IMT2112 Semestre 2024-2
Tarea 3

Elwin van 't Wout

September 30, 2024

Introducción

Muchas de las ecuaciones diferenciales parciales no tienen una solución analítica y debemos aproximar su solución con métodos numéricos. El método de diferencias finitas es uno de los métodos más populares para aproximar la solución de una ecuación diferencial. En esta tarea consideramos el problema de Laplace en una cuadra, dado por

$$\begin{cases} -\nabla \cdot (\alpha \nabla u) + u = f, & 0 < x, y < 1; \\ u = 0, & x = 0, x = 1, y = 0, \text{ or } y = 1; \end{cases} \tag{1}$$

para la función incognita $u = u(x, y)$ y funciones conocidas $f = f(x, y)$ y $\alpha = \alpha(x, y)$ distintas a cero. En este tarea,

$$\alpha(x, y) = x(x-1)y(y-1) + 1,$$
$$f(x, y) = \frac{e^{-\frac{(x-c_x)^2}{2\sigma_x^2} - \frac{(y-c_y)^2}{2\sigma_y^2}}}{2\pi\sigma_x\sigma_y}$$

para constantes $0 < c_x, c_y < 1$ y σ_x, σ_y.
Creamos una malla rectangular con nodos en los puntos $(x_i, y_j) = (ih_x, jh_y)$ en donde h_x y h_y son los pasos de la malla en la dirección x e y. La versión estandar del método de diferencias finitas está dado por el *stencil*

$$\begin{bmatrix} & -\frac{\alpha_{i,j+\frac{1}{2}}}{h_y^2} & \\ -\frac{\alpha_{i-\frac{1}{2},j}}{h_x^2} & \frac{\alpha_{i-\frac{1}{2},j}+\alpha_{i+\frac{1}{2},j}}{h_x^2} + \frac{\alpha_{i,j-\frac{1}{2}}+\alpha_{i,j+\frac{1}{2}}}{h_y^2} + 1 & -\frac{\alpha_{i+\frac{1}{2},j}}{h_x^2} \\ & -\frac{\alpha_{i,j-\frac{1}{2}}}{h_y^2} & \end{bmatrix}. \tag{2}$$

La variable $\alpha(x, y)$ se evalúa en los puntos del medio, por ejemplo $\alpha_{i,j+\frac{1}{2}} = \alpha(x_i, y_{j+\frac{1}{2}}) = \alpha(ih_x, (j+\frac{1}{2})h_y)$.

Después de numerar los nodos, se puede representar la solución numérica como un vector $\mathbf{x}$, la fuente como $\mathbf{b}$ y la diferencia finita como la matriz A. El sistema lineal resultante es

$$A\mathbf{x} = \mathbf{b}$$

en lo cual A es simétrica y positiva definita si $\alpha > 0$. Por lo tanto, se puede usar gradientes conjugados para resolver el sistema lineal.

Tarea

Esta tarea contempla la implementación del método de gradientes conjugados (CG - *Conjugate Gradients*) en paralelo con la librería OpenMP.

1. Expliquen por qué se puede usar el método de CG para resolver el sistema lineal.
   Sugerencia: el teorema de Gershgorin podría ser útil.

2. En esta tarea, deben usar el almacenamiento por *stencils* para la matriz A. Es decir, deben guardar la matriz A en cinco arreglos bidimensionales para los stencils. También deben adecuar los vectores al formato stencil.

3. Dada la malla rectangular con nodos $(x_i, y_j) = (ih_x, jh_y)$, un número fijo de puntos en cada dirección y utilizando el almacenamiento por *stencil*, expliquen el tamaño y formato de todas las variables en CG.

4. Implementen el algoritmo de CG para este sistema lineal en C++.

   (a) El algoritmo está dado en Figura 5.11 del libro de Eijkhout, en lo cual pueden elegir $K = I$ la matriz 6identidad como preacondicionador.

   (b) Eligen los valores $c_x = 0.4$, $c_y = 0.8$, $\sigma_x = 0.2$ y $\sigma_y = 0.1$.

   (c) Eligen un número fijo de iteraciones.

   (d) Imprimen el valor del residuo en cada iteración para revisar la convergencia.

   (e) Expliquen como implementaron las condiciones de borde.

5. Implementen el algoritmo de CG para este sistema lineal con OpenMP.

   (a) Todas las operaciones de álgebra lineal deben ser paralelizadas con OpenMP, como p.ej., la creación de la matriz, la multiplicación matriz por vector, el producto interno y la suma de vectores.

   (b) Expliquen cómo paralelizaron el código y por qué es confiable, sin *race conditions.*

   (c) En el caso de ciclos anidados, expliquen cuales ciclos paralelizaron y por qué esta decisión es eficiente.

6. Miden el tiempo de cómputo y analizen la eficiencia paralela en un experimento de escalabilidad fuerte con cada vez más hilos.

(a) Se puede usar el clúster de Ingeniería UC o un computador personal, como prefieren. En todos casos, mencionen el número de núcleos disponibles.
(b) Pueden medir el tiempo de cómputo p.ej. con el comando `time` en BASH o con las librerías `crono` y `time` en C++.

Evaluación

Entreguen todo el código y las respuestas a las preguntas en una mapa comprimida a través de Canvas.

Los reglamientos del curso se puede encontrar en Canvas. Se destaca que las tareas deben ser hechas de forma individual y deben cumplir las políticas de uso del clúster de Ingeniería UC.