Implementación de la Iteración de Jacobi

César González Segura
Conceptos y Métodos de la Computación Paralela

1. Especificación matemática

El algoritmo de Jacobi es un método iterativo para la estimación de la solución de un sistema lineal de ecuaciones. La iteración se repite hasta que la solución converge a un valor de convergencia ε concreto. La convergencia de la solución está asegurada sólo si la matriz de coeficientes del sistema es de diagonal estrictamente dominante.

Dado un sistema de ecuaciones lineal con n ecuaciones y n incógnitas, expresado mediante álgebra matricial como:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2 \\ \vdots \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n = b_n \end{cases} \rightarrow \quad Ax = b \equiv \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{1n} & a_{2n} & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

Donde A será la matriz de coeficientes, x el vector de incógnitas y b el vector de términos independientes. La resolución del sistema de ecuaciones se obtiene calculando el valor de x, por tanto:

$$Ax = b \equiv A^{-1}Ax = A^{-1}b \equiv \boldsymbol{x} = \boldsymbol{A^{-1}b}$$

Para construir la iteración de Jacobi, se parte de la descomposición de la matriz A en la suma de las matrices D (matriz con los términos en la diagonal principal), L (matriz triangular inferior) y U (matriz triangular superior). A partir de esta descomposición, se puede reescribir el producto anterior como:

$$A = D + L + U \quad \rightarrow \quad Ax = b \equiv \boldsymbol{Dx} + (\boldsymbol{L} + \boldsymbol{U})\boldsymbol{x} = \boldsymbol{b}$$

Se reordena la ecuación y se asignan índices de iteración a cada lado de la ecuación. De esta forma se define la estimación de la solución en el instante siguiente en función de la solución en el instante actual:

$$x = D^{-1}[b - (L + U)x] \quad \rightarrow \quad \boldsymbol{x^{(k+1)}} = \boldsymbol{D^{-1}}\left[\boldsymbol{b} - (\boldsymbol{L} + \boldsymbol{U})\boldsymbol{x^{(k)}}\right]$$

2. Especificación del algoritmo secuencial

Para la especificación inicial de la iteración de Jacobi de forma algorítmica, reescribimos la ecuación tal que:

$$x^{(k+1)} = Tx^{(k)} + C$$

Donde C y T serán matrices formadas a partir de L, U y de la inversa de D de la forma:

$$C = D^{-1}b \qquad T = -D^{-1}(L + U)$$

El cálculo de estas dos ecuaciones implica calcular la inversa de la matriz diagonal. Mientras ninguno de los elementos de la diagonal sea cero, la inversa de la matriz diagonal se puede calcular como:

$$D^{-1} = \begin{bmatrix} d_1 & 0 & \cdots & 0 \\ 0 & d_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & d_n \end{bmatrix}^{-1} = \begin{bmatrix} {}^{1}/_{d_1} & 0 & \cdots & 0 \\ 0 & {}^{1}/_{d_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & {}^{1}/_{d_n} \end{bmatrix}$$

Para obtener la solución al sistema de ecuaciones, se inicializa x con una aproximación inicial ($x^{(0)}$). A partir de este valor se obtiene la solución en la iteración posterior ($x^{(1)}$) y así sucesivamente hasta que el valor de convergencia sea menor que el valor ε. El valor de convergencia se obtiene como:

$$Convergencia = \left\| Ax^{(k+1)} - b \right\| < \varepsilon$$

Se asume que existen las funciones *matvec* para el producto de una matriz con un vector, *matprod* para el producto de dos matrices, *norm2* para obtener la dos-norma de un vector, *matsum* para la suma de dos matrices, *vecsum* para la suma de dos vectores e *isdom* para verificar que una matriz es de diagonal dominante.

Considerando lo anterior, el algoritmo para la iteración de Jacobi puede expresarse como:

Función *Jacobi*
Entradas: Doble A[n,n], Doble b[n], Doble $x^{(0)}$[n], Doble ε, Doble n
Salidas: Doble xs[n]

```
Si no isdom(A) hacer
    Salir
Fin Si

Para i = 0 hasta n – 1 hacer
    Para j = 0 hasta n – 1 hacer
        Si i = j hacer
            D_inv (i, j) ← 1 / A(i, j)
        Si no, hacer
            Si i > j hacer
                L (i, j) ← A(i, j)
            Si no, hacer
                U (i, j) ← A(i, j)
            Fin Si
        Fin Si
    Fin Para
Fin Para

T ← matvec(-D_inv , matsum(L , U))
C ← matvec(D_inv , b)

conv ← ε + 1
x^(actual) ← x^(0)

Mientras conv > ε Hacer
    x^(siguiente) ← vecsum(matvec(T , x^(actual)), C)
    x^(actual) ← x^(siguiente)
    conv ← norm2(vecsum(matvec(A , x), -b)
Fin Mientras

xs ← x^(siguiente)
```

3. Prototipo del algoritmo secuencial

Para comprobar la validez del algoritmo secuencial, se ha implementado un prototipo del mismo utilizando el entorno *MATLAB*.

El prototipo genera un sistema de ecuaciones de manera aleatoria, forzando la matriz de coeficientes del sistema a que sea estrictamente diagonal dominante, y obtiene el resultado del algoritmo para diferentes valores del parámetro de convergencia ε.

Para cada valor del parámetro de convergencia se obtiene el número de iteraciones necesario hasta converger, el tiempo requerido para alcanzar la convergencia en segundos y el error cuadrático medio comparando el resultado del algoritmo con el resultado obtenido al resolver el sistema utilizando el método integrado en *MATLAB*.