Tutorial N°2 : **Setting up a simulation**
with a regular or zoomed grid

and a few words about **Optional exercises** :
nudging, tracers, 1D, different physics,
parallelism, XIOS,
coupling with land-surface schemes (bucket, Orchidee)

LMDZ team

Steps for setting up a simulation : general case

0) **Prerequisits :**

0.1) The **configuration** of your choice is **downloaded in a dedicated folder** -> which components&versions (LMDZ, + other models ? ORC, NEMO, INCA, Reprobus...

0.2) The **model(s) are compiled** with the options of your choice (resolution, IO treatment (IOIPSL, XIOS), running parallel/sequential, rrtm···) **=> gcm.e**

0.3) Initial and boundary files **start*.nc** and **limit.nc** **are available for the chosen grid :** generally existant for regular grids at standard resolutions, to be produced for non-standard grids by compiling and running **ce0l.e**

1) **Create** the simulation folder, in the appropriate directory
   Copy in it (**or create links** to)
   -> ***.def** files, executable(s), scripts for postprocessing
   -> **start*.nc** and **limit.nc** for the chosen grid :
2) **Set your desired parameters** in *.def files: zoom parameters, timestep, dissipation
3) **Run the model** (and the post-processing scripts if needed)

To run a new simulation - with the same gcm.e and grid, start again at **step (1)**
- with different resolution/grid: re-create **start*** and **limit** cf **(0.3)**
- after code changes: recompile cf **(0.2)** => **new gcm.e**

Practical guide :
https://www.lmd.jussieu.fr/~lmdz/pub/Training/Tutorials/Tutorial_2.pdf

1st step : download and unpack working directory :

In LMDZ/LMDZ20201109.trunk/modipsl/modeles/LMDZ :

```
wget http://www.lmd.jussieu.fr/~lmdz/pub/Training/tutorial.tar
tar -xf tutorial.tar
cd TUTORIAL
```

Working directory :

LMDZ/LMDZ20201109.trunk/modipsl/modeles/LMDZ/**TUTORIAL**

imposed

user's choice : *./install_lmdz.sh -name XXX*

user's choice : *mkdir XXX*

```
[asima@ciclad-ng TUTORIAL]$ tree
.
├── DEF
│   ├── config.def
│   ├── gcm.def
│   ├── gcm.def_96x95x39_NPv3.1
│   ├── gcm_zoom_tuto.def
│   ├── guide.def
│   ├── L39.def
│   ├── L47.def
│   ├── L79.def
│   ├── orchidee.def
│   ├── PHYS
│   │   ├── physiq.def_AR4
│   │   ├── physiq.def_NPv0.0
│   │   ├── physiq.def_NPv1.0
│   │   ├── physiq.def_NPv2.0
│   │   ├── physiq.def_NPv3.0
│   │   ├── physiq.def_NPv3.1
│   │   ├── physiq.def_NPv3.2
│   │   ├── physiq.def_NPv4.12
│   │   ├── physiq.def_NPv5.17h
│   │   ├── physiq.def_NPv5.4
│   │   ├── physiq.def_NPv5.5
│   │   ├── physiq.def_NPv5.65
│   │   ├── physiq.def_NPv5.67
│   │   ├── physiq.def_NPv5.70
│   │   ├── physiq.def_NPv5.80b
│   │   ├── physiq.def_NPv5.80bz0
│   │   ├── physiq.def_NPv6.0.10
│   │   ├── physiq.def_NPv6.0.10fallv
│   │   ├── physiq.def_NPv6.0.11trigB
│   │   ├── physiq.def_NPv6.0.12
│   │   ├── physiq.def_NPv6.0.12split
│   │   ├── physiq.def_NPv6.0.12ttop
│   │   ├── physiq.def_NPv6.0.7
│   │   ├── physiq.def_NPv6.0.8
│   │   ├── physiq.def_NPv6.0.9
│   │   └── physiq.def_NPv6.1
│   ├── physiq.def
│   ├── Readme
│   ├── run.def
│   └── traceur.def
├── get_era.sh
├── init.sh
├── Readme
├── reb.sh
├── run_local.sh
└── run_X64_ADA.sh
```

Content of working directory TUTORIAL (1/3)

Readme

```
In the current directory, you may
 1/ compile the model
 2/ create initial and boundary conditions on a zoomed (or regular)
      grid
 3/ run the model

Contains :
==========

init.sh : main script that
 1/ creates initial state and boundary conditions -> INITIAL
 (if running with Orchidee land model : prepares a preliminary
       simulation to produce the corresponding initial state -> SIMU0)
 2/ prepares a first simulation -> SIMU1
 3/ prepares a possible long simulation (enchaine.sh, branche.sh)
DEF : contains default files .def for setup parameters

get_era.sh : to interpolate ERA reanalysis on the model grid
reb.sh : to "rebuild" output file for parallel computation with IOPSL
run_local.sh : to run the model (important for parallel computers)
!run_X64_ADA.sh: the same for ada supercomputer @idris-> now jean-zay!

NB: If you change the horizontal resolution of LMDZ you should
    modify some parameters in DEF/gcm.def :
      --> day_step and iphysiq (in order to satisfy the CFL criteria)
      --> dissipation parameters : tetagdiv, tetagrot, tetatemp
```

```
[asima@ciclad-ng TUTORIAL]$ tree
.
├── DEF
│   ├── config.def
│   ├── gcm.def
│   ├── gcm.def_96x95x39_NPv3.1
│   ├── gcm_zoom_tuto.def
│   ├── guide.def
│   ├── L39.def
│   ├── L47.def
│   ├── L79.def
│   ├── orchidee.def
│   ├── PHYS
│   │   ├── physiq.def_AR4
│   │   ├── physiq.def_NPv0.0
│   │   ├── physiq.def_NPv1.0
│   │   ├── physiq.def_NPv2.0
│   │   ├── physiq.def_NPv3.0
│   │   ├── physiq.def_NPv3.1
│   │   ├── physiq.def_NPv3.2
│   │   ├── physiq.def_NPv4.12
│   │   ├── physiq.def_NPv5.17h
│   │   ├── physiq.def_NPv5.4
│   │   ├── physiq.def_NPv5.5
│   │   ├── physiq.def_NPv5.65
│   │   ├── physiq.def_NPv5.67
│   │   ├── physiq.def_NPv5.70
│   │   ├── physiq.def_NPv5.80b
│   │   ├── physiq.def_NPv5.80bz0
│   │   ├── physiq.def_NPv6.0.10
│   │   ├── physiq.def_NPv6.0.10fallv
│   │   ├── physiq.def_NPv6.0.11trigB
│   │   ├── physiq.def_NPv6.0.12
│   │   ├── physiq.def_NPv6.0.12split
│   │   ├── physiq.def_NPv6.0.12ttop
│   │   ├── physiq.def_NPv6.0.7
│   │   ├── physiq.def_NPv6.0.8
│   │   ├── physiq.def_NPv6.0.9
│   │   └── physiq.def_NPv6.1
│   ├── physiq.def
│   ├── Readme
│   ├── run.def
│   └── traceur.def
├── get_era.sh
├── init.sh
├── Readme
├── reb.sh
├── run_local.sh
└── run_X64_ADA.sh
```

Content of working directory TUTORIAL (2/3)

DEF directory : Parameter files *.def for 3D simulations

run.def : general configuration file :
**.def* files to use, calendar type (*earth_360d,365d,366d*),
restart year, number of days to run *nday* etc

config.def : output, coupling, RRTM, orb. par., GHG, aerosol eff, O_3 etc
gcm.def : grid-dependent param. (day_step, iphysiq, zoom, dissipation) etc.
physic.def : version-specific set of param. (here the 'NPv6.1')
PHYS/physic.def_XXX : available versions of physic.def

guide.def : nudging param.
traceur.def : tracer nb., transport processes, name
orchidee.def : parameters for land model Orchidee
L39, L47, L79.def : vertical discretization etc.

Also seen in run.def : ../DefLists/output.def : output configuration (variables)

See DEF/Readme for details on physic.def_XXX files and references **!**
Hourdin et al. 2006, 2013a, 2013b ; to be updated : 2019)

```
[asima@ciclad-ng TUTORIAL]$ tree
.
├── DEF
│   ├── config.def
│   ├── gcm.def
│   ├── gcm.def_96x95x39_NPv3.1
│   ├── gcm_zoom_tuto.def
│   ├── guide.def
│   ├── L39.def
│   ├── L47.def
│   ├── L79.def
│   ├── orchidee.def
│   ├── PHYS
│   │   ├── physiq.def_AR4
│   │   ├── physiq.def_NPv0.0
│   │   ├── physiq.def_NPv1.0
│   │   ├── physiq.def_NPv2.0
│   │   ├── physiq.def_NPv3.0
│   │   ├── physiq.def_NPv3.1
│   │   ├── physiq.def_NPv3.2
│   │   ├── physiq.def_NPv4.12
│   │   ├── physiq.def_NPv5.17h
│   │   ├── physiq.def_NPv5.4
│   │   ├── physiq.def_NPv5.5
│   │   ├── physiq.def_NPv5.65
│   │   ├── physiq.def_NPv5.67
│   │   ├── physiq.def_NPv5.70
│   │   ├── physiq.def_NPv5.80b
│   │   ├── physiq.def_NPv5.80bz0
│   │   ├── physiq.def_NPv6.0.10
│   │   ├── physiq.def_NPv6.0.10fallv
│   │   ├── physiq.def_NPv6.0.11trigB
│   │   ├── physiq.def_NPv6.0.12
│   │   ├── physiq.def_NPv6.0.12split
│   │   ├── physiq.def_NPv6.0.12ttop
│   │   ├── physiq.def_NPv6.0.7
│   │   ├── physiq.def_NPv6.0.8
│   │   ├── physiq.def_NPv6.0.9
│   │   └── physiq.def_NPv6.1
│   ├── physiq.def
│   ├── Readme
│   ├── run.def
│   └── traceur.def
├── get_era.sh
├── init.sh
├── Readme
├── reb.sh
├── run_local.sh
└── run_X64_ADA.sh
```

Content of working directory TUTORIAL (3/3)

Scripts

init.sh : main script

run_local.sh :
run in parallel & rebuild output on local machine

(needs update for Jean-Zay : run_X64_ADA.sh)
same thing on “ADA” supercomputer at IDRIS

reb.sh : rebuild output, if running in parallel mode

get_era.sh : for nudging :
retrieve and interpolate ERAI reanalysis files

Others, for “production” mode : enchaine.sh, branche.sh

Content of main script init.sh

Remember from TUTORIAL/Readme :

```
init.sh : main script that
 1/ creates initial state and boundary conditions for LMDZ -> INITIAL
    (if running with Orchidee land model : prepares a preliminary
       simulation to produce the corresponding initial state -> SIMU0)
 2/ prepares a first simulation -> SIMU1
```

init.sh (1/2)

```
############################################################
# 0. Setup
############################################################
# standards : 96x95x39, 144x142x79
grid_resolution=48x36x39
an=clim
veget=0
parallel=0
mpi=2
omp=2
machine=local
rrtm=1

############################################################
# 1. Model Compilation (parallel/sequential; consistent with install)
############################################################
```

init.sh (2/2)

```
# 2. Creating initial state and boundary conditions
#####################################################################
        2.1  Getting input files from the web
        ....................................................
   In TUTORIAL, it creates the directory INITIAL
   It copies in it the necessary files : ECDYN.nc, Albedo.nc, Relief.nc, Rugos.nc,
     landiceref.nc, amipbc_sic_YYYY.nc,amipbc_sst_YYYY.nc
        ....................................................
        2.2 Running ce0l.e  (output listing in ce0l.out)  :
It produces initial files start.nc, startphy.nc, and boundary cond. limit.nc
        ....................................................
        2.3 Creating a figure for the grid : grille.pdf
        ....................................................
        (2.4 : if veget=1 -> creating preliminary simulation SIMU0
         to produce the initial files start, startphy and sechiba_rest_in)

# 3. Creating a simulation directory
#########################################################
   In TUTORIAL, it creates the directory SIMU1
   It copies in it the DEF/*def files, and creates links to other necessary files
        (TUTORIAL/gcm.e, TUTORIAL/INITIAL :start.nc, startphy.nc and limit.nc )

# 4. Issuing instructions for running the simulation SIMU1 (and SIMU0 if veget=1)
#########################################################
```

Steps for setting up a simulation using "all-in-one" tutorial.tar

(see : https://www.lmd.jussieu.fr/~lmdz/pub/Training/Tutorials/Tutorial_2.pdf)

0) Download and unpack tutorial.tar

1) Check *.def files, set your desired parameters
Here in particular : the zoom parameters in gcm.def

2) Check/modify setup parameters in init.sh script : grid_resolution, veget, parallel

3) run : ./init.sh ; pay attention at its final instructions about how to run the model

4) check the results :
- visualize grille.pdf , or plot "grille_s" variable from INITIAL/grilles_gcm.nc
- verify that start.nc, startphy.nc and limit.nc files were created in TUTORIAL/INITIAL and the links to those files in TUTORIAL/SIMU1 are OK
IF NOT : Ask for Help

(*Possible Solution :* ulimit -s unlimited , and in TUTORIAL/INITIAL run : ./ce0l)

Now you can run the model : in SIMU1, run : ./gcm.e
and visualize the results : output files in SIMU1: histhf.nc and histday.nc

Proposed exercises :

Nudging

1D

Different physics

Tracers

Parallelism

XIOS

Coupling with surface schemes **(b**ucket, **O**rchidee)