

Input files, forcing data

LMDZ tutorial
January 2023
Lionel GUEZ

Contents

- Introduction
- LMDZ requirements
 - Run-time parameters
 - Initial state and boundary conditions
 - Other forcing data
- Resources for input files

Introduction (1/2)

- This presentation: a few explanations, many pointers to web sites (please download this file so you can click on the links) and to paths on French super-computers.
- A large part of this presentation is a bird's eye inventory of input files for LMDZ.

Introduction (2/2)

- A lot of this, you do not have to know if you use `libIGCM`, because choosing and getting the necessary files will be automated.
 - Note: there are regular training sessions for `libIGCM`.
- The information here may be useful if you have non-standard needs.

Coupling with another model (1/2)

- Soil, vegetation, hydrology: [Orchidée](#)
- Ocean, sea ice, marine biogeochemistry: [Nemo](#)
- Tropospheric chemistry, aerosols: [INCA](#)
- Stratospheric chemistry: [Reprobus](#)
[Marchand et al., JASTP, 2012](#)

Coupling with another model (2/2)

- The sets of input files and input parameters vary with the set of models you couple.

Configurations of LMDZ itself

- 1 or 3 dimensions.
- Nudging or not.
- Realistic or idealized "physical" part of the model.
- The sets of input files and input parameters vary with the above choices.

LMDZ with realistic physics and surface

- LMDZ alone
- 3 dimensions
- No nudging
- Realistic (terrestrial) physics and surface (well, more or less: the continents and relief are there, but no Orchidee)

What is left to choose

- Many possible run-time parameters (for dynamics, physical parameterizations, concentration of greenhouse gases, choice of output)
- Initial state
- Boundary conditions: surface conditions and solar irradiance
- Forcing atmospheric data: ozone field, fields of aerosols

Run-time parameters in `.def` files

- Chosen in text files which have suffix `.def`
- Semantic separation of run-time parameters among six `.def` files (for LMDZ alone, without Orchidee)

INCLUDEDEF lines

- `run.def` is the first file read.
- Other `.def` files (except `tracer.def`) are read only if there is a corresponding `INCLUDEDEF` line in `run.def`. Cf. Distributed `run.def`. For example:
`INCLUDEDEF=physiq.def`

Syntax of `.def` files

(for all `.def` files except `tracer.def`)

- `variable=value`
- Case sensitive
- TRUE or FALSE for logical values
(you can also use `y` or `n`)
- Character values without quotes
- Comment lines start with `#`

The content of `.def` files (1/3)

- `gcm.def`: dynamics
- `physiq.def`: parameterizations
- `vert.def`: vertical discretization
- `config.def`: coupling with other models, forcing data (ozone, aerosols, greenhouse gases CO_2 , CH_4 , N_2O , CFC11, CFC12)

The content of `.def` files (2/3)

- `run.def`: length of the run, dates, calendar
 - Starting date of the first run in a sequence of runs: `dayref` (day number in the year) and `anneeref` (year). You do not change these when you continue a simulation, starting from the end of a previous simulation.
 - Calendar: `calend` (character variable). (See below when we talk about program `ce0l`.)

The content of `.def` files (3/3)

- `tracer.def`: tracers
 - Number identifying the advection scheme, possibly different for horizontal (`hadv`) and vertical advection (`vadv`)
 - Possible: `tracer of tracer`, useful for isotopes
 - For more information on available advection schemes: `dyn3d_common/infotrac.F90`
 - See [presentation on dynamics](#) by E. Millour.

Syntax for `tracer.def`

```
&version=1.0
&lmdz
default hadv=10 vadv=10 phases=g parent=air type=tracer
H20 hadv=14 vadv=14
H20 phases=ls
Aga
```

(4 tracers in this example)

```
&version=<number>
<section name>
default <key>=<val> ...
<tracer name> <key>=<val>
...
```


Syntax for `traceur.def`

- `traceur.def`: old file with old format for tracers
- First line: number of tracers
- Then one line per tracer, with a number identifying the advection scheme

```
4
14 14 H20v
10 10 H20l
10 10 H20i
10 10 Aga
```

```
<hadv> <vadv> <name> [<parent name>]
```

Be careful with `.def` files ! (1/2)

- Parameters have default values in the Fortran source files.
- →The `.def` files that are commonly used do not mention all the parameters.
- Be careful: the distribution of variables among `.def` files is not enforced by the software that reads the files.

Be careful with `.def` files ! (2/2)

- Also, be careful: if you assign values several times to the same variable, the program will not stop, the last assignment will be used.
- Also, be careful: you will not get an error message for a mistyped parameter name in your `.def` files, it will just be ignored.

List of all used parameters

- Each run creates a set of files `used_*.def`. So, if you want to see all the parameter values that were used for your run, be sure to check them.
- `Script` for easier comparison of two sets of `.def` files

Choice of output

- At run-time
- In text files, in XML format (suffix `.xml`)
- You choose which variables to output, at what frequency, with which operation applied to them (average, maximum...).
- See [presentation on output](#).

Initial state and boundary conditions (1/4)

- Separate program: ce0l
(ce0l stands for "création état 0 limites")
- Distributed with the program gcm,
compiled with the same tools

For example:

```
make lmdz_fcm -d 144x142x79 ... ce0l
```

instead of:

```
make lmdz_fcm -d 144x142x79 ... gcm
```

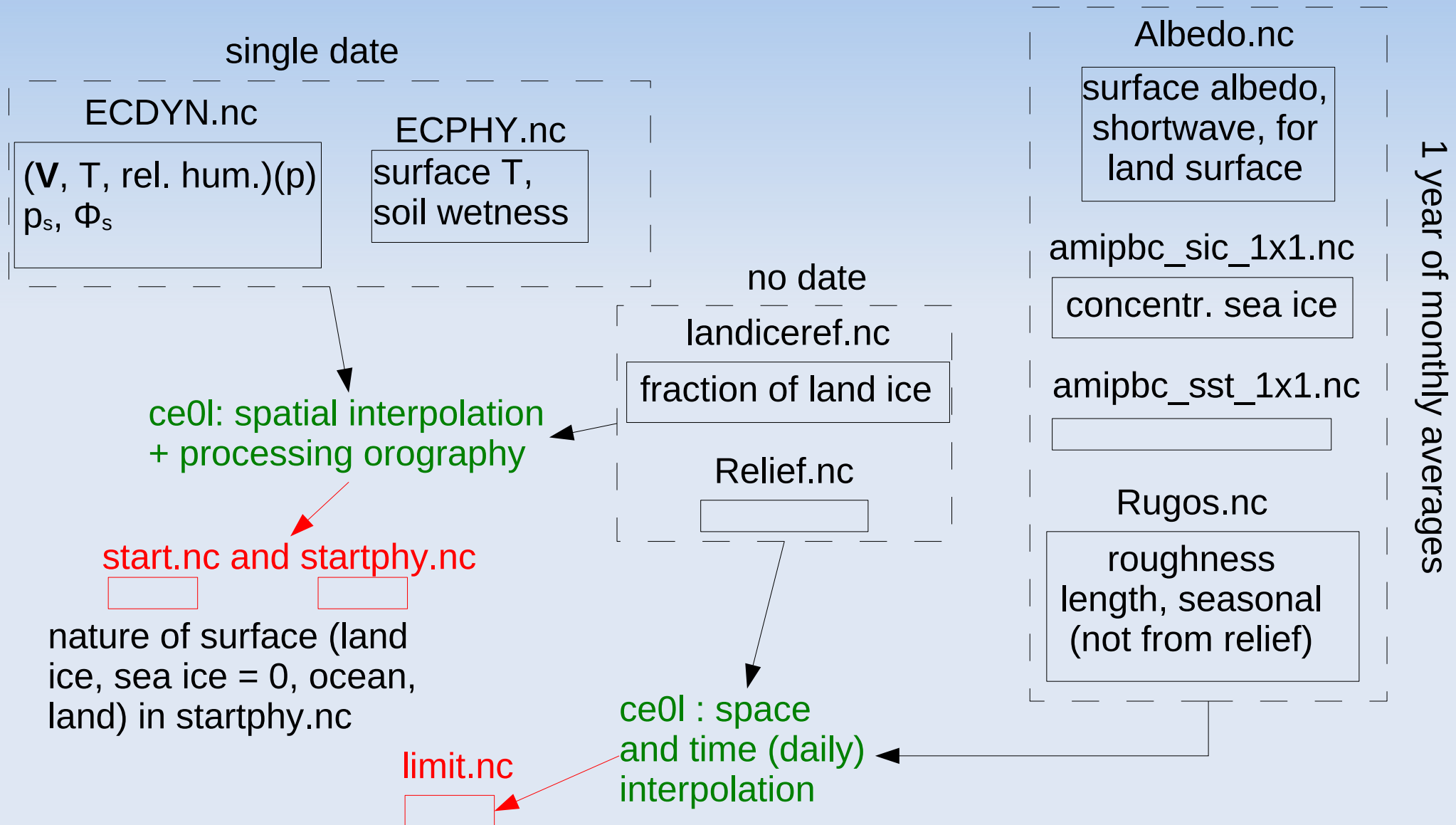
Initial state and boundary conditions (2/4)

- Run ce0l with the same `.def` files than those you want for gcm.
- ce0l creates 2 files for initial state:
`start.nc` for the dynamics part of gcm
`startphy.nc` for the physics part of gcm
- ce0l creates 1 file for 1 year of boundary conditions:
`limit.nc`

Initial state and boundary conditions (3/4)

- Cf. `scripts` to see the content of `limit.nc` or `startphy.nc`.

Initial state and boundary conditions (4/4)



Calendar (1/2)

- Calendar is used in ce0l to create boundary conditions
- For years with constant length, 360 or 365 days:
calend=earth_360d
or
calend=earth_365d

Calendar (2/2)

- For the real calendar (with leap-years), run `ce0l` for each year you want boundary conditions for:
`calend=gregorian`
`anneeref=<the right year>`

Using data from a coupled ocean-atmosphere run

- It is possible to create a file `limit.nc` using SST, sea-ice fraction and land-sea mask from a coupled NEMO-LMDZ run.
- See [documentation IGCMG](#).

Solar forcing: choice of radiative transfer model (1/2)

- Recommended settings:
 - Compile with option `-rad rrtm`
 - `iflag_rrtm = 1` in `physiq.def`
Means we use RRTM and not the old radiative transfer code.
 - `nsw = 6` in `physiq.def`
Number of spectral bands in shortwave domain.

Solar forcing: choice of radiative transfer model (2/2)

- Note: these are not the default settings (for the sake of backward compatibility) so make sure you specify them.

Solar irradiance at top of atmosphere (1/3)

- Either a constant grey solar constant:
`solaire = 1365` (for example)
`ok_suntime_rrtm = n`
in `config.def`

Solar irradiance at top of atmosphere (2/3)

- Or a spectrally and time varying solar irradiance:

`ok_suntime_rrtm = y` in `config.def`

Means we read from a NetCDF file solar irradiance at top of atmosphere as a function of day of the year and spectral band.

Solar irradiance at top of atmosphere (3/3)

- If `ok_suntime_rrtm = y` then gcm needs `solarforcing.nc`, containing variables `tsi`, which is the solar constant, and `ssi_frac`, which is the fraction in each of the nsw bands, daily, for 1 year (`solarforcing.nc` has to be prepared with the right number of days).

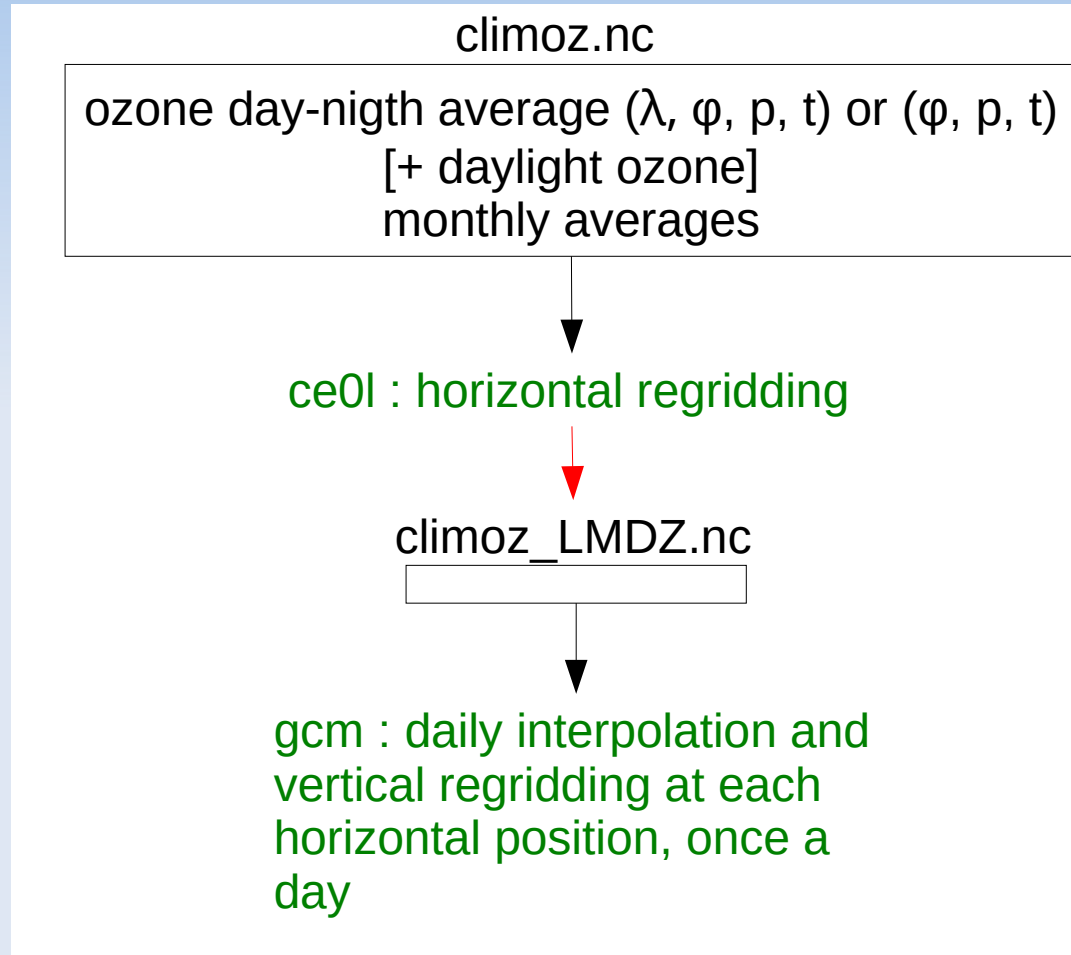
Ozone (1/3)

- Run-time parameter in config.def:
`read_climoz` = - 1, 0, 1 or 2
- 0: analytic expression for the ozone field
(see `phylmd/ozonecm_m.F90`)
- - 1 (use with `solarlong0` = 1000): ozone field made symmetric with respect to the equator, for use with annual mean insolation (used for aquaplanet runs)

Ozone (2/3)

- 1: read day-night average of ozone field from a NetCDF file (this is what you want for CMIP 6 input).
- 2: also read daylight average from the same file (good idea with Reprobus CMIP 5 input and if you run LMDZ with a few model layers in the mesosphere).

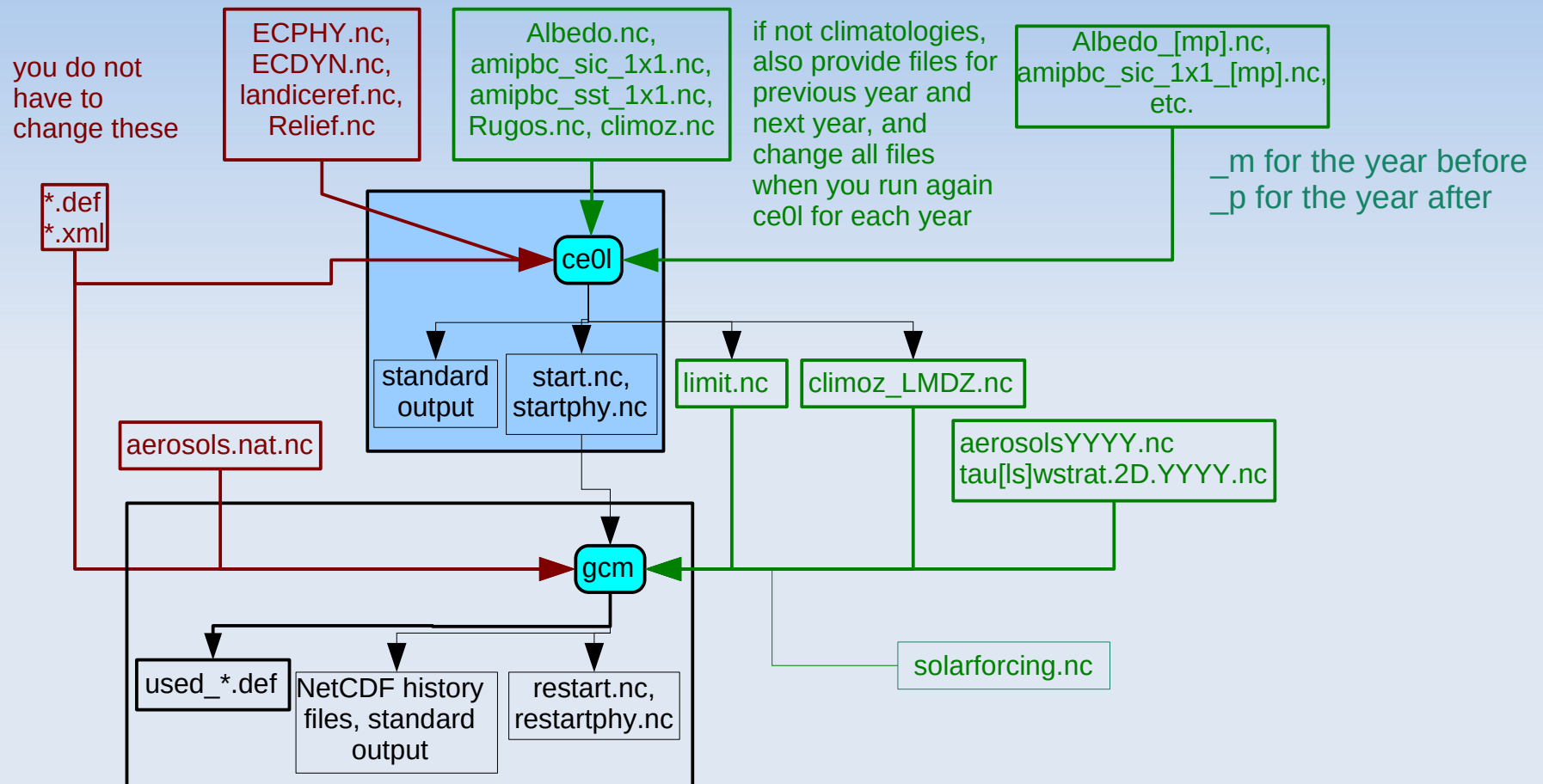
Ozone (3/3)



Aerosols

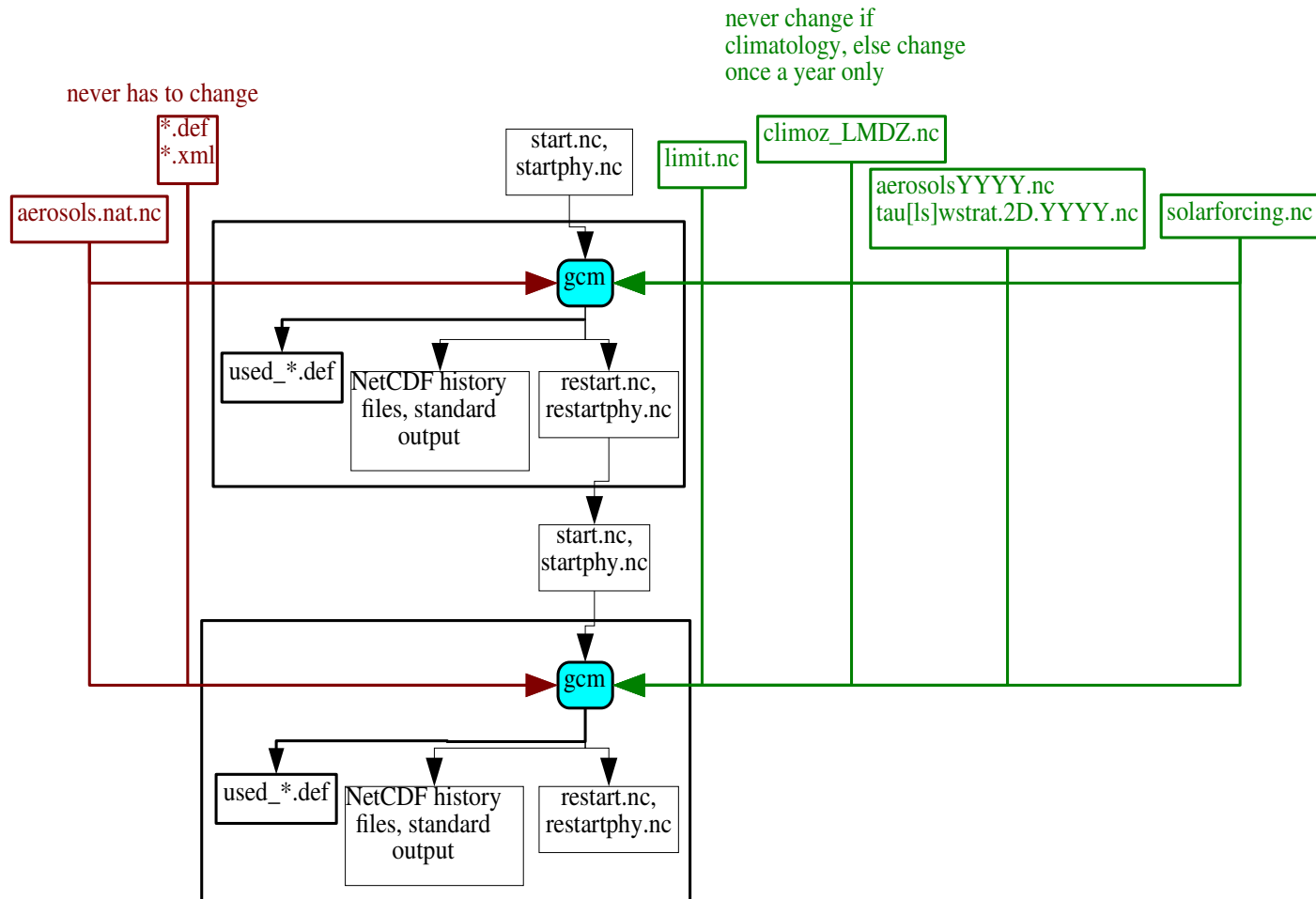
Cf. presentation by Olivier BOUCHER

Putting it all together



See also [Tutorial I. Installing the LMDZ model](#), page 13 for an alternate view.

Chaining runs



Restarting from an arbitrary date (1/3)

- Normally, `gcm` reads the initial date of the run from the file `start.nc`.
- You can set `raz_date = 1` in `run.def`. Then `gcm` uses `dayref` and `anneeref` in `run.def` for the initial date.

Restarting from an arbitrary date (2/3)

- This way, you can start `gcm` at any date from any `restart.nc` and `restartphy.nc`. But watch out for the initial shock if there is a large seasonal difference between the original date of the restart files and the reset date. The shock will come from the discrepancy between the state of the atmosphere and the boundary conditions and declination of the Sun.

Restarting from an arbitrary date (3/3)

- Do not forget to set `raz_date` back to 0 if you want to continue a simulation.

Where do you get the input files ?

Run-time parameters (1/2)

Template `.def` and `.xml` files are distributed with LMDZ source files

- Directory `DefLists`
- `physiq.def_AR4`: physics used for AR4 (old physics)
- `physiq.def_NPv3.2`: new physics of CMIP5
- `physiq.def_NPv6.1.3.newt2mq2m`: physics of CMIP6

Run-time parameters (2/2)

- More details on the history of LMDZ physical parameterizations in the presentation
[LMDZ configurations: uses and tuning](#)

NetCDF input files (1/3)

- See also: [Repository](#) for shared files.
- [Thredds](#) data server

NetCDF input files (2/3)

- If you use LMDZ at the national supercomputing centers [IDRIS](#) and [TGCC](#), or at the [IPSL computing center](#), NetCDF files required by ce01, as well as forcing data (ozone, aerosols, solar irradiance) used for CMIP5 and CMIP6, are available locally there.

NetCDF input files (3/3)

- The root of the arborescence containing all the input files is called **R_IN** in the following pages.
- **R_IN=**
 - `/gpfswork/rech/psl/commun/IGCM` at IDRIS
 - `/ccc/work/cont003/igcmg/igcmg/IGCM` at TGCC
 - `/projsu/igcmg/IGCM` on spirit (IPSL machines)

Boundary conditions (1/2)

- In subdirectory:
`$R_IN/ATM/LIMIT/AMIP.v20220201/original/
360x180`
you will find sea ice and sea surface
temperature from 1869 to 2021, from
`input4MIPS` (variables `siconcbcs` and
`tosbcs`)

Boundary conditions (2/2)

- In subdirectory:
`$R_IN/ATM/INPUT_CE0L`
you will find:
 - `landiceref_GA_HR.new.nc`, recommended for `landiceref.nc`
 - `Relief_GMTED2010_15n015_00625deg_plus_GA.nc`, recommended for `Relief.nc`
 - `Albedo.nc`, `Rugos.nc`

Initial state

In subdirectory:

`$R_IN/ATM/INPUT_CE0L`

you will find:

`ECDYN.nc` and `ECPHY.nc` for a single date. If

you need another date, see:

[Création d'un fichier ECDYN](#) (from ERA interim). Useful for "transpose AMIP" runs or weather forecast runs.

Forcing data for CMIP 5

A lot of data, for pre-industrial, historical or future simulations (with different scenarios).

Forcing data for CMIP 5: ozone

- `$R_IN/ATM/OZONE/HYBRIDE/original/v2.$scenario/tro3_${year}.new.nc`
(see definition of `R_IN`)
- For the scenarios:
 - clim: historical then RCP 8.5
 - RCP26, RCP45, RCP60
- Information on ozone files

Forcing data for CMIP 5: aerosols (1/2)

- `$R_IN/ATM/LMD$resolution/AR5/$scenario/aerosols_11YearsClim_${year}_v5.nc`
- For the resolutions:
 - 128×118
 - 128×88
 - 144×142 (directory `LMD144142`, not `LMD144X142`)
 - 280×280
 - 96×95 (directory `LMD9695`)

Forcing data for CMIP 5: aerosols (2/2)

- And for the scenarios:
HISTORIQUE, RCP26, RCP45, RCP60,
RCP85, esm2_experiment
- Each file contains all aerosol types.
- 1855 for aerosols.nat.nc
- **Informations** on runs of LMDZ-Orchidée-Inca which produced aerosol files

Forcing data for CMIP6

Information on greenhouse gases, tropospheric aerosols, solar irradiation, ozone, stratospheric aerosols and land use.

Forcing data for CMIP 6: ozone

- Ozone input files for ce0l:
`$R_IN/ATM/OZONE/UReading/
historical.v20160711.v2/original`
(see definition of `R_IN`)
- Origin: input4MIPs, variable `vmro3`
- Information on the
processing of ozone files at IPSL

Forcing data for CMIP 6: aerosols

- Aerosol input files from INCA:
\$R_IN/ATM/AEROSOLS/CMIP6/v1/\$resolution/
L79
- Stratospheric aerosols:
\$R_IN/ATM/STRATAERO/CMIP6/v3
- Information on the
processing of aerosol emission at IPSL
and the runs producing aerosol forcing.

Forcing data for CMIP 6: solar

- `$R_IN/ATM/SOLAR/CMIP6/v3.2corr`
- Information on input files