

Modélisation numérique : TD 3

Frédéric Hourdin
hourdin@lmd.ens.fr

23 octobre 2009

1 Calcul numérique de la diffusion

On repart de l'énoncé du TD 2.

On considère une espèce trace dont la concentration $q(x, t)$, en kg/kg d'air par exemple, ne dépend que de la dimension x et du temps, sur un domaine $[0, L]$ en x et $[0, T]$ en temps. On va regarder cette fois l'effet de la diffusion du traceur.

1. On suppose qu'à l'instant initial, le traceur est la superposition de deux sinusoides et d'un mode oscillatoire d'un point de grille sur deux :

$$q_{i,1} = \sin(4\pi x_i/x_{\max}) + \sin(17\pi x_i/x_{\max}) + (-1 + 2 * \text{mod}(i, 2))$$

Tracer la concentration initiale du traceur.

2. Coder un schéma de diffusion numérique avec $\alpha = K\delta t/\delta x^2 = 0.1$

Normalement, il suffit de changer deux lignes (calcul de la concentration initiale et schéma numérique) au programme développé pour le TD2. On pourra supposer $q_{1,n} = q_{2,n}$ et $q_{im,n} = q_{im-1,n}$ (condition de flux nul aux bords) ou $q_{1,n} = 0$ et $q_{im,n} = 0$

3. Sont-ce les petites ou les grandes échelles qui se dissipent le plus rapidement ? Commenter par rapport à ce que vous savez des opérateurs de dissipation.
4. Avec le même champ initial, effectuer un calcul d'advection sur une 100 aine de pas de temps avec les schémas centré et amont du TD2. Commenter.

5. Reprendre le calcul de la dissipation en ne gardant dans le champ initial que la partie oscillation de point de grille : $-1 + 2 * \text{mod}(i, 2)$.

Effectuer le calcul avec $\alpha = 0.05$, $\alpha = 0.2$, $\alpha = 0.25$, $\alpha = 0.3$ et $\alpha = 0.6$.

Effectuer le calcul sur dix pas de temps en sauvant le traceur à chaque pas de temps. Superposer à chaque fois les 10 courbes en utilisant par exemple la commande

```
xmgrace -lengend load q00*
```

Remarque : on pourra sortir les champs en format ascii par exemple tous les 10 pas de temps en utilisant, comme pour le TD2 :

```
character*4 file
do it=1,nt
....
  if (mod(it,10)==1) then
    file="q..."
    write(file(2:4),'(i3.3)') it
```

```
open (10,file=file,form='formatted')
do i=1,imax
  write(10,*) xxx(i),qqq(i)
enddo
close(10)
endif
```

On peut ensuite soit tracer les courbes à l'écran avec la commande

```
xmgrace q*
```

soit également sauvegarder cette figure dans un fichier grace à la commande

```
xmgrace -hardcopy -hdevice JPEG -printfile out.jpg q*
```

(le 'q*' veut dire que vous appliquez la commande à tous les fichiers commençant par la lettre q dans le répertoire où vous vous trouvez, ici q001, q011, q021 ...). Mais n'hésitez pas à utiliser tout autre moyen de votre goût pour sortir les fichiers.

2 Préparation du TD final

Repartir à nouveau du TD2 dans lequel on advectait un champ de traceur gaussien initialement dans une direction unique.

Ajouter une dimension verticale aux différents champs. On utilisera par exemple 100 points en x et 30 en z .

```
integer imax,kmax,i,k
parameter (imax=100,kmax=30)
real xxx(imax),zzz(kmax)
real :: xmax=100000.
real :: zmax=3000.
real :: deltax,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. Prendre comme dans le TD2 une concentration initiale de traceur gaussienne, ne dépendant que de x : $q(x, t = 0) = \exp(-[(x - x_0)/\Delta x]^2)$ avec $\Delta x = L/15$. et $x_0 = L/5$. (ne pas confondre ce Δx avec la taille d'une maille du modèle δx). Tracer la concentration initiale du traceur.
2. Se donner un champs de vent horizontal (en x) dépendant uniquement de z . Il suffit en pratique de spécifier le coefficient $\alpha_x = U\delta t/\delta x$. On posera par exemple $\alpha_x(z) = 0.1 + 0.5z/z_{\max}$. Calculer l'advection horizontale de ce traceur avec un schéma numérique de votre choix et faire quelques sorties 2D à des instants successifs.

Options pour les tracers en 2D

La meilleure solution ...

Utiliser quelque chose que vous connaissez, comme matlab, scilab ou autres.

Fichiers ASCII

On pourra pour ce faire utiliser les lignes fortran suivantes :

```
file="q..."
write(file(2:4),'(i3.3)') it
open (10,file=file,form='formatted')
do k=1,kmax
do i=1,imax
write(10,*) xxx(i),zzz(i),qqq(i)
enddo
enddo
close(10)
```

Puis lancer le logiciel ferret par exemple,

ferret

Puis taper les lignes suivantes :

```
DEFINE AXIS/X=1:100:1 xm
DEFINE AXIS/Z=0.1:3.:0.100 zm
sh da
DEFINE GRID/X=xm/Z=zm gm
FILE/VAR="xlu,zlu,q"/GRID=gm q151
show data
fill q
```

Les mêmes fichiers doivent pouvoir être tracés avec gnuplot.

Fichiers binaires au format grads

Pour ceux qui connaissent le logiciel grads, vous pouvez récupérer sur

<http://www.lmd.jussieu.fr/~hourdin/COURS/Fortran/Grads>

les fichiers

inigrads.F, wrgrads.F et gradsdef.h

Vous ajoutez dans le code, avant la boucle en temps, les lignes

```
character*10 fichier,titre
...
fichier='q'
titre='q'
call inigrads(1,imax ,xxx,1.,0.,xmax,kmax,zzz,0.,zmax,1.,1,0.,1. ,deltat,fichier,titre)
```

qui va initialiser l'écriture d'un fichier grads, avec les coordonnées xxx, et zzz, entre 0 et xmax ou zmax respectivement.

Puis, au moment de l'écriture :

```
if (mod(it,10)==1) then
call wrgrads(1,1,qancien,'q',,q',)
endif
```

les routines inigrads.F et wrgrads.F et le fichier gradsdef.h Vous compilez ensuite

pgf90 monprog.f90 inigrads.F wrgrads.F

Vous lancez enfin grads, puis

```
open q
query file
set mproj off
set t 5
d q
```