



Tarea 3

Pregunta 1

Para utilizar el método de gradientes la matriz debe ser simétrica y definida positiva, por lo que mostraremos estas dos características para nuestra matriz.

Simétrica

Supongamos que $N_y = N_x$.

Sea A la matriz generada por el stencil, de $N^2 \times N^2$ elementos, dadas estas condiciones la matriz A es sparse con 5 diagonales. Considerando que los elementos este y oeste del stencil son iguales, además de la igualdad de los elementos sur y norte podemos concluir que la matriz es simétrica.

Definida positiva

Para mostrar que A es definida positiva utilizaremos el teorema de Gershgorin.

Al ser A una matriz simétrica con 5 diagonales, sabemos que la suma por filas o columnas en valores absolutos es menor que el valor de la diagonal principal, podemos comprobarlo al ver que la formula de definición de la diagonal principal tiene las componentes horizontales y verticales sumado a 1, por lo que el menor valor propio posible siempre será mayor o igual a 1.

Pregunta 2

Parte A

El dominio de la grilla se particionó en bloques de columnas, es decir, se dividen las columnas totales en la cantidad de procesadores.

Parte B

Para almacenar las variables se pueden particionar en la cantidad de procesadores, dejando al ultimo de ellos el resto no divisible.

La forma de particionar es algo importante, por lo que explicaré brevemente como se hará para cada elemento, logrando así un buen balance entre cantidad de envíos y cantidad de datos almacenados en cada procesador, donde es ideal que el número de envíos no sea extremadamente grande y que cada procesador tenga un número similar de datos.

Es importante mencionar que para hacer el método del gradiente conjugado se deben hacer multiplicaciones de matriz por vector, vector por vector (producto punto y calculo de norma) y las sumas correspondientes a estos.

Matrices

Considerando que las matrices que usaré serán 5 de tamaño $N_x \cdot N_y$ con las evaluaciones de alfa para cada punto, es decir, que cada una de estas matrices tendrá una posición del stencil evaluado en todos los puntos de la grilla, cabe notar que estas matrices tienen muchas simetrías, además de que algunas de ellas son iguales a la transpuesta de otras. (Siendo extremadamente eficiente solo debería guardar un cuarto de cada matriz de los costados y generar la central con respecto a eso, pero no lo soy, debido a la cantidad de envíos que se requerirían para implementar esto.)

Para particionar cada matriz y evitar engorrosos envíos entre procesadores para las multiplicaciones de matriz por vector necesarias para el algoritmo, cada matriz será particionada por columnas, es decir, que cada procesador tendrá $\frac{N_x}{p}$ columnas, considerando p el número de procesadores y que si esta división no es exacta el último procesador tendrá el resto. De este modo cada procesador tendrá el mismo dominio de la grilla, por lo que no se necesitará hacer envíos para conocer los valores de ese dominio.

Vectores

Es importante notar que los vectores serán particionados por "bloques". es decir que cada procesador tendrá un trozo continuo de datos de largo $\frac{n}{p}$, donde n es el número de datos y p es el número de procesadores, considerando que si la división no es exacta el último procesador tendrá el resto.

Nuestro vector más importante a particionar es el vector solución para el algoritmo que tiene $N_x \cdot N_y$ datos, que deberán ser trabajados en conjunto con la matriz, que tendrá el mismo dominio de datos en cada procesador, por lo que no se necesitarán hacer envíos innecesarios.

Números

Los números generados a lo largo del procedimiento son particularmente creados en el método del gradiente, por lo que cada uno de estos pertenecerá a su procesador correspondiente, (dominio de matriz y vector correspondiente), por lo que ya vendrán particionados por defecto.

Parte C

Es importante notar que los vectores del algoritmo están guardados en forma de matriz (son básicamente un reshape de los vectores), ya que, de este modo se facilita la multiplicación, porque solo se necesita multiplicar una posición en una matriz por la misma posición en la otra matriz, y como estas tienen las mismas dimensiones, estos elementos están en el mismo procesador y no se necesitan implementar envíos.

Pregunta 3

Parte A

Se consideró $K = I$, por lo que no se hace ningún cálculo presente en el código, sin embargo, esto se debe a que la inversa de la identidad sigue siendo la identidad, por lo que la nueva matriz sigue siendo igual a la matriz no preconditionada.

Parte E

Generación de matrices

Al generar las matrices se hace de forma paralela, pero cada trozo generado en cada procesador se guarda ahí mismo y no debe ser compartido, por lo que no es necesario implementar comunicación.

Multiplicación matriz vector

Para la multiplicación matriz por vector se necesita multiplicar el valor de cada punto de las matrices del stencil por la posición correspondiente en el vector (que tiene forma de matriz), sin embargo, como cada procesador solo tiene un número de columnas (claramente menor al total), por lo que necesitaremos los valores de las posiciones este y oeste, en algunos casos, por ejemplo en las columnas borde de cada procesador sin contar la primera y la ultima.

Para implementar estos intercambios de datos en los bordes de las columnas de los procesadores se implementó un proceso en 4 partes. Inicialmente se envían los datos de toda la ultima columna de cada procesador al procesador siguiente, solo en procesadores impares, por lo que los procesadores pares reciben(sin considerar el primer procesador que no necesita recibir datos), luego este mismo proceso se hace de los procesadores pares a los impares. Luego se envía la primera columna de cada procesador(excepto del primer procesador), al procesador anterior, partiendo por lo procesadores pares y con los procesadores impares recibiendo estos datos; posteriormente se realiza el mismo envío desde los procesadores impares a los pares.

Producto punto

Para este proceso se multiplica cada posición del primer vector por la misma posición del segundo vector y se suma este valor en una variable. Luego se envían todas las variables al procesador cero en donde se suman para luego enviar este valor final a todos los procesadores de vuelta. Es importante que todos los procesadores tengan estos valores, ya que, serán utilizados luego como ponderadores, por lo que nos evitamos así hacer envíos en la función de ponderación.

Ponderación por escalar

Como lo dijimos anteriormente en esta función no se hacen envíos, ya que, se evita con los procesos hechos en la función de producto.

Suma

Para la función de suma solo sumamos los valores de dos elementos en la misma posición, y luego este resultado es guardado en la misma posición anterior pero en una nueva variable, por lo que no se necesitan implementar envíos.