LMDZ tutorial: ICOLMDZ

LMDZ team

December 10-12, 2024

This tutorial focuses on setting up, compiling and running ICOLMDZ. ICOLMDZ is the LMD model that combines the DYNAMICO dynamical core (**https://gitlab.in2p3.fr/ipsl/projets/dynamico/dynamico/** and **https://forge.ipsl.jussieu.fr/dynamico**) and the LMDZ physics package through the ICOSALMDZ interface (**https://trac.lmd.jussieu.fr/LMDZ/browser/ICOSALMDZ**). It is used in global forced and coupled configurations as well as in LAM (Limited Area Model) configurations where it is run over some zoomed region.

This document can be downloaded as a pdf file:

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

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

This tutorial is for users who want to learn the basic steps needed to compile and run ICOLMDZ on their machine. Note that this implies the prerequisite that you can run in parallel on your machine (i.e. that you have already completed the tutorial on running LMDZ in parallel) and that XIOS can be installed on your machine.

WARNING : if you are using a version of the gfortran compiler older than 10., you will need to add the usual **-arch** and **-arch_dir** to the **install_lmdz.sh** commandes below.

1 Running the install_lmdz.sh script and a simple bench

The **install_lmdz.sh** script can download and compile the necessary libraries (NetCDF, IOIPSL, XIOS) and programs (DYNAMICO, ICOSALMDZ, ORCHIDEE and LMDZ), and runs a test simulation. All that is required is to specify the `-icolmdz`, `-xios`and `-parallel mpi_omp` (XIOS is designed to be used in parallel) options:

```
wget http://lmdz.lmd.jussieu.fr/pub/install_lmdz.sh
chmod +x install_lmdz.sh
./install_lmdz.sh -name ICOSALMDZ -r 5341 -icolmdz -xios -parallel mpi_omp -rad oldrad
```

The `-r 5341` will download a version of the LMDZ code that is compatible with the updated DYNAMICO and ICOSALMDZ codes.

As with previous automated installations, you are encouraged to browse through the contents of subdirectories **LMDZ** (e.g. the **compile.sh** script), **DYNAMICO** (this directory containes the source code DYNAMICO dynamical core) and **ICOSALMDZ** (this directory containes the source code for the interface between DYNAMICO and LMDZ) .

The installation should take around 40 minutes and after having compiled the model, will run a simple bench at nbp=10 resolution (which corresponds to a lonlat resolution of about 800km). Once the script has finished running, you can explore the results of bench run that are in the following subdirectory:

```
./ICOSALMDZ/modipsl/modeles/ICOSA_LMDZ/bench_icolmdz_nbp10_79
```

The bench has run for 3 days so you will find 3 days of data to peruse in the histday.nc file

2 Changing resolution

In this section we will change the resolution of the model, from nbp=10 to nbp=20 (approximately from a 800km to a 400km gridsize at the equator) and run a 3-day simulation. We will need to fetch new start and boudary condition files for this resolution:

```
wget http://lmdz.lmd.jussieu.fr/pub/3DBenchs/bench_icolmdz_nbp20_79.tar.gz
tar xvf bench_icolmdz_nbp20_79.tar.gz
```

This will extract (almost) all the files needed to run ICOLMDZ at the higher resolution in a directory called **BENCH_NBP20_L79** . You might want to rename this directory (see later):

```
mv BENCH_NBP20_L79 BENCH_NBP20_L79_reg
```

We then just need to enter the new directory, copy the ICOLMDZ and xios executables and run the bench.sh script:

```
cd BENCH_NBP20_L79_reg
cp ../bin/icosa_lmdz.exe .
cp ../../XIOS/bin/xios_server.exe .
```

If you compare the **run_dynamico.def** in that directory with that from the nbp=10 simulation, you will notice that the parameter **nbp**, has indeed changed value as have some of the time steps and dissipation times. You can then launch the simulation:

```
time ./bench.sh
```

When finished, you should look at the end of the file called **listing** to check that everything went as planned. Ther should be some lines at the end saying

```
PHYREDEM NOUVEAU
....
physiq fin
....
-> report : Performance report : Ratio : 0.286208%
```

If not, ask for help! You can display some results of the simulation by looking at the **histday.nc** file as before.

3 Defining a gaussian zoom

We now extend the previous exercise by zooming on the Paris region using a Schmidt transform. We have to download new start and boundary files again. So, copy the directory where you run the nbp=20 simulaton (from the previous section) in a new directory, download the new needed files in that directory, comment out the 3 lines that define the zoom in the **run_dynamico.def** file (look for "zoom characteristics") and launch the bench script:

```
cp -r BENCH_NBP20_L79_reg BENCH_NBP20_L79_zoom
cd BENCH_NBP20_L79_zoom
wget http://lmdz.lmd.jussieu.fr/pub/3DBenchs/ce0l_icolmdz_nbp20-zoom.tar.gz
tar xvf ce0l_icolmdz_nbp20-zoom.tar.gz
[vi/emacs/nano] run_dynamico.def
time ./bench.sh
```

Once finished, you can compare results obtained on the regular grid and the zoomed grid. As the only difference between the two simulations is the changed grid, differences in the results will come from this change in grid.

Unfortunately, as the outputs are mapped on a regular grid, you cannot actually visualize the results on the native grid. There is an option to output results on the native grid (by using the **remap_output** option in the run.def file) but you'll have to find a way to visualize it with some other software than **ferret** as it cannot cope with specifications of the icosahedral grid of DYNAMICO.

You could also refine the output grid by changing its characteristics in the **context_lmdz.xml** file so that the effect of the zoom may appear.

Any other ideas?