Instituto Tecnológico de Costa Rica
Área Académica de Ingeniería en Computadores

Análisis Numérico para Ingeniería - CE3102

Tarea 2

Pseudocódigo

Implementación en paralelo método de Jacobi

Profesor: Juan Pablo Soto Quirós

Estudiantes:

Adrián Trejos Salazar

Fabián Crawford Barquero

Irene Muñoz Castro

Luis Pedro Morales Rodríguez

Steven Badilla Soto

I Semestre – 2022

Pseudocódigo de la implementación del método de Jacobi paralelizado

Entradas al sistema:

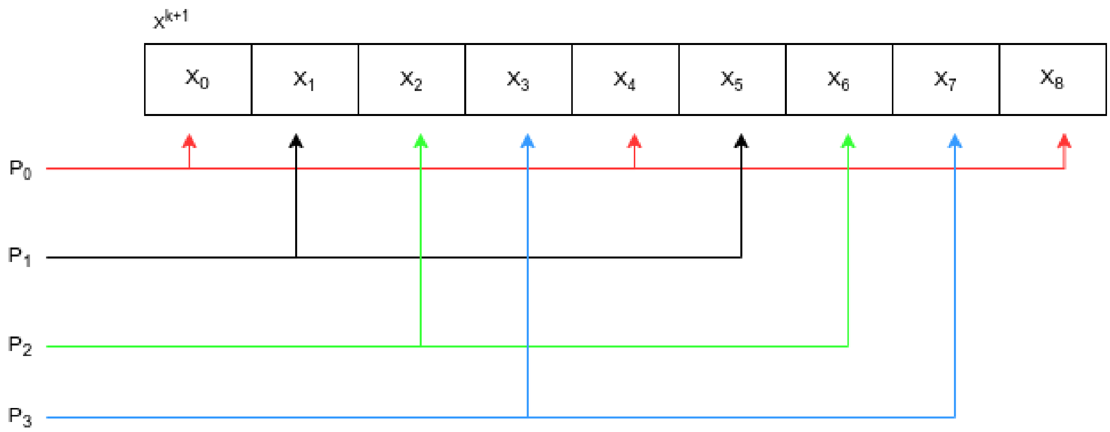
1. A = matriz de coeficientes
2. b = vector de constantes
3. x_0 = vector solución para la primera iteración
4. tol = tolerancia
5. iterMax = Máximo número de iteraciones posibles

Pasos

1. Determinar el número de núcleos lógicos disponibles: ***n***
2. Validar si la matriz es cuadrada. En caso de que no lo sea, terminar con error.
3. Validar si la matriz es diagonalmente dominante. En caso de que no lo sea, terminar con error.
4. Inicializar x_k con el valor de x_0
5. Iniciar proceso iterativo: $for\ k = 0;\ k < iterMax;\ k++$
6. Calcular x_{k+1}: se inician ***n*** procesos en paralelo $p_0,\ p_1 \ldots p_{n-1}$, en donde cada proceso p_j va a calcular $x_{k+1}[i]$ para todos los i en $for\ i = j;\ i < m;\ i += n$, donde ***n*** es el número de procesadores y ***m*** es el tamaño del vector solución.

Es decir si ***n = 4***, el primer proceso p_0, se encargará de calcular los valores $x_{k+1}[0]$, $x_{k+1}[4]$, $_{k+1}[8]$ y así sucesivamente, hasta alcanzar el máximo valor posible de ***i***, menor que ***m***. Al mismo tiempo p_1, se encargará de calcular $x_{k+1}[1]$, $x_{k+1}[5]$, $_{k+1}[9]$ ... esto con cada proceso hasta que se calculen todas las posiciones del vector solución.

Gráficamente, la distribución de tareas del proceso se podría ilustrar de la siguiente manera:



Para calcular cada posición del vector solución se usa la fórmula:

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

7. A x_k se le asigna el valor de x_{k+1}
8. Calcular el error $\| Ax_k - b \|_2$
9. Si el error es menor que ***tol*** o si ***k == iterMax-1***, el algoritmo termina y se retorna x_k ,k y $error$. En caso contrario, se prosigue con la siguiente iteración