# Hands on tutorial #1: First steps with the model

The LMDZ team

February 9, 2026

This first tutorial focuses on installing and making basic first runs using the LMDZ model. This document can be downloaded as a pdf file at
`http://web.lmd.jussieu.fr/~hourdin/COURS/M2/MINI/Intro/Tutorial_1.pdf`
which should ease any copy/paste of command lines to issue.

## 1 Installing a first version

### 1.1 Prerequisites

To run LMDZ, you will need a significant amount of memory, so first ensure this is true. You can use the following command line (that you can also, for convenience, put in your ∼/.bashrc file):

```
ulimit -Ss unlimited
```

### 1.2 Running the `install_lmdz.sh` script

We propose you to create a directory for LMDZ mini projects on your `HOME` directory (but other choices are possible):

```
mkdir -p ~/LMDZmini
cd ~/LMDZmini
```

The first step consists in cloning a git repository containing the model and *blindly* running it :

```
git clone https://lmdz-forge.lmd.jussieu.fr/Git/fhourdin/EnseignementLMDZ.git
cd EnseignementLMDZ/
./exemples.sh
```

The model will be installed automatically in `LMDZ`.

The script should then run smoothly (if it isn't the case, immediately ask for some assistance).

You can take advantage of the installation time to open a second terminal window and explore the downloaded directories and files.

Run `./exemples.sh` as soon as possible. It takes some time du to the donwloads and of the compilation of the netCDF library.

### 1.3 A first glance at the installed files and directory

While installing, you can navigate through the files.

`exemples.sh` will first download [1] `install_lmdz.sh`, the main LMDZ installation script.

`install_lmdz.sh` itself will :

[1] Automatically from the LMDZ web site.

- get files and archives (LMDZ, netCDF library, benches) with the `wget` command from the LMDZ web site.
- install (by default, which means without options) create the `~/LMDZmini/LMDZseq` directory containing LMDZ.
- install `netcdf4_hdf5_seq`, which contains the NetCDF library.
- optionally creates a directory `~/LMDZmini/LMDZ/pub` in which the downloaded files are stored in order to avoid downloading them twice if something went wrong[2].

`LMDZseq` itself contains `modipsl`, which contains the model In `modipsl`, you will find directory `modeles`, containing the `LMDZ` directory.[3]

Once the test bench simulation has been launched (the final step of the `install_lmdz.sh` script), you can visit the LMDZ tree on an other terminal.

```
cd ~/LMDZmini/LMDZseq/modipsl/modeles/LMDZ
```

you will also find a `BENCH32x32x39/` directory from where you will be able to run the model by launching

```
cd ~/LMDZmini/LMDZseq/modipsl/modeles/LMDZ
```

which willl list the outputs of the run (even if the simulation is still running: it indeed takes a few minutes to complete the 3 day-long run on a single processor). Check out the contents of this directory and use your favorite software (Grads, Ferret,...) to browse the contents of the `histday.nc` file.

The source files of the dynamical core are in:

```
~/LMDZmini/LMDZseq/modipsl/modeles/LMDZ/libf/dyn3d
```

including the main programm `gcm.F90`. Physics parameterizations are in:

```
~/LMDZmini/LMDZseq/modipsl/modeles/LMDZ/libf/phylmd
```

physics parameterization driver: `physiq_mod.F90`
The bench simulation is in

```
~/LMDZmini/LMDZseq/modipsl/modeles/LMDZ/BENCH32x32x39
```

You can start looking at the results in `histday.nc`. To know the variables available there :

```
ncdump -h histday.nc  | grep long_name
```

Start to visualize outputs.

## 1.4 Known problems

Installing netCDF is one of the subtile things of the installations. The choice by default was made that `exemples.sh` is compiling the netCDF library from sources. This option is on average the most robust.

If you face problem with it, you can try to use the netCDF library installed on your machine by changin the option `netcdf=0` in `exemples.sh`.

[2] You do not have to do anything special for that.

[3] The default behavior of the `install_lmdz.sh` script is to install the model using the `gfortran` compiler, which is fine on the laptops provided for this training course. In any case, you can switch to another compiler using the `compiler` option of the `install_lmdz.sh` script. Several other features of the script can be modified or unactivated just by editing few parameters in the file or specifying the appropriate option (run `./install_lmdz.sh -h` to learn about these). For example, you can disable the NetCDF download and installation (first operation performed by `install_lmdz.sh`) in case this library is already present on the computer you are using.

# 2 Exploring "physics tendecies"

Open `histday.nc`.

## 2.1 Le rayonnement

On peut commencer par regarder les flux radiatifs descendants au sommet de l'atmosphere (swdntoa) ou au sol (swdnsfc) a differents instants. On peut les moyenner aussi en temps et longitude (plot swdntoa[i=@ave,l=@ave]).

Les tendances associees au rayonnement sont:
- dtswr, K/s pour le SW,
- dtlwr, K/s pour le LW.

Quel est l'impact du rayonnement sur la temperature de l'atmosphere?

## 2.2 La couche limite diffuse

La turbulence de couche limite est prise en compte par une parametrisation des tourbillons diffusifs assurant un transport local dans la couche limite.
Les tendances en temperature et en humidite resultant de la turbulence de couche limite sont:
- dtvdf, K/s pour la temperature,
- dqvdf, kg/kg/s pour l'humidite.

Quel est l'impact de la turbulence sur la temperature et l'humidite de la couche limite?

## 2.3 Les thermiques de couche limite

Le transport non local au sein de la couche limite est pris en compte par une representation explicite des panaches thermiques.
Les tendances en temperature et en humidite resultant de l'effet des thermiques sont:
- dtthe, K/s pour la temperature
- dqthe, kg/kg/s pour l'humidite.

Quel est l'impact des thermiques sur la temperature et l'humidite de l'atmosphere?

## 2.4 La convection profonde precipitante

La convection profonde et les pluies associees sont prises en compte par une parametrisation de la convection nuageuse profonde pouvant atteindre le sommet de la troposphere.
Les tendances en temperature et en humidite resultant de la convection profonde sont:
- dtcon, K/s pour la temperature,
- dqcon, kg/kg/s pour l'humidite.

Quel est l'impact de la convection profonde sur la temperature et l'humidite de l'atmosphere?

## 2.5 Les poches froides

Le schema de convection profonde est couple a une parametrisation des poches froides qui se creent sous les systemes convectifs par l'evaporation des precipitations.
Les tendances en temperature et en humidite resultant de l'effet des poches froides sont:
- dtwak, K/s pour la temperature,
- dqwak, kg/kg/s pour l'humidite.

Quel est l'impact des poches froides sur la temperature et l'humidite de l'atmosphere?

## 2.6 Condensation et evaporation de grande-echelle

Les tendances associees a la condensation et l'evaporation "grande-echelle" sont:
- dtlsc, K/s et dqlsc, kg/kg/s pour la condensation,
- dteva, K/s et dqeva, kg/kg/s pour l'evaporation.

Quel est l'impact de ces processus sur la temperature et l'humidite de l'atmosphere?
Rq: Pour avoir l'effet net, tracez dtlsc+dteva et dqlsc+dqeva.

### 2.7 L'effet de l'orographie sous maille

Les tendances associees a la prise en compte de l'orographie sont:
- dtoro, K/s et dqoro, kg/kg/s pour la trainee,
- dtlif, K/s et dqlif, kg/kg/s pour la portance.
Quel est l'impact de ces processus sur la temperature et l'humidite de l'atmosphere?

### 2.8 Bilan

Comparez l'evolution de la temperature et de l'humidite en un point donne entre la fin et le debut de la simulation avec la somme des tendances dues aux differentes parametrisations physiques.

## 3 Running a sensitivity experiment

In `~/LMDZini/LMDZsec/modipsl/modeles/LMDZ`, create a new directory

```
cd ~/LMDZmini/LMDZseq/modipsl/modeles/LMDZ
mkdir SIMU
cd SIMU
```

Copy required files from `BENCH32x32x39/`

```
cp ../BENCH32x32x39/start.nc .    # initial state of the dynamical core
cp ../BENCH32x32x39/startphy.nc . # initial state of physics parameterizations
cp ../BENCH32x32x39/limit.nc .    # boundary conditions (sea surface temp ...)
cp ../BENCH32x32x39/*.def .       # parameter files
cp ../BENCH32x32x39/gcm.e .       # model executable
```

Then run a new simulation

```
./gcm.e
```

If you do not change anything before rerunning, the output files will be exactly the same as in the first simulations.

More interestingly, you may deactivate the "thermal plume model", the mass flux parameterization of boundary layer convection, by changing `iflag_thermals=18` to `iflag_thermals=0` in `physics.def`.

You can open the two history files coming from the control and sensitivity experiments under `ferret` with

```
use ../BENCH32x32x39/histhf.nc
use histhf.nc
reg/x=0/y=22/k=30:39
set view ul
fill/lev=(-Inf)(-1.8,1.8,0.4)(Inf)/pal=blue_darkred Temp-temp[d=1]
contour/o/color=8 temp
set view ur
plot/l=7 theta[d=1],theta
```

# 4 Installing the Single Column Model (1D configuration)

You may run a 1D or ”Single Column Model” configuration:

```
./exemples.sh -SCM
```

will automatically install the SCM version of the model in `~/LMDZmini/LMDZseq/1D/EXEC/6AL79/ARMCU/REF`. Ceux qui savent qu'ils vont travailler plutôt en 1D peuvent faire la suite avec un ou des cas 1D. De la documentation est disponible dans `~/LMDZmini/LMDZseq/1D/DOC`.

You can run the model by specifying from the begining a list of test cases :

```
./exemples.sh -SCM[:case1,case2,...]
```

Cases should be taken among

```
dry convection          : ayotte
cumulus continent       : arm_cu ARMCY/REF
marine cumulus          : rico RICO/REF rico2  bomex bomex2
strato cumulus          : fire sandufast sanduref sanduslow
ocean. deep convection  : toga twpice
cont.  deep convection  : amma
radiative/conv. eq.     : eq_rd_cv
other                   : hapex92_init dice_bucket gabls4
```

# 5 Installing various Earth climate configurations

```
./exemples.sh --config CONFIG
```

with `CONFIG` in

**`clim`:** Running with imposed climatological SSTs

**`slab`:** Running with a “slab ocean”, a two layer model developed by Francis Codron.

**`aqua`:** Running an aquaplanet, with imposed Sea Surface Temperature (SST) all over the world that depends on latitude only.

**`terra`:** Running a terra planet in which surface temperature is computed with a simple thermal conduction with imposed thermal inertia.

The simulation will be run automatically in a directory named `CONFIG_IMxJMxLM` where `IMxJMxLM` corresponds to the number of grid points in 3 directions.

A first simulation is run automatically on `CONFIG_IMxJMxLM/SIMU1`.

A longer simulation can be run on several months or years, year by year, on `CONFIG_IMxJMxLM/PROD0` by just running `./enchaine.sh`.

# 6 Installing a configuration with idealized physics

The model can be run with a simplified physics, in which only 19 parameters are used to describe the planet to be simulated.

```
./exemples.sh --config SimplePhysics
```

installs a simulation on `SimplePhysics_IMxJMxLM`

## 7 Changing the model grid

A zoom can be applied as well with a zooming factor of typically 2. This zooming factor is defined as the ratio of the smallest grid cell size in x and y to that of the regular grid.

The options to change the grid can be found by running

```
./exemples.sh --help
```

which returns

```
Use : ./exemples.sh [--config|-c bench/tuto/clim/aqua]    Default : bench
                    [--grid|-g IMxJMxLM]                  Default : 96x95x39
                    [--nday|-n number_of_days]            Default : 30
                    [--lon0|-x zoom_center_longitude]     Default : -120
                    [--lat0|-y zoom_center_latitude]      Default : 0
                    [--fact_zoom|-f zoom_factor]          Default : 1
                    [--fact_zoomx|-fx zoom_factor_in_x]   Default : 1
                    [--fact_zoomy|-fy zoom_factor_in_y]   Default : 1
                    [--parallel 0/1]                      Default : 0
```

## 8 Installing a parallel version for longer 3D simulations

```
./exemples.sh --parallel 1
```

This ill automatically install a parallel version of the model in
`~/LMDZmini/LMDZpar/`

For running the model, you can not use `gcm.e` anymore, since you have to specify the number of CPU used for running the code. The parallelism relies on a combination of `OpenMP` and `MPI` approach. When running the model manually on `SIMU`, the number of processes used are specified in the `run_local.sh` command

```
./run_local.sh nMPI nOMP gcm.e
```

with typically `nMPI=2` and `nOMP=2` on a classical laptop with 4 physical CPUs.

# Appendix

## A Changing the outputs

The model outputs are controled in file `config.def` by the following lines (for instance in the standard bench):

```
phys_out_filekeys=       y       y       y       n       n
phys_out_filenames=      histmth histday histhf  histins histLES
phys_out_filelevels=     10      5       4       3       4
phys_out_filetypes=      ave(X)  ave(X)  ave(X) inst(X) inst(X)
phys_out_filetimesteps=   5day   1day    6hr     6hr     6hr
```

Note the `phys_out_filekeys` flags (y/n) which set which file is generated. In the present case, you will obtain 3 files (apart from ORCHIDEE related outputs): `histmth.nc`, `histday.nc` and `histhf.nc`, containing respectively 0 days (should contain 1 record for the whole run, but the output frequency set for `histmth.nc` above is `5day`, whereas the run was only 3 days long), daily and 6-hour averages.

You can find out which variables have been written in a given file by running the command:

```
ncdump -h histday.nc | grep long_name
```

To get a `histmth.nc` with data, you would need to run the model over 5 days. This means you would need to change the value of `nday` in file `run.def` to 5. To run a new simulation, issue command `./gcm.e`; if all goes well, it should end with the message `"Everything is cool"`.

You may also re-run the simulation with a higher output frequency for a few variables (e.g. atmospheric temperature at 2m above the surface: `t2m`, surface pressure: `psol`, precipitation: `precip`, etc.), and other fields types (for instance instantaneous fields) by modifying `config.def`:

```
phys_out_filekeys=       y       y        n       y        n
phys_out_filenames=      histmth histday  histhf  histins  histLES
phys_out_filelevels=     10      5        4       3        4
phys_out_filetypes=      ave(X)  ave(X)  ave(X) inst(X) inst(X)
phys_out_filetimesteps=   5day    1day     6hr     6hr      6hr
flag_t2m=                10      10       5       3        5
flag_psol=               10      10       5       3        4
flag_precip=             10      10       4       3        5
```

Note that unlike other frequencies, 1TS (1 Time Step) is written in capital letters.
You can check in particular the `histins.nc` file and explain the special results.