Advection and Wave Equations

NAME :

This lab work aims at studying several numerical schemes to solve the advection and wave equations. The lecture notes and trial programs can be downloaded at: http://userpages.irap.omp.eu/ rbelmont/.

1 Advection Equation

He we want to solve numerically the 1D advection equation, for a field $u(t,x)$ on the domain $x \in [0,L]$, with a constant velocity v, an initial condition and a set of **periodic** boundary conditions :

$$\frac{\partial u}{\partial t} + v\frac{\partial u}{\partial x} = 0$$
$$u(0,x) = \sin\left(\frac{2\pi x}{L}\right)$$
$$u(t,0) = u(t,L)$$

It is recalled that the CFL parameter for the advection equation is:

$$\lambda = \frac{v\delta t}{\delta x}$$

The found numerical solutions will be compared to the following analytical solution to this equation:

$$T(t,x) = \sin\left(\frac{2\pi(x-vt)}{L}\right)$$

Without loss of generality, we can use $L = 1$m and $v = 1$ ms^{-1}. Also, to start with, we can use $n_x = 128$ points in x, a conservative CFL condition $\lambda = 0.25$ and integrate up to $t_{\max} = 0.1$s. Then, these parameters will be varied.

Note that it can be clever to use integration time multiple of the box crossing time in order to check the behaviour at large time.

1.1 Explicite schemes

An initial python program is proposed in file `adv_expl.py`.
Beware: the proposed code does not use ghost cells to implement the boundary conditions. Instead, it assumes to write specifically the scheme for the boundary points.

Q01− Understand and fill that program with the FTCS scheme for advection.

Q02− Check the effect of the CFL parameter and the scheme stability.

Q03– Find the definition and properties of the explicit **Lax-Friedrich** scheme for advection. Code it.

Q04– Check the effect of the CFL parameter and the stability of the Lax-Friedrich scheme.

Q05– What numerical effect are observed at large time? How is called this phenomenon? Interpret these effect with the consistency error analysis.

Q06– For a time long enough (e.g. $t_{\max} = 10$s) and $\lambda = 0.99$ use a syntax such as: `print "Erreur au max: ", abs(1-max(u)/max(uexacte(t,x)))` to measure the relative error at the Gaussian maximum, for different space resolutions.

n_x	128	256	512	1014
ϵ				

Are these results compatible with the scheme order?

Q07– Implement the **upwind** scheme for advection.

Q08– Check the effect of the CFL parameter and the stability of the upwind scheme.

Q09– What numerical effect are observed at large time ? Interpret these effect with the consistency error analysis.

Q10– Change the velocity sign. What change must be made to conserve the upwind nature of the scheme?

1.2 Schéma implicite de Crank-Nicolson

An initial program is proposed in file `adv_impl.py`. It assumes that the implicit scheme can be written in a matrix form with a cyclic matrix. n that case, its inversion can be performed with algorithms that are much more efficient than in the general case. In python, such matrix inversion is performed with function `cyclic` which requires 5 arguments : the 3 diagonals, the upper-right and lower-left corners, and the right hand side.

$$\begin{pmatrix} b_0 & c_0 & 0 & ... & 0 & 0 & \beta \\ a_0 & b_1 & c_1 & ... & 0 & 0 & 0 \\ 0 & a_1 & b_2 & ... & 0 & 0 & 0 \\ ... & ... & ... & ... & ... & ... & ... \\ 0 & 0 & 0 & ... & b_{n-3} & c_{n-3} & 0 \\ 0 & 0 & 0 & ... & a_{n-3} & b_{n-2} & c_{n-2} \\ \alpha & 0 & 0 & ... & 0 & a_{n-2} & b_{n-1} \end{pmatrix} \begin{pmatrix} u_0 \\ u_1 \\ u_2 \\ ... \\ u_{n-3} \\ u_{n-2} \\ u_{n-1} \end{pmatrix} = \begin{pmatrix} s_0 \\ s_1 \\ s_2 \\ ... \\ s_{n-3} \\ s_{n-2} \\ s_{n-1} \end{pmatrix}$$

Also, it is recalled that the error of the Crank-Nicolson scheme is

$$\epsilon \sim c(dx^2/6)(1-\lambda^2/2)d^3u/dx^3$$

Q11– By adapting the implicit scheme for diffusion, write the Crank-Nicolson scheme for advection for all inner points.

Q12– Write the same scheme at the border using periodic boundary conditions.

Q13– Show that the system can be written in a matrix form $M_{ij}u_j = s_i$ where M_{ij} is cyclique. Write the components a_i of the lower diagonal, b_i on the main diagonal, c_i on the upper diagonal, α the lower left corner, and β the upper right corner. Also calculate the components s_i of the right hand side.

Q14– Understand and fill the proposed program with the Crank-Nicolson implicit scheme with a sinus initial condition of frequency k: $u(0,x) = \sin(2\pi kx/L)$.

Q15– Check the stability and convergence of this scheme (it will be worth using a sparse resolution, e.g. $n_x = 64$).

Q16– How does the numerical solution compare to the analytical one when the time grows long enough? How is this effect called? Interpret this effect in terms of consistency errors. Which solution does lag the other? The numerical one or the analytical one?

Q17– We are now interested in this lag Δx measured at $t_{\max} = 10$s, and with a CFL parameter $\lambda = 1$. This lag will be measured in python with commands such as:

```
nn = 20000; xx = linspace(0,L-dx,nn)
uuexa = interp(xx,x,uexacte(t,x))
uunum = interp(xx,x,u)
print "Dx = ",(argmax(correlate(uunum,uuexa,’full’))-nn+1 )*1.*L/nn
```

Measure this lag for $k = 1$ and different space resolutions and then for $n_x = 256$ and different signal frequencies:

$k = 1, n_x = ...$	32	64	128	256	512
ϵ					

$n_x = 256, k = ...$	1	2	4	6
ϵ				

Is it compatible with the phase error given in the lecture notes for second order, centered in space schemes (in $\Delta x \propto (kdx)^2$)?

Q18– At last, study for $0 < t < 10$s, the time advection of a Gaussienne ($n_x = 128$), and then of a door[1] ($n_x = 256$). Interpret the effects in terms of waves.

1.3 Non-linear advection equation

The linear advection equation is a good case to test the numerical schemes as the numerical solutions can be readily compared to analytical ones. However, numerical solutions are developed in order to solve equations for which there is no simple analytical solution. The Burgers equation is a classical example of non-linear advection equation :

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = 0$$

Such terms naturally appear in the Lagrangian derivative $d/dx = \partial/\partial t + \vec{u}.\vec{\nabla}$ of the hydrodynamical equations.

Q19–Adapt the Lax-Friedrich scheme to solve this equation with a initial sinus wave. Assuming that with such an initial condition where $|u(0,x)| < 1$, the signal always remains smaller than unity: $\forall t > 0, |u(t,x)| < 1$, how must the CFL parameter be chosen to ensure stability?

Q20– What do you observe? Interpret the steepening in terms of variable advection velocities.

[1] $u(0,x) = 1$ for $L/2 - l/2 < x < L/2 + l/2$, and 0 elsewhere

2 Wave equation

Here we want to solve numerically the wave equation on the domain $x \in [0, L]$ for a field $u(t, x)$. It reads:

$$\frac{\partial^2 u}{\partial t^2} - c^2 \frac{\partial^2 u}{\partial x^2} = 0$$

where c is the propagation speed. Once discretized in time and space with steps δt and δx respectively, the CFL parameter for the wave equation is: $\lambda = c\delta t/\delta x$.

This differential equation is second order in time which means that two initial conditions must be provided, for instance the initial field $u(0, x)$ and its initial derivative $u'(0, x)$.

Q21−Write an explicit scheme that is centered in space **and in time**. To that purpose, apply the same kind of methods that the one used for the spatial derivative of the diffusion equation.

Numerically speaking, it is a 2-step method, that is a double recurrence. Two arrays are thus necessary (u_a and u_b for instance). At each time step, they will represent the value of the field at the two previous time steps. The scheme must also be initialized with the values $u(x)$ at the two first time steps. We will assume that the first time step u^1 can be computed as fonction of the initial conditions u_0 and u_0' as:

$$\forall i, \quad u_i^1 = u_i^0 + dt(u')_i^0 + \lambda \left(u_{i+1}^0 - 2u_i^0 + u_{i-1}^0\right)$$

Q22− Implement this numerical scheme with periodic boundary conditions and an initial static Gaussian.

Q23− What numerical effect is observed at large time?

Q24− Implement and test other boundary conditions. For instance, reflective boundary conditions correspond to $du/dx = 0$ at the boundaries.