Introduction to Computational Science, Practical Work

Report must be sent by mail (guillaume.fuhr@univ-amu.fr)
before March,06 2016 in pdf format

Contents

Introduction

The aim of this work will be to study numerical methods and results obtained for an advection-diffusion equation. This equation can be expressed as :

$$\frac{\partial u(x,t)}{\partial t} + V\nabla u(x,t) = C\nabla^2 u(x,t) \tag{1}$$

$$\nabla := \begin{cases} \partial_x \\ \partial_y \\ \partial_z \end{cases}$$

Assuming a domain of size : $[L_x \times L_y] = [2\pi \times 2\pi]$, choosen Δt and Δx must allow a simulation representing $\simeq \mathbf{10}$ times the typical time scale of the considered dynamic.
Diffusion time scale : $\tau_D = L^2/C$
Advection time scale : $\tau_A = L/V$

Values of parameters/grid sizes are not imposed/given, as starting point, you should use mesh size like $[8 \times 8, 16 \times 16, 32 \times 32, \cdots, 256 \times 256]$ and values of C, V in range like $\left[10^{-2}, 10\right]$

1 1D equation

1.1 Diffusion

Considering only diffusive part in 1D :

$$\frac{\partial u(x,t)}{\partial t} = C\frac{\partial^2 u(x,t)}{\partial x^2} \tag{2}$$

1. Resolve numerically discrete equation using advdiff.exe program and using explicit Euler scheme. Using the plots of output data, what kind of boundary conditions are used? How is it implemented in the code?

2. How does system evolves when $C\Delta t/\Delta x^2$ is $< 1/2, = 1/2 - \epsilon, = 1/2, = 1/2 + \epsilon, > 1/2$

3. Calculate numerical growth rate and compare it with the analytic one.

4. For a given Δx, do you observe a limit on time step for numerical stability if you use RK4 scheme? Estimate this with a precision of 10^{-2} and compare this value with the one corresponding to an explicit Euler scheme.

5. Theoretically, there is no constraint on time step with implicit schemes, is it verified numerically? Comments?

1.2 Advection

This time we consider only advection :

$$\frac{\partial u(x,t)}{\partial t} + V\frac{\partial u(x,t)}{\partial x} = 0$$

1. Which implementation of spatial derivative are implemented in the code? is it backward, centered or forward scheme?

2. What is the physical definition of an advection ? Is radial structure conserved in simulations?

3. Is it the case in the simulations for the two following cases :

   - if a sin function is used for $u_0(x)$
   - if an *heavyside* function is used for $u_0(x)$

4. Verify if solution is convergent in the case where you use an RK4 time scheme and centered derivatives in space.

5. How does the system evolves when $V\Delta t/\Delta x$ is $< 1, \simeq 1, = 1, > 1$

6. As in previous section, estimate the stability limit when you consider the RK4 time scheme.

1.3 Advection-Diffusion

Now we consider the full equation :

$$\frac{\partial u(x,t)}{\partial t} + V\frac{\partial u(x,t)}{\partial x} = C\frac{\partial^2 u(x,t)}{\partial x^2}$$

1. What kind of derivate is implemented for advection with implicit Euler scheme ?

2. Calculate numerically diffusive growth rate from simulations. Is your result in agreement with expected value?

3. Runs simulations with centered or backward difference for advection. Calculate growth rate,is it linked to advection coefficient?

4. In previous sections, it has been shown that stability criterion exists for diffusion and advection, what happens if we choose parameters which verifies only one of these criterions?

2 Appendices

2.1 Analytical results

2.1.1 General solution of heat equation on a finite domain

1D case

$$\begin{aligned}
\partial_t u(x,t) &= C\partial_x^2 u(x,t) + Du(x,t) + S(x) && (3)\\
\text{Domaine :} &\quad 0 \le x \le Lx && (4)\\
u(x,0) &= u_0(x) && (5)\\
u(0,t) &= 0 && (6)\\
u(0,Lx) &= 0 && (7)\\
u(x,t) &= \int_0^{Lx} u_0(x)G(x,\xi,t)\mathrm{d}\xi + \int_0^t \int_0^{Lx} S(\xi)G(x,\xi,t-\tau)\mathrm{d}\xi\mathrm{d}\tau && (8)\\
G(x,\xi,t) &= \frac{2}{Lx}e^{(Dt)}\sum_{n=1}^{\infty}\sin\left(\frac{n\pi x}{Lx}\right)\sin\left(\frac{n\pi\xi}{Lx}\right)\exp\left(-\frac{Cn^2\pi^2 t}{Lx^2}\right) && (9)\\
& && (10)
\end{aligned}$$

2D case

$$\begin{aligned}
\partial_t u(x,y,t) &= C\partial_x^2 u(x,t) + C\partial_y^2 u(x,y,t) + S(x,y) && (11)\\
\text{Domaine :} &\quad 0 \le x \le Lx && (12)\\
&\quad -\infty \le y \le \infty && (13)\\
u(x,y,0) &= u_0(x,y) && (14)\\
u(0,y,t) &= 0 && (15)\\
u(Lx,y,t) &= 0 && (16)\\
u(x,t) &= \int_{-\infty}^{\infty}\int_0^{Lx} u_0(\xi,\eta)G(x,y,\xi,\eta,t)\mathrm{d}\xi\mathrm{d}\eta && (17)\\
&\quad + \int_0^t\int_0^{Lx}\int_{-\infty}^{\infty} S(\xi,\eta)G(x,y,\xi,\eta,t-\tau)\mathrm{d}\xi\mathrm{d}\eta\mathrm{d}\tau && (18)\\
G(x,\xi,t) &= G_1(x,\xi,t)G_2(y,\eta,t) && (19)\\
G_1(x,\xi,t) &= \frac{2}{Lx}\sum_{n=1}^{\infty}\sin\left(\frac{n\pi x}{Lx}\right)\sin\left(\frac{n\pi\xi}{Lx}\right)\exp\left(-\frac{an^2\pi^2 t}{Lx^2}\right) && (20)\\
G_2(y,\eta,t) &= \frac{1}{2\sqrt{\pi Ct}}\left[\exp\left(-\frac{(y-\eta)^2}{4Ct}\right) - \exp\left(-\frac{(y+\eta)^2}{4Ct}\right)\right] && (21)
\end{aligned}$$

2.2 Numerical schemes

Supposing this convention for discretisation :

$$f(x,t) \to f(x_i,t_j) \to f(x+i\Delta x, t+j\Delta t) \to f_{t+i}^{x+j}$$

2.2.1 Derivates

And considering an equation expressed as

$$\frac{\partial f(x,t)}{\partial t} = L(t, f(x,t)) \qquad (22)$$

$$\begin{array}{lcl} \text{Forward difference :} & \partial_t f(t) & = \frac{f^{t+1}-f^t}{\Delta t} \\ \text{Backward difference :} & \partial_t f(t) & = \frac{f^t-f^{t-1}}{\Delta t} \\ \text{Centered difference :} & \partial_t f(t) & = \frac{f^{t+1}-f^{t-1}}{2\Delta t} \end{array} \tag{23}$$

2.2.2 explicit Euler

$$\frac{f^{t+1}-f^t}{\Delta t} = L(t, f^t) \tag{24}$$
$$f^{t+1} = f^t + \Delta t L(t, f^t) \tag{25}$$

2.2.3 implicit Euler

$$\frac{f^{t+1}-f^t}{\Delta t} = L(t+\Delta t, f^{t+1}) \tag{26}$$
$$f^{t+1} = f^t + \Delta t L(t+\Delta t, f^{t+1}) \tag{27}$$

2.2.4 Cranck-Nicholson

$$\frac{f^{t+1}-f^t}{\Delta t} = \frac{1}{2}L(t, f^t) + \frac{1}{2}L(t+\Delta t, f^{t+1}) \tag{28}$$
$$f^{t+1} = f^t + \Delta t \left(\frac{1}{2}L(t, f^t) + \frac{1}{2}L(t+\Delta t, f^{t+1})\right) \tag{29}$$

2.2.5 Runge-Kutta 4

$$f^{t+1} = f^t + \frac{\Delta t}{6}\left(k_1 + 2k_2 + 2k_3 + k_4\right) \tag{30}$$
$$\text{with} \tag{31}$$
$$k_1 = \Delta t L(t, f^t) \tag{32}$$
$$k_2 = \Delta t L(t + \frac{\Delta t}{2}, f^t + \frac{1}{2}k_1) \tag{33}$$
$$k_3 = \Delta t L(t + \frac{\Delta t}{2}, f^t + \frac{1}{2}k_2) \tag{34}$$
$$k_4 = \Delta t L(t + \Delta t, f^t + k_3) \tag{35}$$
$$\tag{36}$$

2.3 Code AdvDiff1D

2.3.1 download source and compile

Source code can be download at : **https://github.com/GFuhr/MF_FCM6/zipball/master**
Source code is in folder TP1.

how to compile

- On linux , this program can be compiled via the **make** command.
- To run it on linux,after compilation step through **make**, type **./advdiff.exe**
- On Windows, with any IDE editor such as VisualStudio or Code::Blocks, loading files *advdiff.c* and *advdiff.h*.

2.3.2 Initial Conditions

Time

	$advection$	$diffusion$
$t = 0$	$u^0(x)$	$u^0(x)$

Space

	$advection$	$diffusion$
$x = 0$	$u(t, 0)$	$u(t, 0)$
$x = L_x$		$u(t, L_x)$

Initial Fields

$u^0(x)$	$A \sin(\sigma \frac{\pi}{L_x}(x - x_0))$
$u^0(x)$	$A \exp(-(x - x_0)^2/\sigma^2)$
$u^0(x)$	$\begin{cases} A \text{ if } x \in [x_0 - \sigma/2,\, x_0 + \sigma/2] \\ 0 \text{ else} \end{cases}$

I/O Inputs :
Initial parameters must be entered when you run the program.

- "Nx=?" : number of points in radial direction
- "Npas=?" : number of iterations
- "Nout=?" : every *Nout* iterations, $u(t, x)$ will be write in a file
- "C=?" : diffusion coefficient
- "V=?" : advection coefficient
- "A=?" : Amplitude of the initial field described in 2.3.2
- "x0=?": parameter x_0 defined in 2.3.2
- "sigma=?": parameter σ defined in 2.3.2
- "Dx=?" : radial step Δx
- "Dt=?" : time step Δt

Outputs :
Everytime you run the program, *Nout* output files will be generated **out_XXXX.dat**. Please notice that files will not be erased between runs. Each file is coded in text format (ASCII) and contains $(Nx - 2)$ points with values of u_i^j at time $t = j * Nout$.

2.4 usefull commands

- to measure computation time, command **time** is used :
  time ./bin/h2d_gcc.exe
  Remark : time measure can be considered reliable only if it's greater than 10 seconds.
- How to plot 3D data with gnuplot. In following, we suppose NX=64 and NY=64

```
gnuplot> set dgrid3d 64,64
gnuplot> set hidden3d
gnuplot> splot "H2D_0000.dat" u 1:2:3 with lines
```

- How to plot 1D data with gnuplot.

```
gnuplot> plot "out_0000.dat" with lines
```

- with Octave
  - to read datas use function *load*
  - to plot datas $u(x, y)$, use function *surf*

  ```
  octave> data=load('H2D_0000.dat');
  octave> surf(data)
  ```

- download code in a terminal :
  wget https://github.com/GFuhr/MF_FCM6/zipball/master
- unzip archive :
  unzip master
- put unzipped files in a directory with a "friendly name" :
  mv GFuhr-MF_FCM6* *newname*
- change directory :
  cd
- create directory mkdir
  mkdir directory_name
- delete file :
  rm filename
- list file in a directory :
  ls directory_name
- list file in current directory :
  ls ./