SISTEMAS LINEALES

APROXIMACIÓN NUMÉRICA

INTRODUCCIÓN

RESOLVER UN SISTEMA LINEAL DE ECUACIONES

MÉTODOS DE SOLUCIÓN

MÉTODOS NUMÉRICOS
Ing. Patricio Pugarín Díaz, Mgs.
DEPARTAMENTO DE CIENCIAS EXACTAS - ESPE

CONTENIDO

Título	**Sistemas de Ecuaciones lineales**
Duración	**120 minutos**
Información general	**Resaltar la importancia de resolver un sistema lineal de ecuaciones, con programación, aplicando Métodos Numéricos en problemas de Ingeniería.**
Objetivo	**Conocer las técnicas de los métodos numéricos, con programación numérica, para resolver sistemas lineales de ecuaciones y obtener resultados confiables.**

SISTEMAS DE ECUACIONES LINEALES

RESOLUCIÓN DE SISTEMAS DE ECUACIONES LINEALES:

Sea el sistema de ecuaciones de la forma general:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \cdots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + \cdots + a_{2n}x_n = b_2 \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 + \cdots + a_{3n}x_n = b_3 \\ \vdots \\ \vdots \\ a_{n1}x_1 + a_{n2}x_2 + a_{n3}x_3 + \cdots + a_{nn}x_n = b_{n,} \end{cases}$$

En donde $x_1,\ x_2, x_3, \dots, x_n$ son las incógnitas del sistema;

$a_{ij},\ i = 1,\ 2,\ 3,\ \dots,\ n,\ \ j = 1,\ 2,\ 3,\ \dots,\ n$ son los coeficientes del sistema; y,

$b_i,\ i = 1,\ 2,\ 3,\ \dots,\ n$ son los términos independientes.

El sistema anterior se puede representar matricialmente como:

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3n} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \cdots & a_{nn} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_n \end{pmatrix}$$

Siendo $A = \begin{pmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3n} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \cdots & a_{nn} \end{pmatrix}$= Matriz de coeficientes del sistema

$x = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{pmatrix}$= Vector de soluciones; y $B = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_n \end{pmatrix}$= Vector de términos independientes

Por tanto el sistema matricial se reduce a la Ecuación Matricial del sistema $\boldsymbol{AX} = \boldsymbol{B}$; que es la ecuación que se tiene que resolver.

NOTA:

- El objetivo es construir un algoritmo que permita resolver el problema presentado, dado por la Ecuación Matricial del sistema.
- Es necesario analizar los errores que aparecen cuando se resuelve con ayuda de un computador para controlarlos y reducirlos.

1. SISTEMA DE ECUACIONES LINEALES, TRIANGULAR SUPERIOR.

EJERCICIO:

Establecer un algoritmo de programación para resolver el sistema triangular superior de n ecuaciones con n incógnitas.

(Este ejercicio se desarrollará en clase para establecer una metodología de trabajo para desarrollar programas numéricos).

2. SISTEMA DE ECUACIONES LINEALES, TRIANGULAR INFERIOR.

EJERCICIO:

Establecer un algoritmo de programación para resolver un sistema triangular inferior de n ecuaciones con n incógnitas.

(Este ejercicio se desarrollará en clase para establecer una metodología de trabajo para desarrollar programas numéricos).

EJERCICIOS:

1. Dada una matriz A y dos filas de esta, por ejemplo la fila i y la fila j; más una constante m. Hacer un programa que dé como resultado una matriz B en donde la fila j de la matriz B será el resultado de sumar m veces la fila i de la matriz A a la fila j de la matriz A.
2. Construir un archivo de función que me permita cambiar la fila i con la fila j de una matriz A.

(Estos ejercicios se desarrollarán en clase, con programación numérica, y son funciones para el programa principal a desarrollar).

3. ELIMINACIÓN GAUSSIANA

Se trata de desarrollar un método para resolver un sistema de ecuaciones lineales de la forma $AX = B$, de n ecuaciones con n incógnitas atravez de un sistema equivalente triangular superior (o inferior) de la forma $UX = Y$; es decir:

Se tiene

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3n} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \cdots & a_{nn} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_n \end{pmatrix}$$

Y se quiere construir el sistema triangular superior equivalente $UX = Y$

$$\begin{pmatrix} a_{11}^1 & a_{12}^1 & a_{13}^1 & \cdots & a_{1n}^1 \\ 0 & a_{22}^2 & a_{23}^2 & \cdots & a_{2n}^2 \\ 0 & 0 & a_{33}^3 & \cdots & a_{3n}^3 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & a_{nn}^n \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1^1 \\ b_2^2 \\ b_3^3 \\ \vdots \\ b_n^n \end{pmatrix};$$

En donde $a_{nn}^n x_n = b_n^n$; es decir, $x_n = \frac{b_n^n}{a_{nn}^n}$

Luego, por el algoritmo de sustitución regresiva (para sistema triangular superior), encontramos el resto de las incógnitas o raíces del sistema.

NOTA:

El elemento a_{ij}^r indica el número de veces que se almacena un número en la posición (i, j).

ALGORITMO PARA LA PROGRAMACIÓN. - Se trata de modelar el método pensando ya en la programación en Matlab. Esto se desarrollará conjuntamente con los alumnos ejecutando en el pizarrón.

PIVOTEO TRIVIAL. -

Si $a_{qq}^{q} = 0$, localizamos la primera fila, por debajo de la fila q, por ejemplo la fila k, en la que se tiene $a_{kq}^{q} \neq 0$ y se intercambian las filas k y q; obteniendo un pivote no nulo deseado.

EJERCICIOS

Dado el sistema de ecuaciones lineales, que tiene como soluciones analíticas $x_1 = x_2 = 1$:

$$\begin{cases} 1.133x_1 + 5.281x_2 = 6.414 \\ 24.14x_1 - 1.210x_2 = 22.93 \end{cases}$$

a. Resolver aplicando la eliminación Gaussiana, con 4 cifras significativas para los cálculos a mano (construyendo la matriz ampliada del sistema).

b. Con las mismas consideraciones anteriores, resolver el sistema

$$\begin{cases} 24.14x_1 - 1.210x_2 = 22.93 \\ 1.133x_1 + 5.281x_2 = 6.414 \end{cases}$$

c. Discutir los resultados obtenidos en los literales anteriores.

NOTA:

1. Se puede concluir que una forma de controlar y reducir el error de redondeo es usando como pivote el elemento de mayor magnitud.
2. Esto se conoce como Pivoteo Parcial

PIVOTEO PARCIAL. -

Para reducir la propagación de errores de redondeo se sugiere que se compare el tamaño de todos los elementos la columna q, desde el elemento que está en la diagonal hasta el de la última fila, y localizar la fila en la que está el mayor elemento, en valor absoluto; digamos por ejemplo en la fila k. Procedemos a intercambiar las filas k y q, obteniendo que todos los multiplicadores $m_{rq} < 1,\ r = q+1, q+2, \dots, n$.

EJERCICIO PARA LA CLASE

Aplicando la eliminación Gaussiana y usando solamente los programas combinar filas e intercambio de filas, resolver el sistema lineal de ecuaciones:

$$\begin{pmatrix} -0.04 & 0.04 & 0.12 \\ 0.56 & -1.56 & 0.32 \\ -0.24 & 1.24 & -0.28 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 3 \\ 1 \\ 0 \end{pmatrix}$$

Usando el comando diary, de Matlab, ejecutar y crear un archivo.doc

(Se desarrollará en clase con la participación de los estudiantes)

NOTA:

El sistema de ecuaciones lineales $AX = B$ esta mal condicionado si la matriz A es mal condicionada; es decir, si pequeños cambios en la matriz de coeficientes A producen grandes cambios en la las soluciones del sistema.

EJERCICIO

Resolver el sistema de ecuaciones lineales

$$\begin{pmatrix} 1 & 2 & 1 \\ 10 & 18 & 12 \\ 20 & 22 & 40 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 7 \\ 78 \\ 144 \end{pmatrix}$$

(Se desarrollará en clase con la participación de los estudiantes)

EJERCICIO

Resolver el sistema de ecuaciones lineales

$$\begin{pmatrix} 1 & 2 & 1 \\ 10 & 18 & 12 \\ 20 & 22 & 39 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 7 \\ 78 \\ 144 \end{pmatrix}$$

(Se desarrollará en clase con la participación de los estudiantes)

EJERCICIO

Construir un programa, en Matlab, para resolver un sistema lineal de n ecuaciones con n incógnitas, aplicando la eliminación Gaussiana.

(Construir este programa, en clase, con la participación de los alumnos).

Una propuesta adicional, para este programa es la siguiente:

PROGRAMA: ELIMINACIÓN GAUSS

```
% Método de Gauss para resolver un sistemas de ecuaciones de ornen(nxn)
function x=sistemGaus(A,B)
[n n]=size(A);
Ab=[A';B]';
for k=1:n
    [bb ll]=max(abs(Ab(k:n,k)));
     if bb==0
        error('la matriz es singular')
     end
     m=k+ll-1;
     Ab=intercambiofilas(Ab,k,m);
     for j=k+1:n
     Ab=combinarfilas(Ab,k,j,-Ab(j,k)/Ab(k,k));
     end
end
 x=sistem_Tsuperior(Ab(:,1:n),Ab(:,n+1));
end
```

4. ELIMINACIÓN DE GAUSS JORDAN

Es una variación de la eliminación de Gauss mediante la cual, en la matriz ampliada del sistema, se eliminarán (hacer ceros) los números que están sobre y bajo un pivote. Luego, se debe conseguir una diagonal principal con unos; encontrándose las soluciones del sistema $AX = B$, en la última columna de esta matriz.

NOTA:

Se debe seguir considerando el trabajar con el mayor número de la columna, como pivote, para reducir el error de redondeo.

EJERCICIO.-

Aplicando el método de Gauss Jordan y usando solamente los programas combinar filas e intercambio de filas, resolver el sistema lineal de ecuaciones:

$$\begin{pmatrix} -0.04 & 0.04 & 0.12 \\ 0.56 & -1.56 & 0.32 \\ -0.24 & 1.24 & -0.28 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 3 \\ 1 \\ 0 \end{pmatrix}$$

(Se desarrollará en clase con la participación de los estudiantes)

EJERCICIO

Construir un programa, en Matlab, para resolver un sistema lineal de n ecuaciones con n incógnitas, aplicando Gauss Jordan.

(Construir este programa, en clase, con la participación de los alumnos).

Una propuesta adicional, para este programa es la siguiente:

PROGRAMA: GAUSS JORDAN

```
% sistema de ecuaciones de orden n
function x=GausJ(A,b)

[n n]=size(A);
Ab=[A';b]';

for k=1:n
    [bb ll]=max(abs(Ab(k:n,k)));
    % BB es el màximo valor de la columna, en la posición ll
     if bb==0
        error('la matriz es singular')
     end
    % Vamos a formar una matriz con los maximis ordenados
     m=k+ll-1;
     Ab=intercambiofilas(Ab,k,m);
     % le hacemos triangular superior
     for j=k+1:n
     Ab=combinarfilas(Ab,k,j,-Ab(j,k)/Ab(k,k));
     end
end
% Debemos ahora hacerle a Ab matriz triangular inferir con lo que
% se tendra finalmente una matriz diagonal + la columna de B
for k=n:-1:2

    for j=k-1:-1:1
        Ab =combinarfilas(Ab,k,j,-Ab(j,k)/Ab(k,k));
    end
end
% Matriz diagonal de unos
for k=1:n
    Ab(k,1:n+1)=Ab(k,1:n+1)/Ab(k,k);
end

% Podemos ahora leer directamente las soluciones x del sistema en
%la matriz última Ab
x=Ab(:,n+1);
end
```

5. INVERSA DE MATRICES

Se puede obtener fácilmente a partir de la eliminación de Gauss Jordan; para lo cual construimos la matriz ampliada del sistema de la forma

$$(A \mid I);$$

Siendo I la matriz identidad de orden n. Luego, mediante operaciones entre filas de una matriz, construimos la matriz equivalente de la forma:

$$(I \mid B)$$

De esta última matriz se tiene:

$$Matriz\ Inversa\ de\ A = \boldsymbol{A^{-1}} = \boldsymbol{B}$$

EJERCICIO

Encontrar la inversa de la matriz

$$\begin{pmatrix} -0.04 & 0.04 & 0.12 \\ 0.56 & -1.56 & 0.32 \\ -0.24 & 1.24 & -0.28 \end{pmatrix}$$

(Se desarrollará en clase con la participación de los estudiantes)

NOTA.-

Se puede encontrar la solución de la ecuación matricial $AX = B$, aplicando el método de la inversa (aprendida en el curso de Álgebra Lineal); mediante la aplicación de la fórmula matricial:

$$X = A^{-1} * B$$

Una propuesta adicional, para este programa es la siguiente:

PROGRAMA INVERSA POR GAUSS JORDAN

```
% calculo de la inversa de una matriz con Gauss_jordan
% Ingresa la matriz A
function Ain=inversaG_J(A)
[n1 n2]=size (A);
I=eye(n1);
Ab=[A';I]';
n=n1;

for k=1:n
    [bb l1]=max(abs(Ab(k:n,k)));
    % BB es el màximo valor de la columna, en la posición l1
     if bb==0
        error('la matriz es singular')
     end
    % Vamos a formar una matriz con los maximis ordenados
     m=k+l1-1;
     Ab=intercambiofilas(Ab,k,m);
     % le hacemos triangular superior
     for j=k+1:n
     Ab=combinarfilas(Ab,k,j,-Ab(j,k)/Ab(k,k));
     end
end
```

6. SISTEMAS LINEALES APLICANDO LA FACTORIZACIÓN $PA = LU$

FACTORIZACIÓN DE UNA MATRIZ A.-

Frecuentemente no es posible escalonar una matriz solo con operaciones de eliminación entre filas; sino que se requiere, adicionalmente, intercambiar filas previamente.

Para este tipo de matrices no existe la factorización $A = LU$, lo que se aplica es la factorización $\boldsymbol{PA = LU}$; siendo:

$\boldsymbol{P}$ = Matriz de permutación (guarda la información del cambio de filas)

$\boldsymbol{L}$ = Matriz triangular inferior con diagonal de unos

$\boldsymbol{U}$ = Matriz triangular superior.

NOTA

Es recomendable al realizar la factorización de la matriz A, mantener el criterio de buscar el mayor elemento de una columna para pivote; para reduje el error de redondeo.

EJERCICIO

Encontrar la factorización $PA = LU$ de la matriz:

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 4 & 5 \\ 7 & 8 & 9 \end{pmatrix}$$

NOTA

Aplique el método que aprendió en el curso de Algebra Lineal

(Este ejercicio se desarrollará conjuntamente con los alumnos en la clase)

EJERCICIO

Encontrar la factorización $PA = LU$ de la matriz:

$$A = \begin{pmatrix} 1 & 2 & -2 & 1 \\ 4 & 5 & -7 & 6 \\ 5 & 25 & -15 & -3 \\ 6 & -12 & -6 & 22 \end{pmatrix}$$

(Este ejercicio se desarrollará conjuntamente con los alumnos en la clase)

SOLUCIÓN DE SISTEMAS DE ECUACIONES LINEALES APLICANDO LA FACTORIZACIÓN $\boldsymbol{PA = LU}$

Se tiene que resolver el sistema lineal: $AX = B$ (1)

Por la Factorización de la matriz A, se tiene: $PA = LU$ (2)

Si multiplicamos (1) por P $PAX = PB$

Al remplazar (2) en esta última ecuación y llamando $B_1 = PB$, se tiene

$$LUX = B_1 \qquad (3)$$

Ahora, sea $\boldsymbol{UX = Y}$ (4)

Se tiene, por tanto $\boldsymbol{LY = B_1}$ (5)

Entonces: al resolver la ecuación (5), que es un sistema lineal de ecuaciones triangular inferior, encontramos el vector Y (aplicando el algoritmo de sustitución progresiva).

Con el vector Y, al resolver la ecuación (4), que corresponde a un sistema lineal de ecuaciones triangular superior, encontramos la solución $\boldsymbol{X}$ (aplicando el algoritmo de sustitución regresiva) del sistema lineal inicial $\boldsymbol{AX} = \boldsymbol{B}$.

EJERCICIO

Aplicando la factorización $PA = LU$ y usando solamente los programas para combinar filas, intercambio de filas y para resolver un sistema triangular superior e inferior, encontrar la solución del sistema lineal de ecuaciones:

$$\begin{pmatrix} -0.04 & 0.04 & 0.12 \\ 0.56 & -1.56 & 0.32 \\ -0.24 & 1.24 & -0.28 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 3 \\ 1 \\ 0 \end{pmatrix}$$

(Este ejercicio se desarrollará conjuntamente con los alumnos en la clase)

PROGRAMA: SISTEMA LINEAL APLICANDO LA FACTORIZACIÓN $\boldsymbol{PA = LU}$

```
% Problema
% Solución de un sistema de ecuaciones AX=B, utilizando en lugar de la
% matriz A, su descomposición AP=LU
function [x y]=sollu_sistLUP(A,B)
[L U P]=fac_LUP(A);
[n1 n2]=size(P);
n3=length(B);
if n2~=n3
    disp('no se puede resolver el sistema');
    return
end
B1=(P*B')';
y=(sistemGaus(L,B1));
%disp('La solución del sistema es:')
x=(sistemGaus(U,y))';
end
```

NOTA

Existen otros métodos, llamados iterativos, como el método de Gauss Seidel y que se recomienda hacer una investigación de cómo se aplica al resolver un sistema lineal de ecuaciones:

EJERCICIO

Resolver por el método de Gauss Seidel el sistema lineal de ecuaciones

$$\begin{pmatrix} 3 & -0.1 & -0.2 \\ 0.1 & 7 & -0.3 \\ 0.3 & -0.2 & -10 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 7.85 \\ -19.30 \\ 71.40 \end{pmatrix}$$

(Este ejercicio se desarrollará conjuntamente con los alumnos en la clase)

PROGRAMA: FACTORIZACIÓN $PA = LU$

```
%Problema
% descomposició LUP de una matriz A con pivote parcial
% Ingresa la matriz A
function [L U P]=fac_LUP(A)

[n n1]=size(A);
if n~=n1
    error('No se puede descomponer');
end
L=eye(n);
P=eye(n);
for k=1:n-1
    [m1,m2]=max(abs(A(k:n,k)));
    if m1==0
        disp('la matriz ingresada es singular');
    end
        p=k+m2-1;
        A=intercambiofilas(A,k,p);
        U=A;
        P=intercambiofilas(P,k,p);
        for k=1:n-1
        for j=k+1:n
            factor1=(U(j,k)/U(k,k));
            U(j,k:n)=U(j,k:n)-(U(j,k)/U(k,k))*U(k,k:n);
            L(j,k)=factor1;
        end
        end
end

end
```

REFERENCIAS BIBLIOGRAFICAS

1. Sánchez Juan Miguel, Problemas de Cálculo Numérico para ingenieros con aplicaciones Matlab, McGraw-Hill, Primera edición, 2005.
2. A. Quarteroni, F. Saleri, Cálculo Científico con Matlab y Octave. Springer-Verlag Italia, milano 2006