Taller #7 de Métodos Computacionales
FISI 2028, Semestre 2014 - 20

Profesor: Jaime Forero

Viernes 31 de Octubre, 2014

Importante

- Todo el código fuente y los datos se debe encontrar en un repositorio en github con un commit final hecho antes del medio día del martes 11 de noviembre. El nombre del repositorio debe ser `Apellidos1_Apellidos2_hw7`, por ejemplo si trabajo con Nicolás deberíamos crear un repositorio llamado `Forero_Garavito_hw7`.
- La nota máxima de este taller es de 100 puntos. Se otorgan 1/3 de los puntos si el código fuente es razonable, 1/3 si se puede compilar/ejecutar y 1/3 si da los resultados correctos.
- El miercoles 5 de noviembre habrá una sesión para presentar avances de diferentes grupos en cada una de las preguntas del taller. Hay un bono de 15 puntos para cada grupo que presente un avance significativo.

1. **Cuerda Vibrando** (50 puntos)

   (Ver capítulo 18.2 de Landau, Paez, Bordeianu, *Solución de la ecuación de onda hiperbólica*)

   Vamos a considerar una cuerda de longitud L descrita por la función $u(x,t)$ que corresponde al desplazamiento con respecto a su posición de equilibrio. Después de una perturbación inicial, la evolución de u está dada por

   $$\frac{\partial^2 u(x,t)}{\partial x^2} = \frac{1}{c^2}\frac{\partial^2 u}{\partial t^2} \tag{1}$$

   donde $c = \sqrt{T/\rho}$ es una velocidad de propagación con T la tensión de la cuerda y ρ su densidad [1].

   Las condiciones de contorno corresponden a puntos fijos. Como condición inicial se tiene que la cuerda está estirada de forma triangular, con el máximo ubicado a 8/10 de la longitud total de la cuerda con una altura 1. Es decir,

   $$u(x, t=0) = \begin{cases} 1{,}25x/L & x \leq 0{,}8L \\ 5 - 5x/L & x > 0{,}8L \end{cases} \tag{2}$$

   Integre esta ecuación tomando $T = 40$N y $L = 100$m para $0 < t < 120$s.

   El programa debe estar escrito en C y debe poder ejecutarse como `./string.x rho` donde `rho` es la densidad lineal de la cuerda. Para hacer pruebas pueden tomar $\rho = 0{,}01$kg/m.

   Este programa debe producir un archivo llamado `string_rho.dat` donde se guarda la evolución del sistema en 121 filas (cada fila correspondiente a la evolucion para el tiempo en segundos correspondiente, i.e. la fila 20 es la evolución para $t = 19$s) y 101 columnas (cada columna tiene el valor de y para el x correspondiente, i.e. la columna 99 guarda el valor de u para $x = 98$m). Debe existir otro programa en python que toma como argumento el archivo de datos para producir una gráfica `cuerda_rho.pdf` del mismo tipo de la Figura 18.3 en el libro de Landau, Paez, Bordeianuu. Todo debe poder ejecutarse con un Makefile.

[1] En clase ya resolvimos esta ecuación con diferentes condiciones iniciales: un pulso gaussiano y uno sinusoidal

2. **Shock**. (50 puntos)

   Vamos a trabajar con un fluido unidimensional definido por tres cantidades: la velocidad u, la presión p y la densidad ρ. El gas esá descrito por una ecuación de estado con exponente adiabático $\gamma = 1{,}4$. Vamos a asumir que este fluido se puede describir por las ecuaciones de Euler, que no son nada más que una expresión de la conservación de masa, momentum y energía.

   El objetivo de este punto es resolver las ecuaciones de Euler para la siguiente situación. Hay un tubo que va desde $x = -10$m hasta $x = 10$m y en la mitad hay una membrana rígida. Del lado izquierdo hay un gas es estático $u_I = 0$m/s, tiene densidad alta $\rho_I = 1$kg/m^3 y una presión alta $p_I = 100$kN/m^2. Del lado derecho hay más gas que sigue siendo estático $u_D = 0$m/s pero tienen una densidad baja $\rho_D = 0{,}125$ kg/m^3 y una presión más baja $p_D = 10$kN/m^2.

   Para $t = 0$ la membrana desaparece y el gas es libre de evolucionar siguiendo las ecuaciones de Euler. El objetivo es calcular la presión, densidad y velocidad a lo largo del tubo para el tiempo $t = 0{,}1$s.

   ¿Cómo van a hacer esto? Siguiendo el esquema numérico presentado aquí:

   `http://nbviewer.ipython.org/github/numerical-mooc/numerical-mooc/blob/master/lessons/03_wave/03_05_Sods_Shock_Tube.ipynb`.

   No es necesario que lean las lecciones anteriores del MOOC para resolver el problema. Basta con que vean con cuidado las ecuaciones (8) y (9) que describen el esquema numérico necesario para evolucionar las variables del sistema.

   El programa debe estar escrito en C y debe poder ejecutarse como `./sod_test.x t` donde `t` es el tiempo en segundos al cual se desea evolucionar la solución. Este programa debe producir un archivo llamado `estado_t.dat` donde se guarda en cuatro columnas: la posición a lo largo del tubo, la velocidad, presión y densidad. Debe existir otro programa en python que toma como argumento el archivo de datos para producir una gráfica `estado_t.pdf`. Todo debe poder ejecutarse con un Makefile.