



1. Metodos Iterativos

Los procedimientos iterativos para resolver un sistema de ecuaciones lineales, son utilizados en algunos casos donde los métodos directos (como Gauss-Jordan) no son los adecuados. El primero de ellos conocido como el procedimiento de Jacobi basado en la idea de punto fijo y un segundo procedimiento conocido como método de Gauss-Seidel el cual es una modificación simple del procedimiento de Jacobi.

Por ejemplo, si una matriz es muy grande y rala (los elementos no nulos son una fracción pequeña de la totalidad de elementos de la matriz) entonces el método de eliminación gaussiana no es el mejor método para resolver el sistema porque, en general, tendríamos problemas de lentitud en el manejo de la memoria por tener que almacenar y manipular toda la matriz; además la descomposición de una matriz como la LU deja de ser tan rala como la matriz A . Luego, los métodos iterativos se usan para resolver problemas con matrices muy grandes, ralas y con ciertos patrones ("bandadas"), usualmente problemas que aparecen al discretizar ecuaciones diferenciales en derivadas parciales.

Para resolver numéricamente un sistema $Ax = B$ con A no-singular, los métodos iterativos inician con una aproximación x_0 de la solución x^* para generar una sucesión $\{x_k\}$ que se aproxime a la solución del sistema. En general tales iteraciones se pueden escribir como:

$$x_{k+1} = g_k(x_k)$$

Donde g_k es una sucesión de funciones, (Si $g_k = g$ el método se dice "estacionario").

La solución para x_i esta dada por:

$$x_i = \frac{1}{a_{ii}}(b_i - \sum_{j=1}^{i-1} a_{ij}x_j - \sum_{j=i+1}^n a_{ij}x_j)$$

Si tenemos una aproximación a la solución \mathbf{x}_k entonces,

$$\mathbf{x}_{k+1}(i) = \frac{1}{a_{ii}}(b_i - \sum_{j=1}^{i-1} a_{ij}\mathbf{x}_k(j) - \sum_{j=i+1}^n a_{ij}\mathbf{x}_k(j))$$

1.1. Método de Jacobi

Ahora observe que $A = L + D + U$ donde L y U son matrices triangular inferior y triangular superior de A respectivamente, y D es la diagonal de A . Si además D es no singular se tiene que la iteración de **Jacobi**:

$$\begin{aligned}\mathbf{x}_{k+1} &= \mathbf{x}_k + D^{-1}(B - A\mathbf{x}_k) \\ \mathbf{x}_{k+1} &= D^{-1}(B - (L + U)\mathbf{x}_k)\end{aligned}$$

Como se puede ver, este método iterativo es solo aplicable para sistemas cuadrados y como todo sistema debe tener un criterio de parada:

$$\text{Si } D_k < E \text{ con} \\ D_k = \max(|x_k - x_{k+1}|, |y_k - y_{k+1}|)$$

Graficamente, el método Jacobi es desplazamientos de segmentos de recta como se ve en la figura

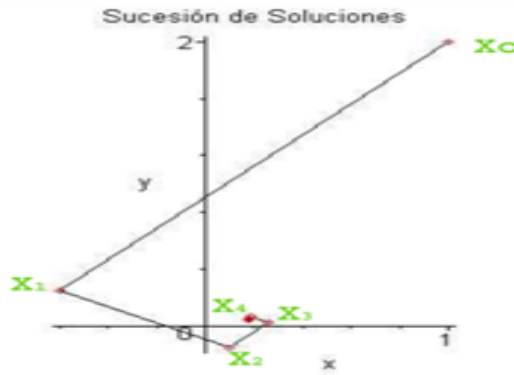


Figura 1: *Sucesión de Soluciones*

Ejemplo

Partiendo de $(x = 1, y = 2)$ aplique dos iteraciones del método de Jacobi para resolver el sistema:

$$\begin{pmatrix} 5x & +2y & = & 1 \\ x & -4y & = & 0 \end{pmatrix}$$

Solución Debemos primeramente despejar de la ecuación la incógnita correspondiente.

$$\begin{aligned} x &= 0.20 + 0.00x - 0.40y \\ y &= 0.00 + 0.25x + 0.00y \end{aligned} \tag{1}$$

Aplicamos la primera iteración partiendo de \mathbf{X}_0

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 0.20 \\ 0.00 \end{pmatrix} + \begin{pmatrix} 0.00 & -0.40 \\ 0.25 & 0.00 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

Utilizando el valor inicial X_0 generamos la primera iteración

$$\begin{pmatrix} x_1 \\ y_1 \end{pmatrix} = \begin{pmatrix} 0.20 \\ 0.00 \end{pmatrix} + \begin{pmatrix} 0.00 & -0.40 \\ 0.25 & 0.00 \end{pmatrix} \begin{pmatrix} x_0 \\ y_0 \end{pmatrix}$$

Luego obtenemos el resultado de la primera iteración $\mathbf{X}_{k+1} = C + T\mathbf{X}_k$

i	x_i	y_i	x_{i+1}	y_{i+1}	D_i
0	1.000	2.000	-0.600	0.250	1.750
1	-0.600	0.250	0.100	-0.150	0.700
2	0.100	-0.150	0.260	0.025	0.175
3	0.260	0.025	0.190	0.065	0.070
4	0.190	0.065	0.174	0.047	0.017
5	0.174	0.047	0.181	0.043	0.007
6	0.181	0.043	0.182	0.045	0.001

Figura 2: *Sucesión de Soluciones*

$$\begin{pmatrix} x_1 \\ y_1 \end{pmatrix} = \begin{pmatrix} -0.60 \\ 0.25 \end{pmatrix}$$

Como se puede ver, se necesita de 7 iteraciones para encontrar una aproximación a la solución del sistema; teniendo en cuenta el criterio de parada se generan las i ésimas iteraciones

Error de Truncamiento: teniendo en cuenta la metodología equivalente del punto fijo, $x = g(x)$ se tiene que la forma recursiva $\mathbf{X}_{k+1} = C + T\mathbf{X}_k$, el error de truncamiento esta dado $E_{k+1} = TE_k$ donde T se conoce como la matriz de transición.

Definición: Sea A una matriz su radio espectral $\rho(A)$ se define como $\rho(A) = \max_{1 \leq i \leq n} \{|\lambda_i|\}$ con λ_i un valor característico de A

1.1.1. Convergencia en Jacobi

Uno de los principales problemas de los métodos iterativos es la garantía de que el método va a converger, es decir, va a producir una sucesión de aproximaciones cada vez efectivamente más próximas a la solución. En el caso del método de Jacobi no existe una condición exacta para la convergencia. Lo mejor es una condición que garantiza la convergencia, pero en caso de no cumplirse puede o no haberla

Criterio: Si la matriz de coeficientes original del sistema de ecuaciones es diagonalmente dominante, el método de Jacobi seguro converge.

Una matriz se dice **matriz diagonalmente dominante**, si en cada uno de los renglones, el valor absoluto del elemento de la diagonal principal es mayor que la suma de los valores abslutos de los elementos restantes del mismo renglón. A veces la matriz de un sistema de ecuaciones no es diagonalmente dominante pero cuando se cambian(utilizando operaciones entre renglones) el orden de las ecuaciones y las incógnitas el nuevo sistema puede tener matriz de coeficientes diagonalmente dominante.

Ejemplo:

$$\begin{pmatrix} 4 & 1 & 1 \\ 2 & 8 & -3 \\ 3 & 2 & 9 \end{pmatrix}$$

Es una matriz dominante porque: $|4| > |1| + |1|$; $|8| > |2| + |-3|$; $|9| > |3| + |2|$
 En el caso que la matriz A no sea diagonalmente dominante entonces, hay revisar todos los posibles ordenamientos de las incógnitas y ver cómo es la matriz resultante. El número de ordenamientos de n variables es $(n-1)!$

Teorema de Convergencia

Sea x_k una sucesión de soluciones por el método iterativo con ecuación $\mathbf{X}_{k+1} = C + T\mathbf{X}_k$ converge con cualquier vector inicial $\mathbf{x}_0 \in \mathbf{R}^n$ a la solución \mathbf{x} si $\rho(A) < 1$

Teniendo en cuenta, que $A = L + D + U$ en el método de Jacobi, la matriz de transición esta definida como: $T = -D^{-1}(L + U)$

Para el caso, del método Gauss-Seidel la matriz de transición esta dada por: $T = (-D^{-1}U)(I + LD^{-1})^{-1}$

Por otro lado como $E_{k+1} = TE_k$ si aplicamos la norma del vector tenemos que $\|E_{k+1}\| \leq \|T\|\|E_k\|$ entonces, la condición suficiente para que el método de Jacobi converga es que $\|T\|_\infty < 1$

Eficiencia de los Métodos Iterativos.

Sea k el número de iteraciones necesarias para alcanzar la precisión deseada, entonces $T(n) = kO(n^2)$

1.2. Método SOR

En el álgebra lineal numérica, el método de sobre-relajación sucesiva (SOR) es una variante del método de Gauss-Seidel para resolver un sistema lineal de ecuaciones, lo que resulta en una convergencia más rápida. Se puede usar un método similar para cualquier proceso iterativo que converja lentamente.

El método SOR (Successive Over Relaxation method), fue ideado simultáneamente por David M. Young, Jr. y por Stanley P. Frankel en 1950 con el propósito de resolver automáticamente sistemas lineales en computadoras digitales. Los métodos de sobre relajación se habían utilizado antes del trabajo de Young y Frankel. Un ejemplo es el método de Lewis Fry Richardson y los métodos desarrollados por R. V. Southwell. Sin embargo, estos métodos fueron diseñados para computación por calculadoras humanas, y requerían cierta experiencia para asegurar la convergencia a la solución que los hacía inaplicables para la programación en computadoras digitales.

Para este caso, el método SOR se obtiene introduciendo un parámetro de aceleración $w \in]1, 2[$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + (L + w^{-1}D^{-1})(B - A\mathbf{x}_k)$$

El método de Gauss-Seidel se obtiene con $w = 1$ en el método **SOR**

$$\mathbf{x}_{k+1} = \mathbf{x}_k + (L + D^{-1})(B - A\mathbf{x}_k)$$

Referencia [http : //rpubs.com/eddyherrera/sistemas_ecuaciones_numericas](http://rpubs.com/eddyherrera/sistemas_ecuaciones_numericas)