US.

[Dentener et al., 2006] Dentener, F., Kinne, S., Bond, T., Boucher, O., Cofala, J., Generoso, S., Ginoux, P., Gong, S., Hoelzemann, J., Ito, A., Marelli, L., Penner, J., Putaud, J.-P., Textor, C., Schulz, M., Werf, G. v. d., and Wilson, J. (2006). Emissions of primary aerosol and precursor gases in the years 2000 and 1750 prescribed data-sets for AeroCom. *Atmos. Chem. Phys.*, 6:4321–4344.

[Derognat et al., 2003] Derognat, C., Beekmann, M., Baeumle, M., Martin, D., and Schmidt, H. (2003). Effect of biogenic volatile organic compound emissions on tropospheric chemistry during the Atmospheric Pollution Over the Paris Area(ESQUIF) campaign in the Ile-de-France region. *J. of Geophys. Res.*, 108(D17):8560.

[Erisman et al., 1994] Erisman, J. W., van Pul, A., and Wyers, P. (1994). Parameterization of surface resistance for the quantification of atmospheric deposition of acidifying pollutants and ozone. *Atmos. Env.*, 28:2595–2607.

[Fecan et al., 1999] Fecan, F., Marticorena, B., and Bergametti, G. (1999). Parameterization of the increase of aeolian erosion threshold wind friction velocity due to soil moisture for arid and semi-arid areas. *Annals of Geophysics*, 17.

[Gelbard and Seinfeld, 1980] Gelbard, F. and Seinfeld, J. H. (1980). Simulation of multicomponent aerosol dynamics. *Journal of colloid and Interface Science*, 78:485–501.

[Gelencser et al., 2002] Gelencser, A., Hoffer, A., Krivacsy, Z., Kiss, G., Molnar, A., and Meszaros, E. (2002). On the possible origin of humic matter in fine continental aerosol. *Journal of Geophysical Research*, 107.

[Gillette, 1977] Gillette, D. A. (1977). Fine particulate emissions due to wind-erosion. *Trans. Am. Soc. Agric. Eng.*, 20:890.

[Ginoux et al., 2001] Ginoux, P., Chin, M., Tegen, I., Prospero, J. M., Holben, B., Dubovik, O., and Lin, S.-J. (2001). Sources and distributions of dust aerosols simulated with the GOCART model. *J. Geophys. Res.*, 106:20255–20273.

[Giorgi, 1986] Giorgi, F. (1986). A particle dry deposition scheme for use in tracer transport models‘. *J. Geophys. Res.*, 91:9794–9806.

[Gomes et al., 2003] Gomes, L., Rajot, J. L., Alfaro, S. C., and Gaudichet, A. (2003). Validation of a dust production model from measurements performed in semi-arid agricultural areas of Spain and Niger. *Catena*, 52:257–271.

[Goossens, 2003] Goossens, D. (2003). Effect of soil crustingon the emission and transport of wind-eroded sediment: field measurements on loamy sandy soil. *Geomorphology*, in press.

[Griffin et al., 1999] Griffin, R. J., Cocker, E. R., Flagan, R. C., and Seinfeld, J. H. (1999). Organic aerosol formation from the oxidation of biogenic hydrocarbons. *J. Geophys. Res.*, 104:3555–3567.

[Guelle et al., 1998] Guelle, W., Balkanski, Y. J., Dibb, J. E., Schulz, M., and Dulac, F. (1998). Wet deposition in a global size-dependent aerosol transport model. 2. Influence of the scavenging scheme on Pb vertical profiles, and deposition. *J. Geophys. Res.*, 103(D22):28875–28891.

[Guenther et al., 2006] Guenther, A., Karl, T., Harley, P., Wiedinmyer, C., Palmer, P. I., and Geron, C. (2006). Estimates of global terrestrial isoprene emissions using MEGAN (Model of Emissions of Gases and Aerosols from Nature). *Atmos. Chem. Phys.*, 6:3181–3210.

[Hall et al., 1999] Hall, S., Khudaish, E., and Hart, A. (1999). Electrochemical oxidation of hydrogen peroxide at platinum electrodes. part iii: Effect of temperature. electrochim. *Electrochim. Acta*, 44:2455–2462.

[Hansen and Reed, 2000] Hansen, M. C. and Reed, B. (2000). A comparison of the igbp discover and university of maryland 1 km global land cover products. *International Journal of Remote Sensing*, 21:1365–1373.

[Harrison and Kito, 1990] Harrison, R. and Kito, A. (1990). Field intercomparison of filter pack and denuder sampling methods for reactive gaseous and particulate pollutants. *Atmos. Environ.*, 24:2633–2640.

[Hauglustaine et al., 2004] Hauglustaine, D. A., Hourdin, F., Jourdain, L., Filiberti, M.-A., Walters, S., Lamarque, J.-F., and Holland, E. A. (2004). Interactive chemistry in the Laboratoire de Meteorologie Dynamique general circulation model: Description and background tropospheric chemistry evaluation. *J. Geophys. Res.*, 109(D04314). doi:10.1029/2003JD003,957.

[Hicks et al., 1987] Hicks, B. B., Baldocchi, D. D., Meyers, T. P., Hosker Jr., R. P., and Matt, D. R. (1987). A preliminary multiple resistance routine for deriving dry deposition velocities from measured quantities. *Water, Air, Soil Pollut.*, 36.

[Hoffman and Calvert, 1985] Hoffman, M. R. and Calvert, J. G. (1985). Chemical transformation modules for eulerian acid deposition models,. EPA/600/3-85/017.

[Horowitz et al., 2003] Horowitz, L. W., Walters, S., Mauzeralles, D. Z., Emmonds, L. K., Rash, P. J., Granier, C., Tie, X., Lamarque, J. F., Schultz, M. G., and P., B. J. (2003). A global simulation of tropospheric ozone and related tracers: Description and evaluation of MOZART, version 2. *J. Geophys. Res.*, 108(D24):4784.

[Iversen and White, 1982] Iversen, J. D. and White, B. R. (1982). Saltation threshold on earth, mars and venus. *Sedimentology*, 29:111–119.

[Jacob, 2000] Jacob, D. J. (2000). Heterogeneous chemistry and tropospheric ozone. *Atmos. Environ.*, 34(34):2131–2159.

[Jung et al., 2002] Jung, C., Kim, Y., and Lee, K. (2002). Analytic solution for polydispersed aerosol dynamics by a wet removal process. *Journal of Aerosol Science*, 33:753–767.

[Kaupp and Umlauf, 1992] Kaupp, H. and Umlauf, G. (1992). Atmospheric gas-particle partitioning of organic compounds: a comparison of sampling methods. *Atmospheric Environment*, 26A:2259–2267.

[Kavouras et al., 1998] Kavouras, I., Mihalopoulos, N., and Stephanou, E. (1998). Formation of atmospheric particles from organic acids produced by forests. *Nature*, 395:683–686.

[Kroll et al., 2006] Kroll, J. H., Ng, N. L., Murphy, S. M., Flagan, R. C., and Seinfeld, J. H. (2006). Secondary organic aerosol formation from isoprene photooxidation. *Environ. Sci. Technol.*, 40:1869–1877.

[Kulmala et al., 1998] Kulmala, M., A., L., and Pirjola, L. (1998). Parameterization for sulfuric acid / water nucleation rates. *J. Geophys. Res.*, 103(No D7):8301–8307.

[Kunit and Puxbaum, 1996] Kunit, M. and Puxbaum, H. (1996). Enzymatic determination of the cellulose content of atmospheric aerosols. *Atmospheric Environment*, 30:1233–1236.

[Lattuati, 1997] Lattuati, M. (1997). *Contribution à l'étude du bilan de l'ozone troposphérique à l'interface de l'Europe et de l'Atlantique Nord: modélisation lagrangienne et mesures en altitude*. Thèse de sciences, Université Paris 6, France.

[Lee and Schwartz, 1983] Lee, Y. N. and Schwartz, S. E. (1983). *Precipitation scavenging, dry deposition and resuspension, vol. 1.*, chapter Kinetics of oxidation of aqueous sulfur (IV) by nitrogen dioxide. Elsevier, New York.

[Loosmore, 2003] Loosmore, G. (2003). Evaluation and development of models for resuspension of aerosols at short times after deposition. *Atmospheric Environment*, 37:639–647.

[Loosmore and Cederwall, 2004] Loosmore, G. and Cederwall, R. (2004). Precipitation scavenging of atmospheric aerosols for emergency response applications: testing an updated model with new real-time data. *Atmos. Env.*, 38:993–1003.

[Louis, 1979] Louis, J. (1979). A parametric model of vertical eddy fluxes in the atmosphere. *Boundary Layer Meteorology*, pages 187–202.

[Louis et al., 1982] Louis, J., Tiedke, M., and Geleyn, J. (1982). A short history of the pbl parametrization at ecmwf. *ECMWF Workshop on Planetary Boundary Layer parametrization*, pages 59–80.

[Madronich et al., 1998] Madronich, S., McKenzie, R. E., Bjorn, L. O., and Caldwell, M. M. (1998). Changes in biologically active ultraviolet radiation reaching the earth's surface. *J. Photochem. Photobiol. B:Biology*, 46:5–19.

[Marticorena and Bergametti, 1995] Marticorena, B. and Bergametti, G. (1995). Modelling the atmospheric dust cycle: 1-Design a soil-derived dust emissions scheme. *J. Geophys. Res.*, 100:16415–16430.

[McRae et al., 1982] McRae, G., Goodin, W., and Seinfeld, J. (1982). Development of a second generation mathematical model for urban air pollution: I. model formulation. *Atmospheric Environment*, 16:679–696.

[Middleton et al., 1990] Middleton, P., Stockwell, W. R., and Carter, W. P. (1990). Agregation and analysis of volatile organic compound emissions for regional modelling. *Atmos. Environ.*, 24:1107–1133.

[Mircea and Stefan, 1998] Mircea, M. and Stefan, S. (1998). A theoretical study of the microphysical parameterization of the scavenging coefficient as a function of precipitation type and rate. *Atmos. Env.*, 32:2931–2938.

[Monahan, 1986] Monahan, E. C. (1986). *In The Role of Air-Sea Exchange in Geochemical Cycling*, chapter The ocean as a source of atmospheric particles, pages 129–163. Kluwer Academic Publishers, Dordrecht, Holland.

[Nenes et al., 1998] Nenes, A., Pilinis, C., and Pandis, S. (1998). ISORROPIA: A new thermodynamic model for inorganic multicomponent atmospheric aerosols. *Aquatic Geochem.*, 4:123–152.

[Nickovic et al., 2001] Nickovic, S., Kallos, G., Papadopoulos, A., and Kakliagou, O. (2001). A model for prediction of desert dust cycle in the atmosphere. *Journal of Geophysical Research*, 106:113–129.

[Odum et al., 1997] Odum, J. R., Jungkamp, T. P. W., Griffin, R. J., Forster, H. J. L., Flagan, R. C., and Seinfeld, J. H. (1997). Aromatics, reformulated gasoline and atmospheric organic aerosol formation. *Environ. Sci. Technol.*, 31:1890–1897.

[Pankow, 1994] Pankow, J. F. (1994). An absorption model of gas/aerosol partition involved in the formation of secondary organic aerosol. *Atmos. Environ.*, 28:189–193.

[Peters and Eiden, 1992] Peters, K. and Eiden, R. (1992). Modelling the dry deposition velocity of aerosol particles to a spruce forest. *Atmos. Environ.*, 26A(14):2555–2564.

[Petroff, 2005] Petroff, A. (2005). *Etude modécaniste du dépôt sec d'Aérosols sur les couverts végétaux*. Thèse de sciences, Université Aix-Marseille II, France.

[Priestley and Taylor, 1967] Priestley, C. H. B. and Taylor, R. J. (1967). On the assessment of surface heat flux and evaporation using large-scale parameters. *Monthly Weather Review*, 2:81–93.

[Pun, 2006] Pun, B. K., S. C.-L. K. (2006). Modeling secondary organic aerosol formation via multiphase partitioning with molecular data. *Environ. Sci. Technol.*, 40:4722–4731.

[Putaud et al., 2004] Putaud, J., Raes, F., Van Dingenen, R., Brggemann, E., Facchini, M., Decesari, S., Fuzzi, S., Gehrig, R., Hglin, C., Laj, P., Lorbeer, G., Maenhaut, W., Mihalopoulos, N., Mller, K., Querol, X., Rodriguez, S., Schneider, J., Spindler, G., ten Brink, H., Trseth, K., and Wiedensohler, A. (2004). A European aerosol phenomenology-2: Chemical characteristics of particulate matter at kerbside, urban, rural and background sites in Europe. *Atmos. Environ.*, 38:2579–2595.

[Reeks and Hall, 2001] Reeks, M. and Hall, D. (2001). Kinetic models for particle resuspension in turbulent flows: theory and measurement. *Journal of Aerosol Science*, 32:1–31.

[Sander et al., 1999] Sander, G.C. and. Parlange, J.-Y., Smith, R.E.and Haverkamp, R., and Hogarth, W. (1999). Estimation of ponding time for constant surface flux. *Hydrology days Ed. AGU Pub. 19*, pages 402–410.

[Schulz et al., 2006] Schulz, M., Textor, C., Kinne, S., Balkanski, Y., Bauer, S., Berntsen, T., Berglen, T., Boucher, O., Dentener, F., Guibert, S., Isaksen, I. S. A., Iversen, T., Koch, D., Kirkevå g, A., Liu, X., Montanaro, V., Myhre, G., Penner, J. E., Pitari, G., Reddy, S., Seland, O., Stier, P., and Takemura, T. (2006). Radiative forcing by aerosols as derived from the AeroCom present-day and pre-industrial simulations. *Atmos. Chem. Phys.*, 6:5225–5246.

[Seinfeld and Pandis, 1997] Seinfeld, J. H. and Pandis, S. N. (1997). *Atmospheric chemistry and physics: From air pollution to climate change*. Wiley-Interscience.

[Shao and Lu, 2000] Shao, Y. and Lu, I. (2000). A simple expression for wind erosion threshold friction velocity. *J. Geophys. Res.*, 105:22,437–22,443.

[Simoneit and Mazurek, 1982] Simoneit, B. and Mazurek, M. (1982). Organic matter of the troposphere, ii natural background of biogenic lipid matter in aerosols over the rural western united states. *Atmospheric Environment*, 16:2139–2159.

[Simpson, 1992] Simpson, D. (1992). Long period modeling of photochemical oxidants in Europe, Calculations for July 1985. *Atmos. Environ.*, 26:1609–1634.

[Slinn, 1983] Slinn, W. (1983). Precipitation scavenging. division of biomedical environmental research, us department of energy, washington, dc. *Atmospheric Sciences and Power Production 1979*.

[Society, 1994] Society, E. I. (1994). Generation of European Emission Data for Episodes (GENEMIS) project, EUROTRAC annual report 1993, part 5. Technical report, EUROTRAC, Garmish-Partenkirchen, Germany.

[Stohl et al., 1996] Stohl, A., Williams, E., Wotawa, G., and Kromp-Kolb, H. (1996). A European inventory of soil nitric oxide emissions and the effect of these emissions on the photochemical formation of ozone in Europe. *Atmos. Environ.*, 30:741–3,755. 3.

[Textor et al., 2006] Textor, C., Schulz, M., Kinne, S., Guibert, S., Bauer, S., Berntsen, T., Berglen, T., Boucher, O., Chin, M., Dentener, F., Diehl, T., Feichter, H., Fillmore, D., Ghan, S., Ginoux, P., Gong, S., Grini, A., Hendricks, J., Horrowitz, L., Isaksen, I., Iversen, T., Koch, D., Krol, M., Lauer, A., Lamarque, J.-F., Liu, X., Montanaro, V., Myhre, G., Penner, J., Pitari, G., Reddy, S., Seland, O., Stier, P., Takemura, T., and Tie, X. (2006). Analysis and quantification of the diversities of aerosol life cycles within AeroCom. *Atmos. Chem. Phys.*, 6:1777–1813.

[Tiedtke, 1989] Tiedtke, M. (1989). A comprehensive mass flux scheme for cumulus parameterization in large-scale models. *Mon. Wea. Rev.*, 117:1779–1800.

[Troen and Mahrt, 1986] Troen, I. and Mahrt, L. (1986). A simple model of the atmospheric boundary layer: Sensitivity to surface evaporation. *Bound.-Layer Meteorol.*, 37:129–148.

[Tsyro, 2002] Tsyro, S. (2002). First estimates of the effect of aerosol dynamics in the calculation of PM_{10} and $PM_{2.5}$. *EMEP report*.

[Vautard et al., 2005a] Vautard, R., Bessagnet, B., Chin, M., and Menut, L. (2005a). On the contribution of natural aeolian sources to particulate matter concentrations in Europe: Testing hypotheses with a modelling approach. *Atmos. Environ.*, 39(18):3291–3303.

[Vautard et al., 2005b] Vautard, R., Honoré, C., Beekmann, M., and Rouil, L. (2005b). Simulation of ozone during the august 2003 heat wave and emission control scenarios. *Atmos. Environ.*, 39(16):2957–2967.

[Verwer, 1994] Verwer, J. (1994). Gauss-seidel iterations for stiff odes from chemical kinetics. *SIAM Journal of Scientific Computing*, 15:1243–1250.

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

[Warren, 1986] Warren, D. R. (1986). Nucleation and growth of aerosols.

[Wesely, 1989] Wesely, M. (1989). Parameterization of surface resistances to gaseous dry deposition in regional-scale numerical models. *Atmos. Environ.*, (23):1293–1304.

[White, 1986] White, B. R. (1986). Encyclopedia of fluid mechanics. *Gulf Publishing, Houston, TX, USA*, pages 239–282.

[Xuan, 2004] Xuan, J. (2004). Turbulence factors for threshold velocity and emission rate of atmospheric mineral dust. *Atmospheric Environment*, 38:1777–1783.

[Zender et al., 2003] Zender, C., Bian, H., and Newman, D. (2003). Mineral dust entrainment and deposition (dead) model. description and 1990s dust climatology. *Journal of Geophysical Research*, 108:4416.

[Zhang et al., 2001] Zhang, L., Gong, S., Padro, J., and Barrie, L. (2001). A size-segregated particle dry deposition scheme for an atmospheric aerosol module. *Atmos. Environ.*, 35(3):549–560.

[Zhang et al., 2007] Zhang, Y., Huang, J.-P., Henze, D. K., and Seinfeld, J. H. (2007). Role of isoprene in secondary organic aerosol formation on a regional scale. *J. Geophys. Res.*

Appendix A

References using CHIMERE

1. Pirovano G., I. Coll, M. Bedogni, S. Alessandrini, M. P. Costa, V. Gabusi, F. Lasry, L. Menut, R. Vautard, 2007, On the influence of meteorological input on photochemical modelling of a severe episode over a coastal area, Atmos. Env. in press

2. Menut L., G.Forêt and G.Bergametti, 2007, Sensitivity of mineral dust concentrations to the model size distribution accuracy, Journal of Geophysical Research, Atmospheres

3. Drobinski P., F. Said, G. Ancellet, J. Arteta, P. Augustin, S. Bastin, A. Brut, J.L. Caccia, B. Campistron, S. Cautenet, A. Colette, B. Cros, U. Corsmeier, I. Coll, A. Dabas, H. Delbarre, A. Dufour, P. Durand, V. Guenard, M. Hasel, N. Kalthoff, C. Kottmeier, A. Lemonsu, F. Lohou, V. Masson, L. Menut, C. Moppert, V.H. Peuch, V. Puygrenier, O. Reitebuch, R. Vautard, 2007, Regional transport and dilution during high pollution episodes in southeastern France: Summary of findings from the ESCOMPTE experiment, Journal of Geophysical Research, Atmospheres in press

4. Deguillaume L., M. Beekmann, L. Menut, Bayesian Monte Carlo analysis applied to regional scale inverse emission modelling for reactive trace gases, Journal of Geophysical Research, Atmospheres, Vol. 112, No. D2, D02307. http://dx.doi.org/10.1029/2006JD007518

5. Konovalov I. B., M. Beekmann, A. Richter, J. P. Burrows, 2005, Inverse modelling of the spatial distributions of NOx emissions on a continental scale using satellite data, Atmposheric Chemistry and Physics , 6, 1747-1770, 2006.

6. Cuvelier, C., Thunis, P., Vautard, R., Amann, M., Bessagnet, B., Bedogni, M., Berkowicz, R., Brandt, J., Brocheton, F., Builtjes, P., Coppalle, A., Denby, B., Douros G., Graf, A., Hellmuth, O., Honoré, C., Hodzic, A., Jonson, J., Kerschbaumer, A., de Leeuw, F., Minguzzi, E., Moussiopoulos, N., Pertot, C., Pirovano, G., Rouil, L., Schaap, M., Stern, R., Tarrason, L., Vignati, E., Volta, M., White, L., Wind, P., Zuber, A., 2006: CityDelta: A model intercomparison to explore the impact of emission reductions in 2010 in European cities in 2010, Atmospheric Environment, in press.

7. Vautard, R., Builtjes, P. H. J., Thunis, P., Cuvelier, K., Bedogni, M., Bessagnet, B., Honoré, C., Moussiopoulos, N., Pirovano G., Schaap, M., Stern, R., Tarrason, L., Van Loon, M., 2006: Evaluation and intercomparison of Ozone and PM10 simulations by several chemistry-transport models over 4 european cities within the City-Delta project, Atmospheric Environment, in press.

8. Hodzic A., B.Bessagnet and R.Vautard, A model evaluation of coarse-mode nitrate heterogeneous formation on dust particles, 1006, Atm.Env, 40, 4158-4171

9. Colette A., Ancellet G., L. Menut and S.R.Arnold, A Lagrangian analysis of the impact of transport and transformation on the ozone stratification observed in the free troposphere during the ESCOMPTE campaign, Atmos. Chem. Phys., 6, 3487-3503, 2006

10. Hodzic A., R. Vautard, P. Chazette, L. Menut and B.Bessagnet, Aerosol chemical and optical properties over the Paris area within ESQUIF project, Atmospheric Chemistry and Physics, 2006, pp.3257-3280. SRef-ID: 1680-7324/acp/2006-6-3257

11. Szopa S., D.Hauglustaine, R.Vautard and L. Menut, Future global tropospheric ozone changes and impact on European air quality, Geophysical Research Letter, 2006, 33, L14805, doi:10.1029/2006GL025860

12. Vautard R., S. Szopa, M. Beekmann, L. Menut, D. A. Hauglustaine, L. Rouil, M. Roemer (2006), Are decadal anthropogenic emission reductions in Europe consistent with surface ozone observations?, Geophys. Res. Lett., 33, L13810, doi:10.1029/2006GL026080

13. Pison I., L. Menut and N.Blond, Inverse modeling of emissions for local photo-oxidant pollution: Testing a new methodology with kriging constraints, Annales Geophysicae, 2006, 24, 1523-1535

14. Hodzic A., R. Vautard, H. Chepfer, P. Goloub, L. Menut, P. Chazette, J.L. Deuze, A. Apituley, P. Couvert, Evolution of aerosol optical thickness over Europe during the August 2003 heat wave as seen from CHIMERE model simulations and POLDER data, Atmospheric Chemistry and Physics, 2006, 1853-1864. SRef-ID: 1680-7324/acp/2006-6-1853

15. Blond, N., Boersma, K. F., Eskes, H. J., van der A, R., Van Rosendael, M., De Smedt, I., Bergametti, G., Vautard, R., 2005. Intercomparison of SCHIAMACHY nitrogen dioxide observations, in-situ measurements and air quality modelling results over Western Europe. J. Geophys. Res., in press.

16. Hodzic, A., Vautard, R., Bessagnet, B., Lattuati, M., and F. Moreto, 2005: On the quality of long-term urban particulate matter simulation with the CHIMERE model. Atmospheric Environment, 39, 5851-5864.

17. Bessagnet B., A.Hodzic, O.Blanchard, M.Lattuati, O.LeBihan, H.Marfaing and L.Rouil, 2005, Origin of particulate matter pollution episodes in wintertime over the Paris basin, Atm.Env., 39, 6159-6174.

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

19. Vautard, R., C. Honoré, M. Beekmann, L. Rouil, 2005: Simulation of ozone during the August 2003 heat wave and emission control scenarios, Atmospheric Environment, in press.

20. Vautard R., B.Bessagnet, M.Chin and L. Menut, On the contribution of natural Aeolian sources to particulate matter concentrations in Europe: testing hypotheses with a modelling approach, Atm. Environ., 39, Issue 18, 3291-3303, 2005

21. Monteiro, A., Vautard, R., Borrego, C., and A. I. Miranda, 2005: Long-term simulations of photo oxidant pollution over Portugal using the CHIMERE model, Atmospheric Environment., 39, 3089-3101.

22. Hodzic A., Vautard, R., Bessagnet, B., Lattuati, M., Moreto, F., Long-term urban aerosol simulation versus routine particulate matter observations. Atmospheric Environment, 39, 5851-5864, 2005

23. Vautard R., B.Bessagnet, M.Chin and L.Menut, On the contribution of natural Aeolian sources to particulate matter concentrations in Europe: testing hypotheses with a modelling approach, Atm. Env., 39, Issue 18, 3291-3303, 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.

25. Bessagnet B., A.Hodzic, R.Vautard, M.Beekmann, S.Cheinet, C.Honoré, C.Liousse and L.Rouil, Aerosol modeling with CHIMERE - preliminary evaluation at the continental scale, Atmospheric Environment, 38, 2803-2817, 2004

26. Blond, N. and R.Vautard, Three-dimensional ozone analyses and their use for short-term ozone forecasts, J. Geophys. Res., 109, D17303, doi:10.1029/2004JD004515, 2004.

27. Wackernagel, H., C. Lajaunie, N. Blond, C. Roth and R. Vautard, Geostatistical risk mapping with chemical transport model output and ozone station data, Ecological Modelling, 179, 177-185, 2004.

28. Hodzic A., Chepfer H., Vautard R., Chazette P., Beekmann M., Bessagnet B., Chatenet B., Cuesta J., Drobinski P., Goloub P., Haeffelin M., Morille Y., 2004: Comparisons of aerosol chemistry-transport model simulations with lidar and sun photometer observations at a site near Paris., J. Geophys. Res., 109, D23201,doi:10.1029/2004JD004735, 2004

29. Konovalov, I., Beekmann, M., Vautard, R., Burrows, J., Richter, A., Nuess, H., and N. Nelansky, 2004: The comparison and evaluation of modelled and GOME measurement derived tropospheric NO2 columns over western and eastern Europe Atmospheric Chemistry and Physics, in press.

30. Labouze, E., C. Honoré, L. Moulay, B. Couffignal, and M. Beekmann, 2004. Photochemical ozone creation potentials: a new set of characterization factors for different gas species at the scale of the Western Europe, Int. Journal of Life Cycle Assessment, Vol.9, Nr.3, 187-195.

31. Monteiro, A., Borrego, C., Vautard, R., and A. I. Miranda, 2004: long-term simulations of photooxidant pollution over Portugal using the CHIMERE model. Atmospheric Environment, submitted.

32. Pison I. and L.Menut, Quantification of the impact of aircraft traffic emissions on tropospheric ozone over Paris area, Atmospheric Env., 38-7, 971-983, 2004

33. Vautard, R., C. Honoré, M. Beekmann, L. Rouil, 2004: Simulation of ozone during the August 2003 heat wave and emission control scenarios, Atmospheric Environment, in press.

34. Zhang, B.-N., R. Vautard and N.-T. Kim Oanh, 2004: Comparison of the UAM-V/SAIMM and CHIMERE/ECMWF Photochemical Modeling System Performance in a Tropical Urban Area, J. Air and Waste Management, in press

35. Beekmann M. and C.Derognat, Monte carlo uncertainty analysis of a regional scale transport chemistry model constrained by measurements from the ESQUIF campaign, Journal of Geophysical Research, ESQUIF special section, 2003

36. Blond, N., Bel, L. and Vautard, R., Three-dimensional ozone data analysis with an air quality model over the Paris area, J. Geophys. Res., 108(D23), 4744, doi:10.1029/2003JD003679, 2003.

37. Derognat C., M.Beekmann, H.Schmidt, R.Vautard, D.Martin, B.Neininger, Effect of biogenic VOC emissions on the tropospheric chemistry during elevated ozone periods in Ile-de-france, Journal of Geophysical Research, ESQUIF special section, 2003

38. Menut L., Adjoint modelling for atmospheric pollution processes sensitivity at regional scale during the ESQUIF IOP2, Journal of Geophysical Research, ESQUIF special section, 2003

39. Sillman S., R. Vautard, L. Menut and D. Kley, O3-NOx-VOC sensitivity and NOx-VOC indicators in Paris: results from models and ESQUIF measurements, Journal of Geophysical Research, ESQUIF special section, 2003

40. Schmidt H. and D.Martin, Continental scale adjoint sensitivity of ozone concentrations in the paris region during the ESQUIF episodes, Journal of Geophysical Research, ESQUIF special section, 2003

41. Vautard R., D. Martin, M. Beekmann, R. Friedrich, A. Jaubertie, D. Kley, M. Lattuati, P. Moral, B. Neininger and J. Theloke, Paris emission inventory diagnostic from the ESQUIF airborne measurements and a chemistry-transport model, Journal of Geophysical Research, ESQUIF special section, 2003

42. Vautard R. et al., A synthesis of the ESQUIF field campaign, Journal of Geophysical Research, ESQUIF special section, 2003

43. Schmidt H., C. Derognat, R.Vautard and M.Beekmann, 'A comparison of simulated and observed ozone mixing ratios for the summer of 1998 in western Europe', Atmospheric Environment, 2001

44. Vautard R., M. Beekmann, J. Roux, D. Gombert, 'Validation of a deterministic forecasting system for the ozone concentrations over the Paris area', Atmospheric Environment, 35 2449-2461, 2001.

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

46. Honoré, C., Vautard, R. and Beekmann, M., Photochemical regimes in urban atmospheres: the influence of dispersion. Geophys. Res. Letters, 27, 1895-1898, 2000.

47. Menut L., R.Vautard, M.Beekmann and C.Honoré, 'Sensitivity of Photochemical Pollution using the Adjoint of a Simplified Chemistry-Transport Model', Journal of Geophysical Research - Atmospheres 105, D12, 15,379-15,402, 2000

48. Menut L., R. Vautard, C. Flamant, C. Abonnel, M. Beekmann, P. Chazette, P. H. Flamant, D. Gombert, D. Guedalia, D. Kley, M.P. Lefebvre, B. Lossec, D. Martin, G. Megie, M. Sicard, P. Perros, G. Toupance, 'Atmospheric pollution over the Paris area: The ESQUIF project', Annales Geophysicae, 18, 1467-1481, 2000

49. Vautard R., M.Beekmann and L. Menut, 'Applications of adjoint modelling in atmospheric chemistry: sensitivity and inverse modelling', Environmental Modeling and Sofware, 15, pp 703-709, 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}
chemprepdir=${chimere_root}/chemprep
domaindir=${chimere_root}/domains
export fnboundir=${chemprepdir}/boundaries_spec
export fnbounsel=${fnboundir}/SOURCE_SELECTOR
checkexist ${fnbounsel}
export chemprepfic=${chemprepdir}/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 ${chemprepdir}
   \rm -rf ${chimere_tmp}/chemprep
   mkdir -p ${chimere_tmp}/chemprep
   \cp -f ${chemprepdir}/src/*.e        ${chimere_tmp}/chemprep
   \cp -f ${chemprepdir}/chemprep.*     ${chimere_tmp}/chemprep
   \cp -f ${chemprepdir}/boun_filter.sh ${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 "+++ 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 "+++=============================================================================================+"
     echo "+++ YOU ARE USING TRACERS....                                                                 +"
     echo "+++ Did you check the TRACER and TRACER_PROFILE files?                                        +"
     echo "+++=============================================================================================+"
    fi
    echo "+++=============================================================================================++"
    echo "+++                    RELAUNCH THIS SCRIPT TO START THE SIMULATION                           ++"
    echo "+++=============================================================================================++"
 exit 0
fi

#================ GEOMETRY

echo "  o  GEOMETRY"
   \rm -rf ${chimere_tmp}/domains
   mkdir -p ${chimere_tmp}/domains
```

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

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

 . ${chimere_tmp}/domains/define_geom.sh
 \cp ${chimere_tmp}/domains/VCOORD_${nlayer_chimere}_${first_layer_pressure}_${top_chimere_pressure} ${

# 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} = 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
    ntotal=`wc -l  ${coordfile} | gawk '{print $1}' `
    nyy=`expr ${ntotal} /  ${nxx}`
    export ideepconv=`gawk -f coord-resol.awk -v nxx=${nxx} -v nyy=${nyy} -v coordfile=${coordfile} ${c
    echo "   o deep convection parameterization automatically selected "
    if [ ${ideepconv} = 0 ] ; then
       echo "   o deep convection parameterization automatically not selected "
    fi
 fi
 if [ ${ideepconv} = 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 "        >> continue the input data preprocessing..."
 else
     echo "  o create Biogenic MEGAN files..."
     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
```

```
nzo=`gawk '$1=="'${dom}'" {print $2}' domainlist.nml`
nme=`gawk '$1=="'${dom}'" {print $3}' domainlist.nml`
dx=`gawk '$1=="'${dom}'" {print $4}' domainlist.nml`
dy=`gawk '$1=="'${dom}'" {print $5}' domainlist.nml`
xbotleft=`gawk '$1=="'${dom}'" {print $6}' domainlist.nml`
ybotleft=`gawk '$1=="'${dom}'" {print $7}' domainlist.nml`

touch COORD
\rm COORD

echo "   making horizontal COORD file"
echo "       geometry definition:"
echo "             nodes="${nzo}"x"${nme}
echo "             resol="${dx}"x"${dy}
echo "             south-west point="${xbotleft}"x"${ybotleft}

gawk -v xbotleft=${xbotleft} -v ybotleft=${ybotleft} -v dx=${dx} -v dy=${dy} -v nzo=${nzo} -v nme=${nme
  for(j=1;j<=nme;j++){
  for(i=1;i<=nzo;i++){
     x=(i-1)*dx
     y=(j-1)*dy
     xlon=xbotleft+x
     xlat=ybotleft+y
     print  xlon "  " xlat >> "COORD"
  }
  }
}'

mv COORD COORD_${dom}
cp COORD_${dom} HCOORD
```

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:

```
#!/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

#  Define the parameters below and run the script by just typing "define_geom"

ptop=700    # Pressure at model top
nlay=8      # Number of layers
pre1=995    # Pressure (hPa) of the first layer top when Ps = 1000 hPa
fout=../data/vgrid/VCOORD.8GEOM700

echo ${nlay} ${pre1} ${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

```
#!/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} =
    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
    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}, \
```

```
    ${chemprepdir}/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.defaul
  gawk '{print $0}END{for(j=1;j<='${trc}';j++){for(i=1;i<='${nhours}';i++) {printf("%s ", 1)};printf

  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_DEP
  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"
echo " gas phase  : " ${mecachim}
echo " aerosols  : " ${aero}
echo " sea salt : " ${seasalt}
echo " carbon   : " ${carb}
echo " pops : " ${pops}
echo " dust  : " ${dust}
echo " dust...out! : " ${dustout}
echo " 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
   mv ANTHROPIC tmp3
   cat tmp3 ${chemprepdir}/data/ANTHROPIC.univ >> ANTHROPIC
fi
if [ ${aero} -eq 1 ] ; then
   mv ANTHROPIC tmp3
   cat tmp3 ${chemprepdir}/data/ANTHROPIC.aero >> ANTHROPIC
  if [ ${carb} = "0" ] ;  then
   mv ANTHROPIC tmp3
   cat tmp3 ${chemprepdir}/data/ANTHROPIC.ppm >> ANTHROPIC
  fi
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
   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
   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
   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" ] ;  t

   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 a 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 ${chemprepdir}/data/PHOTO_PARAMETERS.${meca}
\cp ${chemprepdir}/data/PHOTO_PARAMETERS.${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 ${chemprepdir}/data/DEPO_SPEC.univ         DEPO_SPEC
\cp ${chemprepdir}/data/WETD_SPEC.univ         WETD_SPEC
if [ ${trc} -ge 1 ] ;  then
  cat DEPO_SPEC .trc_DEPO_SPEC > .tmp1
  mv .tmp1  DEPO_SPEC
#  cat WETD_SPEC .trc_WETD_SPEC > .tmp1
#  mv .tmp1  WETD_SPEC
fi
mv WETD_SPEC ${chemprepfic}/WETD_SPEC
mv DEPO_SPEC ${chemprepfic}/DEPO_SPEC

#echo "====================================="
#echo "Other parameters: vegetation, deposition etc."
   \cp ${chemprepdir}/data/VEGETATION.univ        ${chemprepfic}/VEGETATION
   \cp ${chemprepdir}/data/SEA.univ               ${chemprepfic}/SEA
   \cp ${chemprepdir}/data/DEPO_PARS.univ         ${chemprepfic}/DEPO_PARS
   \cp ${chemprepdir}/data/SEMIVOL.univ           ${chemprepfic}/SEMIVOL
   \cp ${chemprepdir}/data/SOIL.univ              ${chemprepfic}/SOIL
   \cp ${chemprepdir}/data/OUTPUT_PARAMETERS  ${chemprepfic}/OUTPUT_PARAMETERS
   \cp ${chemprepdir}/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 ${chemprepdir}/boundaries_spec/${boundmodel}_builder.aero.${soa} >> ${boundmodel}_builde
   if [ ${carb} = "0" ] ;  then
      mv ${boundmodel}_builder tmp3
      cat tmp3 ${chemprepdir}/boundaries_spec/${boundmodel}_builder.ppmcarb >> ${boundmodel}_builder
   fi
   if [ ${seasalt} = "0" ] ;  then
      mv ${boundmodel}_builder tmp3
      cat tmp3 ${chemprepdir}/boundaries_spec/${boundmodel}_builder.aero.so4 >> ${boundmodel}_builde
   fi
   if [ ${seasalt} = "2" ] ;  then
      mv ${boundmodel}_builder tmp3
      cat tmp3 ${chemprepdir}/boundaries_spec/${boundmodel}_builder.nacl >> ${boundmodel}_builder
   fi
   if [ ${dustout} = "0" ] ;  then
      if [ ${dust} = "1" ] ;  then
         mv ${boundmodel}_builder tmp3
         cat tmp3 ${chemprepdir}/boundaries_spec/${boundmodel}_builder.dust >> ${boundmodel}_builder
      fi
   fi
fi
if [ ${seasalt} = "1" ] ;  then
 if [ ${aero} = "1" ] ;  then
   mv ${boundmodel}_builder tmp3
   cat tmp3 ${chemprepdir}/boundaries_spec/${boundmodel}_builder.aero.isal >> ${boundmodel}_builder
 fi
 if [ ${aero} = "0" ] ;  then
   mv ${boundmodel}_builder tmp3
   cat tmp3 ${chemprepdir}/boundaries_spec/${boundmodel}_builder.isal >> ${boundmodel}_builder
 fi
fi
if [ ${dustout} = "1" ] ;  then
   mv ${boundmodel}_builder tmp3
   cat tmp3 ${chemprepdir}/boundaries_spec/${boundmodel}_builder.aero.dustout >> ${boundmodel}_builde
fi
if [ ${carb} = "1" ] ;  then
   mv ${boundmodel}_builder tmp3
   cat tmp3 ${chemprepdir}/boundaries_spec/${boundmodel}_builder.carb >> ${boundmodel}_builder
fi
if [ ${dust} = "1" -a ${aero} = "0" -a ${dustout} = "0" ] ;  then
   mv ${boundmodel}_builder tmp3
   cat tmp3 ${chemprepdir}/boundaries_spec/${boundmodel}_builder.dust >> ${boundmodel}_builder
fi

\mv  ${boundmodel}_builder ${chemprepfic}/${boundmodel}_builder
done
if [ ${mecachim} != "0" ] ;  then
#   echo "    Building MOZART parameters file"
   \cp ${chemprepdir}/boundaries_spec/MOZART2_SPEC.${meca}      ${chemprepfic}/MOZART2_SPEC
#   echo "    Building LMDzINCA parameters file"
   \cp ${chemprepdir}/boundaries_spec/LMDZINCA2_SPEC.${meca}    ${chemprepfic}/LMDZINCA2_SPEC
fi
#echo "====================================="
#echo "Output species list"
\rm tmp2 tmp1
touch tmp1
```

```
touch tmp2
\rm tmp
if [ ${mecachim} != "0" ] ;  then
   mv tmp1 tmp
   cat tmp ${chemprepdir}/output_species/OUTPUT_SPECIES.gas.low > tmp1
   mv tmp2 tmp
   cat tmp ${chemprepdir}/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 ${chemprepdir}/output_species/OUTPUT_SPECIES.aero.${soa} > tmp2
   fi
   if [ ${seasalt} = "1" ] ;  then
      mv tmp2 tmp
      cat tmp ${chemprepdir}/output_species/OUTPUT_SPECIES.isal > tmp2
   fi
   if [ ${seasalt} = "2" ] ;  then
      mv tmp2 tmp
      cat tmp ${chemprepdir}/output_species/OUTPUT_SPECIES.nacl > tmp2
   fi
   if [ ${dust} = "1" ] ;  then
      mv tmp2 tmp
      cat tmp ${chemprepdir}/output_species/OUTPUT_SPECIES.dust > tmp2
   fi
   if [ ${carb} = "1" ] ;  then
      mv tmp2 tmp
      cat tmp ${chemprepdir}/output_species/OUTPUT_SPECIES.carb > tmp2
   fi
   if [ ${pops} = "1" ] ;  then
      mv tmp2 tmp
      cat tmp ${chemprepdir}/output_species/OUTPUT_SPECIES.pops > tmp2
   fi
   if [ ${dustout} = "1" ] ;  then
      mv tmp2 tmp
      cat tmp ${chemprepdir}/output_species/OUTPUT_SPECIES.dustout > tmp2
   fi
   mv tmp2 tmp
   cat tmp .OUTPUT_SPECIES.aero > tmp2

cp tmp2 OUTPUT_SPECIES.full

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=${tstemis}.tar.gz
glcf=${glcfn}.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
```

```
[[ "${stat}" = "y" || "${stat}" = "Y" ]] && doinst=1
echo

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

[ ${#bigdir} -eq 0 ] && bigdir=${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 ${glcf} ]] ; then
   echo -n "One or more Chimere archives are found in your current directory. "
   echo -n "Still want to download? (y/[n]) : "
   read stat
   [[ "${stat}" = "y" || "${stat}" = "Y" ]] && downl=1
fi

# 1. Download CHIMERE source code and data

# 1.0. Chimere sources
[[ ! -f ${modf} || ${downl} -eq 1 ]] && { wget http://www.lmd.polytechnique.fr/chimere/downloads/${modf
       { echo "Can't download ${modf}. Bye!"; exit 1; } }

# 1.1. bigfiles: AEROMIN.bin and AEROORG.bin
[[ ! -f ${bigf} || ${downl} -eq 1 ]] && { wget http://www.lmd.polytechnique.fr/chimere/downloads/${bigf
       { echo "Can't download ${bigf}. Bye!"; exit 1; } }

# 1.2. Data for the Heat Wave test case: emissions
[[ ! -f ${tstemisf} || ${downl} -eq 1 ]] && { wget http://www.lmd.polytechnique.fr/chimere/downloads/${
       { echo "Can't download ${tstemisf}. Bye!"; exit 1; } }

# 1.3. Data for the Heat Wave test case: MM5 meteorology
[[ ! -f ${bigdir}/TEST/METEO/${tstmeteof} && ! -f ${bigdir}/TEST/METEO/${tstmeteo} || ${downl} -eq 1 ]]
       { wget -P ${bigdir}/TEST/METEO http://www.lmd.polytechnique.fr/chimere/downloads/${tstmeteof} ||
         { echo "Can't download ${tstmeteof}. Bye!"; exit 1; } }

# 1.4. Emissions interface: the complete emiSURF preprocessor archive
[[ ! -f ${emisf} || ${downl} -eq 1 ]] && { wget http://www.lmd.polytechnique.fr/chimere/downloads/${emi
       { echo "Can't download ${emisf}. Bye!"; exit 1; } }

# 1.5. Boundary conditions
[[ ! -f ${bc_inca} || ${downl} -eq 1 ]] && { wget http://www.lmd.polytechnique.fr/chimere/downloads/${b
```

```
       { echo "Can't download ${bc_inca}. Bye!"; exit 1; } }
[[ ! -f ${bc_goc} || ${downl} -eq 1 ]] && { wget http://www.lmd.polytechnique.fr/chimere/downloads/$
       { echo "Can't download ${bc_goc}. Bye!"; exit 1; } }
[[ ! -f ${bc_aero} || ${downl} -eq 1 ]] && { wget http://www.lmd.polytechnique.fr/chimere/downloads/
       { echo "Can't download ${bc_aero}. Bye!"; exit 1; } }

# 1.6. GLCF land use classes database
[[ ! -f ${glcf} || ${downl} -eq 1 ]] && { wget http://www.lmd.polytechnique.fr/chimere/downloads/${g
       { echo "Can't download ${glcf}. Bye!"; exit 1; } }

# 1.7. MEGAN data
[[ ! -f ${meganf} || ${downl} -eq 1 ]] && { wget http://www.lmd.polytechnique.fr/chimere/downloads/$
       { echo "Can't download ${meganf}. Bye!"; exit 1; } }

# 2. Uncompress the archives

echo "Uncompressing the archives..."
echo "   ${modf}..."
tar xzvf ${modf} > tar_code.lst
echo "   ${bigf}..."
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; } }
echo "   ${tstemisf}..."
ls ${bigdir}/TEST/EMIS/*.nc >/dev/null 2>/dev/null || { tar xzvf ${tstemisf} -C ${bigdir}/TEST/EMIS
       { echo "Can't extract ${tstemisf} to ${bigdir}/TEST/EMIS. Bye!"; exit 1; } }
echo "   ${tstmeteof}..."
[ -f ${bigdir}/TEST/METEO/${tstmeteof} ] && { gunzip ${bigdir}/TEST/METEO/${tstmeteof} >> tar_heatwv
       { echo "Can't gunzip ${bigdir}/TEST/METEO/${tstmeteof}. Bye!"; exit 1; } }
echo "   ${emisf}..."
tar xjvf ${emisf} > tar_emis.lst
echo "   ${bc_inca}..."
ls ${bigdir}/INCA/INCA* >/dev/null 2>/dev/null || { tar xzvf ${bc_inca} -C ${bigdir} > tar_bic.lst |
       { echo "Can't untar ${bc_inca} to ${bigdir}. Bye!"; exit 1; } }
echo "   ${bc_goc}..."
ls ${bigdir}/GOCART/GOCART* >/dev/null 2>/dev/null || { tar xzvf ${bc_goc} -C ${bigdir} >> tar_bic.l
       { echo "Can't untar ${bc_goc} to ${bigdir}. Bye!"; exit 1; } }
echo "   ${bc_aero}..."
ls ${bigdir}/LMDZAERO/INCA-AEROSOL-clim* >/dev/null 2>/dev/null || { tar xjvf ${bc_aero} -C ${bigdir
       { echo "Can't untar ${bc_aero} to ${bigdir}. Bye!"; exit 1; } }

str=`find ${bigdir} -name "*.nc"`
[[ ${#str} -gt 0 && -d ${bigdir}/MEGAN ]] || { tar xjvf ${meganf} -C ${bigdir} >> tar_megan.lst  ||\
       { echo "Can't untar ${meganf} to ${bigdir}. Bye!"; exit 1; } }

echo -n "Uncompress the GLCF database (about 5G)? (y/[n]) : "
read stat
[[ "${stat}" = "y" || "${stat}" = "Y" ]] && { mkdir -p ${bigdir}/data_glcf ; \
       tar xjvf ${glcf} -C ${bigdir}/data_glcf > tar_glcf.lst ||\
       { echo "Can't untar ${glcf} to ${bigdir}. Bye!"; exit 1; } }

# 3. Move the archives to the bigfiles dir
tarlst="${modf} ${tstemisf} ${emisf} ${bc_inca} ${bc_goc} ${bc_aero} ${glcf} ${bigf} ${meganf}"
echo "Moving ${tarlst} to ${bigdir}/ARCH..."
mv -f ${tarlst} ${bigdir}/ARCH

fi # doinst

echo
echo "------------------------------"
echo "   Configuring Makefile.hdr"
echo "------------------------------"
```

```
# 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 you 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 :

```
mkdir /opt/netcdf-3.6.4-pgf90
```

As an ordinary user :

```
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 :

```
make install
```

C.3.4 ifort 64 bit on Intel EMT64 or AMD Opteron

As root :

```
mkdir /opt/netcdf-3.6.4-ifort
```

As an ordinary user :

```
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 :

```
make install
```

C.3.5 g95 64 bit on Intel EMT64 or AMD Opteron

As root :

```
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 :

```
host1.mycluster.mydomain.edu cpu=2
host2.mycluster.mydomain.edu cpu=2
host3.mycluster.mydomain.edu cpu=2
```

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 you 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

```
ssh localhost
```

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 netcdf 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
edit /etc/profile to set the PATH variable
edit /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
edit /etc/profile to set the PATH variable
edit /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='-O2 -m64'
export CXXFLAGS='-O2 -m64'
export LDFLAGS='-O2'
export FC=ifort
export FCFLAGS='-O2 -m64'
export F77=ifort
export FFLAGS='-O2 -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

```
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 = "" ;
}

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) ;
```

```
lon:units = "degrees_east" ;
lon:long_name = "Longitude" ;
float lat(south_north, west_east) ;
lat:units = "degrees_north" ;
lat:long_name = "Latitude" ;
long Sources_PIG(Sources) ;
Sources_PIG:units = "flag-0-1" ;
Sources_PIG: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 Sources_ilon(Sources) ;
Sources_ilon:units = "cell_number" ;
Sources_ilon: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" ;

// global attributes:
:Title = "CHIMERE SUITE" ;
:Sub-title = "Hourly point emission" ;
:Chimere_type = "EMISSIONS" ;
:Generating_process = "Generated by distribemis" ;
:Conventions = "None" ;
:Domain = "CONT5" ;
:Emission_version = "2008b" ;
:history = "" ;
}
```

E.2 exdomout.nc

```
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) ;
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\n" ;
: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) ;
lat:units = "degrees_north" ;
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\n",
                    "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\n";
}
```

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 ;
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 CH3COE(Time, bottom_top, south_north, west_east) ;
CH3COE: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 CH3O2H(Time, bottom_top, south_north, west_east) ;
CH3O2H: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 oRO2(Time, bottom_top, south_north, west_east) ;
oRO2:units = "molecules/cm3" ;
float oROOH(Time, bottom_top, south_north, west_east) ;
oROOH: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 SO2(Time, bottom_top, south_north, west_east) ;
SO2:units = "molecules/cm3" ;
float SOA(Time, bottom_top, south_north, west_east) ;
SOA:units = "molecules/cm3" ;
float SOAAQ(Time, bottom_top, south_north, west_east) ;
SOAAQ:units = "molecules/cm3" ;
float toPAN(Time, bottom_top, south_north, west_east) ;
toPAN: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 HOX(Time, bottom_top, south_north, west_east) ;
HOX: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 pNH3(Time, bottom_top, south_north, west_east) ;
pNH3: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" ;
}
```