Taller #7. Física Computacional / FISI 2025
Semestre 2013-II.
Profesor: Jaime E. Forero Romero

Octubre 31, 2013

Esta tarea debe resolverse por parejas (i.e. grupos de 2 personas) y la respuesta debe estar en un repositorio de github con un commit final antes del medio día del martes 5 de noviembre del 2013.

El objetivo de este taller es resolver la ecuación de Burgers mediante dos métodos numéricos diferentes.

La ecuación de Burgers es:

$$\frac{\partial u}{\partial t} + cu\frac{\partial u}{\partial x} = 0. \tag{1}$$

Esta ecuación se puede reescribir de la siguiente manera

$$\frac{\partial u}{\partial t} + c\frac{\partial (u^2/2)}{\partial x} = 0. \tag{2}$$

.

Un posible esquema para resolver esta ecuación es uno que calcula las derivadas como diferencias centradas, es decir, la derivadas en el punto i dependen de los valores en $i-1$ y $i+1$:

$$u_i^{j+1} = u_i^{j-1} - \beta\left[\frac{(u^2)_{i+1}^j - (u^2)_{i-1}^j}{2}\right] \tag{3}$$

donde u^2 es la velocidad al cuadrado y $\beta = c/(\Delta x/\Delta t)$ es conocido como el número de Courant-Friederichs-Levy. Para que este esquema funcione numéricamente se debe cumplir la condicion $\beta < 1.0$.

Un segundo esquema más preciso (conocido como el método de Lax-Wendroff) hace uso de una expansión de hasta segundo orden de la función. En este nuevo esquema la solución está dada por:

$$\begin{aligned} u_i^{j+1} &= u_i^j - \frac{\beta}{4}[(u^2)_{i+1}^j - (u^2)_{i-1}^j] \qquad (4) \\ &+ \frac{\beta^2}{8}[(u_{i+1}^j + u_i^j)((u^2)_{i+1}^j - (u^2)_i^j) \end{aligned}$$

$$- \quad (u_i^j + u_{i-1}^j)((u^2)_i^j - (u^2)_{i-1}^j)].$$

1. Escriba un programa que resuelva la ecuación de Burgers usando el esquema numérico de la ecuación 2.
2. Defina arrays con 200 items para las condiciones iniciales y para la solución.
3. Tome la condición inicial como la función sinusoidal $3.0\sin(\pi x)$ donde $0 < x < 2$ y la velocidad del sonido es $c = 1$.
4. Tome las condiciones de contorno fijas donde $u(x = 0) = u(x = 1) = 0$.
5. Encuentre la solución cuando el tiempo total transcurrido es $T = 0.15$. Defina Δt tomando en cuenta la condición de estabilidad $\beta < 1$.
6. Guarde todos los valores de u para todas las iteraciones.
7. Grafique la condición inicial y los valores de u en una gráfica 3D (x, t, u) para ver la evolución de la solución.
8. Repita todos los pasos anteriores pero esta vez usando el esquema de Lax-Wendroff. Mantenga los mismos valores de Δt que utilizó anteriormente.
9. Comente las diferencias entre los resultados el primer método y Lax-Wendroff.

Todo el desarrollo debe estar en un único notebook de Ipython.