

Tutorial I. Installing the LMDZ model



Ionela Musat / Laurent Fairhead

Laboratoire de Météorologie Dynamique

LMDZ Training course

January 13th, 2026

https://lmdz.lmd.jussieu.fr/pub/Training/Tutorials/Tutorial_1.pdf

Tutorial I. Installing the LMDZ model



To install and run the LMDZ model you need to:

- 1/ [get source codes](#): netcdf, IOIPSL, (XIOS*), ORCHIDEE, [LMDZ](#)
- 2/ [compile](#) the codes
- 3/ [run a bench test](#) of 3 days

We have developped a script that does all three things:
[install_lmdz.sh](#)

but there are 2 other ways to install LMDZ.

Tutorial I. Installing the LMDZ model



There are **3 ways** to install LMDZ

Right choice depends on the **machine** you are using, the **type of simulation** (**long** for research projects or **short** for development) you do and **the LMDZ version** you need.

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

- recommended method for **Linux PC, IPSL's spirit cluster or IDRIS**
- **development or short test** run
- **new LMDZ version available every 1-2 months**

2) using `modipsl` and `libIGCM` ⇒ see IPSL Training course, 29-30 January or 5-6 February 2026)

- recommended for **IDRIS, TGCC, CINES**
- **long simulations** with tested reference versions and scripts for launching and monitor them
- will install one of the configurations pre-defined by modipsl, for example LMDZOR_v6
- **new LMDZOR configuration available every 6-12 months**

3) by hand

- recommended for people aiming **challenges (!)**
- you need to develop own scripts to get source codes for each component you need (**IOIPSL, ORCHIDEE, LMDZ**) and link them with the **netcdf library** installed on your machine.

Tutorial I. Using install_lmdz.sh – Contents



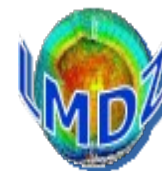
install_lmdz.sh is using standard **shell tools** and **commands** (**wget**, **gunzip**, **tar**, **make**, **cmake**, **Fortran compiler**, **gcc**,...) :

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

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

<http://lmdz.lmd.jussieu.fr/pub/LISM0I.trunk>

Tutorial I. *install_lmdz.sh* options



./install_lmdz.sh -h

-v version	LMDZYYYYMMDD.trunk	20251104.trunk
-r release_nb	LMDZ' svn release : release_nb/last	----
-compiler compiler	gfortran / ifort/ pgf90/ mpif90	gfortran
-parallel mode	parallelism type : none/mpi_omp	none
-d grid_resolution	grid resolution : nlon x nlat x nlev	32x32x39
-bench	launch or not a test bench : 1/0	1
-testing/unstable	LMDZ' svn branch : testing/unstable	testing
-name MODEL	model folder LMDZ\$version\$svn\$optim	LMDZ20251104.trunk
-netcdf PATH	0, 1 or PATH to an existing netcdf	1
-xios	with_xios="y" ; need parallel=mpi_omp	----
-gprof	compile with -pg to enable profiling	----
-cosp	run without, with COSP v1 or COSP v2: NONE/v1 /v2	NONE
-rad	RADIATIVE code : oldrad/rrtm/ecrad	oldrad
-nofcm	compile_with_fcm : 0/1	1
-SCM	install 1D version automatically : 0/1	0
-debug	compile everything in debug mode	none
-opt_makelmdz	call makelmdz_fcm/makelmdz with additional option	none
-physiq	physics' package to use : physiq.def_\${physiq}	physiq.def_NPv6.1
-env_file	arch.env environment file to overwrite the existing one	----
-veget	surface model to run : NONE/CMIP6/xxxx	NONE

Tutorial I. install_lmdz.sh script



Download the script :

```
wget http://lmdz.lmd.jussieu.fr/pub/install_lmdz.sh
```

Make it executable :

```
chmod +x install_lmdz.sh
```

Execute the script, i.e. compile the model and run a 5-days test simulation

```
./install_lmdz.sh -d 32x32x39 -name LMDZseq
```

=> modipsl.**20251104.trunk**.tar.gz => code sources

=> bench_lmdz_**32x32x39**.tar.gz => input files



Principles:

Based on **makefiles** and **pre-processor** directives.

A **unique** procedure to compile the **different executables** (gcm, ce01, 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.

Pre-processor directives allow you to include specific parts of code in your main code. They are governed by what are called CPP keys (such as **CPP_MPI**, **CPP_COUPLE**, **CPP_VEGET**, **INCA**) that are set by the installation script.

Tutorial I. Code Compilation – the makefiles



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

- *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 I. Compilation : main options



makeImdz main options :

[-h] :	help
[-d [[Imx]Jm]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: ioipsi)
[-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

makeImdz_fcm option:

-arch nom_arch : name of target architecture

Tutorial I. Choosing which LMDZ version to work with



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

<http://lmdz.lmd.jussieu.fr/pub/LISM0I.trunk>

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

`lmdz-users@listes.ipsl.fr`

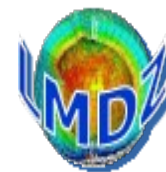
Tutorial I. What you need to run the LMDZ GCM (1)



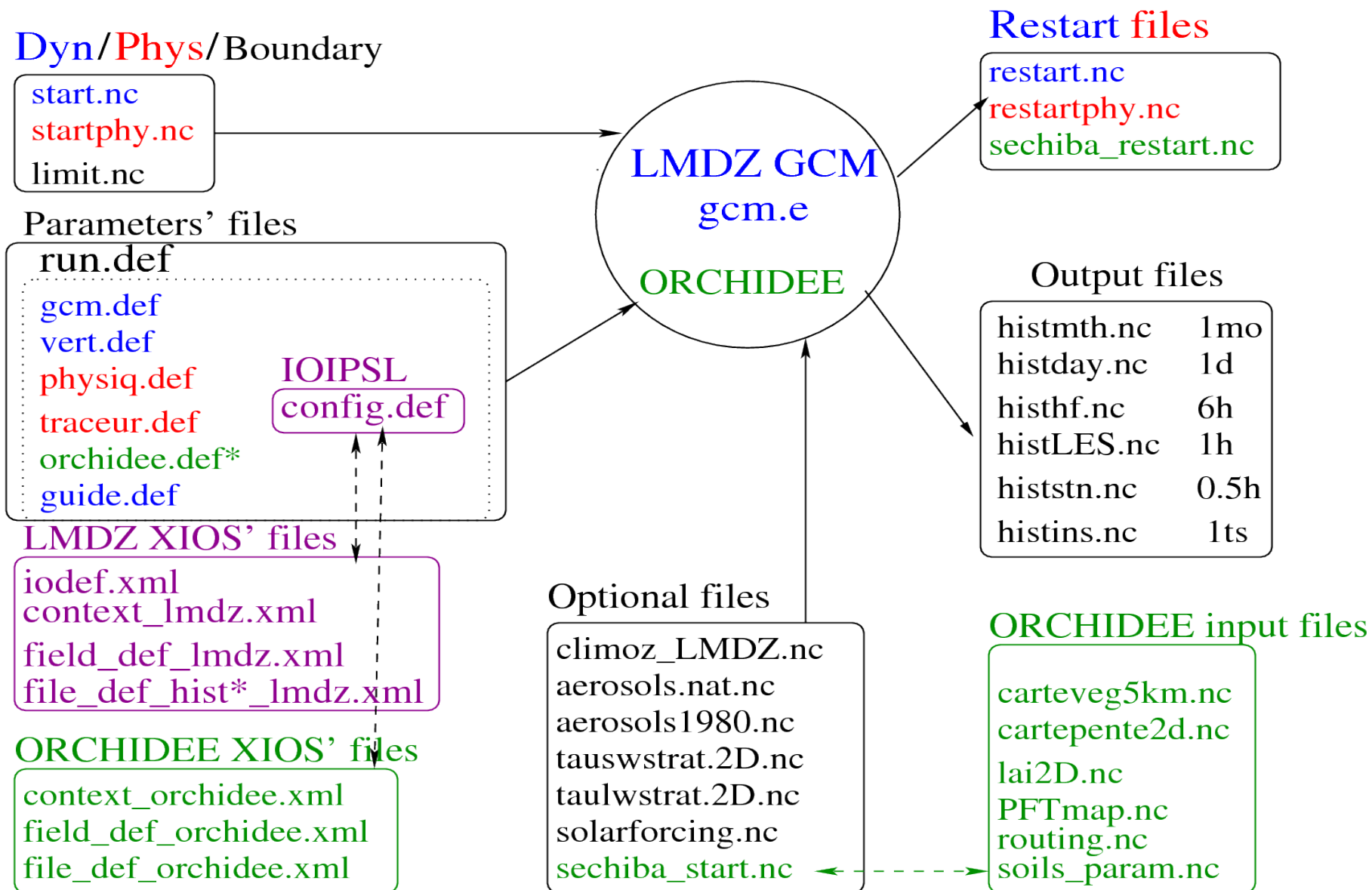
- Executable (LMDZ) file :
 - gcm.e
- Parameters files
 - run.def, gcm.def, vert.def, physiq.def, traceur.def, config.def, etc
- Start files
 - start.nc, startphy.nc
 - These files are created by the [ce0l.e](#) program or may be the result of previous runs
- Boundary conditions file
 - limit.nc
 - Created 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 I. What you need to run the LMDZ GCM (2)



I/O files for a LMDZ run



Tutorial I. Running the model



```
$> cd ~/LMDZseq/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 > listing 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 I. 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

`histhf.nc`, `histday.nc`, `histmth.nc`, etc

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

Tutorial I. Has your run completed successfully ?



NO

You must find out what the problem is.

Look for an error message in the output text file (called listing here), i.e. search for one of the following keywords/phrase:

STOP, *Houston, we have a problem, hgardfou, negative surface pressure, ...*

Different typical errors :

- **Technical problem** : a missing input file, an error in one of the *.def or .xml 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 (integrd routine)** error message.
 - ▶ In any of these cases you will **most probably** have to adjust some flags in the **.def** files (modify dynamic' timestep and teta* flags for a new resolution, for example).
- Otherwise you may have made **some source code modifications** that **might not have been thoroughly tested or validated.**

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



- **Install LMDZ** using [install_lmdz.sh](#)

```
wget http://lmdz.lmd.jussieu.fr/pub/install_lmdz.sh  
chmod +x install_lmdz.sh  
./install_lmdz.sh -d 32x32x39 -name LMDZseq
```

- **Re-compile** and **re-run** an LMDZ simulation : [compile.sh](#) and [bench.sh](#)

```
cd ~/LMDZseq/modipsl/modeles/LMDZ  
./compile.sh  
cd BENCH32x32x39  
./bench.sh
```

- **LMDZ releases** : ask to lmdz-users@listes.ipsl.fr or look here

<http://lmdz.lmd.jussieu.fr/pub/LISMOI.trunk>