Implementación de Jacobi en paralelo

Siendo x^k = [$x_1^{(k)}$, $x_2^{(k)}$, $x_3^{(k)}$,....., $x_m^{(k)}$] la aproximación generada por el método de Jacobi en la k-ésima iteración, entonces la k+1 ésima iteración es x^{k+1} = [$x_1^{(k+1)}$, $x_2^{(k+1)}$, $x_3^{(k+1)}$,....., $x_m^{(k+1)}$] se puede calcular a través de la fórmula:

$$x_i^{(k+1)} = \frac{1}{A_{i,i}} \left(b_i - \sum_{\substack{j=1 \\ j \neq i}}^{m} A_{i,j} x_j^{(k)} \right)$$

Para cada elemento i = 1,2,..,m de x^{k+1} .

Es posible realizar una implementación en paralelo, calculando cada elemento x_i en un procesador (núcleo). Esto se puede representar de forma gráfica así:

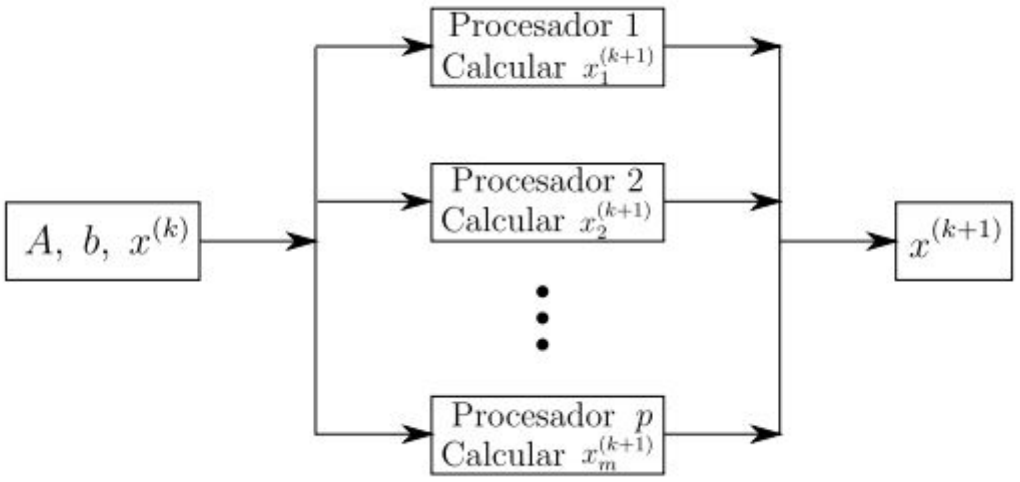

Figura 1: Representación Gráfica de la implementación en paralelo del Método de Jacobi.

Este tipo de implementación se subdivide en varios aspectos:

1) Crear una función que calcule cada elemento x_i en base a la fórmula presentada anteriormente.
2) Parametrizar la función de acuerdo a la variable de interés.
3) Paralelizar el proceso en función del subíndice *i* del vector x, es decir paralelizar el cálculo de x_i en función de i.

I. Creando una función
Se comienza definiendo una función, para este caso llamada Jacobi_parallel, la cual se encarga de calcular el término x_i .

```
function retval = Jacobi_parallel(A, b,x, m,i)
  sum=0;
  for j=1:m
    if(j!=i)
      sum=sum+A(i,j)*x(j);
    endif
  endfor
  retval=(1/A(i,i))*(b(i)-sum);
endfunction
```

Es decir la función antes demostrada tiene cómo retorno un único elemento del vector xk definido por el valor actual de la entrada i.

II. Parametrizando

Se implementa un function handler de variable *i*. El objetivo de este paso es conseguir que todos los parámetros del método de jacobi descrito anteriormente sean interpretados cómo entradas constantes entre los procesos paralelos. Es decir, que la variable del proceso paralelizado sea únicamente i.

```
fun_handler=@(i) Jacobi_parallel(A,b,xk, m,i);
```

III. Paralelizando

El proceso de paralelizar se realiza con la función pararrayfun de la biblioteca Parallel. La función toma cómo argumentos el número de procesadores a utilizar (núcleos), la función de tratamiento y el dato.

En esta implementación la entrada de datos corresponde a la variable i del handler, es decir se va a proveer un vector de tamaño m que funciona cómo indicador (índice) de cual elemento xi de la iteración k+1 está siendo calculado.

```
xk=pararrayfun(nproc,fun_handler,1:m).';
```

El vector que se ingresa cómo parámetro en pararrayfun [1 2 3 …. m] corresponde a la entrada i en la función de Jacobi, que por ser un proceso paralelizado va a ingresar en la Jacobi_parallel cómo cada entrada particular del vector (1, 2, 3, …, m). Es decir se obtendrá un equivalente a:

Jacobi_parallel(A,b,x,m,1)

Jacobi_parallel(A,b,x,m,2)

Jacobi_parallel(A,b,x,m,3)

.

.

.

Jacobi_parallel(A,b,x,m,m)