Wikiprint Book

Title: 1. Create submission directory

Subject: lgcmg_doc - Doc/Setup

Version: 104

Date: 04/23/24 20:58:31

Table of Content

Simulation setup

This chapter describes how to setup your simulation.

1. Create submission directory

The configuration directory contains tools to compile (Makefile and AA_make) and tools to run a simulation, i.e.

- two directories (EXPERIMENTS and GENERAL) allow you to create submission directories for your model configuration;
- if one or several submission directories (e.g. EXP00, OOL_SEC_STO, historical, EXP_AER, etc...) have already been created, you can directly go to the next step.

In the EXPERIMENTS directory you will find subdirectories for each model configuration (included the one you work with). For example:

- IPSLCM5_v5 includes LMDZOR, LMDZ
- IPSLCM5CHT_v5 includes IPSLCM5, LMDZOR, LMDZORINCA
- LMDZOR_v5 includes LMDZ
- LMDZORINCA_v5 includes LMDZOR, LMDZ

Each of these subdirectories may contain a reference experiment (e.g. clim and amip for LMDZOR, NMHC_AER, AER and GES for LMDZORINCA, piControl, historical and cie for IPSLCM5_v5) and the file config.card which will be your simulation's initial setup.

Before preparing the working directory you must know which kind of simulation you want to perform. Then, you must copy the config.card file at the same level as the main Makefile.
For example, to perform a clim experiment with LMDZOR_v5:

```
cd modipsl/config/LMDZOR_v5
cp EXPERIMENTS/LMDZOR/clim/config.card .
```

The header of config.card contains the **JobName** field for which you must specify your simulation's name. Then run the ins_job script that will in first time ask you, if you are working on TGCC, your id group, and then create a directory for your experiment. If you are working on IDRIS it will directly create a directory for your experiment.
In the previous example, a simulation called *DIADEME* is created:

```
cd modipsl/config/LMDZOR_v5
cp EXPERIMENTS/LMDZOR/clim/config.card .
ls
 AA_Make Makefile EXPERIMENTS GENERAL config.card
vi config.card    # Change JobName=DIADEME
../../util/ins_job
ls
 AA_Make Makefile EXPERIMENTS GENERAL DIADEME
```

The config.card file is deleted and a directory called DIADEME is created.

2. Contents of the submission directory

The contents of the new directory are described below.

```
cd DIADEME
ls
 config.card COMP/ PARAM/ POST/ DRIVER/
```

2.1. config.card

The config.card file contains the settings of your simulation configuration. The file contains several sections with the simulation settings (e.g. name, duration, processors' number, post processing, initial state).
Below is a list of the file sections:

2.1.1. The [UserChoices] section

- JobName --> simulation name
- ExperimentName --> experiment name (following the CMIP5 nomenclature for the IPCC simulations)
- SpaceName --> allows you to distribute the simulations of a same experiment on different spaces TEST, DEVT, or PROD
- LongName --> description of your simulation
- TagName --> **do not change this field**; describes to which configuration family your experiment belongs
- ExpType --> **do not change this field**; allows you to find the EXPERIMENTS directory in which you are working
- DateBegin --> simulation start date (yyyy-mm-dd)
- DateEnd --> simulation end date. It must be the last day "included" in your simulation
- PeriodLength --> frequency of the executable run. This parameter can be 1M, 1Y or 10Y
- JobNumProcTot --> number of processors required by your simulation. This parameter depends on the configurations and on the resolutions. For example, a configuration containing LMDZ in 96x95 can use 32 processors at the most.

The parameters ExperimentName and SpaceName are optional. They will only impact the storage architecture of your outputs.
For example :

```
JobName=DIADEME
ExperimentName=REINE
SpaceName=TEST
TagName=LMDZOR


--> the output will be stored in : IGCM_OUT/LMDZOR/TEST/REINE/DIADEME

JobName=DIADEME
TagName=LMDZOR


--> the output will be stored in: IGCM_OUT/LMDZOR/DIADEME
```

The character "_" is not allowed in the variables JobName, ExperimentName and SpaceName

if SpaceName=TEST all output will be store on scratchdir (on curie) or workdir (on ada)

2.1.1.1. PeriodLength

The parameter PeriodLength allows you to determine the continuous length of a simulation period for your configuration, i.e. the frequency at which restart files are created. **The maximum PeriodLength for any LMDZ model is 1M**.

2.1.2. The [Restarts] section

The Restarts section allow to start from an existing simulation. This simulation can be found at the archive machine or at local scratch- or workdir.
Activate by setting **OverRule=y**. All components(e.g. ATM, SRF, etc) will then use the same simulation as restart state.

```
[Restarts]
OverRule=y
RestartDate=1999-12-31                                   # Last day of the experience used as restart for all components
RestartJobName=EXP00                                     # Define restart simulation name for all components
RestartPath=${ARCHIVE}/IGCM_OUT/IPSLCM5A/DEVT/pdControl   # Path Server Group Login
```

The root path for the **RestartPath** depend on the computing center. They are:

```
${ARCHIVE}                          # The storage machine of the computing center
                                    # (CCCSTOREDIR or GAYA). This space can contain
                                    # tar of restarts or
```

```
                                        # usual restarts files
/ccc/store/cont003/dsm/login            # TGCC
/u/rech/ces/login                       # IDRIS

${SCRATCHDIR}                           # The large TGCC workspace (no backup)

/ccc/scratch/cont003/dsm/login          # This kind of space can contain
                                        # usual restarts files

${WORKDIR}                              # The large IDRIS workspace (no backup)

/workgpfs/rech/ces/login                # This kind of space can contain
                                        # usual restarts files
```

libIGCM manages the difference in treatment between a path pointing to restart files that are directly accessible (without pack) and a path pointing to restart files that are in tar format (after pack). The management is made according to the path you provided.

2.1.3. The [ATM], ..., sections of the model components

This section for each of the model components allows you to:

- **define the output frequency**;
- **define whether this component is installed** which will only be considered if you specified **OverRule=n** in the [Restarts] section.

The possible settings for the **RestartPath** options are the same as for the [Restarts] section.

The possible settings for the **WriteFrequency** options are:

- 1M (monthly)
- 5D (5-day)
- 1D (daily)
- HF (6-hour high frequency)
- HF3h (real-time 3-hour frequency - specific to LMDZ)
- HF3hm (3-hour averaged high frequency - specific to LMDZ)
- STN (instantaneous output only for the CFMIP stations - specific to LMDZ).

```
[ATM]
WriteFrequency="1M 1D"                                      # Activate the writing frequency of this component
Restart=y                                                   # If config_Restarts_OverRule == 'n' next 4 params are read
RestartDate=1999-12-31                                      # Last day of the experience used as restart for this component if
RestartJobName=piControl25                                 # Define restart simulation name for this component
RestartPath=${ARCHIVE}/IGCM_OUT/IPSLCM5A/PROD/piControl # Path Server Group Login
```

WriteFrequency specific to the model components

- **LMDZ** : ([ATM]) Each of the frequencies settings 1M, 1D, HF, HF3h, HF3hm, and STN correspond to a given output file. For example, if you specify *1M*, a histmth.nc file will be created. If you want to change the output frequency in the histmth file you must change the corresponding lmdz parameter file. See here.
  ORCHIDEE
  - **[SRF]** : The first frequency corresponds to the output frequency for the sechiba_history.nc file. The available frequencies are: xY, xM, 5D, 1D and xs, where x is an integer and s means seconds. This file is required. If you add HF, a second sechiba_out_2.nc file will be written with the 3H frequency.
  - **[SBG]** : Only one frequency (xY, xM, 5D, 1D or xs) can be specified. It applies to the stomate_history.nc file. The stomate_history_ipcc.nc file always contains daily outputs.
- **INCA** : the section WriteFrequency does not work. Click here to learn more about how to change the writing frequency.

2.1.4. The section [Executable]

This section contains one line for each model component giving the executable's name in the bin/ directory and the executable's name copied to the working directory. You should only change this section if your executable is running in parallel using MPI and OpenMP or if you have changed the

executable's name.

```
[Executable]
#D- For each component, Real name of executable, Name of executable for oasis, optional : number of MPI task, number of Op
ATM= (gcm.e, lmdz.x)
SRF= ("", "")
...
```

Note : (",") indicates that this component has no executable. It is defined in a library linked to another executable (e.g. Orchidee in LMDZOR or Inca in LMDZINCA).

2.1.5. The [Post] section

The options of the [Post] section will allow you to set or disable the frequencies for submitting post processing jobs by changing the 5 following options (see the diagram below).

If you do not wish to run post processing jobs, you must specify **NONE** for both **TimeSeriesFrequency** and **SeasonalFrequency**.

RebuildFrequency and **PackFrequency** should not be disabled except in the case of running in expert mode.


```
RebuildFrequency=1Y          # Frequency of rebuild submission (use NONE for DRYRUN=3)
PackFrequency=1Y             # If absent default to RebuildFrequency.
TimeSeriesFrequency=1Y       # Frequency of post-processing submission (NONE if you don't want)
SeasonalFrequency=2Y         # Seasonal average period (NONE if you don't want,
                             # 2Y at least, 10Y by default)
SeasonalFrequencyOffset=0    # Offset for seasonal average first start dates ;
                             # same unit as SeasonalFrequency
```

2.2. COMP directory

This directory contains the architecture (or map) of each model component. Each map specifies inputs and outputs required by a component.

Input files of each component are organized into different **sections**.

- **[UserChoices]** contains specific options. --> used by the component's drivers (e.g.: lmdz.driver)
- **[InitialStateFiles] Initial conditions** files such as vegetation maps, topography,... --> retrieved by the IGCM_comp_GetInputInitialStateFiles function
- **[BoundaryFiles] Boundary conditions** files such as forcings or a LAI --> retrieved by the IGCM_comp_GetInputBoundaryFiles function
- **[SmoothFiles] Time-varying boundary conditions** files such as aerosols --> retrieved by the IGCM_comp_GetInputSmoothFiles function
- **[ParametersFiles] Parameters** files such as namelist or the run.def file --> retrieved by the IGCM_comp_GetInputParametersFiles function
- **[RestartFiles] Restart** files --> retrieved by the IGCM_comp_GetInputRestartFiles function

2.2.1. The [UserChoices] section

Contains several options which change the simulation setup by *drivers* files of the components (lmdz.driver, opa9.driver, ...). For example :

```
[UserChoices]
# Physics package to use :
# LMDZ_Physics=AP for standard/old physics(default), can be used with LMDZ4_AR5 or LMDZ5/trunk sources
# LMDZ_Physics=NPv3.1 for new physics, to be used with LMDZ5/trunk revision 1554 or later
LMDZ_Physics=AP
```

See the description for LMDZ here.

2.2.2. The [InitialStateFiles] section

Files needed to create initial files. This section is not activated if you chose to start or continue from an existing simulation (Section [Restart] in config.card). The files in this list will be only copied at the startup of your simulation.

```
# ----------------------------------------------------------------
#D- Get initial state (Etat0, carteveg,relief...)
#D- READ AND USE BY GCM FOR ONLY FOR THE FIRST EXECUTION.
# ----------------------------------------------------------------
[InitialStateFiles]    # IGCM_comp_GetInputInitialStateFiles from main Job
List= (SOURCE, DESTINATION)
```

2.2.3. The [BoundaryFiles] section

The files containing the boundary conditions are copied to the working directory.

The files in the **List** list will be copied at each integration period (one 1-month integration per period in general). A job can consist of several periods (PeriodNb).

The files in the **ListNonDel** list will only be copied for the first period of each job. These files will be accessible but will not change during the simulation.

```
# ----------------------------------------------------------------
#D- Get Boundaries Conditions (SST, WIND[X,Y,Z], LAI ...)
#D- READ AND USE BY GCM AT EACH EXECUTION.
# ----------------------------------------------------------------
[BoundaryFiles]        # IGCM_comp_GetInputBoundaryFiles
List=    (SOURCE, DESTINATION)
ListNonDel= (SOURCE, DESTINATION)
```

Be very careful : if there is any space at the end of a line, libIGCM will not taking in account the next line in the list

2.2.4. The [SmoothFiles] section

These are also files containing boundary conditions but their retrieval is only done at specific time integrals and it is not systematic. **1:12:** means that the file will be copied to the working directory at the first integration step and then every 12 iterations until the simulation is finished.

```
# ----------------------------------------------------------------
#D- Get SmoothFiles Conditions (SST, WIND[X,Y,Z], LAI ...)
#D- READ AND USE BY GCM AT EACH EXECUTION but varying in time
# ----------------------------------------------------------------
[SmoothFiles]          # IGCM_comp_GetInputSmoothFiles
List= (SOURCE, DESTINATION, FREQUENCE DE COPIE)
```

2.2.5. The [ParametersFiles] section

The parameter files of the component (namelist, run.def,...)

```
# ----------------------------------------------------------------
#D- Get parameters text files updated by job (.def, namelist ...)
#D- READ AND USE BY GCM AT EACH EXECUTION.
# ----------------------------------------------------------------
[ParametersFiles]      # IGCM_comp_GetInputParametersFiles
List=    (SOURCE, DESTINATION)
```

2.2.6. The [RestartFiles] section

The files providing the restart data. **You must not change this section** it is needed to link the jobs.

```
# ----------------------------------------------------------------
#D- Get restart files (restartphy.nc, orca_restart.nc ...)
#D- READ AND USE BY GCM AT EACH EXECUTION.
# ----------------------------------------------------------------
[RestartFiles]         # IGCM_comp_GetInputRestartFiles
List=    (MODEL OUTPUT NAME, ARCHIVED NAME, MODEL INPUT NAME)
```

2.2.7. The [OutputText] section

This section contains text files which will be produced during the simulation and model input parameter files. You might want to save these files.

```
[OutputText]
List= (NAME OF TEXT1 FILE, NAME OF TEXT2 FILE ....)
```

This files will be saved in tar stored in the output directory

- TGCC :

$STOREDIR/IGCM_OUT/TagName/[SpaceName]/[ExperimentName]/JobName/DEBUG

- IDRIS :

$GAYA*IGCM_OUT/TagName/[SpaceName]/[ExperimentName]/JobName/DEBUG*

2.2.8. The [OutputFiles] section

The netcdf files produced by the simulation are listed in this paragraph. This paragraph is associated with the [Post_*] sections.

```
[OutputFiles]
List = (OUTPUT_FILE_NAME, SAVE_PATH, POSSIBLE ASSOCIATED POST PROCESSING)
```

Refer to this chapter to learn everything about this section.

2.3. DRIVER directory

This directory contains the different drivers (predefined libIGCM functions for the component) of the different configuration's components. These drivers modify the parameter files of each component (.def, namelist, ...) setting the integration times, the outputs, and the forcing files.*

Note : If this directory does not exist the driver files are located in the COMP directory.

2.4. PARAM directory

This directory contains input text files for the configuration's components.

2.5. POST directory

This directory contains configuration files for additional diagnostic output. Click here for more details.

---

3. Set up initial state for the simulation

When you setup a simulation make sure that the list of input files in each card file of the model components and the selected options correspond to your experiment.

There are three different ways to define your simulation's initial conditions:

- *Start using restart files from an existing simulation by setting OverRule=y in the [Restart] section of the config.card file*
- *Start using different restart files from different simulations for each model component by setting Restart=y in each associated part of the config.card file. OverRule=n must be set in config.card.*
- *Use the default section InitialStateFiles in the comp.card file of the model components in config.card you must have OverRule=n and and Restart=n for this case.*

3.1. Example for different restart

3.1.1. Example with OverRule=y

If you wish to use the start state of a given simulation, set in config.card:

```
#========================================================================
#D-- Restarts -
[Restarts]
#D- If you want a GENERAL RULE FOR ALL COMPONENTS RESTARTS, put this flag to 'y'
OverRule=y
#D- Last day of the experience used as restart
RestartDate=1869-12-30
#D- Define restart simulation name
RestartJobName=CD1
#D- Path Server Group Login
RestartPath=${ARCHIVE}/IGCM_OUT/IPSLCM5A/DEVT/pdControl
```

For the same case but if the simulation was performed by someone else, you must give the complete path of the directory, for example:

```
RestartPath=/u/rech/lab/plabxxx/IGCM_OUT/IPSLCM5A/DEVT/pdControl
# or RestartPath=/dmnfs/contxxx/login/IGCM_OUT/IPSLCM5A/DEVT/pdControl
```

3.1.2. Example with OverRule=n and [COMP]/Restart=y

You can also distinguish the setup parameters for each model components. Set `OverRule=n` and use the `Restart`, `RestartDate`, `RestartJobName` and `RestartPath` parameters for each model component (section). For example, use restart files for the atmosphere but not for the surface component. For the surface component the InitialStateFiles will then be used :

```
#D-- ATM -
[ATM]
#
WriteFrequency="1M 1D HF"
# If config_Restarts_OverRule == 'n' all params are read
Restart= y
# Last day of the experience used as restart for this component
RestartDate=1999-12-30
# Define restart simulation name
RestartJobName=2L18
RestartPath=${ARCHIVE}/IGCM_OUT/IPSLCM5A/DEVT/pdControl
#
#D-- SRF -
[SRF]
#
WriteFrequency="1M"
# If config_Restarts_OverRule == 'n' all params are read
Restart= n
# Last day of the experience used as restart for this component
RestartDate=1999-12-30
# Define restart simulation name
RestartJobName=2L18
RestartPath=${ARCHIVE}/IGCM_OUT/IPSLCM5A/DEVT/pdControl
```

3.2. Note for LMDZ

To obtain exactly the same outputs in different simulations, you must choose the same LMDZ `Bands` files. This is explained in `COMP/lmdz.card` with the `LMDZ_NbPeriod_adjust` and `LMDZ_Bands_file_name` parameters.

```
LMDZ_NbPeriod_adjust=0
# To force the use of this Bands file, set LMDZ_NbPeriod_adjust=0 and replace XXXXXXX by Restart Job Name
LMDZ_Bands_file_name=${ARCHIVE}/IGCM_OUT/IPSLCM5/CEPRO0/ATM/Debug/CEPRO0_Bands_96x95x39_3prc.dat_3
```

Click here for more details.

4. Main job of the simulation

The main job contains scripts that will be executed by the system. With libIGCM, this job is unique (in the beginning AA_job and later Job_MYJOBNAME) for all type of configurations. It contains all scripts to initialize a simulation, to summarize the chosen model configuration and to run identical experiments for all model components. It resubmits itself in order to continue the simulation if needed.

The job header depends on the machine type. It contains the job name and the parameters. **Real-times** must be chosen to match the specific classes for the computing machine and according to the simulation length (test or production).

At TGCC you must specify the project number: [#MSUB -A MY_PROJECT](#).

You should change the PeriodNb parameter in the job to change the number of runs in one job (see the example of computation in the next section) :
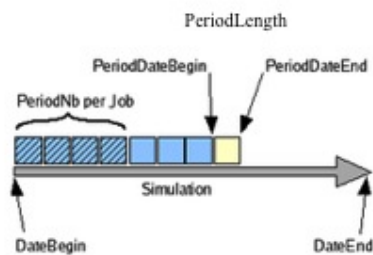
```
#D- Number of execution in one job
PeriodNb=1
```

In some cases, you must set a variable RUN_DIR_PATH in order to avoid deleting the working directory.

```
#D- Define running directory
#D- Default=${TMPDIR} ie temporary batch directory
#RUN_DIR_PATH=/workdir/or/scratchdir/of/this/machine
```

Here is the diagram of the steps in AA_job :

4.1. Choosing PeriodNb

To avoid starting a lot of short jobs which might be queued, the production job starts n integrations (PeriodNb), whose length are <u>*PeriodLength.*</u>

These are calculated as followed:
Time limit = PeriodNb * max(Real time of a PeriodLength)

where Time limit is the requested time in the job header.

At the end of a simulation, the run.card file returns the used CPU time for each simulation step. This will allow you to perform this computation. It is therefore important, for each simulation with a new configuration, to perform a 1-3 month test to estimate beforehand the CPU time.

---

5. Prepare a new experiment

There are two ways to prepare a new working directory for your model configuration:

1. *Start again from the first step described above by copying the desired config.card file to your configuration directory using a new JobName.*
2. *Copy an existing submission directory, delete the files created by the* <u>*simulation*</u>*, and change JobName in config.card.*

For example:

```
cd modipsl/config/LMDZOR_v5
cp -r DIADEME CHOUCROUTE
cd CHOUCROUTE
rm -f Job_DIADEME run.card Script_Output_DIADEME.000001
vi config.card
  JobName=CHOUCROUTE
../../../util/ins_job
```

The ins_job script allows you to create a submission directory from a config.card file or if the directory already exists it allows you to only create the job corresponding to config.card. ins_job will not overwrite a directory or an existing job.

5.1. Post-processing jobs

Jobs headers for post-processing have to be carefully checked, especially elapsed time limits. They are in libIGCM directory (xxx.job) and are adapted for IPSLCM5A with 1Y for RebuildFrequency and PackFrequency. Change time limits if you use larger frequencies.