Tutorial I. Installing the LMDZ model

Ionela Musat

Laboratoire de Météorologie Dynamique

LMDZ Training course, December 2019

Tutorial I. Installing and running the LMDZ model

To install and run the LMDZ model you need to:

1/ get source codes: netcdf, IOIPSL, ORCHIDEE, LMDZ

2/ compile the codes

3/ run a 1-day bench test

We have developped a script that does all three things:

install_lmdz.sh

but there are other ways

Tutorial I. Installing the LMDZ model

There are 3 ways to install LMDZ.
Right choice depends on the machine you are using and the type of simulation (long, test or development) you run.

1) using the install_lmdz.sh script (⇒ this Tutorial)

- the script will download the source codes needed (IOIPSL, ORCHIDEE, LMDZ) and netcdf library and will compile them
- recommended method for Linux PC and short development or test runs.

2) using modipsl and libIGCM (⇒ IPSL Training course)

- you will need to install one of the configuration defined by modipsl (for example LMDZOR_v6).
- recommended for IDRIS, TGCC, CINES and for long simulations, as it provides tested reference versions and scripts for launching and monitor long simulations.

3) by hand

- get source codes for each component you need (IOIPSL, ORCHIDEE, LMDZ) and link them with the netcdf library installed on your machine.

Tutorial 1. Using install_lmdz.sh – Contents

It will do most of the work for you, using standard shell tools and commands (gcc, wget, gunzip, tar, ...) :

- Download the required codes archives
- Choose adequate compiler options and build a Makefile
- Install ancillary libraries (netcdf, modipsl, IOIPSL)
- Install land surface model ORCHIDEE if needed
- Install LMDZ using makelmdz_fcm (or makelmdz) script
- Run a test bench

Further details on LMDZ version (in French), in particular the main modifications between versions:

`http://www.lmd.jussieu.fr/~lmdz/Distrib/LISMOI.trunk`

Tutorial 1. install_lmdz.sh Options

./install_lmdz.sh **-h**

-d grid_resolution:	**choose model grid resolution**	**nlonxnlatxnlev**
-name MODEL:	**choose model folder name**	**LMDZvers-release**
-v version:	choose a version	**YYYMMDD.trunk**
-r release_nb:	choose a "svn release"	**svn-number/ "last"**
-SCM	install **1D** version automatically	
-bench:	launch or not a test bench	**1/0**
- compiler compiler	gfortran / ifort/ pgf90/ mpif90	**default:gfortran**
- parallel mode:	sequential/mixed parallelism	**none/ mpi_omp**
-xios	add **with_xios="y" (need parallel=mpi_omp !)**	
-gprof	compile with **-pg** to enable profiling	
-netcdf PATH:	PATH to an existing netcdf	**netcdfPATH**
-opt_makelmdz version	**makelmdz_fcm**/makelmdz	(compile_with_fcm)

Tutorial 1. install_lmdz.sh script

Download : **wget http://www.lmd.jussieu.fr/~lmdz/pub/install_lmdz.sh**

chmod +x install_lmdz.sh

./install_lmdz.sh -d 32x32x39 -name LMDZ2019 -v 20191106.trunk

Downloads 2 archive files

=> modipsl.trunk.tar.gz => code sources

=> bench_lmdz_32x32x39.tar.gz => input files

Compiles the models

Runs a 1-day test simulation

Tutorial 1. Code Compilation

Principles:

Based on makefiles and pre-processor directives.

A unique procedure to compile the different executables (gcm, ce0l, 1d, …)

An environment which allows the compilation of different configurations (zoom, resolution, physics, ...) from the same directory and source code.

An executable compiled with ORCHIDEE does not need to be recompiled to run without ORCHIDEE.

Tutorial 1. Code Compilation - Preprocessing

Preprocessing:

Set of CPP keys embedded in the source code that allow the inclusion of « extra » code or a choice between differing parts of source code depending on their values, before the code is compiled :

E.g. :
.../libf/phylmd/physiq.F :

```
#ifdef INCA
          ….
          CALL VTb(VTinca)
          calday = REAL(days_elapsed) + jH_cur
          CALL chemini( ….
#endif
```

Some CPP keys used un LMDz :
« System » keys : CPP_MPI, NC_DOUBLE, CPP_OMP
« Configuration » keys : CPP_EARTH, CPP_COUPLE, CPP_VEGET, INCA, REPROBUS
« Output » keys : CPP_IOIPSL, histNMC

Tutorial 1. Code Compilation – the makefiles

Two different but similar scripts can be used to compile LMDz :

- ~~*makegcm*~~*/makelmdz* : using the basic shell and our own scripts.
  - create the *dimensions.h* file using script *makdim* for the required resolution (this allows us to manage multiple resolution from the same directory)
  - create code dependencies with script *create_make_gcm*
  - create the *makefile*
  - define compilation and optimisation options
  - compile and creates the executable *gcm.e*

- *makelmdz_fcm* : used by ***MODIPSL***
  - creates the *dimensions.h file*
  - the *-arch* (needed) option determines the architecture of the target machine. Needed so as to read the right configuration file in the *LMDZ/arch* repertory
  - calls script *fcm* to generate dependencies and compile the code, creates an executable *gcm_RESOLUTION_PAR_....e*

Simple example : ./makelmdz -d 48x32x11 -v false gcm
./makelmdz_fcm -d 48x32x11 -v false gcm

Tutorial 1. Compilation : main options

Principal options :

[-h] : help
[-d [[IMx]JMx]LM] : IM, JM, LM are the x, y, z dimensions (def: 96x72x19)
[-p PHYS] : to compile with libf/phyPHYS physics module, (def: lmd)
[-prod / -dev / -debug] : to compile in production (default) / developpement / debug mode.
[-c false/MPI1/MPI2] : ocean coupling : MPI1/MPI2/false (def: false)
[-v false/true] : with or without vegetation (def: false)
[-chimie INCA/false] : with or without INCA (def: false)
[-parallel none/mpi/omp/mpi_omp] : parallelisation (default: none) : mpi, openmp or mix mpi_openmp
[-g GRI] : grid definition in dyn3d/GRI_xy.h (def: regular)
[-io IO] : choice of I/O library, left to the experts (def: ioipsl)
[-include INCLUDES] : supplementary variables for includes
[-cpp CPP_KEY] : supplementary CPP keys definition
[-filtre NOMFILTRE] : use the filter in libf/NOMFILTRE (def: filtrez)
[-link LINKS] : optional library links

makelmdz_fcm option:
-arch nom_arch : name of target architecture

Tutorial 1. Choosing which LMDZ version to work with

Choose between the different available versions on the LMDZ web site:

`http://www.lmd.jussieu.fr/~lmdz/pub/LISMOI.trunk`

Ask the LMDZ team for more information on which versions are actually used :

lmdz-svp@lmd.jussieu.fr

Tutorial 1. What you need to run the LMDZ GCM (1)

•Executable (LMDZ) file :
- gcm.e

• Parameters files :
run.def, gcm.def, vert.def, physiq.def, traceur.def, config.def, etc

• Start files v:
- start.nc, startphy.nc

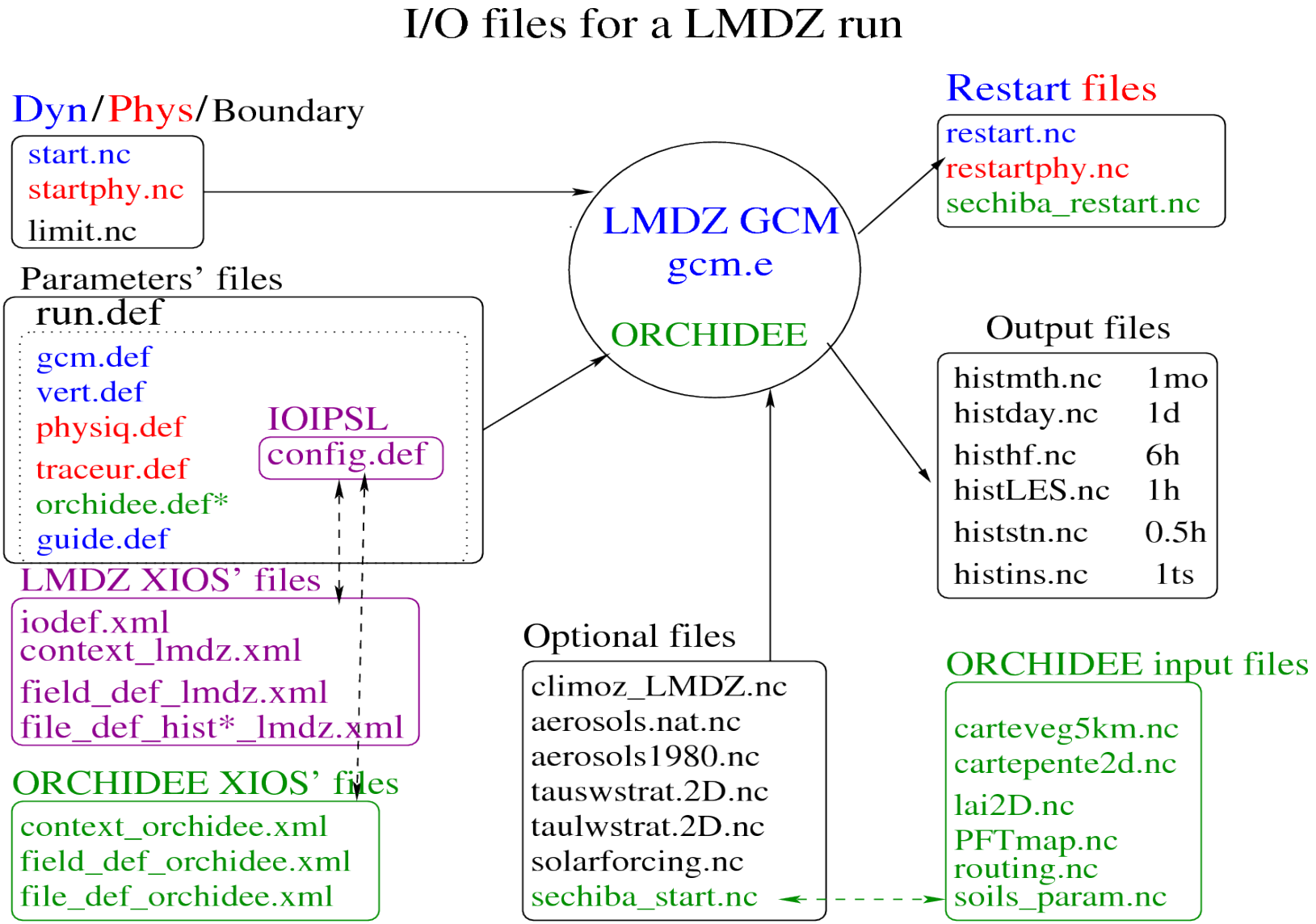
These files are created by the ce0l.e program or may be the result of previous runs

• Boundary conditions file v:
limit.nc
Created by ce0l.e

• Some optional input files v (depending on the simulation) :
- aerosols.nc, climoz_LMDZ.nc, nudging input files (u.nc, v.nc,..), etc

v : these files have to be interpolated on the horizontal grid of the model

Tutorial 1. What you need to run the LMDZ GCM (2)



Tutorial 1. Running the model

```
cd ~/LMDZ2019/modipsl/modeles/LMDZ/BENCH32x32x39
ls
```

```
gcm.e  start.nc startphy.nc limit.nc config.def gcm.def orchidee.def
physiq.def  run.def  traceur.def  vert.def
```

```
./gcm.e
```

or

```
./gcm.e > lmdz.out 2>&1
```

To carry on a simulation that has been run, you have to copy the restart files obtained at the end of the previous run as new initial start files:

```
mv restart.nc start.nc
mv restartphy.nc startphy.nc

./gcm.e
```

Tutorial 1. Has your run completed successfully ?

YES

► you will then have a message saying ***Everything is cool*** on the standard output or in the output text file.

► The code will have created 2 restart files

restart.nc and restartphy.nc

needed to carry on your run

► and some output diagnostic files

histday.nc, histmth.nc, etc. ...

to explore/view using ferret, grads, ...

Tutorial 1. Has your run completed successfully ?

NO

You must find out what the problem is...

Look for an error message in the output text file.
Search for one of the following key words/phrase: ***Houston, we have a problem***, **STOP**, ***hgardfou***, ***integrd: negative surface pressure,*** etc.

Different typical errors :

- technical problem : a missing input file, an error in one of the *.def file

- problem with the model's stability.

Instability in the physics are likely to be detected by hgardfou, which checks the model temperature has realistic values.

Instability in the dynamics most often end up the run with a ***negative surface pressure*** error message.

► **In any of these cases you will most probably have to adjust some flags in the .def files**.

- you have some source code modifications that might not have been thoroughly tested or validated.

Tutorial 1. Installing the LMDZ model: Take-off infos

```
Install LMDZ :
wget http://www.lmd.jussieu.fr/~lmdz/pub/install_lmdz.sh
chmod +x install_lmdz.sh
./install_lmdz.sh -d 32x32x39 -name LMDZ2019
```

Re-compile and re-run an LMDZ simulation : compile.sh and bench.sh

cd ~/LMDZ2019/modipsl/modeles/LMDZ
./compile.sh
cd BENCH32x32x39
./bench.sh

LMDZ releases :
lmdz-svp@lmd.jussieu.fr
http://www.lmd.jussieu.fr/~lmdz/pub/LISMOI.trunk