Laboratorio #11 Métodos Computacionales. 2018-2
Tema: Ecuaciones Diferenciales Parciales

prof. Carolina Garcia sec 3. Daniel Lozano Gómez sec 1 y 2.

En el laboratorio de la clase de hoy se practicarán habilidades para resolver ecuaciones diferenciales parciales aplicables a la Modelación Numérica de la Atmósfera.

Usaremos el modelo más simple y notrivial, este corresponde con la ecuación de advención:

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

en donde c es la velocidad constante de un flujo y $a(x,t)$ es un campo escalar pasivo. La forma discretizada de la ecuación de usando el método Lax-Wendroff es:

$$\frac{a_i^{n+1} - a_i^n}{\tau} = -\frac{c}{2h}(a_{i+1}^n - a_{i-1}^n) + \frac{c^2}{2h^2}(a_{i+1}^n + a_{i-1}^n - 2a_i^n) \tag{2}$$

1. (5 puntos) En un programa de C++ defina un pulso gausiano modulado por la función coseno, así:

$$a(x,t) = \cos\left(k(x - [x_0 + ct]\right) \exp\left[-\frac{[x - (x_0 + ct)]^2}{2\sigma^2}\right] \tag{3}$$

2. (5 puntos) Defina como condición inicial, en tiempo cero en nuestro problema, el pulso anterior, con $\sigma = 0.1$ y $k = \pi/\sigma$. Guarde en un archivo dicho pulso para posiciones desde $-L/2$ a $L/2$, con un total de 50 puntos, para $x_0 = 0$ y $c = 1$. El archivo debe tener dos columnas correspondientes con xa.

3. (20 puntos) Implemente el método de Lax-Wendroff para solucionar la ecuación de advención para un pulso de amplitud $a(x,t)$, para una velocidad de flujo $c = 1$.

4. (5 puntos) Propague el pulso inicial definido anteriormente, a través del flujo dado por la ecuación de advención. Guarde en un archivo el pulso final para un tiempo total de $t_w = 0.02$ con un espaciado de $\tau = 0.015$. El archivo debe tener 2 columnas xa para las posiciones desde $-L/2$ a $L/2$, como se hizo anteriormente.

5. (5 puntos) Descargue o copie el archivo plotxyDos.py del enlace *https://github.com/profcarogarbo /cccpp-p/blob/master/plotxyDos.py.* Ejecute el *script* y guarde la gráfica obtenida a partir de los dos archivos .txt anteriores. Estos deben llamarse *inicial.txt* y *final.txt* respectivamente. Renombre la gráfica a *grafica1.jpg*

6. (5 puntos) Use el comando $*std :: max\_element(arr, arr + tam)$; de la libreria *algorithm* para normalizar los pulsos tanto inicial como final, arr es el nombre del arreglo donde se tiene guardado el pulso y *tam* es el tamaño correspondiente. Vuelva a graficar y guarde el resultado en un archivo llamado *grafica2.jpg*

7. (10 puntos) Cambié el valor de τ con 5 valores diferentes y escriba en un comentario en su código que sucede con el pulso. Guarde las gráficas correspondientes en su carpeta con los nombre *grafica3tn.jpg*, en donde n corresponde con el valor de *tau* utilizado.

8. (3 puntos) Envíe en el link de Sicuaplus la carpeta comprimida con todas las gráficas y códigos con su NOMBRE.zip.

9. (10 puntos) BONO. En un comentario final en su programa o *script*, identifique si hay errores en las siguientes expresiones corrija y cree nuevas variables, según el caso, a excepción de la función *area()*.

- int s1=area(7;
- int s2=area(7)
- Int s3=area(7);
- int s4=area('7)
- `vector v(5); for(int i=0;i<=v.size();++i); cout<<"caribe\n";`
- cin << "*conseguido*!";
- `int x=4; double d=5/(x-2); if(d=2*x+0.5) cout<< "success\n";`