Hands on tutorial #1: First steps with the model

The LMDZ team

January 9-11, 2024

This first tutorial focuses on installing and making basic first runs using the LMDZ model.
This document can be downloaded as a pdf file:

```
wget http://lmdz.lmd.jussieu.fr/pub/Training/Tutorials/Tutorial_1.pdf
```

which should ease any copy/paste of command lines to issue.

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
```

If you are working on a laptop provided by the LMD and dedicated to these training sessions, you have to log in as the default (and sole) user util1 (the associated password will be given during the training session). On those machines, a working folder, named LMDZ, contains all the required archives ; if it's not already there, you can create it:

```
mkdir LMDZ
cd LMDZ
```

You may want to copy this full folder on a USB key at the end of the sessions.

2 Running the install_lmdz.sh script

The first step consists in downloading it from the LMD website and *blindly* running it (after having first set the access permissions to make it executable):

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

In the present case, option `-name LMDZseq` implies that the default version will be installed in directory **LMDZseq**, and option `-d 32x32x39` that a bench testcase at resolution 32×32-L39 will be run.
Compilation should take around ten to fifteen minutes. The test bench is then downloaded (or copied over if already on the disk) and the 3 days long test simulation on a regular 32×32-L39 grid is run ; messages about various downloads (via **wget**) and/or compiler informations are displayed.

The script should then run smoothly (if it isn't the case, immediately ask for some assistance) and end with messages of the likes of:

```
##########################################################
Simulation finished in ...
You have compiled with:
./makelmdz...
You may re-run it with : cd  ...
or ./bench.sh
##########################################################
```

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

`install_lmdz.sh` will check if some archives (LMDZ, NetCDF library, etc.) are on the disk (in the `~/LMDZ` directory and subdirectories) and if not, will try to retrieve them through the network using `wget` command. It will create the **LMDZseq** directory ; inside, you will find subdirectories **modipsl**, which contains the model, and **netcdf4_hdf5_seq**, which contains the NetCDF library.

In **modipsl**, you will find directory **modeles**, containing the **LMDZ** directory.

Once the test bench simulation has been launched (the final step of the **install_lmdz.sh** script), you will also find a **LMDZ/BENCH32x32x39/** directory from where you will be able to 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 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.

3 Preparing things for tomorrow's course

Tomorrow you will need to have run the model (at 32×32-L39 resolution) to investigate the outputs. To prepare things for that you must, in a dedicated directory, run the model for 5 days.

- go to `LMDZseq/modipsl/modeles/LMDZ` and use the file called:

  ```
  bench_lmdz_32x32x39.tar.gz
  ```

  to create a new experiment:

  ```
  mv BENCH32x32x39 BENCH32x32x39_old
  tar -xf bench_lmdz_32x32x39.tar.gz
  mv BENCH32x32x39 CTRL_32x32x39
  cd CTRL_32x32x39
  ```

- change to `nday=5` in `run.def` to ensure that you do a 5 day simulation
- to avoid recompiling the code, just copy the executable you have already compiled before:

  ```
  cp ../BENCH32x32x39_old/gcm.e .
  ```

- finally copy the `config.def_full` file to `config.def` to insure that you have the right diagnostics for tomorrow's course.

  ```
  cp config.def_full config.def
  ```

- Run the model using `./gcm.e > listing 2>&1`.

You can let the simulation run and start the following exercises but you need to be sure that your simulation will have run for tomorrow.

4 Changing the outputs

You can go now back to the 3 day bench (saved in the `BENCH32x32x39_old directory`).

The model outputs are controled in file **config.def** by the following lines:

```
phys_out_filekeys=       y       y       y       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    2hr     6hr     6hr
```

Note the `phys_out_filekeys` flags (y/n) which set which file is generated. In the present case, you will obtain 4 files (apart from ORCHIDEE related outputs): `histmth.nc`, `histday.nc`, `histins.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, 6-hour ans 2-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**:

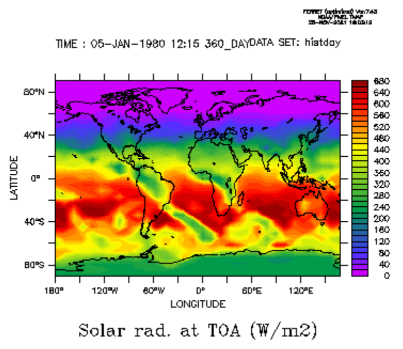
```
phys_out_filekeys=       y       y       y       y       n
phys_out_filenames=      histmth histday histhf  histins histLES
phys_out_filelevels=     10      5       4       3       4
phys_out_filetypes=      ave(X)  ave(X)  ave(X) inst(X) inst(X)
phys_out_filetimesteps=   5day   1day    2hr     6hr     6hr
flag_t2m=                10      10      5       3       5
flag_psol=               10      10      5       3       4
flag_precip=             10      10      4       3       5
```

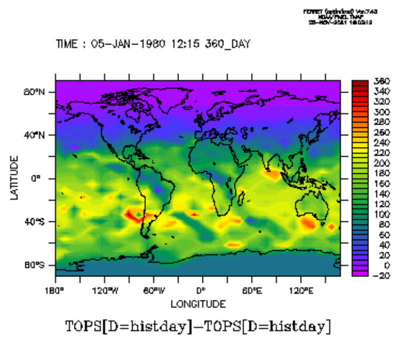
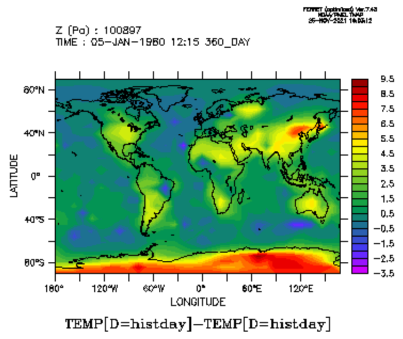
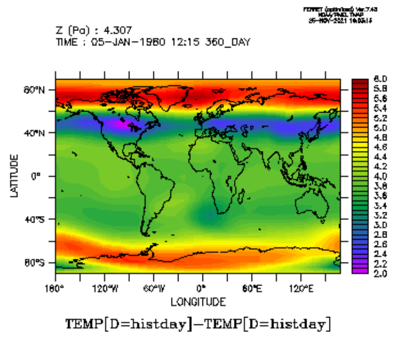
Note that unlike other frequencies, 1TS (1 Time Step) is written in capital letters.
You can check in particular the **histins.nc** file and explain the special results.

5 Making some sensitivity test runs

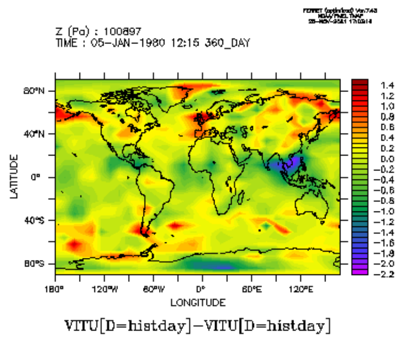
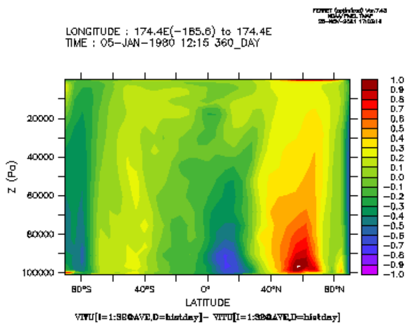
The following 3 exercises will show you how to change some parameters in the different .def files to do some sensitivity runs. You should copy the contents of the `LMDZ/BENCH32x32x39_old/` directory in 3 different directories as it will serve as a starting point for our different exercises and as a reference for comparing results (e.g. `cp -R BENCH32x32x39_old/ Exercice1)`

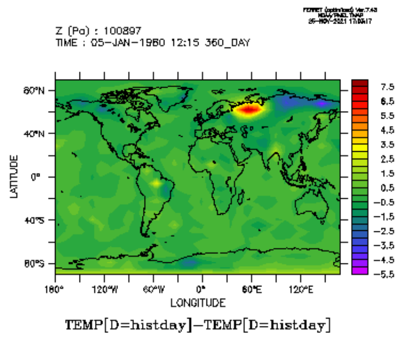
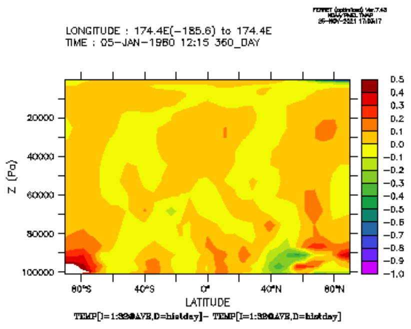
- Exercise 1 : set the solar constant to 2000 (we expect the model to heat up), run the model for five days and display the solar radiation at the top of the atmosphere, the difference on day 5 between the solar radiation at the top of the atmosphere of your simulation and your reference simulation (the one in `LMDZ/BENCH32x32x39_old/`) and compare the temperature of the first and last atmospheric level for day 5 with the reference simulation. You should find a result close to this:



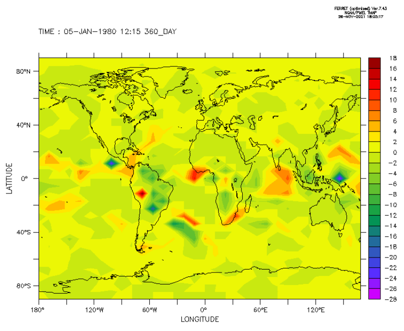

TIME : 05-JAN-1980 12:15 360_DAY
LATITUDE
LONGITUDE
TOPS[D=histday]–TOPS[D=histday]

Z (Pa) : 100897
TIME : 05-JAN-1980 12:15 360_DAY
LATITUDE
LONGITUDE
TEMP[D=histday]–TEMP[D=histday]

Z (Pa) : 4.307
TIME : 05-JAN-1980 12:15 360_DAY
LATITUDE
LONGITUDE
TEMP[D=histday]–TEMP[D=histday]

- Exercise 2 : deactivate the subgrid orography parametrizations by changing the **ok_orodr** and **ok_orolf** flags in **physiq.def**. Run a 5 days simulation and compare the zonal wind component at the first level with the reference simulation, the temperature at the first level and the zonal distributions of the zonal wind component and temperature of the two simulations. You should find a result close to:


Z (Pa) : 100897
TIME : 05-JAN-1980 12:15 360_DAY
LATITUDE
LONGITUDE
VITU[D=histday]–VITU[D=histday]



Z (Pa) : 100897
TIME : 05-JAN-1980 12:15 360_DAY
LATITUDE
LONGITUDE
TEMP[D=histday]–TEMP[D=histday]


- Exercise 3 : activate the Tiedtke convection scheme. Run a 5 day simulation and compare precipitation with the reference simulation. You should get a result close to:



(PRECIP[D=histday]−PRECIP[D=histday])*86400

Hint, running this simulation means that you will need to add a parameter in the physiq.def file

6 Optional: Running an aquaplanet configuration

The simulations run in the previous example included initial and boundary conditions (**start*.nc** and **limit.nc** files). It is also possible to run the model in "aquaplanet" configuration, with idealized initial and boundary conditions (no topography and imposed surface temperatures).

Make an **AQUAPLANET** directory where you will run the model. Copy over the **gcm.e** and ***.def** files from another simulation to this directory. Edit the **gcm.def** file to set

```
read_start=n
iflag_phys=101
```

Note that there are various possibilities for flag **iflag_phys**, values between 101 and 114 correspond to different choices of imposed SSTs (see routines **iniaqua** and **profil_sst** in file **phyaqua_mod.F90** in the **LMDZ/libf/phylmd** directory).

Then run the model:

```
./gcm.e > listing 2>&1
```

Inspect the various outputs (e.g. zonal averages and deviations of meteorological quantities such as surface pressure *psol*) and their temporal evolution, and compare to previously obtained simulations. Experiment changing the length of the run (variable **nday** in the **run.def** file to investigate the evolution of the system from initial settings to a "converged" state.

If time permits, experiment running at a different resolution (e.g. 96×95-L39 ; which means you will have to recompile the model). Note that the number of time steps per day depends on the model resolution, and has to be higher for this resolution than it was strictly required for the 32×32-L39 case. The gcm.def file you've used has already **day_step=480**, which is enough, but you can check that the model will crash if you reduce it significantly (try it !).

7 Optional: Switching to a different version of the model

Using the installation script as we did, we installed a version of the model that is hardcoded in the script. If you search for the line `version=...` in the script, it will give you the version of the model that is installed by default (in our case version `20231022.trunk`, the version dated 22nd October 2023, which corresponds to *svn release* 4729). You can actually change the version of the model you install with the `install_lmdz.sh` script by using the `-v` or `-r` options of the script. This information can moreover be obtained directly from within the LMDZ directory: simply go to the `LMDZseq/modipsl/modeles/LMDZ` directory and execute:

```
svn info
```

Note: you may encounter the following error when typing *svn info*:

```
svn: E155036: Voir la commande 'svn upgrade'
svn: E155036: The working copy at '/home/util1/LMDZ/LMDZ20201109/modipsl/modeles/LMDZ'
is too old (format 10) to work with client version '1.8.8 (r1568071)' (expects format 31).
```

If so, simply follow the suggested cure. Enter: `svn upgrade`. Then try `svn info` again.
If you encounter the following error when typing *svn info*:

```
svn: E155021: The client is too old for the working copy at
'/home/util1/LMDZ/LMDZ20201109/modipsl/modeles/LMDZ'
You need a newer version of the Subversion client.
```

Then you unfortunately won't be able to use svn unless you uprgade to a version at least as recent (version 1.10) as the one used by *install_lmdz.sh.*

Using svn, it is possible to change from model version and select and older or newer `svn release`. To change from current version to *svn release* 4581 (a previous version), go to directory **LMDZ** and issue command:

```
svn update --revision 4581
```

In the **LMDZ** directory, recompile LMDZ. To do so, run[1]
./makelmdz_fcm -arch local -j 8 -rad rrtm -d 32x32x39 -v false gcm

Only routines modified by the svn update or depending on them will be recompiled.

Once the model has been successfully recompiled, run a new simulation. To do so, create a new subdirectory in **LMDZ** (e.g. **BENCH32x32x39.4581**) and copy boundary conditions, initial conditions and parameters files (**limit.nc, start*nc, *.def**) over from directory **BENCH32x32x39**, along with the newly created gcm executable.
As you used **makelmdz_fcm**, it will be **LMDZ/bin/gcm_32x32x39_phylmd_rrtm_seq.e** and you should copy it over to the directory where you will run the model, e.g.:

```
cp LMDZ/bin/gcm_32x32x39_phylmd_rrtm_seq.e BENCH32x32x39.4581/gcm.e
```

Then launch the run[2]:

```
./gcm.e > listing 2>&1
```

Redirection of the model outputs in a text file is usually the best way to keep a trace of the run.

Once the simulation is finished, you can compare, using the **diff** command and visualisation software, the simulation results (compare files **listing**, **hist*.nc restart.nc** and **restartphy.nc** from the two simulations)

Once this test finished, revert to the *trunk* 4729 version of the model by going to directory **LMDZ** and using:

```
svn update --revision 4729
```

Note that using **svn update** without any specific revision number implies updating to the latest version on the branch (which is something you should do regularly if you want to keep up with model updates). Don't forget to recompile the model after any **svn update**!

[1] To avoid typing such a long line over and over, it is more convenient to have it in a script, e.g. **compile.sh** to execute each time a compilation is required.

[2] As for the compilation step, this step can be put in a script, e.g. **bench.sh**, as is the case in **BENCH32x32x39**