Pontificia Universidad Católica de Chile
Instituto de Ingeniería Matemática
IMT2112

Tarea 3

4 de noviembre de 2020

Fernando De Diego Ávila

Preguntas

1)

Recordemos que el método de gradientes conjugados está limitado a matrices simétricas y positivas definidas. Notemos que si $N_x = N_y$, la matriz A del sistema lineal resultante $Ax = b$ es simétrica. Esto, debido a que $-\frac{\alpha_{i+\frac{1}{2},j}}{h^2} = -\frac{\alpha_{(i+1)-\frac{1}{2},j}}{h^2}$. Es decir, el valor del elemento Este del *stencil* del nodo $i + j * N_x$ es igual al valor del elemento Oeste del *stencil* del nodo $(i + 1) + j * N_x$. De la misma forma, $-\frac{\alpha_{i,j+\frac{1}{2}}}{h^2} = -\frac{\alpha_{i,(j+1)-\frac{1}{2}}}{h^2}$. Esto es, el elemento Norte del *stencil* del nodo $i + j * N_x$ es igual al elemento Sur del *stencil* del nodo $i + (j + 1) * N_x$. En ambos casos, esos pares de elementos se encuentran en lugares opuestos de la matriz (con respecto a la diagonal).

A continuación, se presenta una representación visual de la matriz resultante, con la cual se puede apreciar su simetría. Se considera una enumeración de los nodos desde la esquina inferior izquierda, tal como se vio en clases:

$$\begin{bmatrix} \frac{\alpha_{\frac{1}{2},1}+\alpha_{\frac{3}{2},1}}{h^2} + \frac{\alpha_{1,\frac{1}{2}}+\alpha_{1,\frac{3}{2}}}{h^2} + 1 & -\frac{\alpha_{\frac{3}{2},1}}{h^2} & 0 & 0 & \dots & 0 & -\frac{\alpha_{1,\frac{3}{2}}}{h^2} & 0 & \dots \\ -\frac{\alpha_{\frac{3}{2},1}}{h^2} & \frac{\alpha_{\frac{3}{2},1}+\alpha_{\frac{5}{2},1}}{h^2} + \frac{\alpha_{2,\frac{1}{2}}+\alpha_{2,\frac{3}{2}}}{h^2} + 1 & -\frac{\alpha_{\frac{5}{2},1}}{h^2} & 0 & \dots & 0 & 0 & -\frac{\alpha_{2,\frac{3}{2}}}{h^2} & \dots \\ \vdots & \vdots & \vdots & & & & & & \end{bmatrix}$$

Ahora, aplicamos el teorema de Gershgorin para obtener los discos de Gershgorin. En cada diagonal se tiene un valor del tipo $d = \frac{\alpha_{i-\frac{1}{2},j}+\alpha_{i+\frac{1}{2},j}}{h^2} + \frac{\alpha_{i,j-\frac{1}{2}}+\alpha_{i,j+\frac{1}{2}}}{h^2} + 1$, mientras que la suma de los valores absolutos de los elementos de las filas (sin contar la diagonal), tienen un valor de a lo más $r \leq |-\frac{\alpha_{i-\frac{1}{2},j}}{h^2}| + |-\frac{\alpha_{i+\frac{1}{2},j}}{h^2}| + |-\frac{\alpha_{i,j-\frac{1}{2}}}{h^2}| + |-\frac{\alpha_{i,j+\frac{1}{2}}}{h^2}|$.

Notemos que $\alpha_{i,j} > 0$, ya que $x(x-1)y(y-1) \geq 0$ para $x, y \in [0, 1]$. Entonces $r \leq \frac{\alpha_{i-\frac{1}{2},j}}{h^2} + \frac{\alpha_{i+\frac{1}{2},j}}{h^2} + \frac{\alpha_{i,j-\frac{1}{2}}}{h^2} + \frac{\alpha_{i,j+\frac{1}{2}}}{h^2}$, con lo que se obtiene que $d \pm r \geq 1$. Como todos los valores

propios pertenecen a la unión de los discos de Gershgorin, y dichos discos se ubican desde 1 en adelante, se concluye que todos los valores propios son positivos, y así la matriz A es positiva definida.

Como A es simétrica y positiva definida, se concluye que podemos usar el método CG.

2)

Consideraré en este análisis un particionamiento del dominio en bloques de filas, que es lo que implementé en mi tarea. En particular, a cada procesador i le asigné un bloque de $f_i = N_y \,//\, p$ filas, salvo por el último, al cual le asigné $f_{p-1} = N_y \,//\, p + N_y \,\%\, p$ filas, donde p es el número total de procesadores. Por su parte, cada fila tenía largo N_x.

Como se nos pidió guardar la matriz A en formato stencil (en este caso, *five-point stencils*), generé 5 arreglos $2D$ distintos, de tamaño $N_x N_y$. Dichos arreglos los guardé a su vez en un arreglo de largo 5, al que llamé *stencils* (por tanto, podría ser considerada una matriz de 3 dimensiones). Es importante notar que en cada procesador generé las filas de *stencils* que le correspondía. Es decir, en cada procesador i generé (y guardé) 5 arreglos $2D$ de tamaño $f_i N_x$.

Por su parte, los vectores los guardé de la misma forma que un *stencil* (arreglo $2D$). Su particionamiento fue realizado de la misma forma que para el caso de los *stencils*. La única diferencia, es que no guardé los vectores en un arreglo común (no lo consideré necesario). Asigné un espacio para todos los vectores del algoritmo x, r, q, p, etc.

Finalmente, para almacenar los números preferí usar variables de tipo *double* para obtener una alta precisión (y considerando que tenemos memoria suficiente).

3)

Notemos que, en formato *stencil*, para realizar un mat-vec con particionamiento de bloques de filas, basta comunicar los procesadores con sus vecinos inmediatos. En particular, solo se requieren la filas inmediatamente anteriores y/o siguientes a las del procesador. Para realizar esta comunicación, implementé una función llamada *vec-communication*, que consiste en los siguiente:

- Si el procesador es par, recibimos y luego enviamos:
  - Si el procesador no es el último, recibimos la primera fila del siguiente.
  - Si el procesador no es el primero, recibimos la última fila del anterior.
  - Si el procesador no es el último, enviamos la última fila al siguiente.
  - Si el procesador no es el primero, enviamos la primera fila al anterior.
- Si el procesador es impar, enviamos y luego recibimos:
  - Si el procesador no es el primero, enviamos la primera fila al anterior.

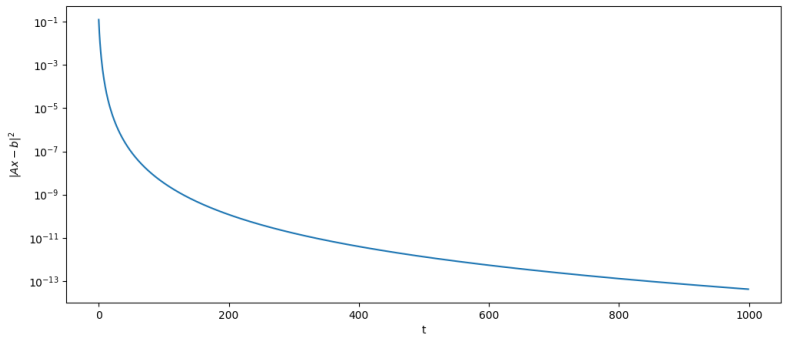
- Si el procesador no es el último, enviamos la última fila al siguiente.
- Si el procesador no es el primero, recibimos la última fila al anterior.
- Si el procesador no es el último, recibimos la primera fila al siguiente.

Esta comunicación fue ejecutada para obtener x antes de realizar el cálculo de r_0 y antes de calcular el residuo de cada iteración, y para obtener p_t antes de calcular $q_t = Ap_t$.

Por otra parte, en cada iteración también fue necesario hacer tres *all-reduce SUM*, dos de los cuales fueron de los resultados locales de los productos puntos, y uno del residuo. Esto lo hice *IN-PLACE*, ya que no había problema de reemplazar los resultados locales por los globales.

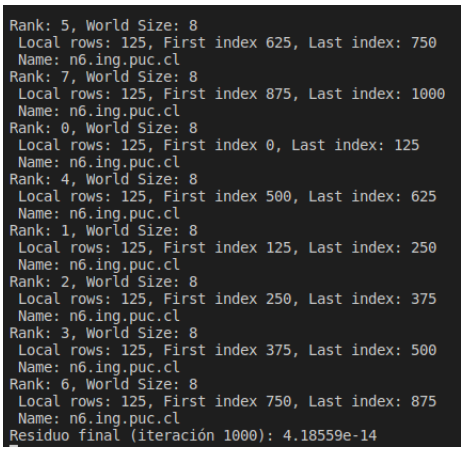
Resultados

Para calcular la efectividad del método, se calcula la norma al cuadrado del residuo $\|Ax-b\|_2^2$. Tanto para funciones puntuales, como para funciones aleatorias, el residuo disminuye en cada iteración. A continuación, se presenta el gráfico del problema con una función puntual y $N = 100$:



Para generar gráficos del mismo tipo para otros casos, basta correr el archivo `plot.py`, entregado en mi solución.

Finalmente, una foto del output generado al correr el programa en el *cluster* de ingeniería es la siguiente:



En la carpeta entregada se encuentra un video que acredita el correcto funcionamiento del programa en el *cluster*.