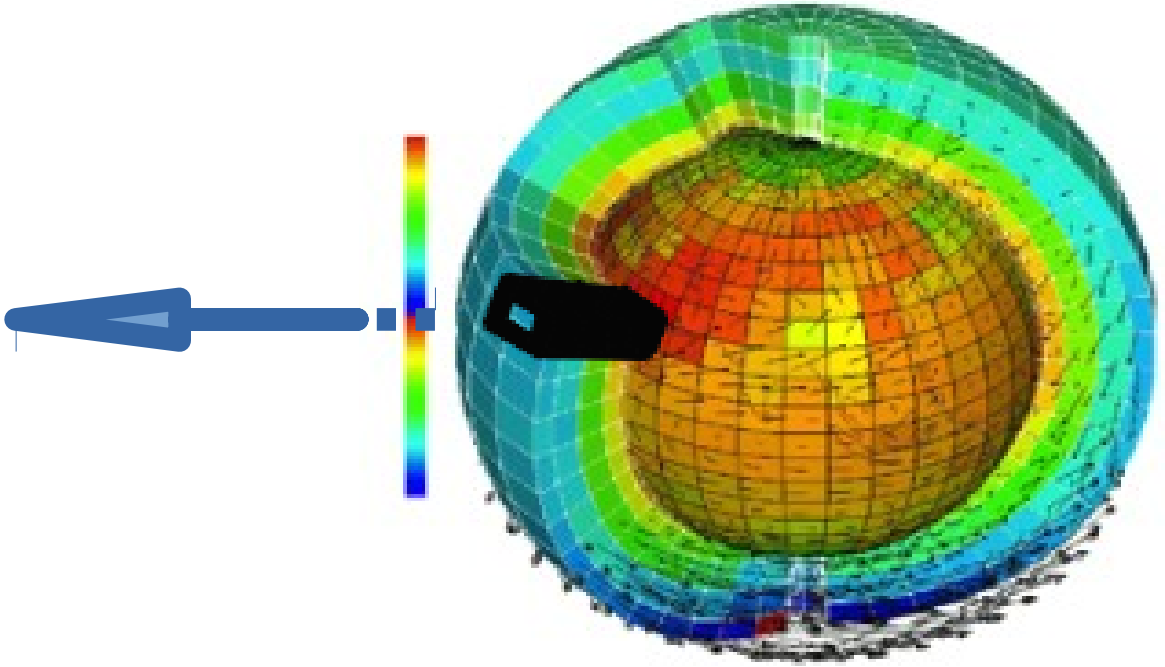
LMDZ Single Column Model



M-P Lefebvre and LMDZ team

How to install 1D model ?

cd LMDZ20201109.trunk

- If you have LMDZ20201109.trunk/**1D** directory, skip this slide
- If not :

wget http://www.lmd.jussieu.fr/~lmdz/pub/1D/1D.tar.gz

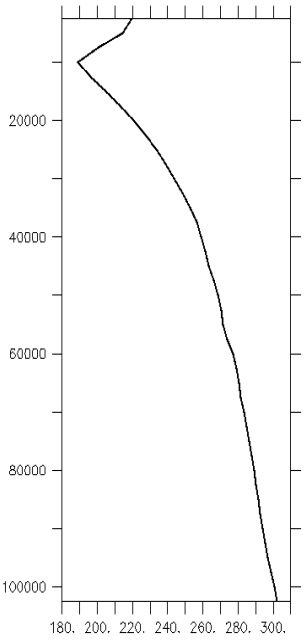
tar xvf 1D.tar.gz

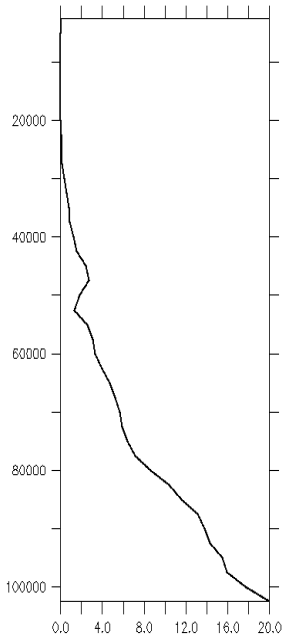
Now you have 1D directory avalaible

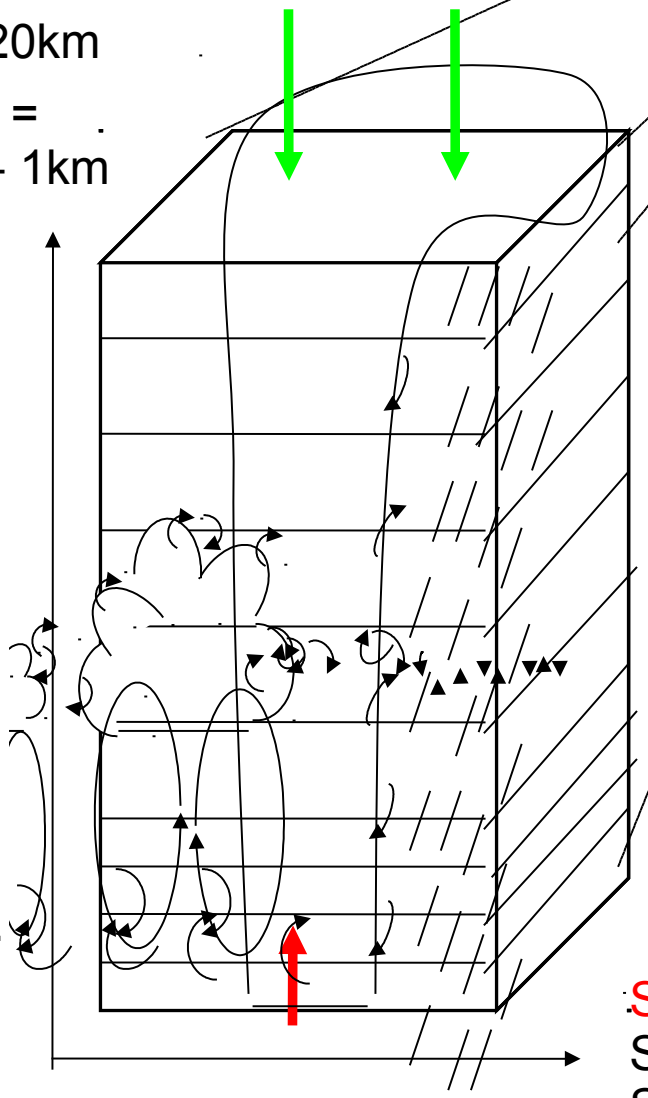
cd 1D

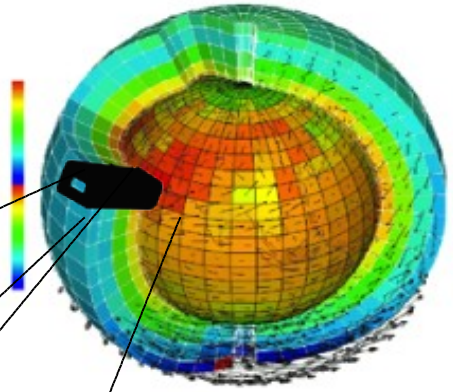
./run.sh

LMDZ in 1D mode









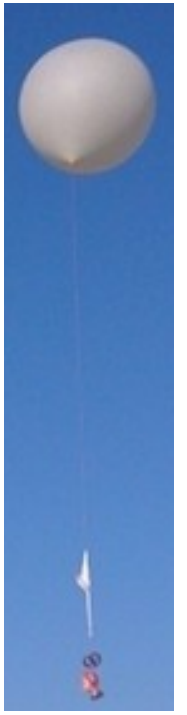
Large scale forcings
(constant or not):
Temperature, humidity,
Wind advection

Surface conditions:
Surface fluxes or
Surface temperature

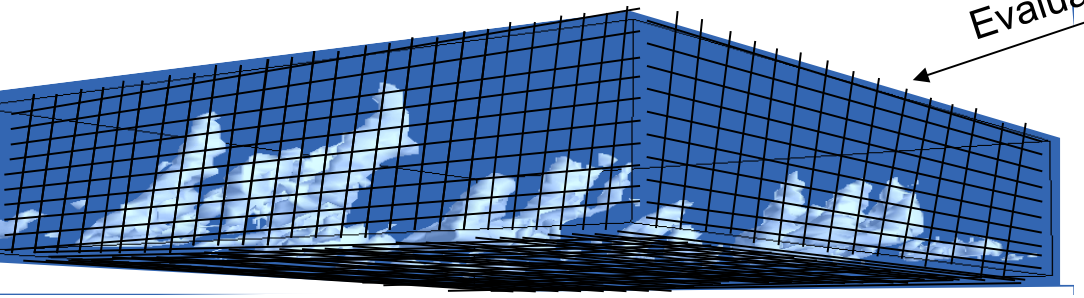
How 1D cases are built ?

Test case, field campaign experiment

Observation



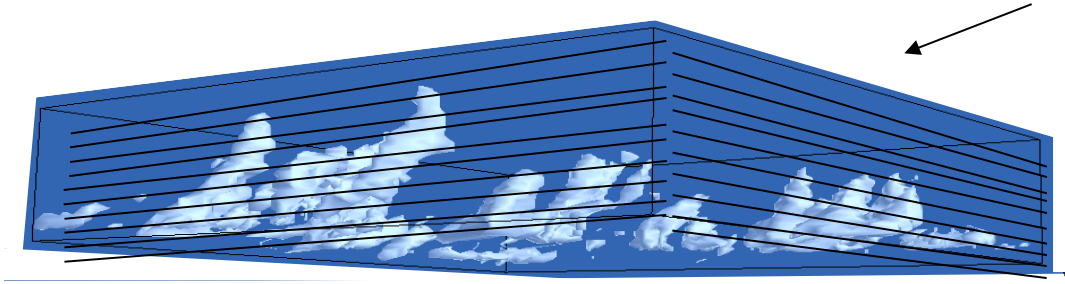
Evaluation



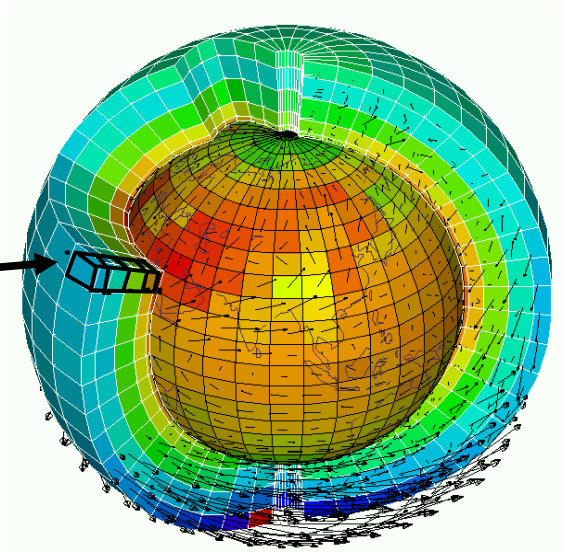
Explicit simulations, Grid cell, 20-100 m

Evaluation

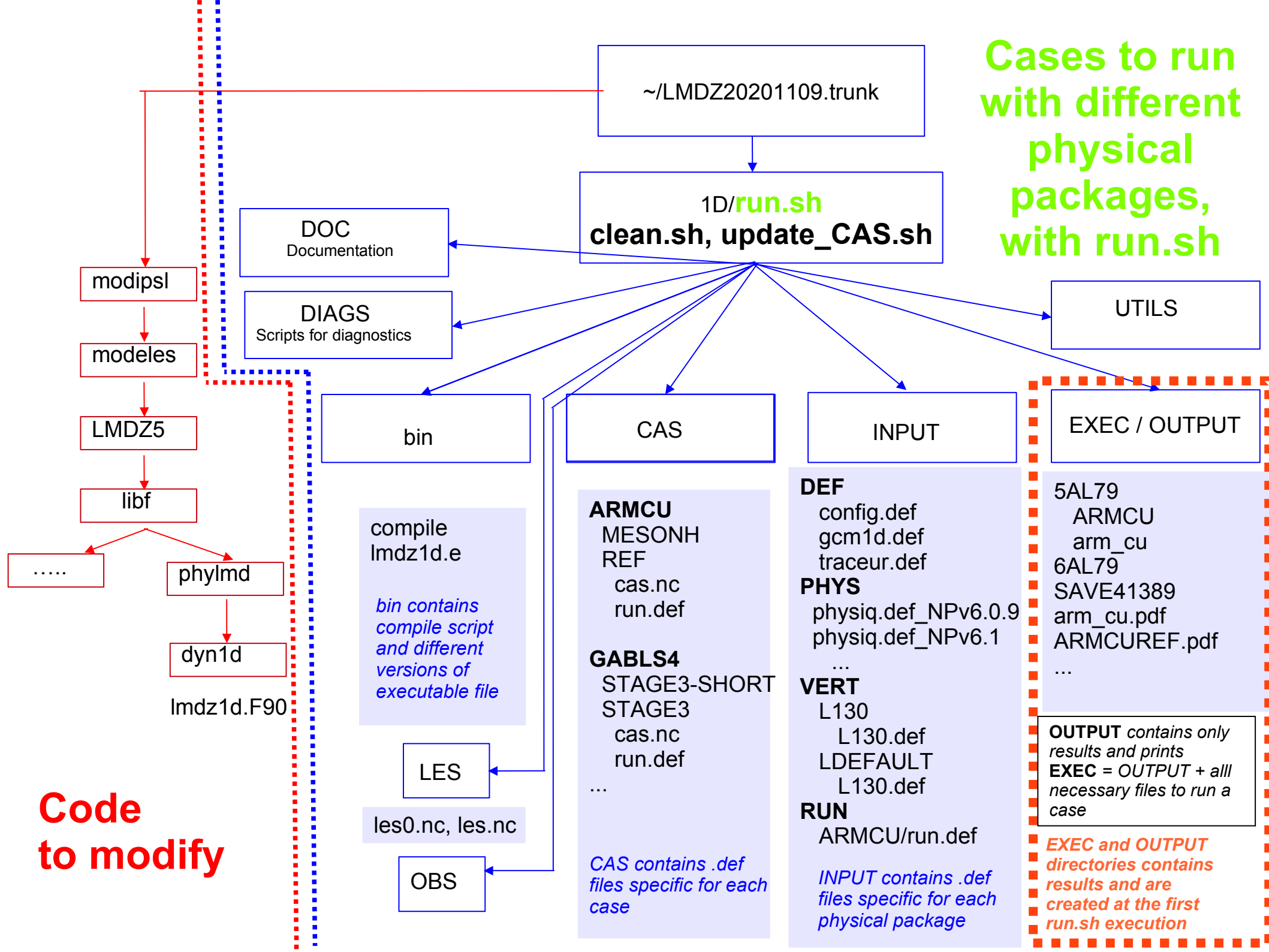
« Large scale » conditions imposed

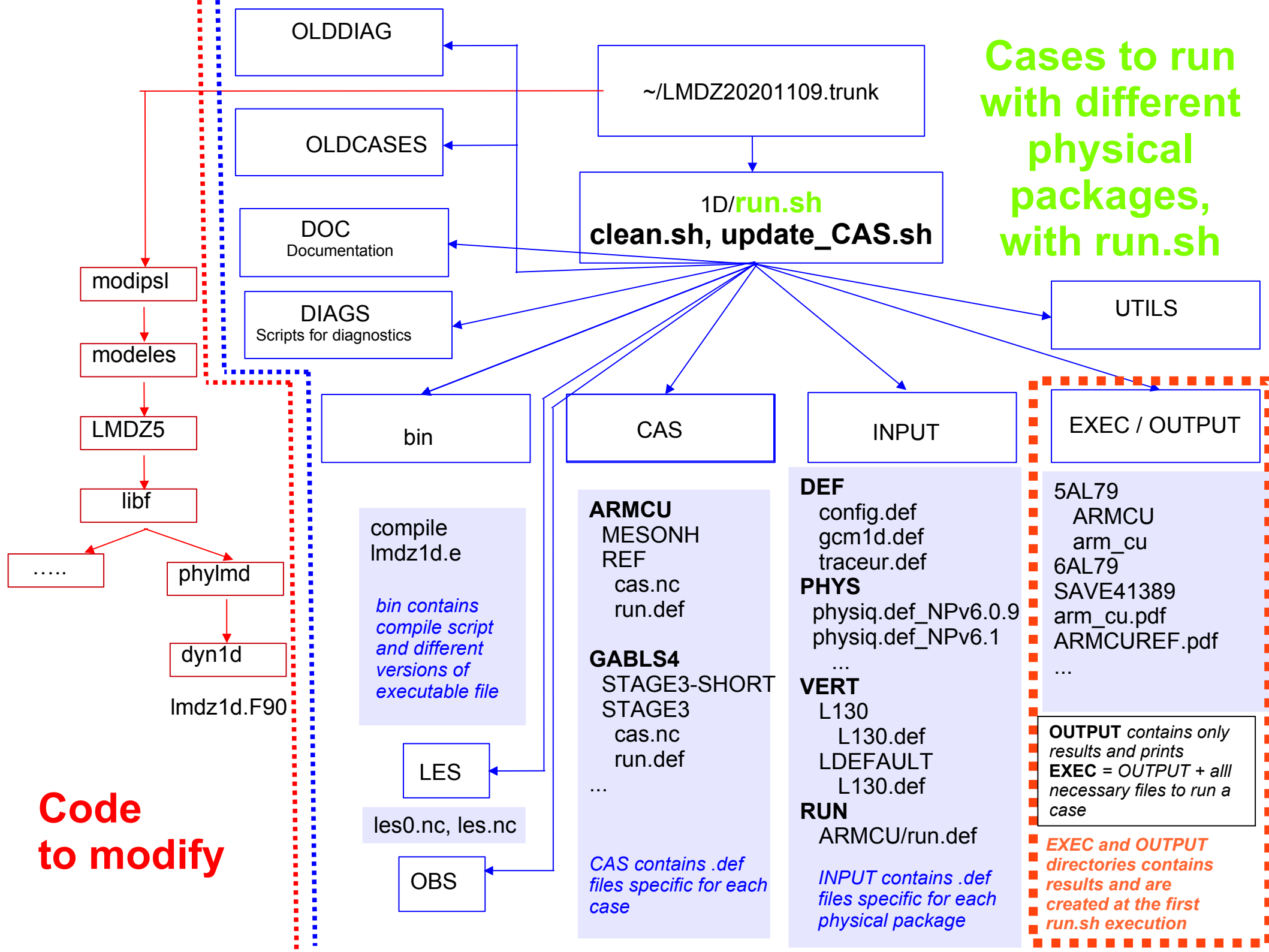


Climate model, parameterizations, « single-column » mode



Courtesy F.Hourdin


~/LMDZ20201109.trunk
1D/run.sh
clean.sh, update_CAS.sh
Cases to run with different physical packages, with run.sh
DOC
Documentation
DIAGS
Scripts for diagnostics
UTILS
modipsl
modeles
LMDZ5
libf
.....
phylmd
dyn1d
lmdz1d.F90
bin
compile
lmdz1d.e
bin contains compile script and different versions of executable file
LES
les0.nc, les.nc
OBS
CAS
ARMCU
MESONH
REF
cas.nc
run.def
GABLS4
STAGE3-SHORT
STAGE3
cas.nc
run.def
...
CAS contains .def files specific for each case
INPUT
DEF
config.def
gcm1d.def
traceur.def
PHYS
physiq.def_NPv6.0.9
physiq.def_NPv6.1
...
VERT
L130
L130.def
LDEFAULT
L130.def
RUN
ARMCU/run.def
INPUT contains .def files specific for each physical package
EXEC / OUTPUT
5AL79
ARMCU
arm_cu
6AL79
SAVE41389
arm_cu.pdf
ARMCUREF.pdf
...
OUTPUT contains only results and prints
EXEC = OUTPUT + alll necessary files to run a case
EXEC and OUTPUT directories contains results and are created at the first run.sh execution
Code to modify


OLDDIAG
~/LMDZ20201109.trunk
OLDCASES
1D/run.sh
clean.sh, update_CAS.sh
DOC
Documentation
DIAGS
Scripts for diagnostics
UTILS
modipsl
modeles
LMDZ5
libf
.....
phylmd
dyn1d
lmdz1d.F90
bin
CAS
INPUT
EXEC / OUTPUT
compile
lmdz1d.e
bin contains compile script and different versions of executable file
LES
les0.nc, les.nc
OBS
ARMCU
MESONH
REF
cas.nc
run.def
GABLS4
STAGE3-SHORT
STAGE3
cas.nc
run.def
...
CAS contains .def files specific for each case
DEF
config.def
gcm1d.def
traceur.def
PHYS
physiq.def_NPv6.0.9
physiq.def_NPv6.1
...
VERT
L130
L130.def
LDEFAULT
L130.def
RUN
ARMCU/run.def
INPUT contains .def files specific for each physical package
5AL79
ARMCU
arm_cu
6AL79
SAVE41389
arm_cu.pdf
ARMCUREF.pdf
...
OUTPUT contains only results and prints
EXEC = OUTPUT + alll necessary files to run a case
EXEC and OUTPUT directories contains results and are created at the first run.sh execution
Cases to run with different physical packages, with run.sh
Code to modify

Common input and output format

In the frame of project DEPHY, we've defined with Météo-France a common format for forcings and output files.

For cases which are up to date ~/CAS : ARMCU, AYOTTE, BOMEX, DYNAMO, GABLS1, GABLS4, IHOP, MPACE, RICO, SANDU, SCMS

+ for these cases, **specify ARMCU/REF** for example in run.sh

+ forcings file is always named **cas.nc**

+ output file is **hourly_std.nc**

+ even if histhf.nc or hourly.nc are also available

For the other cases in **~OLDCASES**/amma, arm_cu, bomex…

+ for these cases, **specify only arm_cu** for example in run.sh

+ forcings file is case_name.nc or prof.inp.001

+ output file is histhf.nc or hourly.nc

Have a look in run.sh and specify

The case(s) ?

listecas="arm_cu ARMCU/REF "

The physics ?

In this example :
You ask for **2** *cases and* **2** *physical packages : you'll get* **4 dataset**

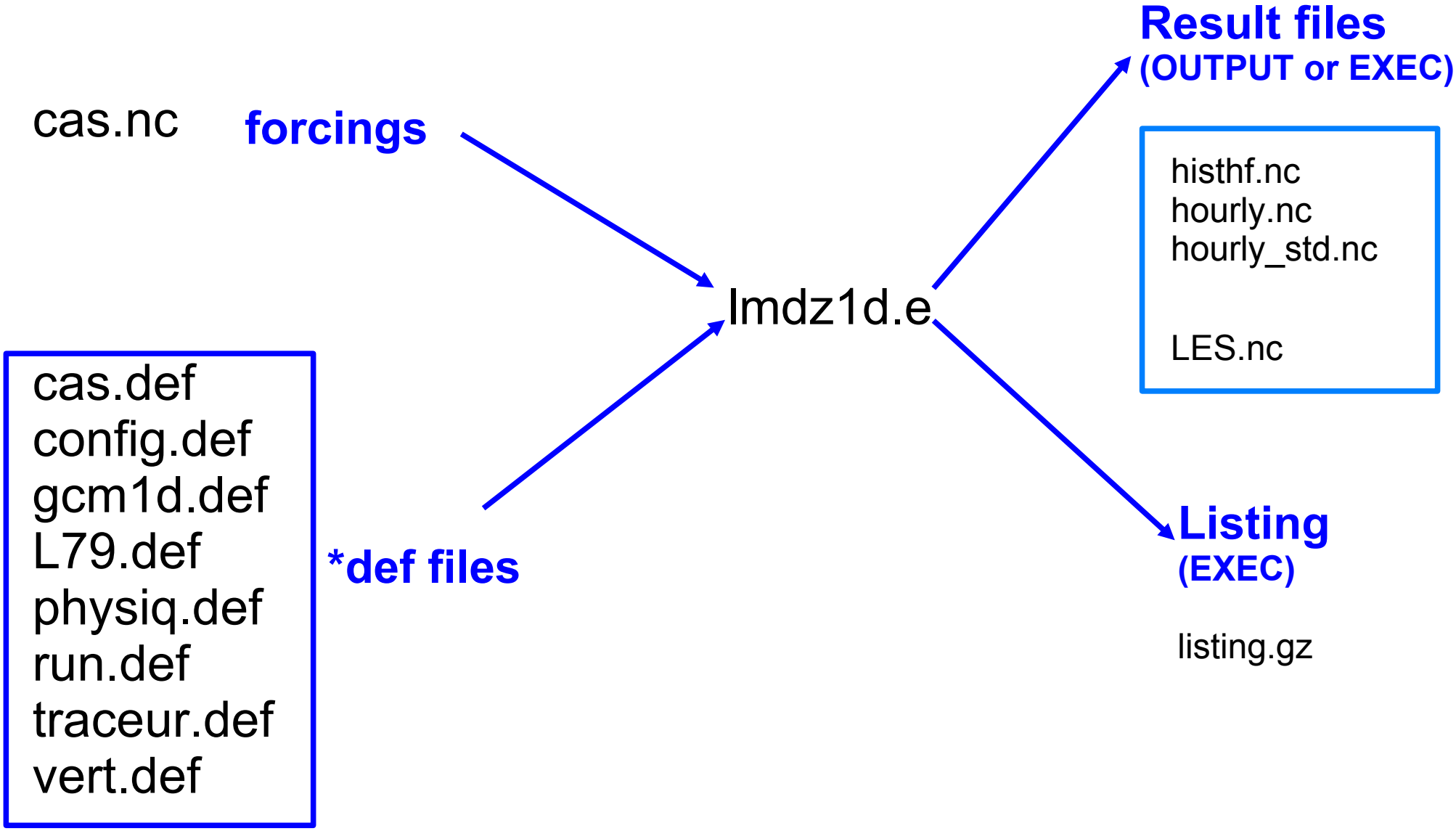
listedef="5A NPv6.1"

The number of levels ?

LLM="79" # imposing the number of **vertical level** (default 130)
day_step="" # number of physical steps per day (default 144, 10 minutes timestep)

flag_output_commun="1" (to get output file in standard output format)

Where are the results ?
In LMD20201109trunk/1D/*EXEC*/$physic/$case/$subcase
Or LMD20201109.trunk/1D/*OUTPUT*/$physic/$case/$subcase



And if the model did not work ?

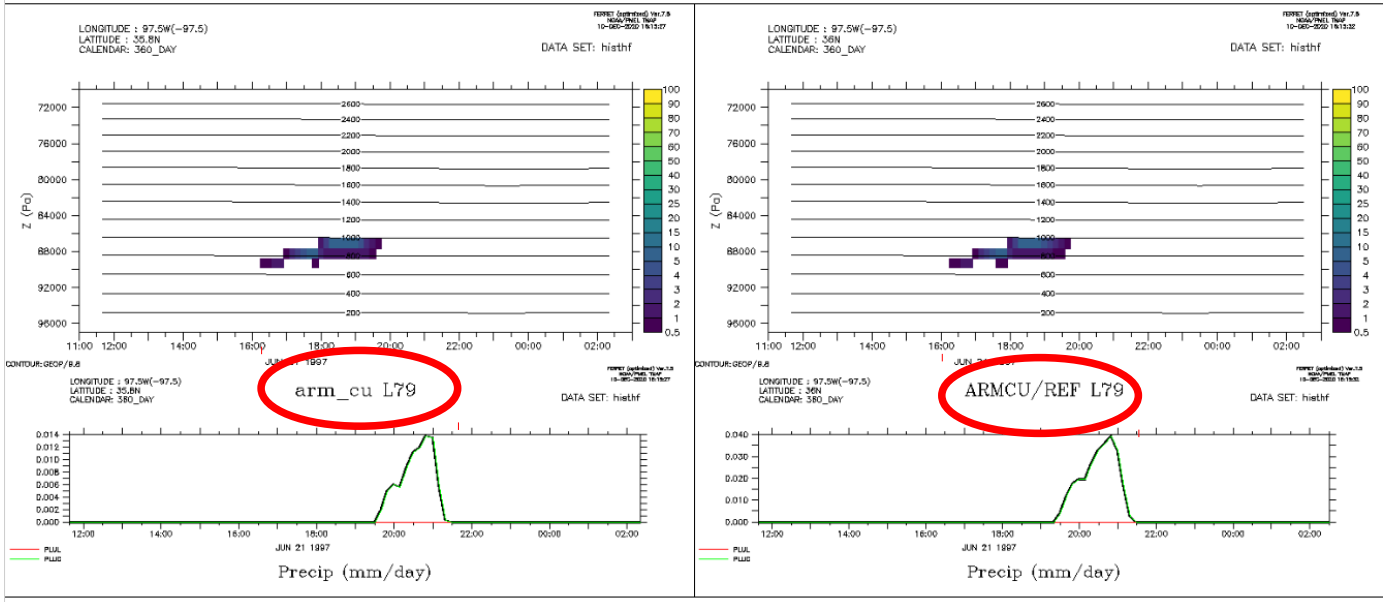
Go in ~EXEC/$physics_name/$case_name :

+ grep cool listing.gz

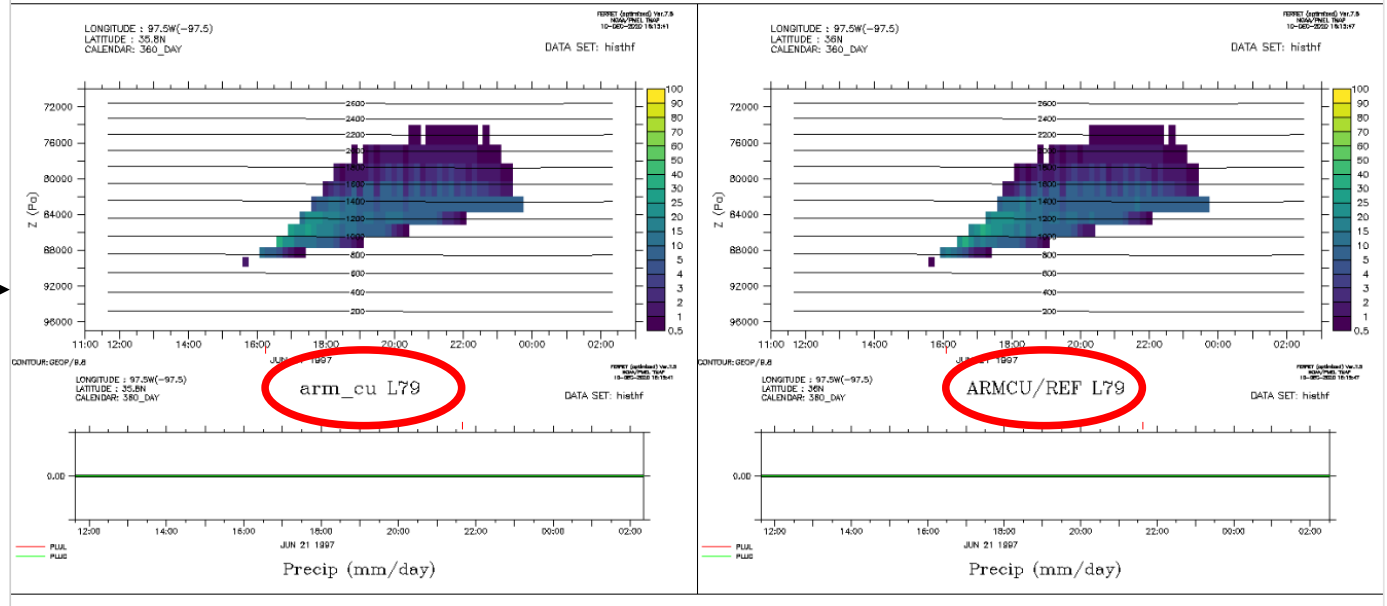
+ check if there are netcdf files

In LMD20201109.trunk/1D/OUTPUT/$physics/$case/$subcase
there are also **automatic diagnostics** :
all.pdf : cloud cover and precipitation

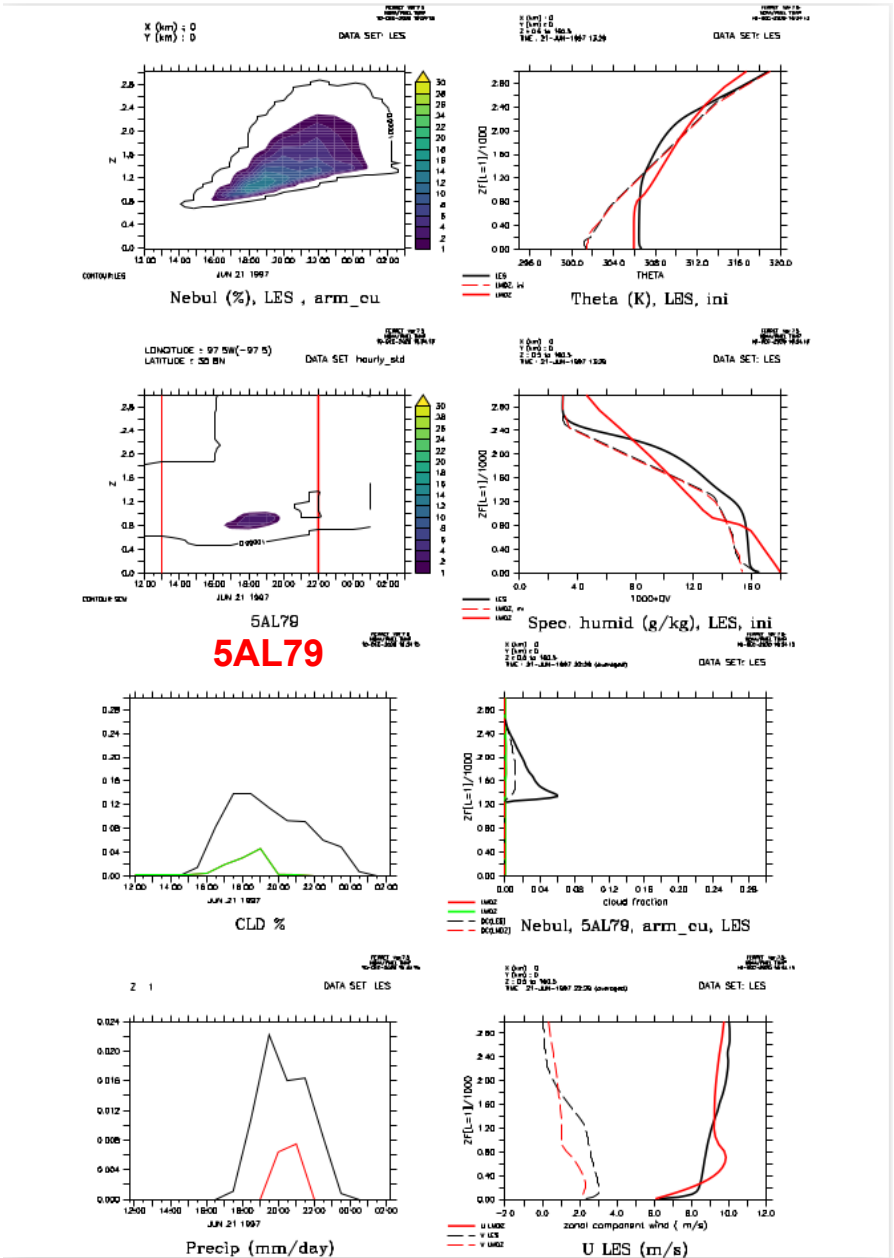
5AL79 →

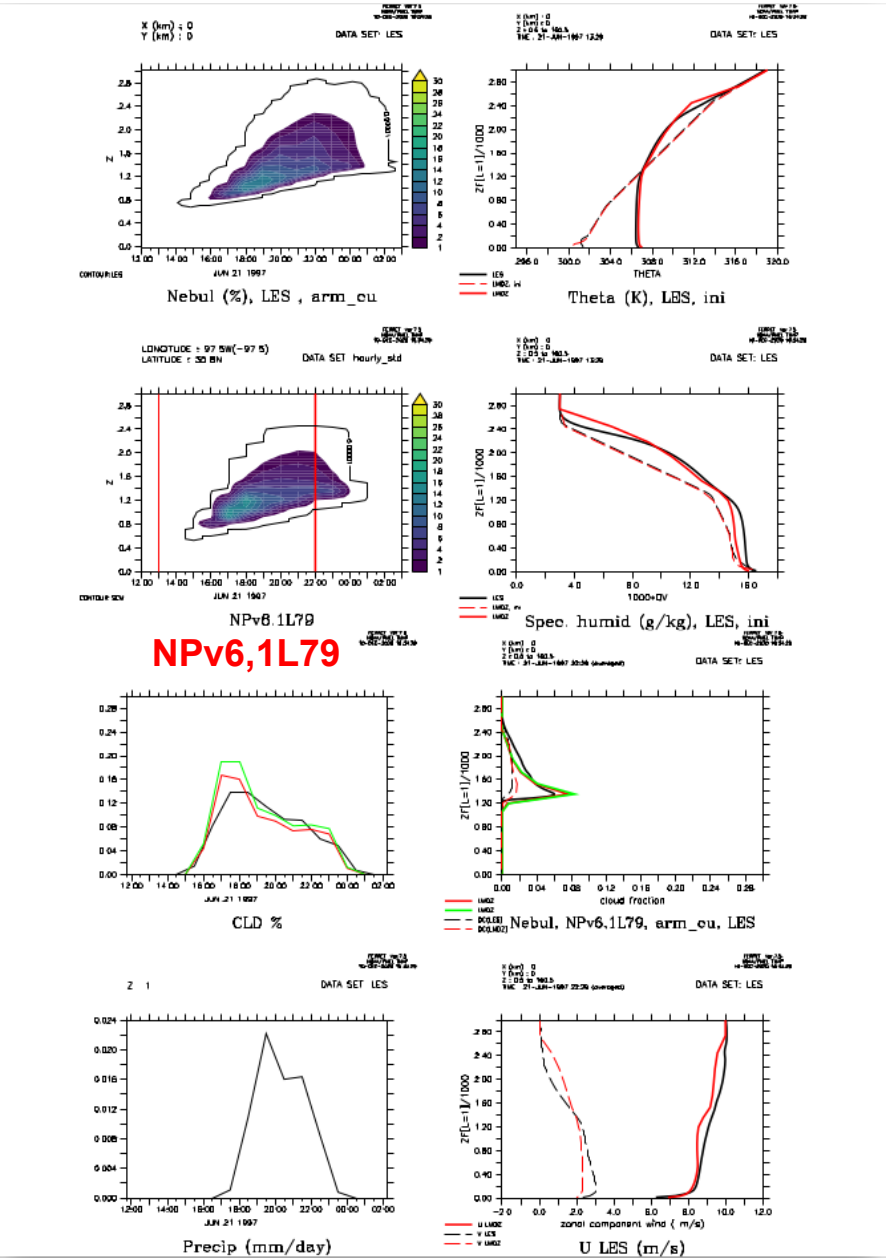


NPv6,1L79 →

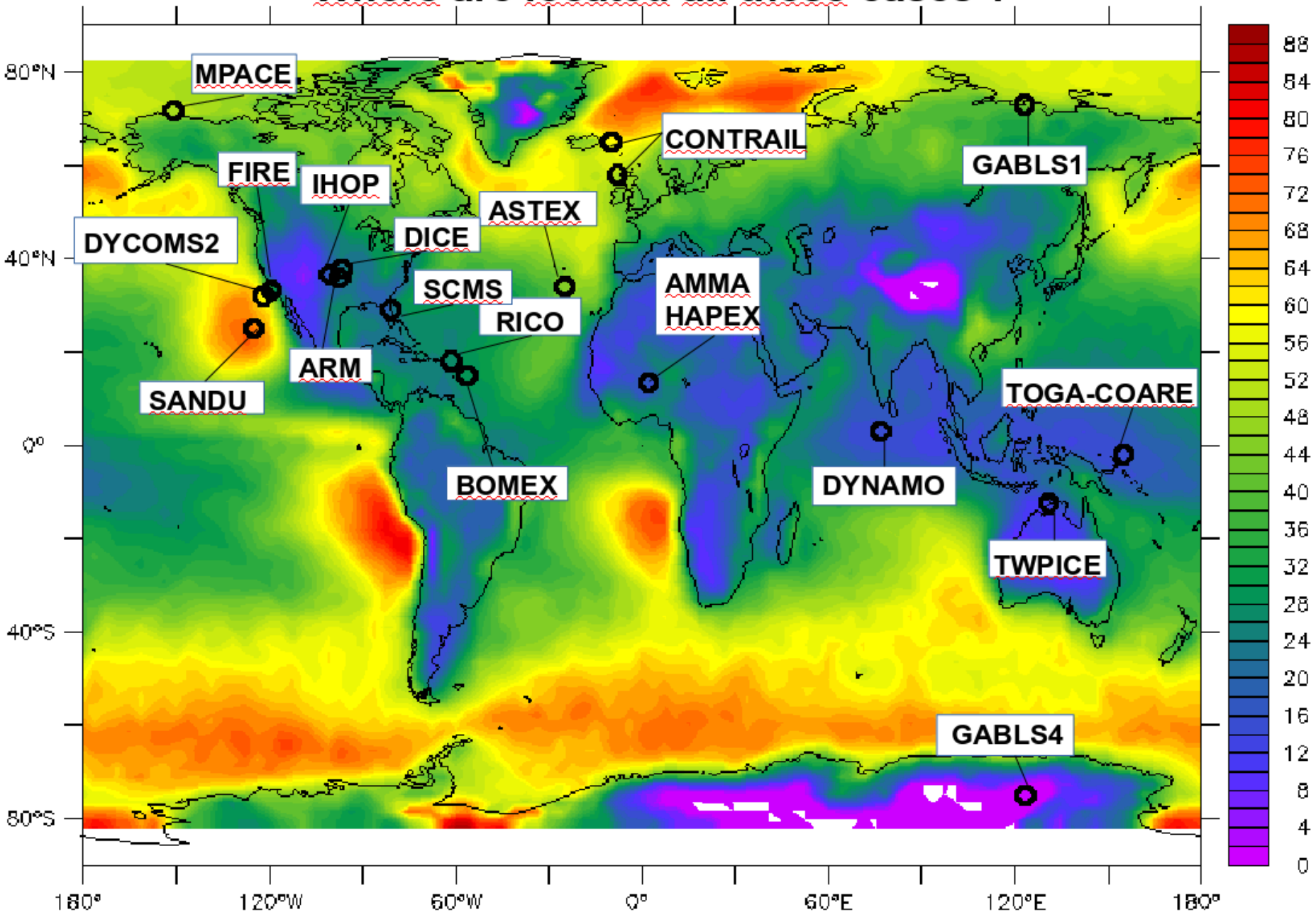


In LMD20201109.trunk/1D/OUTPUT/$physic/$case/$subcase
there are also **automatic diagnostics** :
arm_cu.pdf (& ARMCU.pdf)





Where are located all these cases ?



Background : low cloud cover from Calipso (Chepfer et al. 2008)

Avalaible cases correspond to different meteorological situations

Dry and shallow convection

Arm_cu (diurnal cycle of shallow cumulus over land)
Rico (Rain In Cumulus over Ocean,
shallow precipitating cumulus over sea)
Ayotte (convective boundary layer, sky clear)

Stratocumulus and transition to cumulus

Sandu (transition case with 3 options : variation of SST)
Fire (diurnal cycle of stratocumulus)

Deep convection Over ocean:

Toga
case_e (part of Toga)
TWPICE : off the coast of Darwin

Deep convection Over land:

Hapex : african monsoon
AMMA : african monsoon
Idealized case:
eq_rad_conv (RCE) : radiative and convection scheme active

Recent improvements:

DICE case : characterize boundary layer
In the site of SGP during 3 days/nights
May be **coupled with soil model**

Interaction of a very stable boundary layer with a snow surface :
GABLS1, **GABLS4** and **MPACE**

+ **Cindy Dynamo** case (MJO study)

To conclude: Why use SCMs ?

+ **simplicity**: technical and understanding, usable on any labtop

+ it's a useful tool for **parameterization development**: shallow convection, deep convection, transition from stratocumulus to cumulus, stable boundary layer, radiation...

+ we can **compare results to observations or to explicit simulations** (CRM, LES)

+ then we go back to GCM and test new parameterizations ...

+ hierarchy of models: SCM, LAM, AGCM, GCM ...

Thank you for your attention !