0.1 Método de Gauss-Seidel

Los métodos de Gauss y Cholesky hacen parte de los métodos directos o finitos. Al cabo de un número finito de operaciones, en ausencia de errores de redondeo, se obtiene x^* solución del sistema $Ax = b$.

El método de Gauss-Seidel hace parte de los métodos llamados indirectos o iterativos. En ellos se comienza con $x^0 = (x_1^0, x_2^0, ..., x_n^0)$, una aproximación inicial de la solución. A partir de x^0 se construye una nueva aproximación de la solución, $x^1 = (x_1^1, x_2^1, ..., x_n^1)$. A partir de x^1 se construye x^2 (aquí el superíndice indica la iteración y no indica una potencia). Así sucesivamente se construye una sucesión de vectores $\{x^k\}$, con el objetivo, no siempre garantizado, de que

$$\lim_{k\to\infty} x^k = x^*.$$

Generalmente los métodos indirectos son una buena opción cuando la matriz es muy grande y dispersa o rala (*sparse*), es decir, cuando el número de elementos no nulos es pequeño comparado con n^2, número total de elementos de A. En estos casos se debe utilizar una estructura de datos adecuada que permita almacenar únicamente los elementos no nulos.

En cada iteración del método de Gauss-Seidel, hay n subiteraciones. En la primera subiteración se modifica únicamente x_1. Las demás coordenadas x_2, x_3, ..., x_n no se modifican. El cálculo de x_1 se hace de tal manera que se satisfaga la primera ecuación.

$$\begin{aligned} x_1^1 &= \frac{b_1 - (a_{12}x_2^0 + a_{13}x_3^0 + \cdots + a_{1n}x_n^0)}{a_{11}}, \\ x_i^1 &= x_i^0\,, \quad i = 2, ..., n. \end{aligned}$$

En la segunda subiteración se modifica únicamente x_2. Las demás coordenadas x_1, x_3, ..., x_n no se modifican. El cálculo de x_2 se hace de tal manera que se satisfaga la segunda ecuación.

$$\begin{aligned} x_2^2 &= \frac{b_2 - (a_{21}x_1^1 + a_{23}x_3^1 + \cdots + a_{2n}x_n^1)}{a_{22}}, \\ x_i^2 &= x_i^1\,, \quad i = 1, 3, ..., n. \end{aligned}$$

Así sucesivamente, en la n-ésima subiteración se modifica únicamente x_n. Las demás coordenadas x_1, x_2, ..., x_{n-1} no se modifican. El cálculo de x_n se

hace de tal manera que se satisfaga la n-ésima ecuación.

$$\begin{aligned} x_n^n &= \frac{b_n - (a_{n1}x_1^{n-1} + a_{n3}x_3^{n-1} + \cdots + a_{nn}x_n^{n-1})}{a_{nn}}, \\ x_i^n &= x_i^{n-1}, \quad i = 1, 2, ..., n-1. \end{aligned}$$

Ejemplo 0.1. Resolver

$$\begin{bmatrix} 10 & 2 & -1 & 0 \\ 1 & 20 & -2 & 3 \\ -2 & 1 & 30 & 0 \\ 1 & 2 & 3 & 20 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 26 \\ -15 \\ 53 \\ 47 \end{bmatrix}$$

partiendo de $x^0 = (1,\ 2,\ 3,\ 4)$.

$$\begin{aligned} x_1^1 &= \frac{26 - (2 \times 2 + (-1) \times 3 + 0 \times 4)}{10} = 2.5, \\ x^1 &= (2.5,\ 2,\ 3,\ 4). \\ x_2^2 &= \frac{-15 - (1 \times 2.5 + (-2) \times 3 + 3 \times 4)}{20} = -1.175, \\ x^2 &= (2.5,\ -1.175,\ 3,\ 4). \\ x_3^3 &= \frac{53 - (-2 \times 2.5 + 1 \times (-1.175) + 0 \times 4)}{30} = 1.9725, \\ x^3 &= (2.5,\ -1.175,\ 1.9725,\ 4). \\ x_4^4 &= \frac{47 - (1 \times 2.5 + 2 \times (-1.175) + 3 \times 1.9725)}{20} = 2.0466, \\ x^4 &= (2.5,\ -1.175,\ 1.9725,\ 2.0466). \end{aligned}$$

Una vez que se ha hecho una iteración completa (n subiteraciones), se utiliza el último x obtenido como aproximación inicial y se vuelve a empezar; se calcula x_1 de tal manera que se satisfaga la primera ecuación, luego se calcula x_2... A continuación están las iteraciones siguientes para el ejemplo anterior.

3.0323	−1.1750	1.9725	2.0466
3.0323	−1.0114	1.9725	2.0466
3.0323	−1.0114	2.0025	2.0466
3.0323	−1.0114	2.0025	1.9991

$$
\begin{array}{cccc}
3.0025 & -1.0114 & 2.0025 & 1.9991 \\
3.0025 & -0.9997 & 2.0025 & 1.9991 \\
3.0025 & -0.9997 & 2.0002 & 1.9991 \\
3.0025 & -0.9997 & 2.0002 & 1.9998
\end{array}
$$

$$
\begin{array}{cccc}
3.0000 & -0.9997 & 2.0002 & 1.9998 \\
3.0000 & -1.0000 & 2.0002 & 1.9998 \\
3.0000 & -1.0000 & 2.0000 & 1.9998 \\
3.0000 & -1.0000 & 2.0000 & 2.0000
\end{array}
$$

$$
\begin{array}{cccc}
3.0000 & -1.0000 & 2.0000 & 2.0000 \\
3.0000 & -1.0000 & 2.0000 & 2.0000 \\
3.0000 & -1.0000 & 2.0000 & 2.0000 \\
3.0000 & -1.0000 & 2.0000 & 2.0000
\end{array}
$$

Teóricamente, el método de Gauss-Seidel puede ser un proceso infinito. En la práctica el proceso se acaba cuando de x^k a x^{k+n} los cambios son muy pequeños. Esto quiere decir que el x actual es casi la solución x^*.

Como el método no siempre converge, entonces otra detención del proceso, no deseada pero posible, está determinada cuando el número de iteraciones realizadas es igual a un número máximo de iteraciones previsto.

El siguiente ejemplo no es convergente, ni siquiera empezando de una aproximación inicial muy cercana a la solución. La solución exacta es $x = (1, 1, 1)$.

Ejemplo 0.2. Resolver

$$
\begin{bmatrix} -1 & 2 & 10 \\ 11 & -1 & 2 \\ 1 & 5 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 11 \\ 12 \\ 8 \end{bmatrix}
$$

partiendo de $x^0 = (1.0001,\ 1.0001,\ 1.0001)$.

$$
\begin{array}{ccc}
1.0012 & 1.0001 & 1.0001 \\
1.0012 & 1.0134 & 1.0001 \\
1.0012 & 1.0134 & 0.9660
\end{array}
$$

$$
\begin{array}{ccc}
0.6863 & 1.0134 & 0.9660 \\
0.6863 & -2.5189 & 0.9660 \\
0.6863 & -2.5189 & 9.9541
\end{array}
$$

83.5031	−2.5189	9.9541
83.5031	926.4428	9.9541
83.5031	926.4428	−2353.8586

Algunos criterios garantizan la convergencia del método de Gauss-Seidel. Por ser condiciones suficientes para la convergencia son criterios demasiado fuertes, es decir, la matriz A puede no cumplir estos requisitos y sin embargo el método puede ser convergente. En la práctica, con frecuencia, es muy dispendioso poder aplicar estos criterios.

Una matriz cuadrada es de **diagonal estrictamente dominante por filas** si en cada fila el valor absoluto del elemento diagonal es mayor que la suma de los valores absolutos de los otros elementos de la fila,

$$|a_{ii}| > \sum_{j=1, j\neq i}^{n} |a_{ij}|\,, \quad \forall i.$$

Teorema 0.1. *Si A es de diagonal estrictamente dominante por filas, entonces el método de Gauss-Seidel converge para cualquier x^0 inicial.*

Teorema 0.2. *Si A es definida positiva, entonces el método de Gauss-Seidel converge para cualquier x^0 inicial.*

Teóricamente el método de Gauss-Seidel se debería detener cuando $\|x^k - x^*\| < \varepsilon$. Sin embargo la condición anterior necesita conocer x^*, que es precisamente lo que se está buscando. Entonces, de manera práctica el método de GS se detiene cuando $\|x^k - x^{k+n}\| < \varepsilon$.

Dejando de lado los superíndices, las fórmulas del método de Gauss-Seidel se pueden reescribir para facilitar el algoritmo y para mostrar que $\|x^k - x^*\|$ y $\|x^k - x^{k+n}\|$ están relacionadas.

$$x_i \leftarrow \frac{b_i - \displaystyle\sum_{j=1, j\neq i}^{n} a_{ij}x_j}{a_{ii}},$$

$$x_i \leftarrow \frac{b_i - \displaystyle\sum_{j=1}^{n} a_{ij}x_j + a_{ii}x_i}{a_{ii}},$$

$$x_i \leftarrow x_i + \frac{b_i - A_{i.}\, x}{a_{ii}}.$$

Sean

$$\begin{aligned} r_i &= b_i - A_{i.}\, x, \\ \delta_i &= \frac{r_i}{a_{ii}}. \end{aligned}$$

El valor r_i es simplemente el error, residuo o resto que se comete en la i-ésima ecuación al utilizar el x actual. Si $r_i = 0$, entonces la ecuación i-ésima se satisface perfectamente. El valor δ_i es la modificación que sufre x_i en una iteración.

Sean $r = (r_1, r_2, ..., r_n)$, $\delta = (\delta_1, \delta_2, ..., \delta_n)$. Entonces $x^{k+n} = x^k + \delta$. Además x^k es solución si y solamente si $r = 0$, o sea, si y solamente $\delta = 0$. Lo anterior justifica que el método de GS se detenga cuando $\|\delta\| \leq \varepsilon$. La norma $\|\delta\|$ puede ser la norma euclidiana o $\max |\delta_i|$ o $\sum |\delta_i|$.

Si en el criterio de parada del algoritmo se desea enfatizar sobre los errores o residuos, entonces se puede comparar $\|\delta\|$ con $\varepsilon / \|(a_{11}, ..., a_{nn})\|$; por ejemplo,

$$\|\delta\| \leq \frac{\varepsilon}{\max |a_{ii}|}.$$

El esquema del algoritmo para resolver un sistema de ecuaciones por el método de Gauss-Seidel es:

datos: A, b, x^0, ε, maxit
$x = x^0$
para $k = 1, ...,$maxit
    nrmD$\leftarrow 0$
    para $i = 1, ..., n$
        $\delta_i = (b_i - \text{prodEsc}(A_{i.}, x))/a_{ii}$
        $x_i \leftarrow x_i + \delta_i$
        nrmd$\leftarrow$nrmD$+\delta_i$
    fin-para i
    si nrmD $\leq \varepsilon$ **ent** $x^* \approx x$, **salir**
fin-para k

0.2 Solución por mínimos cuadrados

Consideremos ahora un sistema de ecuaciones $Ax = b$, no necesariamente cuadrado, donde A es una matriz $m \times n$ cuyas columnas son linealmente independientes. Esto implica que hay más filas que columnas, $m \geq n$, y que además el rango de A es n. Es muy probable que este sistema no tenga solución, es decir, tal vez no existe x que cumpla exactamente las m igualdades. Se desea que

$$\begin{aligned} Ax &= b, \\ Ax - b &= 0, \\ \|Ax - b\| &= 0, \\ \|Ax - b\|_2 &= 0, \\ \|Ax - b\|_2^2 &= 0. \end{aligned}$$

Es Posible que lo deseado no se cumpla, entonces se quiere que el incumplimiento (el error) sea lo más pequeño posible. Se desea minimizar esa cantidad,

$$\min \ \|Ax - b\|_2^2 \ . \tag{1}$$

El vector x que minimice $\|Ax-b\|_2^2$ se llama solución por mínimos cuadrados. Como se verá más adelante, tal x existe y es único (suponiendo que las columnas de A son linealmente independientes).

Con el ánimo de hacer más clara la deducción, supongamos que A es una matriz 4×3. Sea $f(x) = \|Ax - b\|_2^2$,

$$\begin{aligned} f(x) =& (a_{11}x_1 + a_{12}x_2 + a_{13}x_3 - b_1)^2 + (a_{21}x_1 + a_{22}x_2 + a_{23}x_3 - b_2)^2 + \\ & (a_{31}x_1 + a_{32}x_2 + a_{33}x_3 - b_3)^2 + (a_{41}x_1 + a_{42}x_2 + a_{43}x_3 - b_4)^2. \end{aligned}$$

0.2.1 Ecuaciones normales

Para obtener el mínimo de f se requiere que las tres derivadas parciales, $\partial f/\partial x_1$, $\partial f/\partial x_2$ y $\partial f/\partial x_3$, sean nulas.

$$\begin{aligned} \frac{\partial f}{\partial x_1} =& 2(a_{11}x_1 + a_{12}x_2 + a_{13}x_3 - b_1)a_{11} \\ & + 2(a_{21}x_1 + a_{22}x_2 + a_{23}x_3 - b_2)a_{21} \end{aligned}$$

$$
\begin{aligned}
&+ 2(a_{31}x_1 + a_{32}x_2 + a_{33}x_3 - b_3)a_{31} \\
&+ 2(a_{41}x_1 + a_{42}x_2 + a_{43}x_3 - b_4)a_{41}.
\end{aligned}
$$

Escribiendo de manera matricial,

$$
\begin{aligned}
\frac{\partial f}{\partial x_1} =& 2(A_1.x - b_1)a_{11} + 2(A_2.x - b_2)a_{21} + 2(A_3.x - b_3)a_{31} \\
&+ 2(A_4.x - b_4)a_{41}.
\end{aligned}
$$

Si B es una matriz y u un vector columna, entonces $(Bu)_i = B_i.u$.

$$
\begin{aligned}
\frac{\partial f}{\partial x_1} &= 2\Big(((Ax)_1 - b_1)a_{11} + ((Ax)_2 - b_2)a_{21} + ((Ax)_3 - b_3)a_{31} \\
&\quad +((Ax)_4 - b_4 a_{41}\Big), \\
&= 2\sum_{i=1}^{4}(Ax - b)_i\, a_{i1}, \\
&= 2\sum_{i=1}^{4}(A._1)_i(Ax - b)_i, \\
&= 2\sum_{i=1}^{4}(A^{\mathrm{T}}{}_1.)_i(Ax - b)_i, \\
&= 2A^{\mathrm{T}}{}_1.(Ax - b), \\
&= 2\big(A^{\mathrm{T}}(Ax - b)\big)_1
\end{aligned}
$$

De manera semejante

$$
\begin{aligned}
\frac{\partial f}{\partial x_2} &= 2\big(A^{\mathrm{T}}(Ax - b)\big)_2\,, \\
\frac{\partial f}{\partial x_3} &= 2\big(A^{\mathrm{T}}(Ax - b)\big)_3
\end{aligned}
$$

Igualando a cero las tres derivadas parciales y quitando el 2 se tiene

$$
\begin{aligned}
\big(A^{\mathrm{T}}(Ax - b)\big)_1 &= 0, \\
\big(A^{\mathrm{T}}(Ax - b)\big)_2 &= 0, \\
\big(A^{\mathrm{T}}(Ax - b)\big)_3 &= 0
\end{aligned}
$$

Es decir,

$$\begin{aligned} A^{\mathrm{T}}(Ax - b) &= 0, \\ A^{\mathrm{T}}\,A\,x &= A^{\mathrm{T}}\,b\,. \end{aligned} \tag{2}$$

Las ecuaciones (2) se llaman **ecuaciones normales** para la solución (o seudosolución) de un sistema de ecuaciones por mínimos cuadrados.

La matriz $A^{\mathrm{T}}A$ es simétrica de tamaño $n \times n$. En general, si A es una matriz $m \times n$ de rango r, entonces $A^{\mathrm{T}}A$ también es de rango r (ver [Str86]). Como se supuso que el rango de A es n, entonces $A^{\mathrm{T}}A$ es invertible. Más aún, $A^{\mathrm{T}}A$ es definida positiva.

Por ser $A^{\mathrm{T}}A$ invertible, hay una única solución de (2), o sea, hay un solo vector x que hace que las derivadas parciales sean nulas. En general, las derivadas parciales nulas son simplemente una condición necesaria para obtener el mínimo de una función (también lo es para máximos o para puntos de silla), pero en este caso, como $A^{\mathrm{T}}A$ es definida positiva, f es convexa, y entonces anular las derivadas parciales se convierte en condición necesaria y suficiente para el mínimo.

En resumen, si las columnas de A son linealmente independientes, entonces la solución por mínimos cuadrados existe y es única. Para obtener la solución por mínimos cuadrados se resuelven las ecuaciones normales.

Como $A^{\mathrm{T}}A$ es definida positiva, (2) se puede resolver por el método de Cholesky. Si $m \geq n$ y al hacer la factorización de Cholesky resulta que $A^{\mathrm{T}}A$ no es definida positiva, entonces las columnas de A son linealmente dependientes.

Si el sistema $Ax = b$ tiene solución exacta, ésta coincide con la solución por mínimos cuadrados.

Ejemplo 0.3. Resolver por mínimos cuadrados:

$$\begin{bmatrix} 2 & 1 & 0 \\ -1 & -2 & 3 \\ -2 & 2 & 1 \\ 5 & 4 & -2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 3.1 \\ 8.9 \\ -3.1 \\ 0.1 \end{bmatrix}.$$

Las ecuaciones normales dan:

$$\begin{bmatrix} 34 & 20 & -15 \\ 20 & 25 & -12 \\ -15 & -12 & 14 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 4.0 \\ -20.5 \\ 23.4 \end{bmatrix}$$

La solución por mínimos cuadrados es:

$$x = (2.0252,\ \ -1.0132,\ \ 2.9728)\,.$$

El error, $Ax - b$, es:

$$\begin{bmatrix} -0.0628 \\ 0.0196 \\ -0.0039 \\ 0.0275 \end{bmatrix}. \quad \diamond$$

Ejemplo 0.4. Resolver por mínimos cuadrados:

$$\begin{bmatrix} 2 & 1 & 3 \\ -1 & -2 & 0 \\ -2 & 2 & -6 \\ 5 & 4 & 6 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 3 \\ 9 \\ -3 \\ 0 \end{bmatrix}.$$

Las ecuaciones normales dan:

$$\begin{bmatrix} 34 & 20 & 48 \\ 20 & 25 & 15 \\ 48 & 15 & 81 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 3 \\ -21 \\ 27 \end{bmatrix}$$

Al tratar de resolver este sistema de ecuaciones por el método de Cholesky; no se puede obtener la factorización de Cholesky, luego $A^{\mathrm{T}}A$ no es definida positiva, es decir, las columnas de A son linealmente dependientes. Si se aplica el método de Gauss, se obtiene que $A^{\mathrm{T}}A$ es singular y se concluye que las columnas de A son linealmente dependientes. $\diamond$

Ejemplo 0.5. Resolver por mínimos cuadrados:

$$\begin{bmatrix} 2 & 1 \\ -1 & -2 \\ -2 & 2 \\ 5 & 4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 3 \\ 0 \\ -6 \\ 6 \end{bmatrix}.$$

Las ecuaciones normales dan:

$$\begin{bmatrix} 34 & 20 \\ 20 & 25 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 48 \\ 15 \end{bmatrix}$$

La solución por mínimos cuadrados es:

$$x = (2,\ \ -1)\,.$$

El error, $Ax - b$, es:

$$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

En este caso, el sistema inicial tenía solución exacta y la solución por mínimos cuadrados coincide con ella. $\diamond$

La implementación eficiente de la solución por mínimos cuadrados, vía ecuaciones normales, debe tener en cuenta algunos detalles. No es necesario construir toda la matriz simétrica $A^{\mathrm{T}}A$ (n^2 elementos). Basta con almacenar en un arreglo de tamaño $n(n+1)/2$ la parte triangular superior de $A^{\mathrm{T}}A$.

Este almacenamiento puede ser por filas, es decir, primero los n elementos de la primera fila, enseguida los $n-1$ elementos de la segunda fila a partir del elemento diagonal, después los $n-2$ de la tercera fila a partir del elemento diagonal y así sucesivamente hasta almacenar un solo elemento de la fila n. Si se almacena la parte triangular superior de $A^{\mathrm{T}}A$ por columnas, se almacena primero un elemento de la primera columna, enseguida dos elementos de la segunda columna y así sucesivamente. Cada una de las dos formas tiene sus ventajas y desventajas. La solución por el método de Cholesky debe tener en cuenta este tipo de estructura de almacenamiento de la información.

Otros métodos eficientes para resolver sistemas de ecuaciones por mínimos cuadrados utilizan matrices ortogonales de Givens o de Householder.

```
function a = interFilas(a, i, j)
  t = a(i,:)
  a(i,:) = a(j,:)
  a(j,:) = t
endfunction
//-----------------------------------------------------------
function x = sistTriSup(a, b)
  // solucion de un sistema triangular superior
  n = size(a,1)
  x = zeros(n,1)
  for i=n:-1:1
    x(i) = ( b(i) - a(i,i+1:n)*x(i+1:n) ) / a(i,i)
  end
endfunction
//-----------------------------------------------------------
function [x, y] = interc(x, y)
  t = x
  x = y
  y = t
endfunction

//===========================================================
//===========================================================

// metodo de Gauss con pivoteo parcial
// no muy eficiente

a0 = [ 2  4 5 -10; 1 2 3 4; -10 8 5 4; -1 1 2 -2]
b = [-14 40 44 -1]'
eps = 1.0e-8;

n = size(a0,1)

a = [a0 b]

for k = 1:n-1
  [vmax , pos] = max( abs(a(k:n,k) ) )
```

```
  if vmax <= eps
    printf('\n Matriz singular o casi singular.\n\n')
    return
  end

  m = k-1 + pos
  if m > k
    a = interFilas(a, k, m)
  end

  for i = k+1:n
    coef = a(i,k)/a(k,k)
    a(i,:) = a(i,:) - coef*a(k,:)
  end

end
if abs( a(n,n) ) <= eps
  printf('\n Matriz singular o casi singular.\n\n')
  return
end

x = sistTriSup(a(:,1:n), a(:,n+1) )

//===========================================================
//===========================================================

// metodo de Cholesky

// resolver  A x = b
// A simetrica y definida positiva
//
// factorizacion de Cholesky  A = U' U,  U triangular superior inevertible
//
// A x = b
// U' U x = b
// sea  y = U x
// U' y = b
// resolver el sistema anterior para obtener  y
```

```
// reoslver  U x = y  para obtener  x


a = [9 2 3; 2 8 -1; 3 -1 5]

b = [36 24 23]'
n = size(a, 1)

y = zeros(n,1);
x = zeros(n,1);

u = chol(a)

y(1) = b(1)/u(1,1)
for i=2:n
  y(i) = ( b(i) - u(1:i-1,i)'*y(1:i-1) )/u(i,i)
end

x(n) = y(n)/u(n,n)
for i=n-1:-1:1
  x(i) = ( y(i) - u(i,i+1:n)*x(i+1:n) )/u(i,i)
end

//===========================================================
//===========================================================

// metodo de Gauss-Seidel

a = [ 10 2 -1 0 ; 1 20 -2 3 ; -2 1 30 0 ; 1 2 3 20]
b = [26   -15  53   47 ]'

x0 = [0 0 0 0]'
n = size(a,1)


x = x0
maxit = 10
for k = 1:maxit
```

```
  for i = 1:n
    res = b(i) - a(i,:)*x
    x(i) = x(i) + res/a(i,i)
  end
end
```