Algoritmos

danizufrique

April 2019

0.1 Sistemas triangulares

Sean la matriz $U \in R^{N\times N}$ **triangular superior** con elementos diagonales no nulos, el vector de incógnitas $x \in R^N$ y el vector de términos independientes b, tenemos el siguiente sistema triangular: $\mathbf{Ux = b}$. Este sistema se resuelve por **sustitución hacia atrás** y el algoritmo para resolverlo es:

$$x_i = \frac{1}{u_{ii}}\left(b_i - \sum_{j=i+1}^{N} u_{ij}x_j\right), \ \ con \ \ i = N, ..., 1$$

Sea la matriz $L \in R^{N\times N}$ **triangular inferior** con elementos diagonales no nulos, el vector de incógnitas $x \in R^N$ y el vector de términos independientes b, tenemos el siguiente sistema triangular: $\mathbf{Lx = b}$. Este sistema se resuelve por **sustitución hacia adelante** y el algoritmo para resolverlo es:

$$x_i = \frac{1}{l_{ii}}\left(b_i - \sum_{j=1}^{i-1} l_{ij}x_j\right), \ \ con \ \ i = 1, ..., N$$

0.2 Métodos de Gauss y Gauss-Jordan. Pivotaje

Los datos son $N \geq 1$, $A \in R^{N\times N}$, $b \in R^N$.

Suponemos que $a_{kk}^{(k)} \neq 0 \ con \ k = 1, ..., N$ (en caso contrario hemos terminado y no es posible llegar a un sistema triangular equivalente). Definimos recursivamente los multiplicadores:

$$m_{ik} := \frac{a_{ik}^{(k)}}{a_{kk}^{(k)}}, \ \ con \ \ i = k+1, ..., N$$

$$a_{ij}^{(k+1)} := a_{ij}^{(k)} - m_{ik}akj^{(k)}, \ \ con \ \ i = k+1, ..., N; \ j = k, ..., N$$

$$b_i^{(k+1)} := b_i^{(k)} - m_{ik}b_k^{(k)}, \ \ con \ \ i = k+1, ..., N$$

0.3 Factorización Doolittle y Crout

0.3.1 Doolittle

Sea $i = 1, ..., N$, entonces

$$u_{ij} = a_{ij} - \sum_{k=1}^{i-1} l_{ik}u_{kj}, \ \ con \ j = i, ..., N$$

Y supuesto que $u_{ii} \neq 0$

$$l_{ji} = \frac{1}{u_{ii}}\left(a_{ji} - \sum_{k=1}^{i-1} l_{jk}u_{ki}\right), \ \ con \ j = i+1, ..., N$$

0.3.2 Crout

Aprovechar algoritmo Doolittle, ya que si $A^T = LU$, entonces $A = U^T L^T$.

0.4 Cholesky

Para todo $j = 1, ..., N$

$$i = 1, ..., j-1 \;\Rightarrow\; u_{ij} = \frac{1}{u_{ii}} \left(a_{ij} - \sum_{k=1}^{i-1} u_{ki} u_{kj} \right)$$

$$u_{jj} = \sqrt{a_{jj} - \sum_{k=1}^{j-1} u_{kj}^2}$$

Además, se eliminan los elementos de debajo de la diagonal principal, ya que hay simetría.

0.5 Jacobi y Gauss-Seidel

0.5.1 Jacobi

$$i = 1, ..., N \Rightarrow x_{ni} = \frac{1}{a_{ii}} \left(b_i - \sum_{j=1 \wedge j \neq i}^{N} a_{ij} x_{n-1\ j} \right)$$

0.5.2 Gauss-Seidel

$$i = 1, ..., N \Rightarrow x_{ni} = \frac{1}{a_{ii}} \left(b_i - \sum_{j=1}^{i-1} a_{ij} x_{nj} - \sum_{j=i+1}^{N} a_{ij} x_{n-1\ j} \right)$$