Métodos de Simulación Física: Segundo Parcial

Nombre____ Cédula ________________________

Instrucciones generales

El parcial está diseñado para desarrollarse en 3 horas, aproximadamente. Pasado ese tiempo, debe hacerse un primer envío al correo jdmunozcsimulacion@gmail.com colocando el subject "Segundo Parcial Simulación: [NOMBRE], [CÉDULA]", reemplazando los espacios de [NOMBRE] y [CÉDULA] con su nombre y su cédula, respectivamente. El envío debe contener todos los códigos .cpp, las gráficas en .jpg como attachments, y los datos que se le pidan como parte del texto. Luego, pueden hacer un segundo envío antes de las 11:59pm del día domingo 23 de junio. El primer envío tiene en la nota un peso del 80%, y el segundo, del 20%.

Buena suerte y buen pulso!!

Problema a desarrollar

Como comentamos en clase, es bien sabido que un modelo de lattice-Bolzmann LBGK para fluidos, con regla de evolución

$$f_i(\vec{x} + \vec{v}_i \Delta t, t + \Delta t) - f_i(\vec{x}, t) = -\frac{1}{\tau}\left[f_i(\vec{x}, t) - f_i^{eq}(\vec{x}, t)\right]$$

y función de equilibrio

$$f_i^{eq} = w_i \rho \left(1 + 3\vec{v}_i \cdot \vec{U} + \frac{9}{2}\left(\vec{v}_i \cdot \vec{U}\right)^2 - \frac{3}{2}\left\|\vec{U}\right\|^2\right)$$

cumple la ecuación de advección-difusion,

$$\frac{\partial \rho}{\partial t} + \nabla \cdot \left(\rho \vec{U}\right) = D \nabla^2 \vec{U} \quad , \quad \text{con} \quad D = \frac{1}{3}\left(\tau - \frac{1}{2}\right)\Delta t$$

si el campo de velocidades $\vec{U}$ se toma impuesto desde el exterior (en vez de calcularlo por una sumatoria) y si la densidad ρ se calcula como

$$\rho = \sum_i f_i \quad .$$

1. Condición Inicial

a) (16pts) Tome como punto de partida el modelo de lattice-Boltzmann realizado en clase para fluidos y defina un espacio de simulación con $L_x = 256$ y $L_y = 64$ con condiciones de frontera periódicas. Elimine la función de imponer campos y modifique las funciones J_x y J_y que calculan las componentes de la densidad de momentum para que retornen el valor cero (0). Imponga como condición inicial una distribución gaussiana en dirección x y uniforme en dirección y (lo que se logra imponiendo que las funciones f_i tomen como valores iniciales los valores de equilibrio para $U_x = U_y = 0$ y $\rho = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{ix-\mu}{\sigma}\right)^2\right]$, con $\mu = L_x/8$ y $\sigma = L_y/9$. Implemente una función en la clase LatticeBoltzmann que imprima en un archivo (digamos Densidad.dat) las tres cantidades ix iy ρ (calculada con las funciones f_{new}) separadas por espacios y finalizado en return para todas las celdas del espacio de simulación. Finalmente, corra la evolución del sistema para tmax=1 y grafique (en gnuplot, o en Python) el archivo producido como una gráfica de densidades codificadas por color. Recuerde que esto se puede hacer entrando a gnuplot y corriendo los comandos

```
set pm3d map
set size ratio -1
set terminal jpeg enhanced
set output "Densidad.jpg"
splot "Densidades.dat"
```

Compruebe que la densidad inicial aparece como una franja vertical.

De este primer punto, envíe:

- *(10pts) El programa .cpp que hace la simulación (ojalá con animación)*
- *(6pts) El .jpg que muestra la distribución de densidad inicial*

2. Difusión Pura

b) (12pts) Cambie $\mu = L_x/2$ y construya una función en la clase LatticeBoltzmann que retorne el valor de la varianza de la distribución marginal en x. En otras palabras, que calcule

$$\sigma^2 = \frac{1}{N} \sum_{\substack{todas\ las \\ celdas}} \rho(ix, iy, true) * (ix - x_{\text{prom}})^2 \quad ,$$

con

$$x_{\text{prom}} = \frac{1}{N} \sum_{\substack{todas\ las \\ celdas}} \rho(ix, iy, true) * ix \quad \text{y} \quad N = \sum_{\substack{todas\ las \\ celdas}} \rho(ix, iy, true) \quad .$$

Luego, deje correr $t_{\text{max}} = 1000$ pasos y compruebe que la varianza crece linealmente con el tiempo con pendiente $2D$.

De este segundo punto, envíe:

- *(7pts) El programa .cpp*
- *(3pts) La gráfica .jpg de $\sigma^2(t)$ hasta t=1000.*
- *(2pts) El valor de la pendiente*

3. Advección-Difusión

c) (12pts) Vuelva a colocar $\mu = L_x/8$ e imponga un flujo de Poiseuille (un campo de velocidades externas con perfil parabólico) en el espacio de simulación. Esto se hace modificando la función J_x para que devuelva el valor

$$J_x = \frac{\rho}{2} \frac{\Delta p}{\eta} \, iy\left(L_y - 1 - iy\right) \quad ,$$

con $\Delta p = 0.001$ y $\eta = 10$. Imprima y grafique en gnuplot o en Pythonel mapa de densidades para t=500, t=1000 y t=2000. Observe cómo la banda que originalmente era vertical se va anchando en el centro y deformándose.

De este tercer punto, envíe:

- *(6pts)El programa .cpp o .ipynb*
- *(6pts) Las gráficas .jpg para* $t = 500, 1000\ y\ 2000$.

4. Instalar un detector

Finalmente, construya una función dentro de la clase LatticeBoltzmann (un "Detector") que devuelva la densidad total en la posición $ix = 120$, es decir que devuelva el valor

$$Detector = \sum_{iy=0}^{Ly-1} \rho(ix = 120, iy, true)$$

Grafique la salida del detector en función del tiempo para $0 \leq t \leq 5000$. Además, grafique en gnuplot el mapa de densidades para t=5000.

De este tercer punto, envíe:

- *(4pts) El programa .cpp*
- *(3pts)La gráfica .jpg de densidades para* $t = 5000$
- *(3pts)La gráfica .jpg que muestra la respuesta del detector en función del tiempo.*