LDMZ tutorial: tracers

LMDZ team

January 2023

This tutorial focuses on using and adding tracers in LMDZ.
This document can be downloaded as a pdf file (so you could copy/paste command lines from it):

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

1 Prerequisits

You should have executed the mandatory part of Tutorial #2.

2 Experimenting with tracers

2.1 Radon and lead

Go to the directory TUTORIAL/SIMU1 where you have run a simulation according to Tutorial #2. In the file `traceur.def`, you can see that two trace species, named RN, for radon, and PB, for lead, are already defined in addition to vapor and condensed water. Visualize these two tracers in the file `histday.nc`. The NetCDF variables RN and PB are numbers of atoms per unit mass, in kg^{-1}. Usually, activities are preferred for comparison with observations. The activities per unit mass, in Bq kg^{-1}, are:

$$A_{\mathrm{Rn}} = \frac{\mathrm{RN}}{4{,}765 \cdot 10^5}$$
$$A_{\mathrm{Pb}} = \frac{\mathrm{PB}}{1{,}028 \cdot 10^9}$$

(The numerical values that appear in these fractions are the lifetimes of radon and lead, in s.) You can take a look at LMDZ info number 7, figure 19, or the plot in the General introduction of the LMDZ model (§ 4 "Operating modes") to check that you have sensible results. A recent work on the modeling of tracers with LMDZ is Pilon et al. (2015, QJRMS). If you want to see activities per unit volume, you should divide by the mass density, using NetCDF variables pres, temp and ovap in the file `histhf.nc`. (The Ferret color palette in figure 19 of LMDZ info number 7 is saz2.)

2.2 Inserting new tracers

We will now explain how to modify the code in order to add idealized tracers. As an example, we will add two tracers, and call them `NewTr1` and `NewTr2`. We will define a domain of the horizontal grid in which the two tracers will initially have the same constant value. The first tracer will be transported by the boundary layer and convective sub-grid-scale motion, in addition to large-scale advection. The second tracer will only be transported by large-scale advection.

We will have to modify the Fortran program but let us first consider the run-time parameters that we have to modify. At run-time, we decide to include the tracers `NewTr1` and `NewTr2` in the simulation by changing the file `tracer.def`. Append one line for each tracer at the end, as here below :

```
&version=1.0
&lmdz
default    type=tracer  phases=g    hadv=10   vadv=10   parent=air
H2O                                 hadv=14   vadv=14
H2O                     phases=ls
RN,PB
NewTr1
NewTr2
```

Note that you also have in DEF a file called `traceur.def` (French word for "tracer") which is the old format of the file. It still works, so you could use it by changing in plus the number of tracers in the beginning, from 5 to 7. However we recommend to rename it as `OLDtracer.def`, to be sure to use the new `tracer.def` file.

Now let us turn to the Fortran program. The only file we need to modify is :

`modipsl/modeles/LMDZ/libf/phylmd/traclmdz_mod.F90`

Here are the changes you should make in that file:

- Declare two new module variables, `id_NewTr1` and `id_NewTr2`, with type integer. These are the identifying numbers of the tracers in the program. You can take a previous declaration (`id_pcsat`, `id_pcocsat` ...) as a template. (Do not forget the OpenMP directives.)
- All remaining changes will be made in the procedure `traclmdz_init`, which is inside the module `traclmdz_mod`. In order to define `id_NewTr1` and `id_NewTr2`, the program will scan the file `traceur.def`, looking for NewTr1 and NewTr2. So you should initialize `id_NewTr1` and `id_NewTr2` to 0 before the loop "`DO iq = 1, nqtot`" beginning at line (approx) 180. There is a comment just above, saying "Recherche des traceurs connus", which means "looking for known tracers".
- In the body of this loop, in the `SELECT CASE` block, set `id_NewTr1` to the value of index `it` if the "name" of `tracers(iq)` is NewTr1. Same for NewTr1. You can add the tests near line (approx) 194, for instance, after the test for "pcq0" (that you can take as an example). ATTENTION : in the `SELECT CASE` part of the code, the tests on the tracer names are based on comparisons made in lower case, so use the `CASE("newtr1")` for the tracer `NewTr1`.
- For NewTr2, just after setting `id_NewTr2`, deactivate convective and boundary layer transport by setting `conv_flg(it)` and `pbl_flg(it)` to 0.
- Finally, we will initialize the tracers. There is a loop on tracers, beginning at line 294, which tests whether the initial tracer field, read from the file `start.nc`, is zero everywhere. At this point, the tracer field would also be zero if it was not found in `start.nc`. There is a comment just above the test which says "Initalize tracer that was not found in restart file" (the typo "Initalize" is in the code!). In the body of the test, for our two tracers, change the value at the surface in some horizontal domain. The value of tracers is in the variable `tr_seri`. The first dimension of `tr_seri` is for the horizontal position, the second dimension is for the vertical level and the third dimension identifies the tracer. The index of the surface in the vertical dimension is 1. Use variables `xlat` (latitudes) and `xlon` (longitudes) to choose the horizontal domain.

In summary, after making those changes, svn diff should give you something like this:

```
$ svn diff traclmdz_mod.F90
Index: traclmdz_mod.F90
===================================================================
--- traclmdz_mod.F90 (revision 2304)
+++ traclmdz_mod.F90 (working copy)
@@ -44,6 +44,9 @@
   INTEGER,SAVE :: id_be       ! Activation et position du traceur Be7 [ id_be=0 -> desactive ]
 !$OMP THREADPRIVATE(id_be)

+  INTEGER,SAVE :: id_NewTr1, id_NewTr2  ! Activation et position des nouveaux traceurs
+!$OMP THREADPRIVATE(id_NewTr1, id_NewTr2)
+
   INTEGER,SAVE :: id_pcsat, id_pcocsat, id_pcq ! traceurs pseudo-vapeur CL qsat, qsat_oc, q
 !$OMP THREADPRIVATE(id_pcsat, id_pcocsat, id_pcq)
   INTEGER,SAVE :: id_pcs0, id_pcos0, id_pcq0   ! traceurs pseudo-vapeur CL qsat, qsat_oc, q
@@ -172,6 +175,7 @@
 ! ----------------------------------------
      id_rn=0; id_pb=0; id_aga=0; id_be=0; id_o3=0
     id_pcsat=0; id_pcocsat=0; id_pcq=0; id_pcs0=0; id_pcos0=0; id_pcq0=0
+    id_NewTr1=0; id_NewTr2=0
     it = 0
```

```
     DO iq = 1, nqtot
        IF(.NOT.(tracers(iq)\%isInPhysics)) CYCLE
@@ -188,6 +192,8 @@
          CASE("pcs0");    id_pcs0   = it
          CASE("pcos0");   id_pcos0  = it
          CASE("pcq0");    id_pcq0   = it
+         CASE("newtr1");  id_NewTr1 = it
+         CASE("newtr2");  id_NewTr2 = it ; conv_flg(it)=0 ; pbl_flg(it)=0
          CASE DEFAULT
            WRITE(lunout,*) 'This is an unknown tracer in LMDZ : ', trim(tracers(iq)\%name)
        END SELECT
@@ -295,6 +301,9 @@
                    tr_seri(i,:,it) = 100.
                 END IF
              END DO
+          ELSE IF (it==id_NewTr1 .or. it==id_NewTr2) THEN
+               where (xlat >= 40. .and. xlat <=45 .and. xlon >= 0. .and.  &
+                       xlon <= 5. ) tr_seri(:, 1, it) = 1.
           ELSE
              ! No specific initialization exist for this tracer
              tr_seri(:,:,it) = 0.
```

Now you'll want to recompile the model.
Before doing that :

```
# From the folder modeles/LMDZ/libf/phylmd, go to modeles/LMDZ/TUTORIAL
cd ../../TUTORIAL
# Rename the old executable gcm.e (maybe you'll want to use it in another exercise):
mv gcm.e gcm_orig.e
# Remove the link to it, located in SIMU1 :
rm SIMU1/gcm.e
```

Recompile the model using the `compile.sh` script you find in `TUTORIAL` :
ATTENTION, it was created by `init.sh` with the options coherent with your TUTORIAL setup, and thus it is different from the script with the same name, produced by `install_lmdz.sh` in `modeles/LMDZ` (you may want to compare the two of them).
Make sure you are in TUTORIAL folder, then run compile.sh :

```
./compile.sh gcm
```

Rename the newly created `gcm.e` in `gcm_tracers.e` :

```
mv gcm.e gcm_tracers.e
```

Now go again in SIMU1, to run the simulation. Rename the restart files that you produced when running the model in the frame of Tutorial2, and create a link to the new executable :

```
cd SIMU1
mv restart.nc start.nc
mv restartphy.nc startphy.nc
ln -s ../gcm_tracers.e gcm.e
```

Run the model :

```
./gcm.e > listing
```

Visualize the two new tracers in `histday.nc` and examine the difference between them.