Numerical modeling, tutorial class 1

Frédéric Hourdin, hourdin@lmd.ens.fr

January 7, 2025

Radioactive decay

The goal of this first tutorial class is both to get familiar with the computer system used (Linux, fortran, graphical package, ...) and explore a very simple numerical issue concerning temporal integration.

From appearances to mathematics

Many atmospheric/oceanic constituents (or trace species) are subject to a sink : chemical reaction with an other species, removal of aersols by rainfall (scavenging), radioactive decay ...

If the sink is a linear function of the tracer, the time evolution equations reads :

$$\frac{\partial q}{\partial t} = -\lambda q \tag{1}$$

Numerics

The goal is to compute successive values $q^{(n)}$ of the concentration q starting from an initial concentration $q^{(0)}$ (assuming that the exact solution is unknown). Time n and $n+1$ are separated by a constant time-step δt.

By estimating the time derivative in Eq. 1 as a finite difference between time n and $n+1$, and estimating the sink at the beginning, at the end or at the middle of the time-step, one can derive 3 possible estimations of $q^{(n+1)}$ as a function of $q^{(n)}$.

Computer world : first program

Choose a formulation based on a ”right” time derivative known as Euler implicit scheme (the most natural one, between n et $n+1$ with a sink computed at time n) in order to write a fortran code computing the tracer concentration at successive time steps.

For numerical application : $\lambda = 1/1000$ s^{-1}, $q^{(0)} = 1$ (arbitrary units) and $\delta_t = 60$ s

How to write and execute a fortran program

Create a file prog.f90 (for this, use an ASCII editor such as gedit) containing the fortran lines.

For beginners, use the following

```
Implicit none ! Very important. Forces to declare expicitely all
              ! variables
Integer n
Do n=1,100
   print*,'Time ',n
Enddo
End
```

You may gain time by going through the introduction to Fortran with examples available at `https://web.lmd.jussieu.fr/~hourdin/COURS/FORTRAN/`

Then compile `> gfortran prog.f90`

and run the program `> ./a.out`

In this exemple, 'n' is an integer which designates the successive time steps, in the time loop. It is preferable to use a floating number for the time variable itself. It is also preferable to use long enough names for the variables and include comments (lines starting by a "!").

One can use the following exemple :

```
Implicit none
! Declarations
!--------------
Integer nt,ntmax
Real time,deltat

! Initialisations
!----------------
ntmax=100
time=0.
deltat=60.

! time loop
!----------------
Do nt=1,ntmax
   time=time+deltat
   print*,'time ',time
Enddo
End
```

Once the fortran code has been 1) written, 2) compiled and 3) executed,

plot the results with a graphical package of your choice. For this, you can for instance write the output variables time and concentration in the loop with the following fortran lines

```
...
open(11,file='qapprox',form='formatted')
```

```
Do nt=1,ntmax
...
   write(11,*) time,qnew
Enddo
close(11)
```

It is then possible to plot the results with the xmgrace command.

```
> xmgrace -legend load qapprox
```

For those who did introduce a fortran array 'real q(n)' containing the 'n' consecutive values of the concentration, write a new version of the program without any array, either overwriting the value of 'q' or using to values, 'qold' and 'qnew' for instance.

Behavior of the numerical scheme

In addition to the approximate solution, compute the exact solution (which is known in our particular case). The exponential function in fortran is exp(x).

Plot the exact and approximate solutions on the same graph.

You can open, in the the same code, one file for the approximate and one for the exact solution.

```
open(10,file='qexact',form='formatted')
open(11,file='qapprox',form='formatted')
do nt=1,ntmax
   write(11,*) time,qapprox
   write(10,*) time,qexact
```

then plot the two curves, for instance with the command :

```
> xmgrace -legend load qexact qapprox
```

Change the time step of integration. Try for instance to compute the concentration over 10 time steps of 600 seconds each. The 10 time steps of 1000 seconds, the 10 time steps of 120 seconds and 10 time steps of 3000 seconds.

To avoid modifying the code each time, you can read the entry parameters interactively by adding the following lines :

```
character*20 fichier ! declaring a string of 20 characters
...
print*,'number of time steps'
read(*,*) ntmax
print*,'time steps in seconds'
read(*,*) deltat
print*,'Name of output file'
read(*,*) fichier
open(11,file=fichier,form='formatted')
```

Then answer the questions asked when running the code :

```
> a.out
> 10
> 600
> qapporche600
```

Plot each time the results;

Explain the results by comparing the relationship between $q^{(n)}$ and $q^{(n+1)}$ used for the numerical algorithm to that of the exact solution.

Centered and left derivative

Redo the same series of curves by using the centered derivative or the left derivative, for the finite difference.

The work can be done for instance in separate sub-directories.

Analyzes

For each scheme, analysis the scheme behavior with more or less long time steps.

Comment on the accuracy of the various schemes at short time steps, fron the figures, and by using limited developments of the exact and approximate solutions of $q^{(n+1)}/q^{(n)}$ for $\alpha = \lambda\delta t << 1$.

For long time steps, compare the character more or less "physical" (in link with the world of appearances or physical theories) of the various schemes.

Write a small report

Write a very small report consisting on two significant figures with legenda. 1 page maximum. To be given back together with the report on TD2.

To save files under xmgrace, go to the menu 'print setup'. Select a 'device' such as 'PNG' then 'accept' then return to the rolling menu to execute the 'print' command.

You can use for instance 'openoffice.org3' or 'latex' for the text.