Introducción a la Programación y Análisis Numérico

Año 2022

Práctica 9: Sistemas de Ecuaciones Lineales.

Ej. 1: Métodos directos: Eliminación Gaussiana/ Método de Gauss-Jordan. Pivoteo Parcial

a) Considere los siguientes sistemas de ecuaciones $Ax = b$, donde

$$A_1 = \begin{bmatrix} 0 & 1 & -1 \\ 2 & -1 & -1 \\ 1 & 1 & -1 \end{bmatrix}, b_1 = \begin{bmatrix} 2 \\ 0 \\ 1 \end{bmatrix}$$

$$A_2 = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & -1 & 1 \end{bmatrix}, b_2 = \begin{bmatrix} 2 \\ 3 \\ 1 \end{bmatrix}.$$

i. Para ambos sistemas, calcule el determinante de A haciendo uso de la función `det()` de MATLAB/OCTAVE. ¿Qué puede decir del sistema A_2?

ii. ¿Para cuál de los sistema puede asegurar la existencia de una solución única? Resuelva el sistema de ecuaciones mediante eliminación Gaussiana con *pivoteo parcial*.

b) Estimación a-priori del error. Análisis perturbativo:

Dado el sistema $Ax = b$, suponga que el término b se perturba por una cantidad δb.

El efecto de la perturbación sobre la solución será $x + \delta x$, con δx acotada por

$$\frac{\| \delta x \|}{\| x \|} \leq cond(A) \frac{\| \delta b \|}{\| b \|}$$

Si la matriz A es la que es perturbada por una cantidad δA, el efecto de tal perturbación sobre la solución será $x + \delta x$ con δx acotada por

$$\frac{\| \delta x \|}{\| x + \delta x \|} \leq cond(A) \frac{\| \delta A \|}{\| A \|}$$

Inspeccione la función `cond()` de MATLAB/OCTAVE. Calcule los valores de `cond(A1)` y `cond(A2)`. ¿Qué indican estos resultados?

c) Sean los sistemas de ecuaciones $Ax = b$ determinados por:

$$A_1 = \begin{bmatrix} 10^{-15} & 1 \\ 1 & 10^{11} \end{bmatrix}, A_2 = \begin{bmatrix} 10^{-14,6} & 1 \\ 1 & 10^{15} \end{bmatrix}, A_3 = \begin{bmatrix} 10^{15} & 1 \\ 1 & 10^{-14,6} \end{bmatrix}$$

y

$$b_1 = \begin{bmatrix} 1 + 10^{-15} \\ 10^{11} + 1 \end{bmatrix}, b_2 = \begin{bmatrix} 1 + 10^{-14,6} \\ 10^{15} + 1 \end{bmatrix}, b_3 = \begin{bmatrix} 10^{15} + 1 \\ 1 + 10^{-14,6} \end{bmatrix}.$$

Calcule los determinantes de las matrices A para cada sistema. Conociendo los determinantes de dichas matrices y sin resolver los sistemas, ¿qué puede decir de las soluciones de los mismos? ¿Qué puede decir a partir del número de condición?

Ej. 2: Factorización LU

a) Encuentre algebraicamente la factorización $PA = LU$ de las siguientes matrices:

$$A = \begin{bmatrix} 1 & 2 & -1 \\ 3 & 6 & 2 \\ -1 & 1 & 4 \end{bmatrix}, B = \begin{bmatrix} 0 & 1 & 4 \\ -1 & 2 & 1 \\ 1 & 3 & 3 \end{bmatrix}.$$

b) De las líneas de código proporcionadas al final de la práctica, utilice las necesarias para implementar como función la resolución de un sistema de ecuaciones lineal $Ax = b$ mediante factorización LU y sustitución hacia adelante y hacia atrás. El programa debe devolver como variables de salida las matrices L, U, P y el vector solución x.

c) Utilizando el programa implementado en el inciso anterior, encuentre la factorización LU y la solución de los sistemas de ecuaciones $Ax = b$ para las matrices de coeficientes:

$$A_1 = \begin{bmatrix} 0 & 1 & 2 \\ 1 & 2 & 3 \\ 2 & 3 & 2 \end{bmatrix}, A_2 = \begin{bmatrix} 2 & 1 & 0 \\ 0 & 3 & 2 \\ 1 & 2 & 4 \end{bmatrix}, A_3 = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 0 & -2 \\ -1 & 2 & 3 \end{bmatrix}$$

y

$$b_1 = \begin{bmatrix} 2 \\ 4 \\ 5 \end{bmatrix}, b_2 = \begin{bmatrix} 4 \\ 12 \\ 17 \end{bmatrix}, b_3 = \begin{bmatrix} 1 \\ -2 \\ 3 \end{bmatrix}$$

d) MATLAB/OCTAVE cuenta con una función predefinida que permite obtener la descomposición LU. La expresión **lu(A)** devuelve las matrices L y U o L, U y P, dependiendo de la cantidad de argumentos de salida explicitados. Utilice la función predefinida para verificar las descomposiciones realizadas y mediante **y = L\(P*b)** y **x = U\y** determine el vector solución x. Compare con los resultados del inciso c). Las diferencias obtenidas, ¿a qué considera que pueden deberse?

e) Sea la siguiente matriz tri-diagonal A:

$$A = \begin{bmatrix} 2 & 2 & 0 & 0 & 0 & 0 \\ 1 & 5 & 4 & 0 & 0 & 0 \\ 0 & 4 & 8 & 4 & 0 & 0 \\ 0 & 0 & 6 & 8 & 6 & 0 \\ 0 & 0 & 0 & 6 & 4 & 3 \\ 0 & 0 & 0 & 0 & 4 & 9 \end{bmatrix}$$

i. Realice una función para su resolución que explote la naturaleza tri-diagonal de la matriz, reduciendo así el número de operaciones. **Ayuda**: revise la página 306 del libro Métodos numéricos para ingenieros, Chapra - Canale, Quinta edición.

ii. Resolver los sistemas de ecuaciones $Ax = b$ donde

$$b_1 = \begin{bmatrix} 4 \\ 10 \\ 16 \\ 20 \\ 13 \\ 13 \end{bmatrix}, b_2 = \begin{bmatrix} 8 \\ 20 \\ 32 \\ 40 \\ 26 \\ 26 \end{bmatrix}, b_3 = \begin{bmatrix} 12 \\ 30 \\ 48 \\ 60 \\ 39 \\ 39 \end{bmatrix}$$

Ej. 3: Métodos iterativos: Gauss-Seidel y Jacobi.

Método de Jacobi: dada A de $n \times n$ con elementos diagonales no nulos. A partir de un vector $\mathbf{x}^{(0)}$ de orden n, se obtiene el vector solución de forma iterativa. Es decir, para $k = 0, 1, 2, ...$

$$x_i^{(k)} = \frac{-\sum_{\substack{j=1 \\ j\neq i}}^{n} a_{ij} x_j^{k-1} + b_i}{a_{ii}}, \quad \textit{para } i = 1, \ldots n$$

Método de Gauss-Seidel: dada A de $n \times n$ con elementos diagonales no nulos. A partir de un vector $\mathbf{x}^{(0)}$ de orden n, se obtiene el vector solución de forma iterativa. Es decir, para $k = 0, 1, 2, ...$

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

En general como una primera aproximación suele tomarse el vector nulo, es decir, $\mathbf{x}^{(0)} = 0$. Además, como todo método iterativo necesitan de un criterio de corte para las iteraciones. Utilizaremos $\frac{||\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)}||}{||\mathbf{x}^{(k)}||} < \epsilon$, para una cierta tolerancia ϵ preestablecida.

a) Realice un programa en MATLAB/OCTAVE que permita resolver un sistema de ecuaciones del tipo $Ax = b$ mediante

i. El método iterativo de *Jacobi*.

ii. El método iterativo de *Gauss-Seidel*.

b) Utilice los programas desarrollados en el inciso anterior para resolver el sistema:

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

cuya solución única esta dada por $x = [1, -1]^T$

c) Considere una barra de metal de 10 m de longitud que es sometida a una fuente de calor en un extremo por un tiempo determinado. Al cabo de ese tiempo, la fuente de calor cesa y se miden temperaturas de 10°C y 200°C en ambos extremos. Se desea encontrar la temperatura en los puntos intermedios x_1, x_2, x_3 y x_4 ubicados a 2 m equidistantes unos de otros. Asumiendo que la temperatura en cada punto es el promedio de las temperaturas medidas en los puntos vecinos,

i. Escriba el sistema de ecuaciones $Ax = b$ que representa el problema en cuestión.

ii. Encuentre las temperaturas en cada punto utilizando los métodos de *Jacobi* y *Gauss-Seidel* programados en el inciso *a*.

d) Considere el sistema de ecuaciones dado, con solución exacta $x = (1, 1)$:

$$\begin{bmatrix} 1 & -5 \\ 7 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} -4 \\ 6 \end{bmatrix}.$$

i. Utilizando la aproximación inicial $(x_1, x_2) = (0, 0)$, resuelva el mismo sistema mediante los métodos de *Gauss-Seidel* y *Jacobi*.

ii. ¿Qué puede decir de la convergencia de estos métodos en este caso? ¿Por qué ocurre este fenómeno.

e) Sean los siguientes sistemas de ecuaciones:

$$\begin{bmatrix} 3{,}8 & 1{,}6 & 0{,}9 \\ -0{,}7 & 5{,}4 & 1{,}6 \\ 1{,}5 & 1{,}1 & 3{,}2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 15{,}5 \\ 10{,}3 \\ 3{,}5 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 1 \\ -1 & 1 & 0 \\ 1 & 2 & -3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 4 \\ 1 \\ -4 \end{bmatrix}.$$

y

$$\begin{bmatrix} 1 & 0{,}5 & 0{,}5 \\ 0{,}5 & 1 & 0{,}5 \\ 0{,}5 & 0{,}5 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 2 \\ 2 \\ 2 \end{bmatrix}.$$

i. Resuelva los sistemas de ecuaciones dados utilizando los métodos de *Gauss-Seidel* y *Jacobi*. ¿Qué puede decir de la convergencia de estos métodos para los sistemas dados?

ii. Los elementos de la diagonal de la matriz nos proporcionan un criterio suficiente pero no necesario de convergencia. Si el elemento de la diagonal en módulo es mayor a la suma de los módulos de los demás elementos de la fila, es decir, si se cumple:

$$|a_{i,i}| > \sum_{\substack{j=1 \\ j \neq i}}^{n} |a_{i,j}|$$

la convergencia está garantizada [1]. Calcule dicho criterio para las matrices del inciso anterior.

[1] Notar que al ser un criterio suficiente pero no necesario, los métodos pueden converger aún cuando no se cumpla, pero la convergencia estará garantizada cuando la condición se satisfaga.

Líneas de código para la factorización LU con pivoteo y para la resolución del sistema de ecuaciones por sustitución:

```
%factorizacion PA = LU
%y resolucion del sistema mediante sustitucion hacia adelante y hacia atras
clc; clear; close all;

M = load('matriz.dat');
n = length(M(:,1));
A = M(:,1:3);
b = M(:,4);
P = eye(n);
L = eye(n);
U = zeros(n);

%permutacion de la matriz a:
%la permutacion se realiza mirando el maximo abs(value) por columna
for i = 1: n-1
  [val_a, index_a] = max(abs(A(i:n,i)));

  if(abs(val_a) <= 1E-8)
    disp('elemento nulo en la diagonal')
    stop
  else
    A([i index_a+i-1],:) = A([index_a+i-1,i], :);
    P([i index_a+i-1],:) = P([index_a+i-1,i],:);

  endif
endfor

U(1,:) = A(1,:);
L(:,1) = A(:,1)./U(1,1);

for i = 2: n-1;
  U(i,i) = A(i,i) - sum((L(i,1:i-1))'.*U(1:i-1,i));
  for j = i+1: n;
    U(i,j) = A(i,j)-sum((L(i,1:i-1))'.*U(1:i-1,j));
    L(j,i) = 1/U(i,i)*(A(j,i)-sum((L(j,1:i-1))'.*U(1:i-1,i)));
  endfor
endfor

U(n,n) = A(n,n) - sum((L(n,1:n-1))'.*U(1:n-1,n));

%sustitución hacia adelante

b = P*b;
for i = 2: n
  suma = b(i);
    for j =1: i-1
      suma = suma - L(i,j)*b(j);
    end
    b(i) = suma;
end

%sustitución hacia atrás

x(n)  = b(n)/U(n,n);
for i = n-1: -1: 1
  suma = 0;
  for j = i + 1: n
    suma = suma + U(i,j)*x(j);
  end
  x(i) = (b(i) - suma)/U(i,i);
end
```