ESA/CNES contracts "Mars Environment Models"
WP11.1 Deliverable Report

Ref: ESA 11369/95/NL/JG(SC)
CNES "Base de données atmosphériques martiennes"

[DRAFT] User Manual for the LMD Martian Mesoscale Model

A. Spiga

Laboratoire de Météorologie Dynamique
Institut Pierre Simon Laplace
Université Pierre et Marie Curie
Paris, France

Contact: spiga@lmd.jussieu.fr, forget@lmd.jussieu.fr

December 13, 2008

Contents

Chapter I

Introducing the model

☞ Please first read the document "Design and Performance of the LMD Martian Mesoscale Model" to know what the model is, what kind of results can be obtained and how these results compare with available data or independant simulations

To be completed with description of the dynamics/physics driver

Chapter II

First steps toward running the model

This chapter is meant for first time users of the LMD Martian Mesoscale Model. We describe how to install the model on your system, compile the program and run a test case. Experience with either the terrestrial WRF mesoscale model or the LMD Martian GCM is not absolutely required, although it would help you getting more easily through the installation process.

II.1 Prerequisites

II.1.1 General requirements

In order to install the LMD Martian Mesoscale Model, please ensure that:

- your computer is connected to the internet;
- your OS is Linux[1] with a decent set of basic commmands (`sed`, `awk`, ...);
- your Fortran compiler is the PGI commercial compiler `pgf90` or the GNU free compiler[2] `g95`;
- your C compiler is `gcc` and C development libraries are included;
- `bash`, `m4` and `perl` are installed on your computer;
- `NETCDF` libraries have been compiled *on your system*.

☞ You might also find useful – though not mandatory – to install on your system:

- the `ncview` utility[3], which is a nice tool to visualize the contents of a NETCDF file;
- the `IDL` demo version[4], which is used by the plot utilities provided with the model.

Three environment variables associated with the `NETCDF` libraries must be defined:

```
declare -x NETCDF=/disk/user/netcdf
declare -x NCDFLIB=$NETCDF/lib
declare -x NCDFINC=$NETCDF/inc
```

☞ All command lines in the document are proposed in `bash`.

You also need the environment variable `$LMDMOD` to point at the directory where you will install the model (e.g. `/disk/user/MODELS`):

```
declare -x LMDMOD=/disk/user/MODELS
```

☞ Please check that $\sim$ 200 Mo free disk space is available in `/disk`.

[1] The model was also successfully compiled on MacOSX; "howto" information is available upon request.
[2] Sources and binaries available on `http://www.g95.org`
[3] `http://meteora.ucsd.edu/~pierce/ncview_home_page.html`
[4] `http://www.ittvis.com/ProductServices/IDL.aspx`

II.1.2 Parallel computations

Parallel computations with the Message Passing Interface (MPI) standard are supported by the ARW-WRF mesoscale model. If you want to use this capability in the LMD Martian Mesoscale Model, you would have the installation of MPICH2 as a additional prerequisite.

Please download the current stable version of the sources (e.g. `mpich2-1.0.8.tar.gz`) on the MPICH2 website `http://www.mcs.anl.gov/research/projects/mpich2` and install the MPICH2 utilities by the following commands:

```
mkdir $LMDMOD/MPI
mv mpich2-1.0.8.tar.gz $LMDMOD/MPI
cd $LMDMOD/MPI
tar xzvf mpich2-1.0.8.tar.gz
cd mpich2-1.0.8
./configure --prefix=$PWD --with-device=ch3:nemesis > conf.log 2> conferr.log &
# please wait...
make > mk.log 2> mkerr.log &
declare -x WHERE_MPI=$LMDMOD/MPI/mpich2-1.0.8/bin
```

☞ Even if you add the `$LMDMOD/MPI/mpich2-1.0.8/bin` directory to your `$PATH` variable, defining the environment variable `$WHERE_MPI` is still required to ensure a successful compilation of the model.

II.1.3 Compiling the terrestrial WRF model

The LMD Martian Mesoscale Model is based on the terrestrial NCEP/NCAR ARW-WRF Mesoscale Model. As a first step towards the compilation of the Martian version, we advise you to check that the terrestrial model compiles on your computer with either `g95` or `pgf90`.

On the ARW-WRF website `http://www.mmm.ucar.edu/wrf/users/download/get_source.html`, you will be allowed to freely download the model after a quick registration process (click on "New users"). Make sure to download the version 2.2 of the WRF model and copy the `WRFV2.2.TAR.gz` archive to the `$LMDMOD` folder.

Then please extract the model sources and configure the compilation process:

```
cd $LMDMOD
tar xzvf WRFV2.2.TAR.gz
cd WRFV2
./configure
```

The `configure` script analyzes your architecture and proposes you several possible compilation options. Make sure to choose the "single-threaded, no nesting" option related to either `g95` (should be option 13 on a 32 bits Linux PC) or `pgf90` (should be option 1 on a 32 bits Linux PC).

The next step is to compile the WRF model by choosing the kind of simulations you would like to run. A simple and direct test consists in trying to compile the idealized case of a 2D flow impinging on a small hill:

```
./compile em_hill2d_x > log_compile 2> log_error &
```

☞ In case you encounter problems compiling the ARW-WRF model, please read documentation on the website `http://www.mmm.ucar.edu/wrf/users`, contact the WRF helpdesk or search the web for your error message.

If the compilation was successful (the file `log_error` should be empty or only reporting few warnings), you should find in the `main` folder two executables `ideal.exe` and `run.exe` that would allow you to run the test simulation:

```
cd test/em_hill2d_x
./ideal.exe
./wrf.exe
```

During the simulation, the time taken by the computer to perform integrations at each dynamical timestep is displayed in the standard output. The simulation should end with a message `SUCCESS COMPLETE WRF`. The model results are stored in a `wrfout` data file you might like to browse with a `NETCDF`-compliant software such as `ncview`.

☞ If you compiled the model with `g95`, `ideal.exe` will probably complain about an error reading the namelist. Please move the line `non_hydrostatic` below the line `v_sca_adv_order` in the `namelist.input` file to solve the problem.

II.2 Compiling the Martian model

II.2.1 Extracting and preparing the sources

To start the installation of the Martian mesoscale model, download the archive `LMD_MM_MARS.tar.gz` (click on `http://www.lmd.jussieu.fr/~aslmd/LMD_MM_MARS/LMD_MM_MARS.tar.gz` or use the `wget` command). Copy the sources in the `$LMDMOD` directory and extract the files:

```
cp LMD_MM_MARS.tar.gz $LMDMOD
cd $LMDMOD
tar xzvf LMD_MM_MARS.tar.gz
```

Execute the `prepare` script that would do some necessary preparatory tasks for you: deflate the various compressed archives contained into `LMD_MM_MARS`, download the ARW-WRF sources from the web, apply a (quite significant) "Martian patch" to these sources and build the final structure of your `LMD_MM_MARS` directory:

```
cd $LMDMOD/LMD_MM_MARS
./prepare
```

Please check the contents of the `LMD_MM_MARS` directory:

- seven `bash` scripts: `build_static`, `copy_model`, `makemeso`, `prepare`, `prepare_ini`, `prepare_post`, `save_all`;
- the sources directory `SRC`;
- the static data directory `WPS_GEOG`;
- the simulation utilities directory `SIMU`.

and check that the `LMD_MM_MARS/SRC` directory contains:

- the model main sources in `WRFV2`,
- the preprocessing sources in `WPS` and `PREP_MARS`,
- the postprocessing sources in `ARWpost`,
- three `tar.gz` archives and two information text files.

II.2.2 Main compilation step

In order to compile the model, execute the `makemeso` compilation script in the `LMD_MM_MARS` directory

```
cd $LMDMOD/LMD_MM_MARS
./makemeso
```

and answer to the questions about

1. compiler choice (and number of processors if using MPI)
2. number of grid points in longitude [61]
3. number of grid points in latitude [61]
4. number of vertical levels [61]
5. number of tracers [1]
6. number of domains [1]

☞ On the first time you compile the model, you will probably wonder what to reply to questions 2 to 6 ... type the answers given in brackets to compile an executable suitable for the test case given below.

☞ Suppose you compiled a version of the model for a given set of parameters 1 to 6 to run a specific compilation. If you would like to run another simulation with at least one of parameters 1 to 6 subject to change, the model needs to be recompiled[5] with `makemeso`.

☞ When you use parallel computations, please bear in mind that with 2 (resp. 4, 6, 8, 16) processors the whole domain would be separated into 2 (resp. 2, 3, 4, 4) tiles over the latitude direction and 1 (resp. 2, 2, 2, 4) tile over the longitude direction. Thus make sure that the number of grid points minus 1 in each direction could be divided by the aforementioned number of tiles over the considered direction.

☞ If you use grid nesting, note that no more than 4 processors can be used.

The `makemeso` is an automated script which performs the following serie of tasks:

- determine if the machine is 32 or 64 bits;
- ask the user about the compilation settings;
- create a corresponding directory `$LMDMOD/LMD_MM_MARS/DIRCOMP`;

  ☞ For example, a `DIRCOMP` directory named `g95_32_single` is created if the user requested a `g95` compilation of the code for single-domain simulations on a 32bits machine.

- generate with `copy_model` a directory `DIRCOMP/WRFV2` containing links to `SRC/WRFV2` sources;

  ☞ This method ensures that any change to the model sources would be propagated to all the different `DIRCOMP` installation folders.

- execute the WRF `configure` script with the correct option;
- tweak the resulting `configure.wrf` file to include a link towards the Martian physics;
- calculate the total number of horizontal grid points handled by the LMD physics;
- duplicate LMD physical sources if nesting is activated;

  ☞ The model presently supports 3 nests, but more nests can be included by adaptating the following files:

  ```
  $LMDMOD/LMD_MM_MARS/SRC/WRFV2/call_meso_inifis3.inc
  $LMDMOD/LMD_MM_MARS/SRC/WRFV2/call_meso_physiq3.inc
  $LMDMOD/LMD_MM_MARS/SRC/WRFV2/mars_lmd/libf/duplicate3
  $LMDMOD/LMD_MM_MARS/SRC/WRFV2/mars_lmd/libf/generate3
  $LMDMOD/LMD_MM_MARS/SRC/WRFV2/mars_lmd/makegcm*  ## search for 'nest'
  ```

[5] This necessary recompilation each time the number of grid points, tracers and domains is modified is imposed by the LMD physics code. The WRF dynamical core alone is much more flexible.

- compile the LMD physical packages with the appropriate `makegcm` command and collect the compiled objects in the library `liblmd.a`;

  ☞ During this step that could be a bit long, especially if you defined more than one domain, the `makemeso` script provides you with the full path towards the text file `log_compile_phys` in which you can check for compilation progress and possible errors. In the end of the process, you will find an error message associated to the generation of the final executable. Please do not pay attention to this, as the compilation of the LMD sources is meant to generate a library of compiled objects called `liblmd.a` instead of a program.

- compile the modified Martian ARW-WRF solver, including the `liblmd.a` library;

  ☞ When it is the first time the model is compiled, this step could be quite long. The `makemeso` script provides you with a `log_compile` text file where the progress of the compilation can be checked and a `log_error` text file listing errors and warnings during compilation. A list of warnings related to `grib` utilities (not used in the Martian model) may appear and have no impact on the final executables.

  ☞ The compilation with `g95` might be unsuccessful due to some problems with files related to terrestrial microphysics. Please type the following commands:

  ```
  cd $LMDMOD/LMD_MM_MARS/SRC
  tar xzvf g95.tar.gz
  cp -f g95/WRFV2_g95_fix/* WRFV2/phys/
  cd $LMDMOD/LMD_MM_MARS
  ```

  then recompile the model with the `makemeso` command.

- change the name of the executables in agreements with the settings provided by the user.

  ☞ If you choose to answer to the `makemeso` questions using the aforementioned parameters in brackets, you should have in the `DIRCOMP` directory two executables:

  ```
  real_x61_y61_z61_d1_t1_p1.exe
  wrf_x61_y61_z61_d1_t1_p1.exe
  ```

  The directory also contains a text file in which the answers to the questions are stored, which allows you to re-run the script without the "questions to the user" step:

  ```
  ./makemeso < makemeso_x61_y61_z61_d1_t1_p1
  ```

II.3 Running a simple test case

We suppose that you had successfully compiled the model at the end of the previous section and you had used the answers in brackets to the `makemeso` questions.

In order to test the compiled executables, a ready-to-use test case (with pre-generated initial and boundary conditions) is proposed in the `LMD_MM_MARS_TESTCASE.tar.gz` archive you can download at `http://www.lmd.jussieu.fr/~aslmd/LMD_MM_MARS/LMD_MM_MARS_TESTCASE.tar.gz`. This test case simulates the hydrostatic atmospheric flow around Arsia Mons during half a sol with constant thermal inertia, albedo and dust opacity.

☞ Though the simulation reproduces some reasonable features of the mesoscale circulation around Arsia Mons (e.g. slope winds), it should not be used for scientific purpose, for the number of grid points is unsufficient for single-domain simulation and the integration time is below the necessary spin-up time.

To launch the test simulation, please type the following commands, replacing the `g95_32_single` directory with its corresponding value on your system:

```
cp LMD_MM_MARS_TESTCASE.tar.gz $LMDMOD/LMD_MM_MARS/
tar xzvf LMD_MM_MARS_TESTCASE.tar.gz
cd TESTCASE
ln -sf ../g95_32_single/real_x61_y61_z61_d1_t1_p1.exe wrf.exe
tar xzvf wrfinput.tar.gz
nohup wrf.exe > log_wrf &
```

☞ If you compiled the model using MPICH2, the command to launch a simulation is slightly different:

```
[simulation on 2 processors on 1 machine]
mpd &      # first-time only (or after a reboot)
           # NB: may request the creation of a file .mpd.conf
mpirun -np 8 wrf.exe < /dev/null &      # NB: mpirun is only a link to mpiexec
tail -20 rsl.out.000?     # to check the outputs

[simulation on 16 processors in 4 connected machines]
echo barry.lmd.jussieu.fr > ~/mpd.hosts
echo white.lmd.jussieu.fr >> ~/mpd.hosts
echo loves.lmd.jussieu.fr >> ~/mpd.hosts
echo tapas.lmd.jussieu.fr >> ~/mpd.hosts
ssh barry.lmd.jussieu.fr   # make sure that ssh to other machines
                           # is possible without authentification
mpdboot -f ~/mpd.hosts -n 4
mpdtrace
mpirun -l -np 16 wrf.exe < /dev/null &   # NB: mpirun is only a link to mpiexec
tail -20 rsl.out.00??     # to check the outputs
```

Chapter III

Setting the simulation parameters

In this chapter, we describe how to set the various parameters defining a given simulation. As could be inferred from the content of the `TESTCASE` directory, two parameter files are needed to run the model:

1. The parameters related to the dynamical part of the model can be set in the file `namelist.input` according to the ARW-WRF namelist formatting.
2. The parameters related to the physical part of the model can be set in the file `callphys.def` according to the LMD-MGCM formatting.

III.1 Dynamical settings

`namelist.input` controls the behavior of the dynamical core in the LMD Martian Mesoscale Model. Compared to the file the ARW-WRF users are familiar with[1], the `namelist.input` in the LMD Martian Mesoscale Model is much shorter. The only mandatory parameters in this file are information on time control[2] and domain definition.

The minimal version of the `namelist.input` file corresponds to standard simulations with the model. It is however possible to modify optional parameters if needed, as is the case in the `namelist.input` associated to the Arsia Mons test case (e.g. the parameter `non_hydrostatic` is set to false to assume hydrostatic equilibrium, whereas standard simulations are non-hydrostatic).

A detailed description of the `namelist.input` file is given below[3]. Comments on each of the parameters are provided, with the following labels:

- `(*)` denotes parameters not to be modified,
- `(r)` indicates parameters which modification implies a new recompilation of the model,
- `(n)` describes parameters involved when nested domains are defined,
- `(p1)`, `(p2)`, `(p3)` mention parameters which modification implies a new processing of initial and boundary conditions (see next chapter),
- `(*d)` denotes dynamical parameters which modification implies non-standard simulations – please read `SRC/WRFV2/run/README.namelist` and use with caution.

If omitted, the optional parameters would be set to their default values indicated below.

[1] A description of this file can be found in `SRC/WRFV2/run/README.namelist`.

[2] More information on the adopted Martian calendar: `http://www-mars.lmd.jussieu.fr/mars/time/solar_longitude.html`

[3] You may find the corresponding file in `SIMU/namelist.input_full`.

-- file: namelist.input_full --

```
&time_control
start_year       = 2024,    !! (p1) Start Martian Year (20XX for MY XX)
start_month      = 07,      !! (p1) Start Martian Month
start_day        = 01,      !! (p1) Start Martian Day
start_hour       = 06,      !! (p1) Start Martian Hour (at longitude 0)
end_year         = 2024,    !! (p1) End Martian Year (20XX for MY XX)
end_month        = 07,      !! (p1) End Martian Month
end_day          = 02,      !! (p1) End Martian Day
end_hour         = 06,      !! (p1) End Martian Hour (at longitude 0)
history_interval    = 37,   !! Frequency of outputs (37 --> 3700s = 1 Martian hour)
frames_per_outfile  = 24,   !! Size of time dimension in files
restart          = .false.  !! (*) Output restart files ?
restart_interval = 8880     !! (*) Frequency of output restart files ?
io_form_history  = 2        !! (*) Choice of NETCDF for ouputs
io_form_restart  = 2        !! (*) Choice of NETCDF for ouputs
io_form_input    = 2        !! (*) Choice of NETCDF for ouputs
io_form_boundary = 2        !! (*) Choice of NETCDF for ouputs
debug_level      = 0        !! (*) Verbose level
!!
!! OPTIONAL
!!
interval_seconds = 3700     !! (p2) Frequency of large-scale fields update (s)
input_from_file = T,        !! (n)(p2) Initialize a given domain with an input file
/


&domains
time_step   = 50            !! Dynamical timestep
dx = 20000,                 !! (p2) Horizontal resolution
dy = 20000,                 !! (p2) Horizontal resolution (should be equal to dx)
e_we   = 51,                !! (r)(p2) Number of longitude grid points
e_sn   = 51,                !! (r)(p2) Number of latitude grid points
e_vert = 61,                !! (r)(p2) Number of vertical levels
p_top_requested = 5         !! (p3) Chosen value of pressure at the top of the model
!!
!! OPTIONAL
!!
time_step_fract_num = 0     !! Additional fraction to time_step: numerator
time_step_fract_den = 1     !! Additional fraction to time_step: denominator
num_metgrid_levels = 26     !! (p1) number of vertical levels in GCM inputs (+1)
force_sfc_in_vinterp = 8    !! (p3) Number of levels hardwired in the PBL
                            !!         NB: decrease this parameter when low model top
max_dz = 1500.              !! (p3) Maximal interval (m) between vertical levels
eta_levels = -1.            !! (p3)  Specify a list of e_vert eta levels
max_dom = 1                 !! (r)(n)(p2) Total number of domains
grid_id                = 1, !! (n)(p2) Identification of the domain
parent_id              = 0, !! (n)(p2) Associated parent domain
i_parent_start         = 0, !! (n)(p2) x-position of the bottom-left nest corner
j_parent_start         = 0, !! (n)(p2) x-position of the bottom-left nest corner
parent_grid_ratio      = 1, !! (n)(p2) Ratio of horizontal resolution parent/nest
parent_time_step_ratio = 1, !! (n) Ratio of time step parent/nest
/
```

```
&physics
!!
!! OPTIONAL
!!
radt = 1,                   !! Ratio between physical and dynamical time step
mars =  0,                  !! (r)(p2) Configuration of tracers:
                            !!         0: no tracers, 1: water vapor + ice, 2: dust
init_TI = 0.,               !! (p3) Define constant thermal inertia value
init_AL = 0.,               !! (p3) Define constant albedo value
init_U = 0.,                !! (p3) Define constant ini/bdy zonal wind value
init_V = 0.,                !! (p3) Define constant ini/bdy meridional wind value
init_WX = 0.,               !! (p3) \  Ini/bdy wind profile is everywhere equal to
init_WY = 0.,               !! (p3) /  the wind profile @ grid pt (init_WX,init_WY)
init_MU = 0.,               !! (p3) Multiply ini & bdy zonal wind by init_U
init_MV = 0.,               !! (p3) Multiply ini & bdy meridional wind by init_V
init_LES = F,               !! (p3) Uniform domain initialization for LES
/

&dynamics
!!
!! OPTIONAL
!!
time_step_sound = 6,        !! Ratio of time step dynamic/acoustic integration
                            !!   NB: an increase could help solve instabilities
non_hydrostatic = T,        !! Integrate in non-hydrostatic/hydrostatic mode
pd_scalar = F,              !! Positive-definite advection scheme for tracers
!!
diff_opt = 1                !! (*d) Diffusion option [set to 0 if LES or GCM]
km_opt = 4                  !! (*d) Eddy coefficient option
diff_6th_factor = 2,        !! (*d) Knievel numerical diffusion [set to 0 if LES]
diff_6th_opt = 0.2,         !! (*d) Knievel numerical coeff. [set to 0.5 if GCM]
smdiv = 0.1,                !! (*d) Divergence damping
emdiv = 0.01,               !! (*d) External-mode filter for mass coord. model
epssm = 0.1,                !! (*d) Time off-centering for vertical sound waves
h_mom_adv_order = 5,        !! (*d) Horizontal momentum advection order
v_mom_adv_order = 3,        !! (*d) Vertical momentum advection order
h_sca_adv_order = 5,        !! (*d) Horizontal scalar advection order
v_sca_adv_order = 3,        !! (*d) Vertical scalar advection order
/

&bdy_control
!!
!! OPTIONAL
!!
specified = T,              !! (n)(p3) Boundary conditions specified by GCM
nested = F,                 !! (n)(p3) Boundary conditions from parent domain
periodic_x = F,             !! (p3) Periodic boundary conditions over x
periodic_y = F,             !! (p3) Periodic boundary conditions over y
open_xs = F,                !! (p3) Open boundary conditions @ western boundary
open_xe = F,                !! (p3) Open boundary conditions @ eastern boundary
open_ys = F,                !! (p3) Open boundary conditions @ southern boundary
open_ye = F,                !! (p3) Open boundary conditions @ northern boundary
spec_bdy_width = 5          !! (p3) Width of transition zone with specified=T
                            !!      NB: spec_bdy_width must equal relax_zone+1
relax_zone = 4              !! (p3) Width of relaxation zone with specified=T
/
```

```
&grib2
/

&fdda
/

&namelist_quilt              !! (*)
nio_tasks_per_group = 0,     !! (*)
nio_groups = 1,              !! (*)
/                            !! (*)

                     -- end file:  namelist.input_full --
```

☞ Please pay attention to rigorous syntax while editing your personal `namelist.input` file to avoid reading error.

☞ To modify the default values (or even add personal parameters) in the `namelist.input` file, edit the `SRC/WRFV2/Registry/Registry.EM` file. You will then have to recompile the model with `makemeso` ; answer `y` to the last question.

In case you run simulations with `max_dom` nested domains, you have to set `max_dom` parameters wherever there is a "," in the above list. Here is an example of the resulting syntax of the `time_control`, `domains` and `bdy_control` categories in `namelist.input`:

```
&time_control
start_year       = 2024, 2024, 2024,
start_month      =   01,   01,   01,
start_day        =   49,   49,   49,
start_hour       =   06,   06,   06,
end_year         = 2024, 2024, 2024,
end_month        =   01,   01,   01,
end_day          =   54,   54,   54,
end_hour         =   06,   06,   06,
history_interval   = 74,   74,   74,
frames_per_outfile = 12,   12,   12,
restart          = .false.
restart_interval = 8880,
io_form_history  = 2
io_form_restart  = 2
io_form_input    = 2
io_form_boundary = 2
debug_level      = 0
input_from_file  = T, T, T,
/



&domains
time_step   = 40
dx     = 36000,  9000,  2250,
dy     = 36000,  9000,  2250,
e_we   =   153,   157,   157,
e_sn   =   153,   157,   157,
e_vert =    61,    61,    61,
p_top_requested = 3
```

```
 max_dom = 3
 grid_id                = 1,   2,   3,
 parent_id              = 0,   1,   2,
 i_parent_start         = 0,  57,  57,
 j_parent_start         = 0,  57,  57,
 parent_grid_ratio      = 1,   4,   4,
 parent_time_step_ratio = 1,   2,   2,
 /

 &bdy_control
 specified = T, F, F
 nested    = F, T, T
 /
```

III.2 Physical settings

`callphys.def` controls the behavior of the physical parameterizations in the LMD Martian Mesoscale Model. The organization of this file is exactly similar to the corresponding file in the LMD Martian GCM, which user manual can be found at `http://web.lmd.jussieu.fr/~forget/datagcm/user_manual.pdf`.

Please find in what follows the contents of `callphys.def`:

```
                          -- file:  callphys.def --


General options
~~~~~~~~~~~~~~~
tracer    (Run with or without tracer transport ?)
F
diurnal   (Diurnal cycle ?  if diurnal=F, diurnal averaged solar heating)
T
season    (Seasonal cycle ? if season=F, Ls stays constant like in "start")
T
lwrite    (want some more output on the screen ?)
F
stats     (Saving statistics in file "cumul" ?)
F
calleofdump (Saving EOF profiles in file "profiles" for Climate Database ?)
F
Dust scenario. Used if the dust is prescribed (i.e. if tracer=F or active=F)
~~~~~~~~~~~~~
iaervar  (=1 Dust opt.deph read in startfi; =2 Viking scenario; =3 MGS scenario
4        (=4 Mars Year 24 from TES assimilation)
iddist  (Dust vertical distribution: =0: old distrib. (Pollack90)
3       (=1: top set by "topdustref"; =2: Viking scenario; =3 MGS scenario )
topdustref (Dust top altitude (km). Matter only if iddist=1)
55.
Physical Parameterizations :
~~~~~~~~~~~~~~~~~~~~~~~~~~
callrad   (call radiative transfer ?)
T
callnlte (call NLTE radiative schemes ?   matter only if callrad=T)
F
callnirco2 (call CO2 NIR absorption ?   matter only if callrad=T)
T
```

```
calldifv  (call turbulent vertical diffusion ?)
T
calladj   (call convective adjustment ?)
F
callcond  (call CO2 condensation ?)
T
callsoil  (call thermal conduction in the soil ?)
T
calllott  (call Lott's gravity wave/subgrid topography scheme ?)
F
Radiative transfer options :
~~~~~~~~~~~~~~~~~~~~~~~~~~
iradia    (the rad.transfer is computed every "iradia" physical timestep)
37
callg2d   (Output of the exchange coefficient mattrix ? for diagnostic only)
F
rayleigh  (Rayleigh scattering : should be =F for now)
F
Tracer (dust water, ice and/or chemical species) options (use if tracer=T) :
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
dustbin    (DUST: Transported dust ? (if >0, uses q(1) to q(dustbin))
0
active     (DUST: Radiatively active dust ? (uses q(1) to q(dustbin))
F
doubleq    (DUST: needs dustbin=1, use mass q(1) and nb q(2) mr to predict dust size ?)
F
lifting    (DUST: lifted by GCM surface winds ?)
F
dustdevil  (DUST: lifted by dust devils ?)
F
scavenging (DUST: Scavenging by CO2 snowfall ?)
F
sedimentation (DUST/WATERICE: Gravitationnal sedimentation ?)
F
iceparty   (WATERICE: Water cycle includes water ice mixing ratio q(nqmx-1))
F
activice   (WATERICE: Radiatively active transported atmospheric water ice ?)
F
water      (WATER: Compute water cycle using q(nqmx) )
F
caps       (WATER: put the current permanent caps at both poles)
F
photochem  (PHOTOCHEMISTRY: chemical species included)
F
Thermospheric options (relevant if tracer=T) :
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
calthermos  (call thermosphere ?)
F
thermoswater  (WATER: included without cycle only if water=F)
F
callconduct  (call thermal conduction ?     matter only if calthermos=T)
F
calleuv  (call EUV heating ?                matter only if calthermos=T)
F
callmolvis  (call molecular viscosity ?     matter only if calthermos=T)
F
```

```
callmoldiff  (call molecular diffusion ?    matter only if callthermos=T)
F
thermochem  (call thermospheric photochemistry ?  matter only if callthermos=T)
F
solarcondate (date for solar flux calculation: 1985 < date < 2002))
1993.4       (Solar min=1996.4 ave=1993.4 max=1990.6)



                        -- end file:  callphys.def --
```

- ☞ Note that in the given example the convective adjustment, the gravity wave parameterization, and the NLTE schemes are turned off, as is usually the case in typical Martian tropospheric mesoscale simulations.
- ☞ `iradia` sets the frequency (in dynamical timesteps) at which the radiative computations are performed.
- ☞ Modifying `callphys.def` only implies to recompile the model if the number of tracers is different.
- ☞ If you run a simulation with, say, 3 domains, please ensure that you defined three files `callphys.def`, `callphys_d2.def` and `callphys_d3.def`.

Chapter IV

Preprocessing utilities

In the previous chapter, we decribed the simulation settings in the `namelist.input` file. We saw that any modification of the parameters labelled with `(p1)`, `(p2)` or `(p3)` implies the initial and boundary conditions and/or the domain definition to be recomputed prior to running the model again. As a result, you were probably unable to change many of the parameters of the Arsia Mons test case (proposed in section II.3) in which the initial and boundary conditions – as well as the domain of simulation – were predefined.

In this chapter, we describe the installation and use of the preprocessing tools to define the domain of simulation, calculate an initial atmospheric state and prepare the boundary conditions for the chosen simulation time. This necessary step would eventually allow you to run your own simulations at the specific season and region you are interested in, with a complete ability to modify any of the parameters in `namelist.input`.

IV.1 Installing the preprocessing utilities

First and foremost, since the preprocessing utilities could generate (or involve) files of quite significant sizes, it is necessary to define a directory where these files would be stored. Such a directory (e.g. `/bigdisk/user`) must be linked as follows

```
ln -sf /bigdisk/user $LMDMOD/TMPDIR
```

A second prerequisite to the installation of the preprocessing tools is that the LMD Martian Mesoscale Model was compiled at least once. If this is not the case, please compile the model with the `makemeso` command (see section II.2.2).

The compilation process created an installation directory adapted to your particular choice of compiler+machine. The preprocessing tools will also be installed in this directory. Please type the following commands:

```
cd $LMDMOD/LMD_MM_MARS/g95_32_single/   ## or any install directory
ln -sf ../prepare_ini .
./prepare_ini
```

The script `prepare_ini` plays with the preprocessing tools an equivalent role as the `copy_model` with the model sources : files are simply linked to their actual location in the `SRC` folder. Once you have executed `prepare_ini`, please check that two folders were generated: `PREP_MARS` and `WPS`.

In the `PREP_MARS` directory, please compile the programs `create_readmeteo.exe` and `readmeteo.exe`, using the compiler mentionned in the name of the current installation directory:

```
echo $PWD
cd PREP_MARS/
./compile [or] ./compile_g95
ls -lt create_readmeteo.exe readmeteo.exe
cd ..
```

In the `WPS` directory, please compile the programs `geogrid.exe` and `metgrid.exe`:

```
cd WPS/
./configure   ## select your compiler + 'NO GRIB2' option
./compile
ls -lt geogrid.exe metgrid.exe
```

Apart from the executables you just compiled, the preprocessing utilities include `real.exe`, which was compiled by the `makemeso` script along with the mesoscale model executable `wrf.exe`. `real.exe` should be copied or linked in the simulation directory (e.g. `TESTCASE` for the Arsia Mons test case) to be at the same level than `namelist.input`.

☞ Even though the name of the executable writes e.g. `real_x61_y61_z61_d1_t1_p1.exe`, such program is not related to the specific `makemeso` parameters – contrary to the `wrf.exe` executable. We just found that renaming the (possibly similar if the model sources were not modified) `real.exe` was a practical way not to confuse between executables compiled at different moments.

IV.2 Running the preprocessing utilities

When you run a simulation with `wrf.exe`, the program attempts to read the initial state in the files `wrfinput_d01`, `wrfinput_d02`, ... (one file per domain) and the parent domain boundary conditions in `wrfbdy_d01`. The whole chain of data conversion and interpolation needed to generate those files is summarized in the diagram next page. Three distinct preprocessing steps are necessary to generate the final files. As is described in the previous section, some modifications in the `namelist.input` file [e.g. start/end dates labelled with `(p1)`] requires a complete reprocessing from step 1 to step 3 to successfully launch the simulation, whereas other changes [e.g. model top labelled with `(p3)`] only requires a quick reprocessing at step 3, keeping the files generated at the end of step 2 the same.

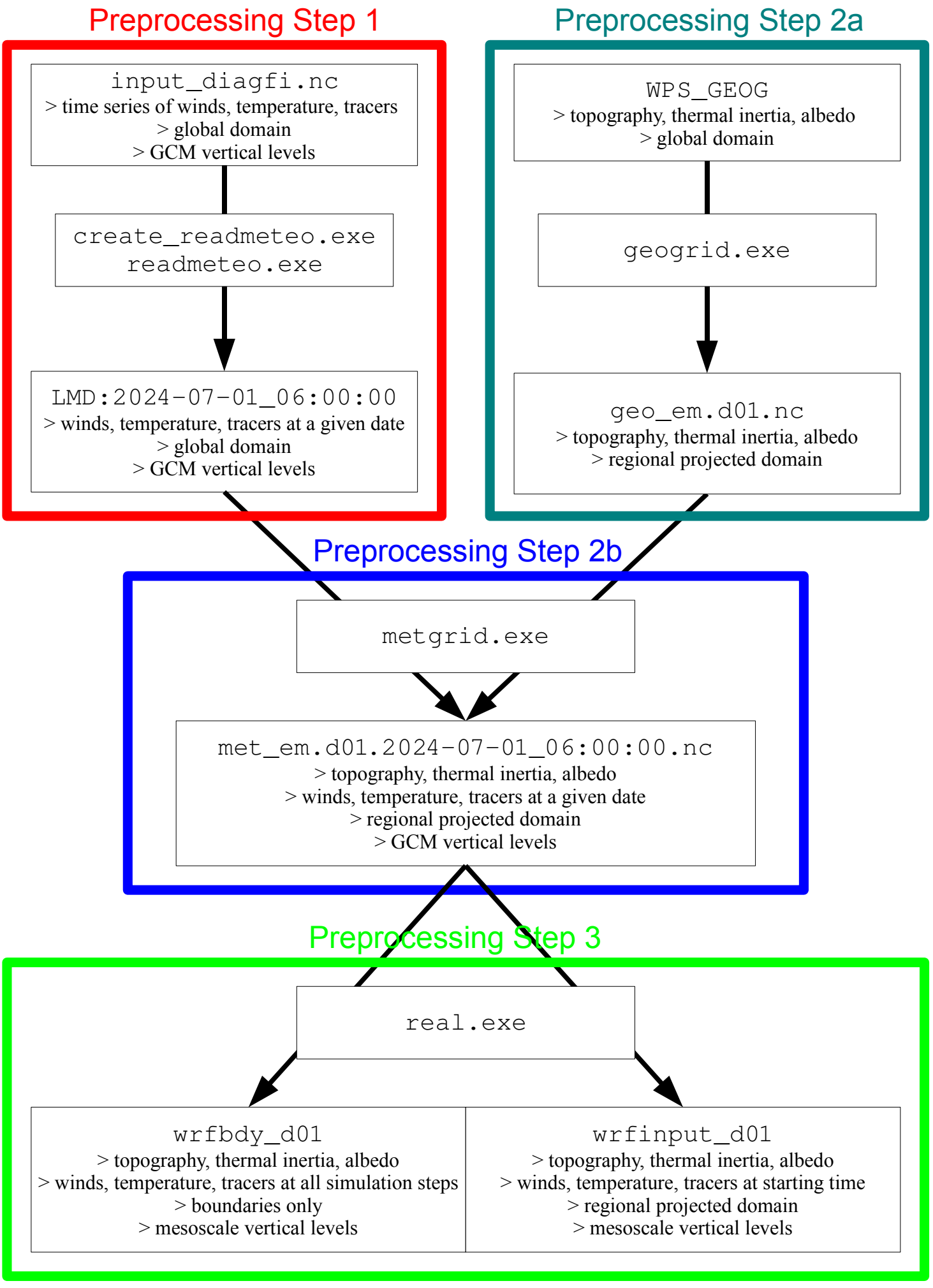
IV.2.1 Input data

IV.2.1.1 Static data

All the static data (topography, thermal inertia, albedo) needed to initialize the model are included in the `$LMDMOD/LMD_MM_MARS/WPS_GEOG` directory. By default, only coarse-resolution datasets[1] are available, but the directory also contains sources and scripts to install finer resolution datasets:

- 32 and/or 64 pixel-per-degree (ppd) MOLA topography [*Smith et al.*, 2001],
- 8 ppd MGS/Thermal Emission Spectrometer (TES) albedo [*Christensen et al.*, 2001],
- 20 ppd TES thermal inertia [*Putzig and Mellon*, 2007]

[1] Corresponding to the fields stored in the file `surface.nc` known by LMD-MGCM users: `http://web.lmd.jussieu.fr/~forget/datagcm/datafile/surface.nc`


Preprocessing Step 1
input_diagfi.nc
> time series of winds, temperature, tracers
> global domain
> GCM vertical levels
create_readmeteo.exe
readmeteo.exe
LMD:2024-07-01_06:00:00
> winds, temperature, tracers at a given date
> global domain
> GCM vertical levels
Preprocessing Step 2a
WPS_GEOG
> topography, thermal inertia, albedo
> global domain
geogrid.exe
geo_em.d01.nc
> topography, thermal inertia, albedo
> regional projected domain
Preprocessing Step 2b
metgrid.exe
met_em.d01.2024-07-01_06:00:00.nc
> topography, thermal inertia, albedo
> winds, temperature, tracers at a given date
> regional projected domain
> GCM vertical levels
Preprocessing Step 3
real.exe
wrfbdy_d01
> topography, thermal inertia, albedo
> winds, temperature, tracers at all simulation steps
> boundaries only
> mesoscale vertical levels
wrfinput_d01
> topography, thermal inertia, albedo
> winds, temperature, tracers at starting time
> regional projected domain
> mesoscale vertical levels

The role of the `build_static` script is to automatically download these datasets from the web (namely PDS archives) and convert them to an acceptable format for a future use by the preprocessing utilities:

```
cd $LMDMOD/LMD_MM_MARS
./build_static
```

- ☞ Please install the `octave` free software[2] on your system to be able to use the `build_static` script. Another solution is to browse into each of the directories contained within `WPS_GEOG`, download the data with the shell scripts and execute the `.m` scripts with either `octave` or the commercial software `matlab` (just replace `#` by `%`).
- ☞ If you do not manage to execute the `build_static` script, converted ready-to-use datafiles are available upon request.
- ☞ The building of the MOLA 64ppd topographical database can be quite long. Thus, such a process is not performed by default by the `build_static` script. If the user would like to build this database, please remove the `exit` command in the script, just above the commands related to the MOLA 64ppd.
- ☞ The resulting `WPS_GEOG` can reach a size of several hundreds of Mo. You might move such a folder in a place with more disk space available, but then be sure to create in `$LMDMOD/LMD_MM_MARS` a link to the new location of the directory.

IV.2.1.2 Meteorological data

The preprocessing tools generate initial and boundary conditions from the `diagfi.nc` outputs of LMD-MGCM simulations. If you would like to run a mesoscale simulation at a given season, you need to first run a GCM simulation and output the meteorological fields at the considered season. For optimal forcing at the boundaries, we advise you to write the meteorological fields to the `diagfi.nc` file at least each two hours. Please also make sure that the following fields are stored in the NETCDF `diagfi.nc` file:

```
netcdf diagfi {
dimensions:
  Time = UNLIMITED ; // (72 currently)
  index = 100 ;
  latitude = 49 ;
  longitude = 65 ;
  altitude = 32 ;
variables:
  float Time(Time) ;
    Time:long_name = "Time" ;
    Time:units = "days since 0000-00-0 00:00:00" ;
  float controle(index) ;
    controle:title = "Control parameters" ;
  float latitude(latitude) ;
    latitude:units = "degrees_north" ;
    latitude:long_name = "North latitude" ;
  float longitude(longitude) ;
    longitude:long_name = "East longitude" ;
    longitude:units = "degrees_east" ;
  float aps(altitude) ;
    aps:title = "hybrid pressure at midlayers" ;
    aps:units = "Pa" ;
  float bps(altitude) ;
    bps:title = "hybrid sigma at midlayers" ;
    bps:units = "" ;
```

[2] Available at `http://www.gnu.org/software/octave`

```
float phisinit(latitude, longitude) ;
  phisinit:title = "Geopotential at the surface" ;
float emis(Time, latitude, longitude) ;
  emis:title = "Surface emissivity" ;
  emis:units = "w.m-1" ;
float tsurf(Time, latitude, longitude) ;
  tsurf:title = "Surface temperature" ;
  tsurf:units = "K" ;
float ps(Time, latitude, longitude) ;
  ps:title = "surface pressure" ;
  ps:units = "K" ;
float co2ice(Time, latitude, longitude) ;
  co2ice:title = "co2 ice thickness" ;
  co2ice:units = "kg.m-2" ;
float temp(Time, altitude, latitude, longitude) ;
  temp:title = "temperature" ;
  temp:units = "K" ;
float u(Time, altitude, latitude, longitude) ;
  u:title = "Zonal wind" ;
  u:units = "m.s-1" ;
float v(Time, altitude, latitude, longitude) ;
  v:title = "Meridional wind" ;
  v:units = "m.s-1" ;
float tsoil(Time, altitude, latitude, longitude) ;
  tsoil:title = "soil temperature" ;
  tsoil:units = "K" ;
float q01(Time, altitude, latitude, longitude) ;
  q01:title = "ice" ;
  q01:units = "kg/kg" ;
float q02(Time, altitude, latitude, longitude) ;
  q02:title = "h2o" ;
  q02:units = "kg/kg" ;
```

☞ If the fields `emis`, `co2ice`, `q01`, `q02`, `tsoil` are missing in the `diagfi.nc` file, they are replaced by respective default values 0.95, 0, 0, 0, tsurf.

An example of input meteorological file `diagfi.nc` file can be downloaded at `http://web.lmd.jussieu.fr/~aslmd/LMD_MM_MARS/diagfi.nc.tar.gz`. Please deflate the archive and copy the `diagfi.nc` file in `$LMDMOD/TMPDIR/GCMINI`. Such a file can then be used to define the initial and boundary conditions, and we will go through the three preprocessing steps.

IV.2.2 Preprocessing steps

IV.2.2.1 Step 1: Converting GCM data

The programs in the `PREP_MARS` directory convert the data from the NETCDF `diagfi.nc` file into separated binary datafiles for each date contained in `diagfi.nc`, according to the formatting needed by the preprocessing programs at step 2. These programs can be executed by the following commands:

```
cd $LMDMOD/LMD_MM_MARS/your_install_dir/PREP\_MARS
echo 1 | ./create_readmeteo.exe     # drop the "echo 1 |" if you want control
./readmeteo.exe < readmeteo.def
```

If every went well with the conversion, the directory `$LMDMOD/TMPDIR/WPSFEED` should contain files named `LMD:`.

IV.2.2.2 2: Interpolation on the regional domain

In the `WPS` directory, the `geogrid.exe` program allows you to define the mesoscale simulation domain to horizontally interpolate the topography, thermal inertia and albedo fields at the domain resolution and to calculate useful fields such as topographical slopes.

Please execute the commands:

```
cd $LMDMOD/LMD_MM_MARS/your_install_dir/WPS
ln -sf ../../TESTCASE/namelist.wps .   # test case
./geogrid.exe
```

The result of `geogrid.exe` – and thus the definition of the mesoscale domain – can be checked in the NETCDF file `geo_em.d01.nc`. A quick check can be performed using the command line

```
ncview geo_em.d01.nc
```

if `ncview` is installed, or the `IDL` script `out_geo.pro`

```
idl
IDL> out_geo, field1='TOPO'
IDL> out_geo, field1='TI'
IDL> SPAWN, 'ghostview geo_em.d01_HGT_M.ps &'
IDL> SPAWN, 'ghostview geo_em.d01_THERMAL_INERTIA.ps &'
IDL> exit
```

if the demo version of `IDL` is installed. Of course if your favorite graphical tool supports the NETCDF standard, you might use it to check the domain definition in `geo_em.d01.nc`.

If you are unhappy with the results or you want to change the location of the mesoscale domain on the planet, the horizontal resolution, the number of grid points ..., please modify the parameter file `namelist.wps` and execute again `geogrid.exe`. Here are the contents of `namelist.wps`:

```
&share
 wrf_core = 'ARW',                    !!   [do not modify: choice of dynamical core]
 max_dom = 1,                         !! number of simulation domains
 start_date = '2024-01-04_02:00:00'   !! YYYY-MM-DD_HH:mm:ss start date
 end_date   = '2024-01-04_12:00:00'   !! YYYY-MM-DD_HH:mm:ss end date
 interval_seconds = 3700              !! frequency of GCM updates [1 Mars hour = 3700 s]
 io_form_geogrid = 2,                 !!   [do not modify: choice of NETCDF outputs]
 debug_level = 0,                     !! verbose level of the programs
 opt_output_from_geogrid_path='./'    !! location of the geogrid outputs
/


&geogrid
 parent_id         =   1,         !! number identifying the related parent domain
 parent_grid_ratio =   1,         !! ratio between parent and nested domains
 i_parent_start    =   1,         !! x-position of the southwest corner of nest
 j_parent_start    =   1,         !! y-position of the southwest corner of nest
 e_we              =  61,         !! number of longitude grid points
 e_sn              =  61,         !! number of latitude grid points
 geog_data_res     = 'gcm'        !! choice of static data sources
                                  !! NB: possible: '64ppd', '32ppd', ...
                                  !! NB: please glance at geogrid/GEOGRID.TBL
 dx = 20000,                      !! resolution (meters) in the x-dimension
 dy = 20000,                      !! resolution (meters) in the y-dimension
 map_proj = 'mercator',           !! map projection: 'mercator', 'lambert' or 'polar'
 ref_lat   =  -12.,               !! north latitude of the center of the domain
 ref_lon   =  239.,              !! east longitude of the center of the domain
 truelat1  =  0.0,                !! (lambert or polar) lat position of projection cone
 truelat2  =  0.0,                !!   [do not modify]
 stand_lon =  0.0,                !! (lambert or polar) lon position of projection cone
 geog_data_path = './WPS_GEOG',   !!   [do not modify: symbolic link in the WPS folder]
/
```

```
&metgrid
 fg_name = './WPSFEED/LMD'        !!   [do not modify: symbolic link in the WPS folder]
 io_form_metgrid = 2,             !!   [do not modify: choice of NETCDF outputs]
 opt_output_from_metgrid_path='./WRFFEED/current'  !!   [do not modify: symbolic link]
/
```

- ☞ No input meteorological data are actually needed to execute `geogrid.exe`.
- ☞ More details about the database and more options of interpolation could be found in the file `geogrid/GEOGRID.TBL`.
- ☞ Defining several domains yields distinct files `geo_em.d01.nc`, `geo_em.d02.nc`, `geo_em.d03.nc`...

Once the `geo_em` file(s) are generated, the `metgrid.exe` program performs a similar horizontal interpolation of the meteorological fields to the mesoscale domain as the one performed by `geogrid.exe` for the surface data. Then the program writes the results in `met_em` files and also collects the static fields and domain parameters included in the `geo_em` file(s) Please type the following commands:

```
cd $LMDMOD/LMD_MM_MARS/your_install_dir/WPS
./metgrid.exe
```

If every went well, the directory `$LMDMOD/TMPDIR/WRFFEED` should contain the `met_em.*` files.

IV.2.2.3 Step 3: Vertical interpolation on mesoscale levels

The last step is to execute `real.exe` to perform the interpolation from the vertical levels of the GCM to the vertical levels defined in the mesoscale model. This program also prepares the final initial state for the simulation in files called `wrfinput` and the boundary conditions in files called `wrfbdy`.

To successfully execute `real.exe`, you need the `met_em.*` files and the `namelist.input` file to be in the same directory as `real.exe`. Parameters in `namelist.input` controlling the behavior of the vertical interpolation are those labelled with `(p3)` in the detailed list introduced in the previous chapter.

Please type the following commands to prepare files for the Arsia Mons test case (or your personal test case if you changed the parameters in `namelist.wps`):

```
cd $LMDMOD/TESTCASE
ln -sf $LMDMOD/WRFFEED/met_em* .
./real.exe
```

The final message of the `real.exe` should claim the success of the processes and you are now ready to launch the integrations of the LMD Martian Mesoscale Model again with the `wrf.exe` command as in section II.3.

- ☞ When you modify either `namelist.wps` or `namelist.input`, make sure that the common parameters are exactly similar in both files (especially when running nested simulations) otherwise either `real.exe` or `wrf.exe` command will exit with an error message.

Chapter V

Starting simulations from scratch

V.1 Running your own GCM simulations

To be completed

V.2 Complete simulations with `runmeso`

To be completed

Chapter VI

Outputs

VI.1 Postprocessing utilities and graphics

To be completed. Do-it-all `idl` scripts would be described here !

VI.2 Modify the outputs

To be completed. Though the method is different, we kept all the convenient aspects of `writediagfi`

Chapter VII

Frequently Asked Questions

- ☞ Which timestep should I choose to avoid crashes of the model ?
- ☞ In the Martian simulations, why can't I define boundaries each 6 hours as on Earth ?
- ☞ Help ! I get strange assembler errors or ILM errors while compiling !
- ☞ Is it possible to run the model on a specific configuration that is not supported ?
- ☞ Why do I have to define four less rows in the parent domain when performing nested runs ?
- ☞ I am kind of nostalgic of early/middle Mars. How could I run mesoscale simulations at low/high obliquity ?
- ☞ Why `real.exe` is crashing when the model top pressure is lower than 2 Pa ?
- ☞ Can I use the two-way nesting ?

> To be completed.