Numerical modeling : Tutorial #3

Frédéric Hourdin
hourdin@lmd.ens.fr

January 21, 2025

Available on

```
wget http://www.lmd.jussieu.fr/~hourdin/COURS/M2/TD/TD3e.pdf
```

1 Preparing the code : horizontal advection in a 2D framework

We consider first the advection of a Gaussian distribution of tracer in the x direction such as in Tutorial #2, but assuming that the tracer is defined on a bi-dimensional (x, z) domain.

Let us take a domain of horizontal length $L = 3000$ m and height $H = 1000$ m with 300 points on the horizontal and 100 on the vertical which leads to an isotropic and constant resolution : $\delta x = \delta z = 10$ m.

The horizontal wind is assumed to vary linearly with altitude $U(z) = U_{\rm min} + (U_{\rm max} - U_{\rm min})\, z/H$ with $U_{\rm min} = 2$ m s^{-1} et $U_{\rm max} = 10$ m s^{-1}.

Like in Tutorial #2, we assume a Gaussian distribution in x : $q(x, z, t = 0) = exp(-[(x - x_0)/\lambda]^2)$ with $\lambda = L/15.$ et $x_0 = L/5.$.

For this first question, you can use the code available on the M2 web site :

```
wget http://www.lmd.jussieu.fr/~hourdin/COURS/M2/TD3.tar.gz
tar xvf TD3.tar.gz
cd TD3
gedit TD3maquette.F90
```

1.1 First plots from ASCII files

In order to make the first plots, the 'TD3maquette.F90' program makes use of an home-made method, similar to the one used in Tutorial #2 :

```
integer, parameter :: imax=300,kmax=100
real xxx(imax),zzz(kmax),qqq(imax,kmax)
character*5 fichier
fichier="e...."
write(file(2:5),'(i4.4)') it
open (10,file=fichier,form='formatted')
do k=1,kmax
do i=1,imax
   write(10,*) xxx(i),zzz(k),qqq(i,k)
enddo
enddo
close(10)
```

Once compiled, the program can be executed.ASCII files like 'e0101' will then appear in your current directory. You can derive plots with the ferret software. The first time you activate ferret on a terminal, you should run

```
conda activate FERRET
```


Then, to use ferret :

```
ferret
```

Then

```
DEFINE AXIS/X=10:3000:10 xm
DEFINE AXIS/Z=10:1000:10 zm
sh da
DEFINE GRID/X=xm/Z=zm gm
FILE/VAR="xlu,zlu,q"/GRID=gm e0001
FILE/VAR="xlu,zlu,q"/GRID=gm e0701
show data
fill q[d=1]
fill/o/lev=(0.05,1,0.05)/pal=orange_blue q[d=2]
contour/o/lev=(0.1,1,0.1) q[d=2]
```

You can also run directly under ferret the same lines in a 'script' available from the archive :

```
go trace.jnl
```

The same plots can be made with gnuplot for those who are familiar with it.

Ferret systematically saves all the commands you entered with your keyboard in a file named ferret.jnl which becomes ferret.jnl.∼N∼ (with N=1, 2, ...). When you are happy with a figure, you can the save the last ferret.jnl file in a file named for instance mybeautifulfigure.jnl (mv ferret.jnl mybeautifulfigure.jnl).

You can then create a gif image with the following commands :

```
ferret
go mybeautifulfigure.jnl
frame/file=mybeautifulfigure.gif
quit
```

1.2 Use of the netcdf library

It becomes rapidly clear that this way of doing is not efficient enough. In order to automatize the inputs/outputs, we will now use the self-documented format 'netcdf'.

For this, start by replacing in file "TD3maquette.F90", the line

```
#undef IONC
```

by

```
#define IONC
```

The lines in between

```
#ifdef IONC
```

and

```
#endif IONC
```

are then considered as part of the fortran code (which was not the case before) when applying the 'cpp' pre-processor. The pre-processing is done automatically before the compilation for files ending by '.F90' rather than '.f90'.

The program will then call the subroutines "iotd_ini" which creates a netcdf file, "iotd_ecrit" which writes a variable in this file and "iotd_fin" which closes the file. Those subroutines are available on your current directory as you can see with :

```
ls iotd*
```

Those subroutines call other subroutines available in the 'netcdf' library. If you compile 'TD3maquette.F90' directly with gfortran, you will see that the compilation is unsuccessful. You must compile simultaneously the iotd* subroutines and 'link' them with the netcdf library. This will be done automatically with the script 'compile.x' available on your directory

```
./compile.x TD3maquette.F90
```

which creates an executable 'TD3maquette.exe' rather than 'a.out'. After executing the program, you will find the 'coucou.nc' on your directory. You can look at the content of this file with the following commands :

```
ncdump -h coucou.nc
ncdump coucou.nc | more
```

It can also be open with ferret

```
use coucou.nc
```

You can see the content of the files under ferret with command

```
show data
```

In order to plot the tracer plume at time 5 :

```
fill/l=500 qexact
```

Note that all the time steps have been stored in the coucou.nc file so that the file is quite big. You can also draw the wind vector over (/o or /overlay) the concentration:

```
vector/o/l=500 u,w
```

1.3 Horizontal advection

Code horizontal tracer advection with an upstream scheme (like in Tutorial 2) and plot tracer concentrations at two time steps.

2 2D plume computation : explicit computation and parameterization

In this part, we take the same parameters as previously except that the tracer is initially zero in the domain. Instead, a continuous point source is assumed in a mesh of the domain. In practice:

$$q(imax/6, 3*kmax/4) = 1. \tag{1}$$

2.1 Explicit computation

We will considered the previous wind field (horizontal only with vertical shear) as the large scale flow and add a perturbation on it, both in x and z.

Both perturbations will fluctuate in both space in time. In order to simplify the advection computation (and impose that the wind is non divergent in both the horizontal and vertical directions), the perturbation of the horizontal wind is assumed to depend on the vertical direction only. It will be prescribed as

$$U'(x,t) = U(x,t) \times [\sin(3.2k/kmax - t/20.)] \tag{2}$$

where k is the index of the vertical layer and $U(x,t)$ is the wind of the previous questions.

Similarly, the vertical wind W is assumed to depend on the horizontal direction and time only:

$$W(x,t) = W_{\max}\left[0.4\ \sin(k_1\pi\frac{x}{L} - t/\tau_1) + sin(k_2 * \pi\frac{x}{L} - t/\tau_2)\right] \tag{3}$$

with $W_{\max} = 1.5$ m s^{-1}, $k_1 = 33$, $\tau_1 = 50$ s, $k_2 = 10$ et $\tau_2 = 66$ s or, in fortran:


```
www(1:imax)=wmax*(0.2*sin(30.*pi*xxx(1:imax)/xmax-temps/50.)+sin(10.*pi*xxx(1:imax)/xmax-temps/6
```

The fortran code for those wind fluctuation is already included as comments in 'TD3maquette.F90'.

It is recommended to compute, at each time step $\alpha_x = \delta t U(X)/\delta x$ and $\alpha_z = \delta t W(X)/\delta z$. Do the computation over 3000 time steps, and plot the results at successive times to illustrate the behavior of the plume.

2.2 Computation with diffusion

We now add turbulent diffusion on the vertical with diffusivity K_z=2 m^2/s (it should be done on the vertical as well since the horizontal mesh is very small, but it is not useful here).

In order to prepare the following questions, code vertical diffusion in two steps. First compute the tracer flux:

$$F_q(x,z) = \rho\overline{w'q'} = -K_z\frac{\partial q}{\partial z} \tag{4}$$

before taking its vertical divergence.

Compute again the time evolution of the plume over 3000 time steps, with the same parameter values.

You will consider the result of this simulation with fluctuations of the wind and small scale diffusion as "the explicit simulation". You may plot the concentration at successive time steps with ferret, as:

```
use coucou
set v ul
shade/lev=(0.001)(0.002)(0.005)(0.01,0.2,0.01)(0.2,0.6,0.05)(Inf)/pal=rainbow/l=1000 qn
set v ur
shade/lev=(0.001)(0.002)(0.005)(0.01,0.2,0.01)(0.2,0.6,0.05)(Inf)/pal=rainbow/l=2000 qn
set v ll
shade/lev=(0.001)(0.002)(0.005)(0.01,0.2,0.01)(0.2,0.6,0.05)(Inf)/pal=rainbow/l=2500 qn
set v lr
shade/lev=(0.001)(0.002)(0.005)(0.01,0.2,0.01)(0.2,0.6,0.05)(Inf)/pal=rainbow/l=3000 qn
frame/file=ok.gif
```

where 'qn' is the name of the tracer field in 'coucou.nc'.

Save 'coucou.nc' as 'expl.nc' for instance.

2.3 Comparison of the two computations

Compare the averaged tracer concentration for the explicit computation averaged between time 2000 and 3000, with simulation run with the unperturbed wind for various value of the vertical diffusivity between 2 and 50 m^2/s.

You can compare the fields by either by plotting vertical profiles in $x = L$

```
use 'expl.nc'
plot/title='Qn, explicit' qn[d=1,i=300,l=2000:3000@ave]
plot/o/title='Kz=2' qn[d=2,i=300,l=2000:3000@ave]
```

or by plotting bi-dimensional maps:

```
fill qn[i=300,l=2000:3000@ave]
```

Comment the results.

How could you use those results to justify the replacement of the wind fluctuations by a diffusive formulation ? How does the value of the required (tuned) diffusivity compare with the actual small scale diffusivity of 2 m^2/s ? Can you relate the coefficient K_z to the values of the vertical fluctuating wind W of the explicit computation ?