UdeMM - Ingeniería de Sistemas

Simulación de modelos en derivadas parciales

Trabajo Práctico 3

Año 2020

Módulo 3

Problemas de Advección

1. Burgers con MacCormack

Considere la ecuación de Burgers 1D:

$$\frac{\partial u}{\partial t} = -u\frac{\partial u}{\partial x} \tag{1}$$

Es posible representar esta expresión en forma conservativa,

$$\frac{\partial u}{\partial t} = -\frac{\partial F}{\partial x} \tag{2}$$

si tomamos

$$F = \frac{u^2}{2}$$

La condición inicial se encuentra descripta por la función escalón (función *Heaviside*) siguiente:

$$u(x,0) = \begin{cases} 1 & 0 \leq x < 2 \\ 0 & 2 \leq x \leq 4 \end{cases} \tag{3}$$

(A) Cree una interfaz **Jupyter** e importe los siguientes paquetes Python requeridos en el desarrollo del presente cuaderno (instalar previamente JSAnimation):

```
%matplotlib inline
import numpy
from matplotlib import pyplot
from matplotlib import animation
from JSAnimation.IPython_display import display_animation
```

(B) Complete la función `u_initial` la cual genere un vector con la condición inicial dada:

```
def u_initial(nx):
    #codigo aqui
    return u
```

(C) Escriba la función `computeF` la cual genere un vector $F(u)$.

(D) Empleando el esquema de MacCormack

$$u_i^* = u_i^n - \frac{\Delta t}{\Delta x}\left(F_{i+1}^n - F_i^n\right) \quad \text{(predictor)} \tag{4}$$

$$u_i^{n+1} = \frac{1}{2}\left(u_i^n + u_i^* - \frac{\Delta t}{\Delta x}\left(F_i^* - F_{i-1}^*\right)\right) \quad \text{(corrector)} \tag{5}$$

complete la función que se propone a continuación:

```
def maccormack(u,nt,dt,dx):
    un = numpy.zeros((nt,len(u)))
    un[:] = u.copy()
    ustar = u.copy()
    for n in range(1,nt):
        F = computeF(u)
        ustar =
        Fstar = computeF(ustar)
        un[n] =
        u = un[n].copy()
    return un
```

(E) Procure correr el siguiente fragmento de código:

```
nx = 81
nt = 70
dx = 4.0/(nx-1)

def animate(data):
    x = numpy.linspace(0,4,nx)
    y = data
    line.set_data(x,y)
    return line,

u = u_initial(nx)
sigma = .5
dt = sigma*dx

un = maccormack(u,nt,dt,dx)

fig = pyplot.figure();
ax = pyplot.axes(xlim=(0,4),ylim=(-.5,2));
line, = ax.plot([],[],lw=2);

anim = animation.FuncAnimation(fig,animate,frames=un,
                               interval=50)
display_animation(anim,default_mode='once')
```

1.1. Añadir amortiguamiento

Las oscilaciones que se observan en el frente de la onda de choque son características de los métodos de segundo orden. Una alternativa para atenuar su efecto consiste en recurrir a la amortiguación numérica. Un término de amortiguamiento que habitualmente se emplea con MacCormack es

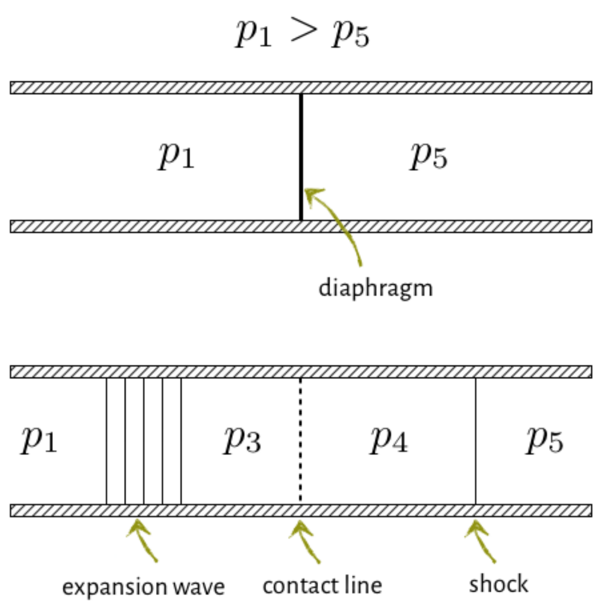
$$\epsilon \left(u_{i+1}^n - 2u_i^n + u_{i-1}^n\right) \tag{6}$$

(F) Procure agregar este término al paso *predictor* de la función `maccormack` y experimente con valores de $0 < \epsilon < 1$. Encuentre un valor de ϵ que logre disipar las oscilaciones sin afectar en gran medida el resto de la solución.

2. Trabajo de código: Tubo de choque

Un tubo de choque consiste en un dispositivo ideal que genera una onda de choque 1D en un gas compresible. El problema admite una solución analítica de las ecuaciones de Euler, resultando un ensayo prototipo útil para estimar la preciśion de esquemas numéricos. Consideremos un tubo que presenta dos regiones conteniendo un gas a diferentes presiones, ambas separadas por un diafragma rígido. El gas se encuentra inicialmente en reposo, y la región a la izquierda del diafragma a una presión mayor con respecto a la región derecha. En el instante $t = 0$ el diafragma se retira y se genera en consecuencia una onda de choque: el gas a alta presión avanza hacia la región de baja presión y se establece un flujo transitorio 1D el cual consiste en:

- una onda de choque que viaja hacia la derecha;
- una onda de expansión que viaja hacia la izquierda;
- una discontinuidad en movimiento.



2.1. Ecuaciones de Euler

Las ecuaciones de Euler gobiernan el movimiento de un flujo invíscido (no viscoso) a través de las leyes de conservación de la masa, cantidad de movimiento y energía. Consideremos un flujo 1D con velocidad u en la dirección x. Las ecuaciones de Euler para un fluido con densidad ρ y presión p se escriben:

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x}(\rho u) = 0 \tag{7}$$

$$\frac{\partial}{\partial t}(\rho u) + \frac{\partial}{\partial x}(\rho u^2 + p) = 0 \tag{8}$$

$$\frac{\partial}{\partial t}(\rho e_T) + \frac{\partial}{\partial x}(\rho u e_T + pu) = 0 \tag{9}$$

siendo $e_T = e + u^2/2$ la energía total por unidad de masa, igual a la energía interna más la energía cinética. En formulación vectorial tendremos

$$\frac{\partial}{\partial t}\underline{\mathbf{u}} + \frac{\partial}{\partial x}\underline{\mathbf{f}} = 0 \tag{10}$$

siendo $\underline{\mathbf{u}}$ el vector de variables conservativas:

$$\underline{\mathbf{u}} = \begin{bmatrix} \rho \\ \rho u \\ \rho e_T \end{bmatrix} \tag{11}$$

y $\underline{\mathbf{f}}$ el vector de flujos:

$$\underline{\mathbf{f}} = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ (\rho e_T + p) u \end{bmatrix} \tag{12}$$

2.2. El método de Richtmyer

De la misma forma que en el esquema de MacCormack, el esquema de Richtmyer es un método de dos pasos, dado por:

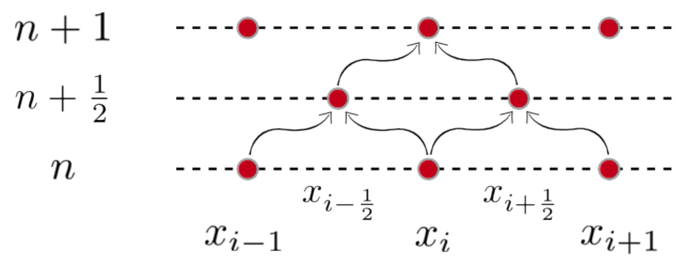
$$\underline{\mathbf{u}}_{i+\frac{1}{2}}^{n+\frac{1}{2}} = \frac{1}{2}\left(\underline{\mathbf{u}}_{i+1}^{n} + \underline{\mathbf{u}}_{i}^{n}\right) - \frac{\Delta t}{2\Delta x}\left(\underline{\mathbf{f}}_{i+1}^{n} - \underline{\mathbf{f}}_{i}^{n}\right) \tag{13}$$

$$\underline{\mathbf{u}}_{i}^{n+1} = \underline{\mathbf{u}}_{i}^{n} - \frac{\Delta t}{\Delta x}\left(\underline{\mathbf{f}}_{i+\frac{1}{2}}^{n+\frac{1}{2}} - \underline{\mathbf{f}}_{i-\frac{1}{2}}^{n+\frac{1}{2}}\right) \tag{14}$$

Los vectores de flujo empleados en el segundo paso se obtienen al evaluar las funciones de flujo en la salida del primer paso:

$$\underline{\mathbf{f}}_{i+\frac{1}{2}}^{n+\frac{1}{2}} = \underline{\mathbf{f}}\left(\underline{\mathbf{u}}_{i+\frac{1}{2}}^{n+\frac{1}{2}}\right) \tag{15}$$

El primer paso es un **predictor** de la solución, consistente con un esquema de Lax-Friedrichs. El segundo paso es un **corrector** en el cual se aplica un esquema de salto de rana (leap-frog). La siguiente figura ilustra la grilla de cálculo para el esquema de Richtmyer, donde el índice asociado al tiempo intermedio $n + 1/2$ requerirá de una variable temporaria (auxiliar) en el código.



2.3. Condiciones del ensayo

Considere un tubo que abarque una longitud comprendida entre $x = -10\,\mathrm{m}$ y $x = 10\,\mathrm{m}$ con la membrana situada en $x = 0\,\mathrm{m}$, y la siguiente condición inicial:

$$\underline{IC}_L = \begin{bmatrix} \rho_L \\ u_L \\ p_L \end{bmatrix} = \begin{bmatrix} 1{,}0\, kg/m^3 \\ 0\, m/s \\ 100\, kN/m^2 \end{bmatrix} \tag{16}$$

$$\underline{IC}_R = \begin{bmatrix} \rho_R \\ u_R \\ p_R \end{bmatrix} = \begin{bmatrix} 0{,}125\, kg/m^3 \\ 0\, m/s \\ 10\, kN/m^2 \end{bmatrix} \tag{17}$$

siendo $\underline{IC}_L$ el vector representativo de la densidad, velocidad y presión iniciales en el lado izquierdo de la membrana, mientras que $\underline{IC}_R$ representa el vector para la densidad, velocidad y presión iniciales en el lado derecho de la membrana.
En el instante $t = 0\,\mathrm{s}$ la membrana se retira. Calcule la densidad, velocidad y presión a través del tubo de choque en el instante $t = 0{,}01\,\mathrm{s}$ empleando el método de Richtmyer. Tenga en cuenta los parámetros siguientes en su simulación:

```
nx = 81
dx = .25
dt = .0002
gamma = 1.4
```