Métodos iterativos de solución de SEL

Método de Gauss-Seidel

MAT-251

Dr. Alonso Ramírez Manzanares
CIMAT A.C.
e-mail: alram@cimat.mx
web: http://www.cimat.mx/~alram/met_num/

Dr. Salvador Botello Rionda
CIMAT A.C.

Descomposición de Jacobi

- En el método de Jacobi, descomponíamos la matriz como

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}$$

$$A = \begin{bmatrix} a_{11} & 0 & & 0 \\ 0 & a_{22} & & \\ & & & 0 \\ 0 & & 0 & a_{nn} \end{bmatrix} - \begin{bmatrix} 0 & & 0 \\ -a_{21} & & \\ & & \\ -a_{n1} & -a_{n,n-1} & 0 \end{bmatrix}$$

$$- \begin{bmatrix} 0 & -a_{12} & -a_{1n} \\ & & \\ & & -a_{n-1,n} \\ 0 & & 0 \end{bmatrix}$$

$$= D - L - U.$$

Método Iterativo de Jacobi

- Ejemplo para el SEL con solución exacta $\mathbf{x} = (1, 2, -1, 1)^t.$

$$
\begin{array}{lrcrcrcrcl}
E_1: & 10x_1 & - & x_2 & + & 2x_3 & & & = & 6, \\
E_2: & -x_1 & + & 11x_2 & - & x_3 & + & 3x_4 & = & 25, \\
E_3: & 2x_1 & - & x_2 & + & 10x_3 & - & x_4 & = & -11, \\
E_4: & & & 3x_2 & - & x_3 & + & 8x_4 & = & 15
\end{array}
$$

- Para tener un sistema

$$
\begin{array}{lclclclclcl}
x_1 & = & & & \frac{1}{10}x_2 & - & \frac{1}{5}x_3 & & & + & \frac{3}{5}, \\
x_2 & = & \frac{1}{11}x_1 & & & + & \frac{1}{11}x_3 & - & \frac{3}{11}x_4 & + & \frac{25}{11}, \\
x_3 & = & -\frac{1}{5}x_1 & + & \frac{1}{10}x_2 & & & + & \frac{1}{10}x_4 & - & \frac{11}{10}, \\
x_4 & = & & & -\frac{3}{8}x_2 & + & \frac{1}{8}x_3 & & & + & \frac{15}{8}.
\end{array}
$$

Fórmula iterativa de Gauss-Seidel

- Teniendo la siguiente fórmula para el calculo iterativo de cada x_i :

$$x_i^{(k)} = \frac{\sum_{\substack{j=1 \\ j\neq i}}^{n} \left(-a_{ij} x_j^{(k-1)}\right) + b_i}{a_{ii}}, \qquad \text{for } i = 1, 2, \ldots, n.$$

- Pero, nótese que cuando calculamos una incógnita ***i***, de hecho ya calculamos antes todas las $x_1,\ldots,x_{i-1}$, por lo que podemos acelerar la convergencia usando:

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

Ejemplo del método de Gauss-Seidel

- Veamos un ejemplo para solución de este SEL con solución exacta $\mathbf{x} = (1, 2, -1, 1)^t$.

$$\begin{array}{rcrcrcrcr}
10x_1 & - & x_2 & + & 2x_3 & & & = & 6, \\
-x_1 & + & 11x_2 & - & x_3 & + & 3x_4 & = & 25, \\
2x_1 & - & x_2 & + & 10x_3 & - & x_4 & = & -11, \\
& & 3x_2 & - & x_3 & + & 8x_4 & = & 15.
\end{array}$$

- el cuál dá origen a las siguientes fórmulas iterativas:

$$\begin{array}{rclclclcl}
x_1^{(k)} & = & & & \frac{1}{10}x_2^{(k-1)} & - & \frac{1}{5}x_3^{(k-1)} & & & + & \frac{3}{5}, \\
x_2^{(k)} & = & \frac{1}{11}x_1^{(k)} & & & + & \frac{1}{11}x_3^{(k-1)} & - & \frac{3}{11}x_4^{(k-1)} & + & \frac{25}{11}, \\
x_3^{(k)} & = & -\frac{1}{5}x_1^{(k)} & + & \frac{1}{10}x_2^{(k)} & & & + & \frac{1}{10}x_4^{(k-1)} & - & \frac{11}{10}, \\
x_4^{(k)} & = & & & -\frac{3}{8}x_2^{(k)} & + & \frac{1}{8}x_3^{(k)} & & & + & \frac{15}{8}.
\end{array}$$

Ejemplo del método de Gauss-Seidel

- Si utilizamos x(0) = (0,0,0,0)T e iteramos obtenemos

k	0	1	2	3	4	5
$x_1^{(k)}$	0.0000	0.6000	1.030	1.0065	1.0009	1.0001
$x_2^{(k)}$	0.0000	2.3272	2.037	2.0036	2.0003	2.0000
$x_3^{(k)}$	0.0000	-0.9873	-1.014	-1.0025	-1.0003	-1.0000
$x_4^{(k)}$	0.0000	0.8789	0.9844	0.9983	0.9999	1.0000

- para la condición de paro:

$$\|\mathbf{x}^{(5)} - \mathbf{x}^{(4)}\|_\infty = 0.0008 < 10^{-3},$$

Gauss-Seidel vs. Jacobi (ambos con la misma condición de paro)

- Jacobi

k	0	1	2	3	4	5	6	7	8	9	10
$x_1^{(k)}$	0.000	0.6000	1.0473	0.9326	1.0152	0.9890	1.0032	0.9981	1.0006	0.9997	1.0001
$x_2^{(k)}$	0.0000	2.2727	1.7159	2.053	1.9537	2.0114	1.9922	2.0023	1.9987	2.0004	1.9998
$x_3^{(k)}$	0.0000	−1.1000	−0.8052	−1.0493	−0.9681	−1.0103	−0.9945	−1.0020	−0.9990	−1.0004	−0.9998
$x_4^{(k)}$	0.0000	1.8750	0.8852	1.1309	0.9739	1.0214	0.9944	1.0036	0.9989	1.0006	0.9998

- Gauss-Seidel

k	0	1	2	3	4	5
$x_1^{(k)}$	0.0000	0.6000	1.030	1.0065	1.0009	1.0001
$x_2^{(k)}$	0.0000	2.3272	2.037	2.0036	2.0003	2.0000
$x_3^{(k)}$	0.0000	−0.9873	−1.014	−1.0025	−1.0003	−1.0000
$x_4^{(k)}$	0.0000	0.8789	0.9844	0.9983	0.9999	1.0000

Algoritmo (1 de 2)

To solve $A\mathbf{x} = \mathbf{b}$ given an initial approximation $\mathbf{x}^{(0)}$:

INPUT the number of equations and unknowns n; the entries a_{ij}, $1 \leq i, j \leq n$ of the matrix A; the entries b_i, $1 \leq i \leq n$ of $\mathbf{b}$; the entries XO_i, $1 \leq i \leq n$ of $\mathbf{XO} = \mathbf{x}^{(0)}$; tolerance TOL; maximum number of iterations N.

OUTPUT the approximate solution $x_1, \ldots, x_n$ or a message that the number of iterations was exceeded.

Algoritmo (2 de 2)

Step 1 Set $k = 1$.

Step 2 While $(k \leq N)$ do Steps 3–6.

Step 3 For $i = 1, \ldots, n$

$$\text{set } x_i = \frac{-\sum_{j=1}^{i-1} a_{ij} x_j - \sum_{j=i+1}^{n} a_{ij} XO_j + b_i}{a_{ii}}.$$

Step 4 If $||\mathbf{x} - \mathbf{XO}|| < TOL$ then OUTPUT $(x_1, \ldots, x_n)$;
(*The procedure was successful.*)
STOP.

Step 5 Set $k = k + 1$.

Step 6 For $i = 1, \ldots, n$ set $XO_i = x_i$.

Step 7 OUTPUT ('Maximum number of iterations exceeded');
(*The procedure was successful.*)
STOP.

¿Es necesario guardar la matriz?

Forma matricial del método de Gauss-Seidel

- Tomamos la ecuación iterativa

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

- y multiplicamos todo por a_{ii} y agrupamos los términos asociados a la misma iteración:

$$a_{i1}x_1^{(k)} + a_{i2}x_2^{(k)} + \cdots + a_{ii}x_i^{(k)} = -a_{i,i+1}x_{i+1}^{(k-1)} - \cdots - a_{in}x_n^{(k-1)} + b_i,$$

- Quedando el conjunto de todas las ecuaciones i=1,...,n como

$$\begin{aligned}
a_{11}x_1^{(k)} &= -a_{12}x_2^{(k-1)} - a_{13}x_3^{(k-1)} - \cdots - a_{1n}x_n^{(k-1)} + b_1,\\
a_{21}x_1^{(k)} + a_{22}x_2^{(k)} &= -a_{23}x_3^{(k-1)} - \cdots - a_{2n}x_n^{(k-1)} + b_2,\\
&\vdots\\
a_{n1}x_1^{(k)} + a_{n2}x_2^{(k)} + \cdots + a_{nn}x_n^{(k)} &= b_n.
\end{aligned}$$

Forma matricial del método de Gauss-Seidel

- Este sistema de ecuaciones partiendo de que la matriz fué descompuesta como ***A = D - L - U***

$$
\begin{aligned}
a_{11}x_1^{(k)} \quad &= \quad -a_{12}x_2^{(k-1)} - a_{13}x_3^{(k-1)} - \cdots - a_{1n}x_n^{(k-1)} + b_1, \\
a_{21}x_1^{(k)} + a_{22}x_2^{(k)} \quad &= \quad -a_{23}x_3^{(k-1)} - \cdots - a_{2n}x_n^{(k-1)} + b_2, \\
\vdots \quad & \\
a_{n1}x_1^{(k)} + a_{n2}x_2^{(k)} + \cdots + a_{nn}x_n^{(k)} \quad &= \quad b_n.
\end{aligned}
$$

- genera la fórmula matricial iterativa

$$(D - L)\mathbf{x}^{(k)} = U\mathbf{x}^{(k-1)} + \mathbf{b}$$

Forma matricial del método de Gauss-Seidel

- Dada la fórmula iterativa

$$(D - L)\mathbf{x}^{(k)} = U\mathbf{x}^{(k-1)} + \mathbf{b}$$

- si $(D - L)^{-1}$ existe, entonces usamos $T_g = (D - L)^{-1}U$ y $c_g = (D - L)^{-1}\mathbf{b}$

$$\mathbf{x}^{(k)} = T_g\mathbf{x}^{(k-1)} + \mathbf{c}_g,$$

¿Cuándo funciona esto?

- Dado que usamos $T_g = (D - L)^{-1}U$, debemos de invertir ***D-L***, y su determinante es $a_{11}*a_{22}*...*a_{nn}$.

- Por lo tanto necesitamos que a_{ii} sea diferente de cero.

- En el ejemplo mostrado, se sugiere que el metódo de Gauss-Seidel es superior al de Jacobi, esto sucede la mayoría de las veces, pero hay casos donde el método de Jacobi converge y el método de Gauss-Seidel no. *Nótese que en el método de Jacobi su matriz asociada era* $\boldsymbol{T_j} = \boldsymbol{D}^{-1}(\boldsymbol{L}+\boldsymbol{U})$.

- Podemos decir que si la matriz **A** es estrictamente diagonal dominante, entonces para cualquier vector **b** y cualquier estimación inicial $\boldsymbol{x}^{(0)}$ los 2 métodos convergen a la solución única del sistema $\boldsymbol{Ax}=\boldsymbol{b}$.

Vector de residuo

- Sea **x˜** una aproximación de la solución real ***x*** del sistema, entonces hay un vector residual **r** = **b**- ***A*x**˜.
- Lo que se hace en los métodos de GS y Jacobi es generar una sucesión de aproximaciones tal que **r** converja rápidamente a cero.
- Definimos $\mathbf{r}_i^{(k)} = (r_{1i}^{(k)}, r_{2i}^{(k)}, \ldots, r_{ni}^{(k)})^t$ como el vector residual del método de GS correspondiente al vector solución $\boldsymbol{x}_i^{(k)}$ aproximado, que esta definido por:

$$\mathbf{x}_i^{(k)} = (x_1^{(k)}, x_2^{(k)}, \ldots, x_{i-1}^{(k)}, x_i^{(k-1)}, \ldots, x_n^{(k-1)})^t.$$

- La **m**-ésima componente de $\boldsymbol{r}_i^{(k)}$ es: $$r_{mi}^{(k)} = b_m - \sum_{j=1}^{i-1} a_{mj} x_j^{(k)} - \sum_{j=i}^{n} a_{mj} x_j^{(k-1)},$$

$$r_{mi}^{(k)} = b_m - \sum_{j=1}^{i-1} a_{mj} x_j^{(k)} - \sum_{j=i+1}^{n} a_{mj} x_j^{(k-1)} - a_{mi} x_i^{(k-1)},$$

Vector de residuo

- De $$r_{mi}^{(k)} = b_m - \sum_{j=1}^{i-1} a_{mj} x_j^{(k)} - \sum_{j=i+1}^{n} a_{mj} x_j^{(k-1)} - a_{mi} x_i^{(k-1)},$$

- En particular la i-ésima componente de $r_i^{(k)}$ es

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

- y por lo tanto

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

- y el paso de GS es:

$$x_i^{(k)} = \frac{1}{a_{ii}} \left[b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(k-1)} \right],$$

Vector de residuo

- por lo tanto

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

- se puede escribir como $a_{ii}x_i^{(k-1)} + r_{ii}^{(k)} = a_{ii}x_i^{(k)}.$

- o sea

$$x_i^{(k)} = x_i^{(k-1)} + \frac{r_{ii}^{(k)}}{a_{ii}}.$$

- Nótese entonces, que el paso de GS escoge $\boldsymbol{x}_i^{(k)}$ en función del residuo.

Deducción de relajación

Ahora corre hasta ***i*** y no ***i***-1

- Analicemos ahora para la entrada del vector residual en el siguiente cálculo

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

- asociado al vector $\mathbf{x}_{i+1}^{(k)} = (x_1^{(k)}, \ldots, x_i^{(k)}, x_{i+1}^{(k-1)}, \ldots, x_n^{(k-1)})^t.$

- pero en GS $x_i^{(k)} = \frac{1}{a_{ii}} \left[b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(k-1)} \right],$

- por lo tanto el paso GS hace $r_{i,i+1}^{(k)} = 0.$ lo cual puede no ser muy correcto (agresivo,ya que no intenta hacer todo el vector ***r*** cero sino solo una entrada).

Deducción de relajación

- Por lo tanto no usamos el paso tal cual,

$$x_i^{(k)} = x_i^{(k-1)} + \frac{r_{ii}^{(k)}}{a_{ii}} .$$

- sino que lo relajamos

$$x_i^{(k)} = x_i^{(k-1)} + \omega \frac{r_{ii}^{(k)}}{a_{ii}},$$

El metodo SOR (Succesive Over Relaxation)

- En este método de relajación usamos un peso ω

$$x_i^{(k)} = (1-\omega)x_i^{(k-1)} + \frac{\omega}{a_{ii}}\left[b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k-1)}\right]$$

- con $\omega = 1$ tenemos el método de Gauss-Seidel, cuando $0 < \omega < 1$ tenemos sub-relajación (algunas veces es mejor para sistemas que no convergen con GS) y si $\omega > 1$ tenemos sobre-relajación (la cúal se usa para acelerar la convergencia de sistemas que si convergen con GS). Hay teoría que sugiere $1 < \omega < 2$ en los casos de sobre relajación.

- TAREA, ESCRIBIR EL SISTEMA MATRICIAL DEL MÉTODO SOR.

Pequeño ejemplo:

- El SEL

$$
\begin{array}{rcrcrcr}
4x_1 & + & 3x_2 & & & = & 24, \\
3x_1 & + & 4x_2 & - & x_3 & = & 30, \\
 & & -x_2 & + & 4x_3 & = & -24
\end{array}
$$

con solución exacta $(3, 4, -5)^t$

Gauss-Seidel

k	0	1	2	3	4	5	6	7
$x_1^{(k)}$	1	5.250000	3.1406250	3.0878906	3.0549316	3.0343323	3.0214577	3.0134110
$x_1^{(k)}$	1	3.812500	3.8828125	3.9267578	3.9542236	3.9713898	3.9821186	3.9888241
$x_1^{(k)}$	1	-5.046875	-5.0292969	-5.0183105	-5.0114441	-5.0071526	-5.0044703	-5.0027940

SOR with $\omega = 1.25$

k	0	1	2	3	4	5	6	7
$x_1^{(k)}$	1	6.312500	2.6223145	3.1333027	2.9570512	3.0037211	2.9963276	3.0000498
$x_2^{(k)}$	1	3.5195313	3.9585266	4.0102646	4.0074838	4.0029250	4.0009262	4.0002586
$x_3^{(k)}$	1	-6.6501465	-4.6004238	-5.0966863	-4.9734897	-5.0057135	-4.9982822	-5.0003486