Numerical modeling : tutorial #4

Frédéric Hourdin
hourdin@lmd.ens.fr

5 décembre 2012

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

You can start from tutorial #2 in which a gassian tracer field was advected horizontally in 1D.

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}.

```
integer, parameter :: imax=300,kmax=100
integer :: i,k
real xxx(imax),zzz(kmax)
real :: xmax=3000.
real :: zmax=1000.
real :: detlax,deltaz
real :: qancien(imax,kmax),qnouveau(imax,kmax)
deltax=xmax/float(imax)
deltaz=zmax/float(kmax)
do i=1,imax
   xxx(i)=i*deltax
enddo
do k=1,kmax
   zzz(k)=k*deltaz
enddo
```

1. Let us take, as in tutorial #2, a Gaussian initial concentration, depending on x only : $q(x, z, t = 0) = exp(-[(x - x_0)/\lambda]^2)$ with $\lambda = L/15.$ and $x_0 = L/5.$.
   Plot the initial tracer concentration.
2. Choose the time step δt considering the maximum velocity encountered in the domain. You could define an array that depends on the vertical layer only to store the CFL coefficient $\alpha_x(z) = \delta t\ U(z)/\delta x$.
   Compute horizontal advection with a first order upstream scheme and make 2D plots at various times (see help at the end).

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

In addition to the horizontal wind, we assume now that there is a vertical wind, which fluctuates in both time and space. In order to simplify the advection computation (and impose that the wind is non divergeant in both the horizontal and vertical directions), the vertical wind W is assumed to depend on the horizontal direction and time only :

On prendra comme formule :

$$W(x,t) = W_{\mathrm{max}} \left[0,2\ \sin(k_1\pi\frac{x}{L} - t/\tau_1) + sin(k_2 * \pi\frac{x}{L} - t/\tau_2)\right] \tag{2}$$

with $W_{\mathrm{max}} = 5$ m s^{-1}, $k_1 = 30$, $\tau_1 = 50$ s, $k_2 = 10$ et $\tau_2 = 66$ s or, in fortran :

```
www=wmax*(0.2*sin(30.*pi*xxx(i)/xmax-temps/50.)+sin(10.*pi*xxx(i)/xmax-temps/66.))
```

It is recomanded to compute, at each time step $\alpha_z(x) = \delta t W(X)/\delta z$. Do the computation over 2000 time steps, and plot the results at successive times to illustrate the behavior of the plume.

2.2 Computation with diffusion

We now assume $W(z) = 0$ but we add a compution of vertical diffusion on the vertical, with a diffusion coefficient constant in time and space $K_z = 50$ m^2/s.

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

$$F_q(x,z) = \rho\overline{w'q'} = -K_z\frac{\partial q}{\partial z} \tag{3}$$

before taking its vertical divergence.

Compute again the time evolution of the plume over 2000 time steps, with the same parameter values.

2.3 Comparison of the two computations

Compare the averaged tracer comcentration for the two computation between time 1000 and 2000, either by ploting vertical profiles in $x = L$ or by ploting bi-doensional maps.

If possible, fix the tracer iso-contours with evenly space values such as (0,001 ; 0,002 ; 0,005 ; 0,01 ; 0,02 ; 0,05 ; 0,1 ; 0,2 ; 0,5)

Comment the results.

Can you relate the coefficient K_z to the values of the vertical fluctuating wind W of the explicit computation.

Options for 2D graphs

The best solution ...

Use something you know, such as matlab, scilab, gnuplot or others.

2D ASCII output

You can use the following fortran lines :

```
integer, parameter :: imax=300,kmax=100
real xxx(imax),zzz(kmax),qqq(imax,kmax)
character*5 fichier
fichier="q...."
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)
```

And run the ferret graphical software,

```
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 q151
show data
fill q
```

Ferret systematically sves all the command 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 get 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 majoliefigure.jnl
> frame/file=mafigure.gif
> quit
```