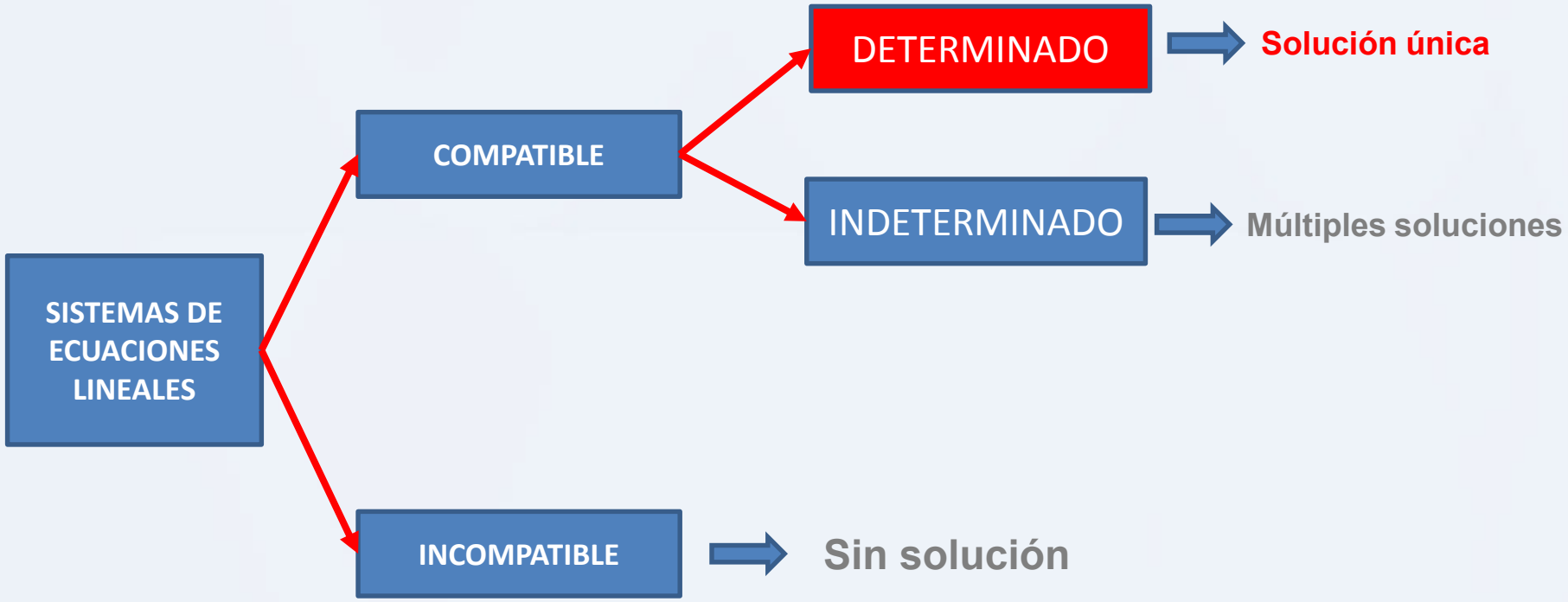
Solución numérica de sistemas de ecuaciones lineales

Solución de sistemas de ecuaciones lineales

Casos que se pueden presentar:



Solución numérica de sistemas de ecuaciones lineales

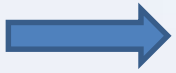
Solución de sistemas de ecuaciones lineales

La estructura matricial que tendría cada una de las alternativas seria parecida a lo siguiente:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$$

Compatible determinado (solución única)

$$\begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 4 \\ 5 \\ 0 \end{bmatrix}$$



Compatible indeterminado (Múltiples soluciones)

$$\begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$$

Inompatible (Sin solución)

Solución numérica de sistemas de ecuaciones lineales

Solución de sistemas de ecuaciones lineales

Con los métodos que se verán en este capítulo se trata de obtener la ***solución única*** de un sistema de ecuaciones lineales (sistema compatible determinado) y en caso contrario indicar que el sistema o tiene soluciones múltiples o no tiene solución.

Solución numérica de sistemas de ecuaciones lineales

Utilización de métodos numéricos para sistemas de ecuaciones lineales

Como se comentó al inicio del capítulo los métodos numéricos para la solución de este tipo de sistemas solamente nos sirven cuando el sistema tiene una solución única (Compatible determinado).

Es recomendable antes de aplicar cualquier método en verificar el valor del determinante de la matriz A.

$$AX = B$$

$$\det\left(A\right) \neq 0$$

Si dicho determinante es igual a cero el sistema no tiene una solución única y no es aplicable la utilización de método numéricos

Solución numérica de sistemas de ecuaciones lineales

Método de Gauss-Jordan

El Método de ***Gauss–Jordan*** o también llamado eliminación de Gauss–Jordan, es un método que sirve para varias cosas (por ejemplo obtener la inversa), pero en nuestra materia lo usaremos para resolverse sistemas de ecuaciones lineales con n números de variables.

$$2x + y - z = 1$$
$$x - 2y + 3z = 6$$
$$x + y - 2z = -3$$

Solución numérica de sistemas de ecuaciones lineales

Método de Gauss-Jordan

Por el método de Gauss se van obteniendo soluciones mediante la reducción del sistema dado a otro equivalente en el que cada ecuación tiene una incógnita menos que la anterior.

$$\begin{bmatrix} 2 & 1 & -1 \\ 1 & -2 & 3 \\ 1 & 1 & -2 \end{bmatrix} = \begin{bmatrix} 1 \\ 6 \\ -3 \end{bmatrix} \Rightarrow \begin{bmatrix} 2 & 1 & -1 \\ 1 & -2 & 3 \\ 0 & 3 & -5 \end{bmatrix} = \begin{bmatrix} 1 \\ 6 \\ -9 \end{bmatrix}$$

$$(r_3 - r_2) \rightarrow$$

Solución numérica de sistemas de ecuaciones lineales

Método de Gauss-Jordan

El método de Gauss transforma la matriz de coeficientes en una matriz triangular superior, por medio del pivoteo.

$$\begin{bmatrix} 2 & 1 & -1 \\ 0 & 5 & 7 \\ 0 & 0 & 4/3 \end{bmatrix} = \begin{bmatrix} 1 \\ -11 \\ 4 \end{bmatrix}$$

Solución numérica de sistemas de ecuaciones lineales

Método de Gauss-Jordan

Después se resuelve por el llamado método de ***sustitución hacia atrás***, es decir en la última fila se puede encontrar fácilmente el valor de Z, conocido ese valor se puede ir al segunda fila para conocer Y, finalmente conocido Y se puede ir al primera fila para conocer X

$$\begin{bmatrix} 2 & 1 & -1 \\ 0 & 5 & 7 \\ 0 & 0 & 4/3 \end{bmatrix} = \begin{bmatrix} 1 \\ -11 \\ 4 \end{bmatrix}$$

$$2X+(2)-(3)=1 \quad \Rightarrow \quad X=(1-2+3)/2=1$$

$$5Y-7(3)=-11 \quad \Rightarrow \quad Y=(-11+21)/5=2$$

$$(4/3)Z=4 \quad \Rightarrow \quad Z=3$$

Solución numérica de sistemas de ecuaciones lineales

Método de Gauss-Jordan

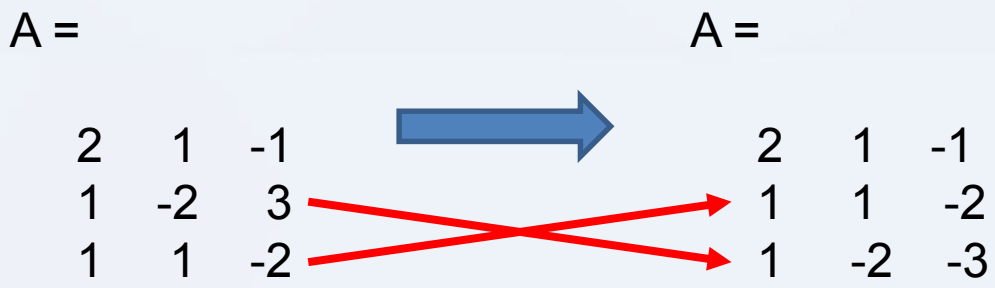
Otra forma de resolver el sistema es seguir transformándola hasta tener una matriz identidad donde los valores de las variables serán igual a los elementos de el vector B modificado.

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} \Rightarrow \begin{aligned} X &= 1 \\ Y &= 2 \\ Z &= 3 \end{aligned}$$

Solución numérica de sistemas de ecuaciones lineales

Método de Gauss-Jordan

Intercambiar elementos de una fila a otra



Solución numérica de sistemas de ecuaciones lineales

Método de Gauss-Jordan

Ejemplo de cambiar renglon

```
function Ma = cambiar_renglon(a,b,Ma)
   %a y b => Renglones que se quien cambiar de posición
   %Ma => Matriz que se quiere transformar
    V1=Ma(a,:);
    V2=Ma(b,:);
    Ma(a,:)=V2;
    Ma(b,:)=V1;
end
```

Solución numérica de sistemas de ecuaciones lineales

Método de Gauss-Jordan

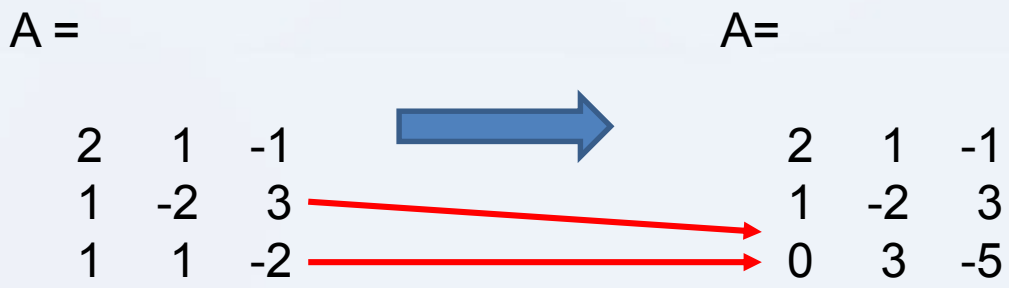
Ejemplo de cambiar columna

```
function Ma = cambiar_columna(a,b,Ma)
   %a y b => Columnas que se quieren cambiar de posición
   %Ma => Matriz que se quiere transformar
    V1=Ma(:,a);
    V2=Ma(:,b);
    Ma(:,a)=V2;
    Ma(:,b)=V1;
end
```

Solución numérica de sistemas de ecuaciones lineales

Método de Gauss-Jordan

Convertir a cero un elemento de una matriz con base en un elemento contenido en la misma columna



Solución numérica de sistemas de ecuaciones lineales

Método de Gauss-Jordan

El código de la función seria.

```
function Ma = convertir_ceros(a,b,c,Ma)
    %a => filas donde esta el pivote
    %b => fila cuyo elemento se quiere cambiar a cero
    %c => columna donde se harán las sustituciones

    Rp=Ma(a,:);
    Rc=Ma(b,:);
    if ( Ma(a,c) ~= 0 )
          Ma(b,:)=Rc+(-1)*(Ma(b,c)/Ma(a,c))*Rp;
    end
end
```

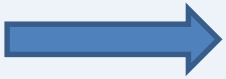
Solución numérica de sistemas de ecuaciones lineales

Método de Gauss-Jordan

Obtener la posición del máximo elemento en valor absoluto de una matriz

A =

$$\begin{matrix} 2 & 1 & -1 \\ 1 & -2 & 3 \\ 1 & 1 & -2 \end{matrix}$$



ans = 3

Solución numérica de sistemas de ecuaciones lineales

Método de Gauss-Jordan

```
function Mf =maximo_matriz(Ma )
   Mf=zeros(1,3);
   n=size(Ma,1);
   vmax=0;     fmax=0; cmax=0;
   for i=1:n
      for j=1:n

         if (abs(Ma(i,j)) > abs(vmax))

            vmax=Ma(i,j);
            fmax=i;
            cmax=j;
         end
      end
   end
   Mf=[vmax fmax cmax];
end
```

Debe ser modificado para pedir el máximo de una submatriz

A =

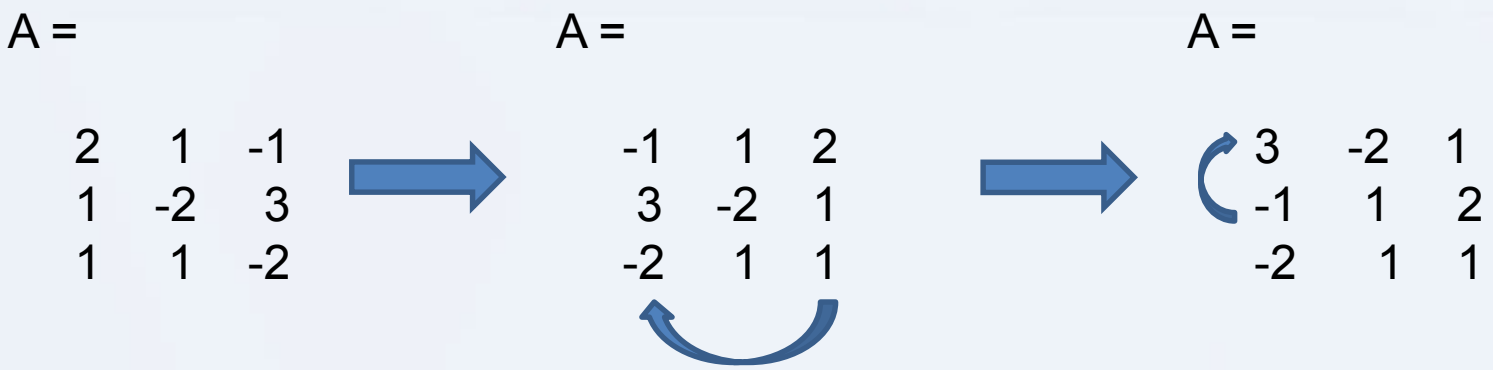
2	1	-1
1	-2	3
1	1	-2

Solución numérica de sistemas de ecuaciones lineales

Método de Gauss-Jordan

Con base en un elemento pivote, acomodar dicho elemento en cierta posición moviendo filas y columnas.

Ejemplo: El elemento de máximo valor absoluto colocarlo en la celda (1,1)



Solución numérica de sistemas de ecuaciones lineales

Método de Gauss-Jordan

```
function Mf = acomodar_maximo( a,b,vmax,Ma )
    Mf=Ma;
    Mf=cambiar_fila(vmax(2),a,Ma);
    Mf=cambiar_columna(vmax(3),b,Mf);
 end
```

Solución numérica de sistemas de ecuaciones lineales

Método de Gauss-Jordan

Si al aplicar este método en cada transformación se van utilizando **números racionales** el **error** por la aplicación del método será ***cero***. Pero si utilizamos números decimales se puede presentar un error debido a **redondeo** o ***truncamiento***.

Racional	2/3
Decimal a 5 digitos	0.66667
Redondeo	0.667
Truncamiento	0.666

Solución numérica de sistemas de ecuaciones lineales

Método de Gauss-Jordan por estrategias de pivoteo total

Para disminuir dicho error se puede aplicar una variante al método de Gauss_Jordan llamado ***pivoteo total.***

Esta metodología se puede considerar un ***Gauss-Jordan mejorado***, que al tomar como pivote el máximo elemento en valor absoluto de una matriz o submatriz y colocarlos sobre la diagonal principal, nos permite disminuir los errores.

Nota: Existen otras técnicas de pivoteo el parcial y el parcial escalonado.

Solución numérica de sistemas de ecuaciones lineales

Método de Gauss-Jordan por pivoteo total

La idea de este método consiste en lo siguiente:

Analizamos los elementos de la matriz y nos enfocamos en el que tenga mayor valor absoluto.

$$A = \left[\begin{array}{ccc|c} 8 & 1 & 6 & 4 \\ 3 & 5 & 7 & 1 \\ 4 & 9 & 2 & 7 \end{array}\right]$$

En este caso el valor seleccionado es 4 que se encuentra en la celda (3,2)

No debe ser tomada en cuenta en el análisis de máximo valor

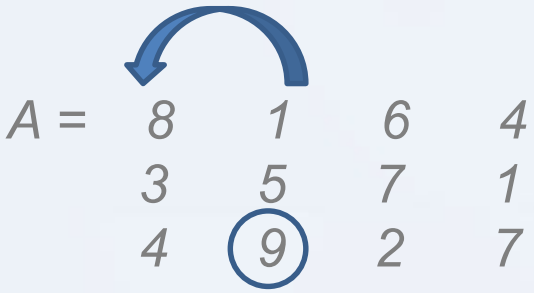
$$A\left(1 \le i \le \mathrm{n}, 1 \le j \le \mathrm{n}\right)$$

$$A\left(1 \le i \le 3, 1 \le j \le 3\right)$$

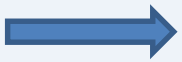
Solución numérica de sistemas de ecuaciones lineales

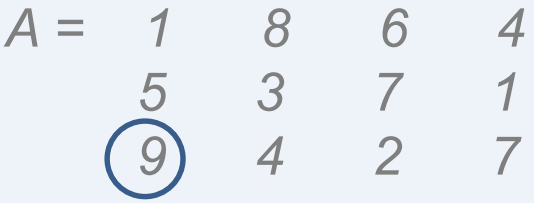
Método de Gauss-Jordan por pivoteo total

Posteriormente debemos intercambiar las filas y columnas de tal forma que la fila que contenga el (pivote) se ubique en la parte superior de la matriz y sobre la diagonal principal.




```
>>C=cambiar_columna(2,1,A)
```





$$A_{C1} \Leftrightarrow A_{C2}$$

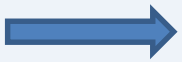
Solución numérica de sistemas de ecuaciones lineales

Método de Gauss-Jordan por pivoteo total

Finalmente se debe realizar un cambio de renglones para colocar el elemento de mayor valor absoluto en la celda (1,1)

$$A = \begin{matrix} 1 & 8 & 6 & 4 \\ 5 & 3 & 7 & 1 \\ \textcircled{9} & 4 & 2 & 7 \end{matrix}$$

>>C=cambiar_renglon (1,3,A)



$$A = \begin{matrix} \textcircled{9} & 4 & 2 & 7 \\ 5 & 3 & 7 & 1 \\ 1 & 8 & 6 & 4 \end{matrix}$$

$$A_{R1} \Leftrightarrow A_{R2}$$

Solución numérica de sistemas de ecuaciones lineales

Método de Gauss-Jordan por pivoteo total

Una vez acomodada la matriz se deben eliminar los elementos que están por debajo de ella como en el Gauss Jordan tradicional.

$$A = \begin{matrix} \textcircled{9} & 4 & 2 & 7 \\ 5 & 3 & 7 & 1 \\ 1 & 8 & 6 & 4 \end{matrix}$$

```
>>A=convertir_ceros(1,2,1,A)
>>A=convertir_ceros(1,3,1,A)
```

```
A =

    9.00000   4.00000   2.00000   7.00000
    0.00000   0.77778   5.88889  -2.88889
    0.00000   7.55556   5.77778   3.22222
```

Solución numérica de sistemas de ecuaciones lineales

Método de Gauss-Jordan por pivoteo total

En Matlab se puede dar formato de fracciones a los resultados de la siguiente forma:

```
>>format rat;
>> A
```

```
A =

    9     4     2     7
    0   7/9  53/9 -26/9
    0  68/9  52/9  29/9
```

Format short nos permite regresar al formato de 4 decimales

Solución numérica de sistemas de ecuaciones lineales

Método de Gauss-Jordan por pivoteo total

Posteriormente se vuelve a repetir el proceso a partir del segundo renglón buscando el elemento de máximo valor absoluto sea colocado sobre la diagonal principal en la celda (2,2).

$A =$

9.00000	4.00000	2.00000	7.00000
0.00000	0.77778	5.88889	-2.88889
0.00000	7.55556	5.77778	3.22222

No se deben tomar en cuenta en el análisis el renglón y columna 1

$$A\left(2 \le i \le n, 2 \le j \le n\right)$$

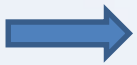
$$A\left(2 \le i \le 3, 2 \le j \le 3\right)$$

Solución numérica de sistemas de ecuaciones lineales

Método de Gauss-Jordan por pivoteo total

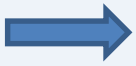
Una vez acomodada la matriz se deben eliminar los elementos que están por debajo de ella como en el Gauss-Jordan tradicional.

```
>>C=cambiar_columna(2,1,A)
```



```
A =
   9.00000   4.00000   2.00000   7.00000
   0.00000   0.77778   5.88889  -2.88889
   0.00000   7.55556   5.77778   3.22222
```

```
>>A=convertir_ceros(1,3,1,A)
```



```
A =
   9.00000   2.00000   4.00000   7.00000
   0.00000   5.88889   0.77778  -2.88889
   0.00000   0.00000   6.79245   6.05660
```

Para nuestro caso ya se triangulo la matriz y se puede aplicar la sustitución hacia atrás ó seguir manipulando la matriz hasta obtener la matriz identidad del lado izquierdo.

Solución numérica de sistemas de ecuaciones lineales

Método de Gauss-Jordan por pivoteo total

La solución para nuestro ejemplo es:

$A =$

$$\begin{matrix} 9.00000 & 2.00000 & 4.00000 & 7.00000 \\ 0.00000 & 5.88889 & 0.77778 & -2.88889 \\ 0.00000 & 0.00000 & 6.79245 & 6.05660 \end{matrix}$$

$$Z = 6.05660 / 6.79245 = 0.89167$$

$$Y = -2.88889 - 0.77778(0.89167) = -0.60833$$

$$X = 7 - 2(0.89167) - 4(-0.60833) = 0.51667$$

Solución numérica de sistemas de ecuaciones lineales

Método de Gauss-Jordan por pivoteo total

Si se sigue manipulando la matriz para encontrar la identidad tendríamos lo siguiente:

```
>> A=convertir_ceros(2,1,2,A)
>> A=convertir_ceros(3,2,3,A)
>> A=convertir_ceros(3,1,3,A)
>> A(1,:)=A(1,:)/A(1,1)
>> A(2,:)=A(2,:)/A(2,2)
>> A(3,:)=A(3,:)/A(3,3)
```

```
A =
   1.00000   0.00000   0.00000   0.51667
   0.00000   1.00000   0.00000  -0.60833
   0.00000   0.00000   1.00000   0.89167
```

Solución numérica de sistemas de ecuaciones lineales

Método de Gauss-Jordan por pivoteo total

Tarea: Aplicando Gauss-Jordan con Pivote total encontrar la solución del siguiente sistema de ecuaciones lineales.

$$w+3x-4y+3z=-4$$
$$7w-8x+3y+2z=0$$
$$w+3x+4y+5z=-10$$
$$2x+y+7z=-19$$

Se deben poner en la tarea todas las transformaciones realizadas

Solución numérica de sistemas de ecuaciones lineales

Métodos de descomposición LU (Crout y Doolittle)

La descomposición LU consiste en encontrar dos matrices L y U construidas de tal forma que se cumpla que:

$$A=L*U$$

Este método nos permite reducir las iteraciones y errores ocasionados por el pivoteo en el método de ***Gauss-Jordan*** y Adicionalmente puede ser usado para otros procesos como la obtención de la ***inversa***.

Solución numérica de sistemas de ecuaciones lineales

Métodos de descomposición LU

Existen dos estructuras o versiones que deben tener las matrices que desean encontrarse.

Versión Crout

$$L = \begin{bmatrix} L_{11} & 0 & 0 \\ L_{21} & L_{22} & 0 \\ L_{31} & L_{32} & L_{33} \end{bmatrix}$$

$$U = \begin{bmatrix} 1 & U_{12} & U_{13} \\ 0 & 1 & U_{23} \\ 0 & 0 & 1 \end{bmatrix}$$

Versión Doolitle

$$L = \begin{bmatrix} 1 & L_{12} & L_{13} \\ 0 & 1 & L_{23} \\ 0 & 0 & 1 \end{bmatrix}$$

$$U = \begin{bmatrix} U_{11} & 0 & 0 \\ U_{21} & U_{22} & 0 \\ U_{31} & U_{32} & U_{33} \end{bmatrix}$$

Se puede observar que las matrices se intercambian al cambiar de versión

Solución numérica de sistemas de ecuaciones lineales

Métodos de descomposición LU

Construir la factorización LU de la matriz A utilizando la definición de Crout.

$$A = \begin{bmatrix} -1 & 3 & 2 \\ 3 & -4 & 1 \\ 2 & 5 & -2 \end{bmatrix} \Rightarrow A = LU = \begin{bmatrix} -1 & 3 & 2 \\ 3 & -4 & 1 \\ 2 & 5 & -2 \end{bmatrix}$$

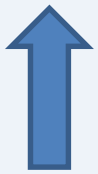
Se buscan encontrar dos matrices triangulares con las características indicadas anteriormente que al multiplicarse nos den la matriz A

Solución numérica de sistemas de ecuaciones lineales

Métodos de descomposición LU

El método parte de una primera aproximación al definir la matriz A como:
A=LU donde L=I (matriz identidad) y U=A

$$A = LU = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -1 & 3 & 2 \\ 3 & -4 & 1 \\ 2 & 5 & -2 \end{bmatrix}$$



Esas matrices pueden ser manipuladas para ir poco a poco transformándose en las matrices L y U definitivas

Solución numérica de sistemas de ecuaciones lineales

Métodos de descomposición LU

La matriz U no esta como se pide en la definición Crout (debe ser una matriz triangular superior) por lo cual se deben eliminar U_{21}, U_{31} U_{32} y convertir la diagonal principal en unos.

$$A = LU = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -1 & 3 & 2 \\ 3 & -4 & 1 \\ 2 & 5 & -2 \end{bmatrix}$$

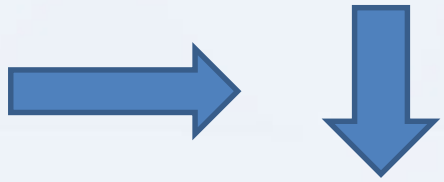
$$L = \begin{bmatrix} L_{11} & 0 & 0 \\ L_{21} & L_{22} & 0 \\ L_{31} & L_{32} & L_{33} \end{bmatrix} \qquad U = \begin{bmatrix} 1 & U_{12} & U_{13} \\ 0 & 1 & U_{23} \\ 0 & 0 & 1 \end{bmatrix}$$

Solución numérica de sistemas de ecuaciones lineales

Métodos de descomposición LU

La transformación se hace utilizando dos métodos. La matriz U se transforma utilizando el método tradicional de Gauss Jordán (convirtiendo elementos a cero por renglón), Pero la matriz L transforma diferente. Utiliza el valor negativo del pivote utilizado para la transformación U, pero aplicando las operaciones a las columnas).

$$A = LU = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -1 & 3 & 2 \\ 3 & -4 & 1 \\ 2 & 5 & -2 \end{bmatrix}$$



L **U**

$$U \Rightarrow \quad R_C = R_C - (Factor) * R_P$$

$$L \Rightarrow \quad C_P = C_P + (Factor) * C_C$$

P=Renglon Pivote

C=Renglon a Cambiar

Solución numérica de sistemas de ecuaciones lineales

Métodos de descomposición LU

Para convertir en cero el primer elemento del segundo renglón, tenemos:

$$A = LU = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -1 & 3 & 2 \\ 3 & -4 & 1 \\ 2 & 5 & -2 \end{bmatrix}$$

L ***U***

Renglon Pivote=1
Renglon a cambiar=2

$$U \Rightarrow R_2 = R_2 - (-3) * R_1 \qquad L \Rightarrow C_1 = C_1 + (-3) * C_2$$

$$\text{Factor LU} = U(2,1) / U(1,1) = (3 / (-1)) = -3$$

Solución numérica de sistemas de ecuaciones lineales

Métodos de descomposición LU

Transformando las formulas a formato Matlab:

$$\text{FactorLU} = \frac{\text{U}(2,1)}{\text{U}(1,1)} = \frac{3}{-1} = -3$$

$$\text{U} \Rightarrow \text{R}_2 = \text{R}_2 - (-3) * \text{R}_1 \Rightarrow U(2,:) = \text{U}(2,:) - factorLU * \text{U}(1,:)$$

$$\text{L} \Rightarrow \text{C}_1 = \text{C}_1 + (-3) * \text{C}_2 \Rightarrow L(:,1) = L(:,1) + factorLU * \text{L}(:,2)$$

Solución numérica de sistemas de ecuaciones lineales

Métodos de descomposición LU

$$U(2,:) = [3,-4,1] - (-3)[-1,3,2] = [0,5,7]$$

$$L(:,1) = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + (-3)\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ -3 \\ 0 \end{bmatrix}$$

$$A = LU = \begin{bmatrix} 1 & 0 & 0 \\ -3 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} -1 & 3 & 2 \\ 0 & 5 & 7 \\ 2 & 5 & -2 \end{bmatrix}$$

Solución numérica de sistemas de ecuaciones lineales

Métodos de descomposición LU

Modificando el valor del elemento U(3,1) a cero: $\text{FactorLU} = \frac{\text{U}(3,1)}{\text{U}(1,1)} = \frac{2}{-1} = -2$

$$U(3,:) = [2,5,-2] - (-2)[-1,3,2] = [0,11,2]$$

$$L(:,1) = \begin{bmatrix} 1 \\ -3 \\ 0 \end{bmatrix} + (-2)\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ -3 \\ -2 \end{bmatrix}$$

$$A = LU = \begin{bmatrix} 1 & 0 & 0 \\ -3 & 1 & 0 \\ -2 & 0 & 1 \end{bmatrix}\begin{bmatrix} -1 & 3 & 2 \\ 0 & 5 & 7 \\ 0 & 11 & 2 \end{bmatrix}$$

Solución numérica de sistemas de ecuaciones lineales

Métodos de descomposición LU

Modificando el valor del elemento U(3,2) a cero:

$$\text{FactorLU} = \frac{U(3,2)}{U(2,2)} = \frac{11}{5}$$

$$U(3,:) = [0,11,2] - (11/5)[0,5,7] = [0,0,-67/5]$$

$$L(:,2) = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} + (11/5) \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 11/5 \end{bmatrix}$$

$$A = LU = \begin{bmatrix} 1 & 0 & 0 \\ -3 & 1 & 0 \\ -2 & 11/5 & 1 \end{bmatrix} \begin{bmatrix} -1 & 3 & 2 \\ 0 & 5 & 7 \\ 0 & 0 & -67/5 \end{bmatrix}$$

Solución numérica de sistemas de ecuaciones lineales

Métodos de descomposición LU

Lo siguiente que nos faltaría seria ***convertir en unos*** los elementos de la diagonal de U. Para la matriz U para renglón se debe dividir dicho reglón entre el valor asignado a la diagonal principal que le corresponda. Para la matriz L, en lugar de dividir se debe multiplicar cada una de las columnas por los valores encontrados en la diagonal de U.

$$U = \begin{bmatrix} -1 \\ 5 \\ -67/5 \end{bmatrix} \rightarrow \begin{bmatrix} -1 & 3 & 2 \\ 0 & 5 & 7 \\ 0 & 0 & -67/5 \end{bmatrix}$$

$$\begin{bmatrix} -1 & 5 & -67/5 \end{bmatrix}$$

$$L = \begin{bmatrix} 1 & 0 & 0 \\ -3 & 1 & 0 \\ -2 & 11/5 & 1 \end{bmatrix}$$

Solución numérica de sistemas de ecuaciones lineales

Métodos de descomposición LU

Aplicando las operaciones correspondientes tenemos:

$$L = \begin{bmatrix} -1 & 0 & 0 \\ 3 & 5 & 0 \\ 2 & 11 & -67/5 \end{bmatrix} \qquad U = \begin{bmatrix} 1 & -3 & -2 \\ 0 & 1 & 7/5 \\ 0 & 0 & 1 \end{bmatrix}$$

Solución numérica de sistemas de ecuaciones lineales

Métodos de descomposición LU

Ejemplo: Transformación LU para un sistema de 3x3.

```
function [L,U]=LUpasos(A)
    n=size(A,2);        U=A;        L=eye(n);
    factor=U(2,1)/U(1,1);
    U(2,:)=U(2,:)-factor*U(1,:);
    L(:,1)=L(:,1)+factor*L(:,2);
    factor=U(3,1)/U(1,1)
    U(3,:)=U(3,:)-factor*U(1,:);
    L(:,1)=L(:,1)+factor*L(:,3);
    factor=U(3,2)/U(2,2)
    U(3,:)=U(3,:)-factor*U(2,:);
    L(:,2)=L(:,2)+factor*L(:,3);
    factorR1=U(1,1);
    U(1,:)=U(1,:)/factorR1;
    L(:,1)=L(:,1)*factorR1;
    factorR1=U(2,2);
    U(1,:)=U(2,:)/factorR1;
    L(:,1)=L(:,2)*factorR1;
    factorR1=U(3,3);
    U(3,:)=U(3,:)/factorR1;
    L(:,3)=L(:,3)*factorR1;
end
```

Solución numérica de sistemas de ecuaciones lineales

Métodos de descomposición LU

Tarea: Obtener las matrices L y U que son producto de la descomposición de la siguiente matriz.

Ma =

$$\begin{bmatrix} 1 & 3 & -4 & 3 \\ 7 & -8 & 3 & 2 \\ 1 & 3 & 4 & 5 \\ 0 & 2 & 1 & 7 \end{bmatrix}$$

Se deben poner en la tarea todas las transformaciones realizadas

Solución numérica de sistemas de ecuaciones lineales

Solución de sistemas lineales por medio de A=LU

Sea el sistema:

$$(A)(x) = b$$

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} b_{11} \\ b_{21} \\ b_{31} \end{bmatrix} \Rightarrow (A)(x) = (L)(U)(\mathrm{x}) = (L)\left[(U)(\mathrm{x})\right]$$

$$A = LU = (L)(U)$$

Solución numérica de sistemas de ecuaciones lineales

Solución de sistemas lineales por medio de A=LU

Una posible estrategia de solución consiste en: Crear una matriz Y de apoyo, que servirá como intermediaria para posteriormente calcular los valores de las X.

$$\text{Si} \quad y=(\mathrm{U})(\mathrm{x}) \quad \Longrightarrow \quad (A)(x)=(L)(y)=b$$

$$(L)(y)=b \quad \Longrightarrow \quad \begin{bmatrix} 1 & 0 & 0 \\ L_{21} & 1 & 0 \\ L_{31} & L_{32} & 1 \end{bmatrix} \begin{bmatrix} y_{11} \\ y_{21} \\ y_{31} \end{bmatrix} = \begin{bmatrix} b_{11} \\ b_{21} \\ b_{31} \end{bmatrix}$$

Solución numérica de sistemas de ecuaciones lineales

Solución de sistemas lineales por medio de A=LU

Una vez con los valores encontrados los valores de la ***matriz y***, las incógnitas del sistema inicial se resuelven con sustitución hacia atrás:

Se habia propuesto anteriormente que $y = (\mathrm{U})(\mathrm{x})$

$$(\mathrm{U})(\mathrm{x}) = \mathrm{y} \quad \Rightarrow \quad \begin{bmatrix} U_{11} & U_{12} & U_{13} \\ 0 & U_{22} & U_{23} \\ 0 & 0 & U_{33} \end{bmatrix} \begin{bmatrix} x_{11} \\ x_{21} \\ x_{31} \end{bmatrix} = \begin{bmatrix} y_{11} \\ y_{21} \\ y_{31} \end{bmatrix}$$

Solución numérica de sistemas de ecuaciones lineales

Solución de sistemas lineales por medio de A=LU

Ejemplo:

$$\begin{bmatrix} 1 & 0 & 0 \\ -1 & 5 & 0 \\ 2 & 3 & 3 \end{bmatrix}\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} 2 \\ 8 \\ 7 \end{bmatrix}$$

$$y = \begin{bmatrix} 2 \\ 2 \\ 3 \end{bmatrix}$$

$$y(1) = \left(2 - \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \right) / 1 = 2$$

$$y(2) = \left(8 - \begin{bmatrix} -1 & 5 & 0 \end{bmatrix}\begin{bmatrix} 2 \\ 0 \\ 0 \end{bmatrix} \right) / 5 = 2$$

$$y(3) = \left(7 - \begin{bmatrix} 2 & 3 & 3 \end{bmatrix}\begin{bmatrix} 2 \\ 10 \\ 0 \end{bmatrix} \right) / 3 = 3$$

Solución numérica de sistemas de ecuaciones lineales

Solución de sistemas lineales por medio de A=LU

Programa para sustitución hacia adelante (se debe tener una matriz diagonal inferior)

```
function resultados = sustitucion_hacia_adelante(Ma,B)
    n=size(Ma,1);
    resultados=zeros(n,1);
    for i=1:n
        resultados(i)=   (B(i)-Ma(i,:)*resultados) / Ma(i,i);
    end
end
```

Solución numérica de sistemas de ecuaciones lineales

Solución de sistemas lineales por medio de A=LU

Ejemplo:

$$\begin{bmatrix} 1 & 2 & -1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 2 \\ 2 \\ 3 \end{bmatrix}$$

$$x = \begin{bmatrix} 2 \\ 2 \\ 3 \end{bmatrix}$$

$$x(3) = \left(3 - \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \right) / 1 = 3$$

$$x(2) = \left(2 - \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}\begin{bmatrix} 0 \\ 0 \\ 3 \end{bmatrix} \right) / 1 = 2$$

$$x(1) = \left(2 - \begin{bmatrix} 1 & 2 & -1 \end{bmatrix}\begin{bmatrix} 0 \\ 2 \\ 3 \end{bmatrix} \right) / 1 = 1$$

Solución numérica de sistemas de ecuaciones lineales

Solución de sistemas lineales por medio de A=LU

Ejemplo: Use la factorización LU, para obtener la solución del siguiente sistema:

$$Ax = b \Rightarrow \begin{bmatrix} 1 & 2 & -1 \\ -1 & 3 & 1 \\ 2 & 1 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 2 \\ 8 \\ 7 \end{bmatrix}$$

```
>> A=[1 2 -1; -1 3 1 ;2 1 1 ];
>> b=[2;8;7];
>> [L,U]=metodoLU(A)
>> y = sustitucion_hacia_adelante(L,b);
>> x=sustitucion_hacia_atras(U,y);
```

Solución numérica de sistemas de ecuaciones lineales

Solución de sistemas lineales por medio de A=LU

Tarea: Aplicando transformaciones LU, obtener la solución del siguiente sistema.

$$
\begin{aligned}
w + 3x - 4y + 3z &= -4 \\
7w - 8x + 3y + 2z &= 0 \\
w + 3x + 4y + 5z &= -10 \\
2x + y + 7z &= -19
\end{aligned}
$$

Se deben poner en la tarea todas las transformaciones realizadas

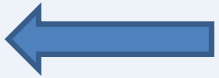
Solución numérica de sistemas de ecuaciones lineales

Método de Jacobi

Este método nos permite resolver sistemas de ecuaciones con base en una ecuación de recurrencia y una aproximación inicial.

$$Ax = B$$

$$x_{i+1} = c + Bx_i$$



Ecuación de recurrencia

X Son las incógnitas que deben asignarse en una aproximación inicial

El método terminara cuando: $x_{i+1} \approx x_i$

Solución numérica de sistemas de ecuaciones lineales

Método de Jacobi

Ejemplo: Resolver por medio del método de ***Jacobi*** el siguiente sistema de ecuaciones lineales con una tolerancia de 0.001 unidades.

$$Ax = B \longrightarrow \begin{bmatrix} 3 & 1 & 1 \\ -1 & 4 & 1 \\ -2 & 1 & 4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 8 \\ 10 \\ 12 \end{bmatrix}$$

Partiendo de la aproximación inicial de: $x_1 = 0, \quad x_2 = 0, \quad x_3 = 0$

Solución numérica de sistemas de ecuaciones lineales

Método de Jacobi

El primer paso es realizar la multiplicación de A por X e igualarlo a B.

$$\begin{bmatrix} 3 & 1 & 1 \\ -1 & 4 & 1 \\ -2 & 1 & 4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 8 \\ 10 \\ 12 \end{bmatrix}$$

$$3x_1^0 + x_2^0 + x_3^0 = 8$$

$$-x_1^0 + 4x_2^0 + x_3^0 = 10$$

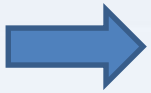
$$-2x_1^0 + x_2^0 + 4x_3^0 = 12$$

Solución numérica de sistemas de ecuaciones lineales

Método de Jacobi

El segundo paso es despejar en cada una de las 3 ecuaciones la variable que corresponda al número de ecuación:

$$3x_1^0 + x_2^0 + x_3^0 = 8$$
$$-x_1^0 + 4x_2^0 + x_3^0 = 10$$
$$-2x_1^0 + x_2^0 + 4x_3^0 = 12$$



$$x_1^0 = \frac{1}{3}\left(8 - x_2^0 - x_3^0\right)$$
$$x_2^0 = \frac{1}{4}\left(10 + x_1^0 - x_3^0\right)$$
$$x_3^0 = \frac{1}{4}\left(12 + 2x_1^0 - x_2^0\right)$$

Solución numérica de sistemas de ecuaciones lineales

Método de Jacobi

El tercer paso es cambiar el índice de las variables despejadas por el valor correspondiente a i+1 que para nuestro ejemplo seria 0+1=1.

$$x_1^1 = \frac{1}{3}\left(8 - x_2^0 - x_3^0\right)$$

$$x_2^1 = \frac{1}{4}\left(10 + x_1^0 - x_3^0\right)$$

$$x_3^1 = \frac{1}{4}\left(12 + 2x_1^0 - x_2^0\right)$$

Estas serán finalmente las ecuaciones de recurrencia

Solución numérica de sistemas de ecuaciones lineales

Método de Jacobi

Asignando los valores iniciales tenemos: $x_1^0 = x_2^0 = x_3^0 = 0$

$$x_1^1 = \frac{1}{3}\left(8 - x_2^0 - x_3^0\right) = \frac{1}{3}\left(8 - 0 - 0\right) = \frac{8}{3}$$

$$x_2^1 = \frac{1}{4}\left(10 + x_1^0 - x_3^0\right) = \frac{1}{4}\left(10 + 0 - 0\right) = \frac{10}{4}$$

$$x_3^1 = \frac{1}{4}\left(12 + 2x_1^0 - x_2^0\right) = \frac{1}{4}\left(12 + 2\left(0\right) - \left(0\right)\right) = 3$$

Solución numérica de sistemas de ecuaciones lineales

Método de Jacobi

Al comparar valores vemos que no cumple con las tolerancias.

$$\left|x_1^0 - x_1^1\right| \geq 0.001 \quad \Rightarrow \quad \left|8/3 - 0\right| \geq 0.001$$

$$\left|x_2^0 - x_2^1\right| \geq 0.001 \quad \Rightarrow \quad \left|10/4 - 0\right| \geq 0.001$$

$$\left|x_3^0 - x_3^1\right| \geq 0.001 \quad \Rightarrow \quad \left|3 - 0\right| \geq 0.001$$

Solución numérica de sistemas de ecuaciones lineales

Método de Jacobi

Al no cumplirse con las condiciones se vuelve a realizar una nueva iteración:

$$x_1^1 = 8/3 \qquad x_2^1 = 10/4 \qquad x_3^1 = 3$$

$$x_1^2 = \frac{1}{3}\left(8 - x_2^0 - x_3^0\right) = \frac{1}{3}\left(8 - (10/4) - 3\right) = 5/6$$

$$x_2^2 = \frac{1}{4}\left(10 + x_1^0 - x_3^0\right) = \frac{1}{4}\left(10 + (8/3) - 3\right) = 29/12$$

$$x_3^2 = \frac{1}{4}\left(12 + 2x_1^0 - x_2^0\right) = \frac{1}{4}\left(12 + 2(8/3) - (10/4)\right) = 89/24$$

$$|8/3 - 5/6| \le 0.001 \qquad |10/4 - 29/12| \ge 0.001 \qquad |3 - 89/24| \ge 0.001$$

Solución numérica de sistemas de ecuaciones lineales

Método de Jacobi

Después de 13 iteraciones se llega a lo siguiente:

$$x_1^{13} = 0.9999 \approx 1$$

$$x_2^{13} = 1.9999 \approx 2$$

$$x_3^{13} = 3.0000 \approx 3$$

Solución numérica de sistemas de ecuaciones lineales

Método de Jacobi

Código del método de Jacobi para el caso exclusivo de sistemas de 3x3

```
function [V1,VR]= Jacobi( A,B,Vo )
    X11=(B(1,1)-A(1,2)*Vo(2,1)-A(1,3)*Vo(3,1))/A(1,1);
    X12=(B(2,1)-A(2,1)*Vo(1,1)-A(2,3)*Vo(3,1))/A(2,2);
    X13=(B(3,1)-A(3,1)*Vo(1,1)-A(3,2)*Vo(2,1))/A(3,3);
    V1=[X11; X12 ; X13];
    VR=V1-Vo;
 end
```

Solución numérica de sistemas de ecuaciones lineales

Método de Jacobi

El método muchas veces diverge y para disminuir esa posibilidad se debe acomodar la matriz para que se convierta en una **matriz diagonalmente dominante**.

$$\left|x_{i\ j=i}\right| > \sum\left(\left|x_{ij}\right|\right),\ \text{j=1,2...n},\quad \text{i} \neq \text{j}$$

$$\begin{bmatrix} 3 & 1 & 1 \\ 2 & 8 & 5 \\ -2 & 4 & 7 \end{bmatrix}$$

Una matriz se dice **matriz diagonalmente dominante**, si en cada uno de los renglones, el valor absoluto del elemento de la diagonal principal es mayor que la suma de los valores absolutos de los elementos restantes del mismo renglón.

Solución numérica de sistemas de ecuaciones lineales

Método de Jacobi

Son diagonalmente dominantes:

$$\begin{bmatrix} 4 & 1 \\ 3 & 8 \end{bmatrix}, \begin{bmatrix} 4 & 1 & 1 \\ 2 & 8 & -3 \\ 3 & 2 & 9 \end{bmatrix}, \begin{bmatrix} -6 & 1 & 2 \\ 1 & 3 & 0 \\ 3 & 2 & -9 \end{bmatrix}$$

No son diagonalmente dominantes:

$$\begin{bmatrix} 4 & 4 \\ 3 & 8 \end{bmatrix}, \begin{bmatrix} 4 & 1 & 3 \\ 2 & 8 & 1 \\ 3 & -10 & 2 \end{bmatrix}, \begin{bmatrix} 4 & 1 & 1 \\ 2 & 8 & -7 \\ 3 & -10 & 20 \end{bmatrix}$$

Solución numérica de sistemas de ecuaciones lineales

Método de Jacobi

Por ejemplo la siguiente matriz puede volverse diagonalmente divergente, al permutar la primera y segunda columna.

$$\begin{bmatrix} 3 & 12 & -1 \\ 11 & -4 & 3 \\ -3 & -2 & -12 \end{bmatrix} \rightarrow \begin{bmatrix} 12 & 3 & -1 \\ -4 & 11 & 3 \\ -2 & -3 & -12 \end{bmatrix}$$

Solución numérica de sistemas de ecuaciones lineales

Método de Gauss-Seidel

El método de Gauss-Seidel es muy semejante al método de Jacobi. Pero mientras que en el de Jacobi se utiliza el valor de todas las incógnitas para determinar una ***nueva aproximación***, en el de Gauss-Seidel se va utilizando los valores de las incógnitas recién calculadas en la ***misma iteración***, y no en la siguiente.

$$x_1^1 = \frac{1}{3}\left(8 - x_2^0 - x_3^0\right)$$

$$x_2^1 = \frac{1}{4}\left(10 + x_1^0 - x_3^0\right)$$

$$x_3^1 = \frac{1}{4}\left(12 + 2x_1^0 - x_2^0\right)$$

Jacobi

$$x_1{}^1 = f(x_1{}^0, x_2{}^0, ..x_i{}^0)$$

$$x_2{}^1 = g(x_1{}^0, x_2{}^0, ..x_i{}^0)$$

...

$$x_i{}^1 = h(x_1{}^0, x_2{}^0, ..x_i{}^0)$$

$$x_1^1 = \frac{1}{3}\left(8 - x_2^0 - x_3^0\right)$$

$$x_2^1 = \frac{1}{4}\left(10 + x_1^1 - x_3^0\right)$$

$$x_3^1 = \frac{1}{4}\left(12 + 2x_1^1 - x_2^1\right)$$

Gauss Seidel

$$x_1{}^1 = f(x_1{}^0, x_2{}^0, ..x_i{}^0)$$

$$x_2{}^1 = g(x_1{}^1, x_2{}^0, ..x_i{}^0)$$

$$x_3{}^1 = h(x_1{}^1, x_2{}^1, ..x_i{}^0)$$

Solución numérica de sistemas de ecuaciones lineales

Método de Gauss-Seidel

Resolver por Gauss_Seidel el mismo ejercicio utilizado para ejemplificar el método del Jacobi.

$$Ax = B \longrightarrow \begin{bmatrix} 3 & 1 & 1 \\ -1 & 4 & 1 \\ -2 & 1 & 4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 8 \\ 10 \\ 12 \end{bmatrix}$$

Solución numérica de sistemas de ecuaciones lineales

Método de Gauss-Seidel

Aplicando valores iniciales:

$$x_1^{(1)} = 0 \qquad x_2^{(1)} = 0 \qquad x_3^{(1)} = 0$$

$$x_1^1 = \frac{1}{3}\left(8 - x_2^0 - x_3^0\right) = \frac{1}{3}(8 - 0 - 0) = \frac{8}{3}$$

$$x_2^1 = \frac{1}{4}\left(10 + x_1^1 - x_3^0\right) = \frac{1}{4}\left(10 + \left(\frac{8}{3}\right) - 0\right) = \frac{38}{12}$$

$$x_3^1 = \frac{1}{4}\left(12 + 2x_1^1 - x_2^1\right) = \frac{1}{4}\left(12 + 2\left(\frac{8}{3}\right) - \left(\frac{38}{12}\right)\right) = \frac{85}{24}$$

$$|8/3 - 0| \le 0.001 \qquad |38/12 - 0| \ge 0.001 \qquad |85/24 - 0| \ge 0.001$$

Solución numérica de sistemas de ecuaciones lineales

Método de Gauss-Seidel

Este método necesitaría de 9 iteraciones para encontrar la solución del sistema:

$$x_1^9 = 1.0001 \approx 1$$
$$x_2^9 = 2.0001 \approx 2$$
$$x_3^9 = 3.0000 \approx 3$$

Solución numérica de sistemas de ecuaciones lineales

Método de Gauss-Seidel

Código del método de Gauss-Seidel para el caso exclusivo de sistemas de 3x3

```
function [V1,VR]= Jacobi( A,B,Vo )
    X11=(B(1,1)-A(1,2)*Vo(2,1)-A(1,3)*Vo(3,1))/A(1,1);
    X12=(B(2,1)-A(2,1)*Vo(1,1)-A(2,3)*Vo(3,1))/A(2,2);
    X13=(B(3,1)-A(3,1)*Vo(1,1)-A(3,2)*Vo(2,1))/A(3,3);
    V1=[X11; X12 ; X13];
    VR=V1-Vo;
 end
```

Solución numérica de sistemas de ecuaciones lineales

Métodos de Krylov

Krylov creo varios métodos numéricos, dos de los mas conocidos son los relacionados con el polinomio característico y el de las potencias.

El ***Método de polinomio característico*** se utiliza para matrices cuadradas y nos permite obtener los eigenvalores característico (valores característicos, valores propios o eigenvectores) de dicha matriz.

La ecuación que define el método es la siguiente:

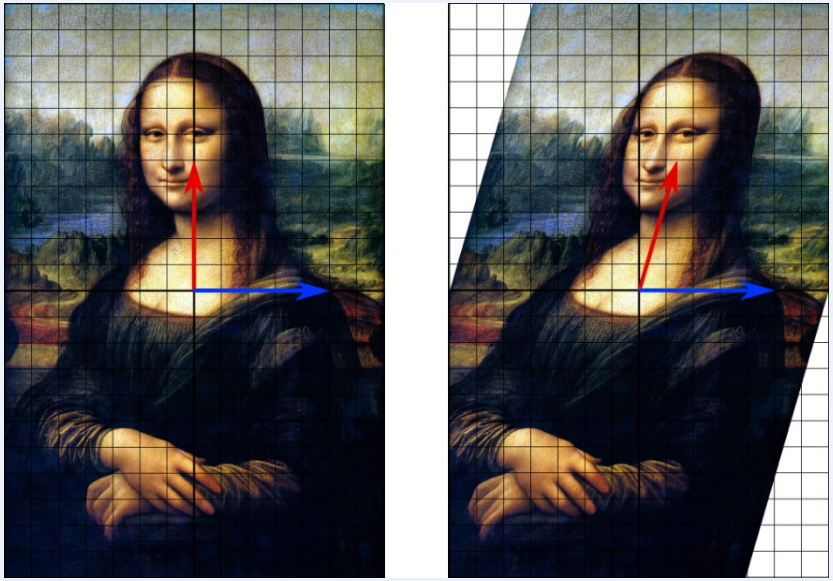
$$\left|A-\lambda I\right| = \begin{vmatrix} (a_{11}-\lambda) & a_{12} & a_{13} \\ a_{21} & (a_{22}-\lambda) & a_{23} \\ a_{31} & a_{32} & (a_{33}-\lambda) \end{vmatrix} = 0$$

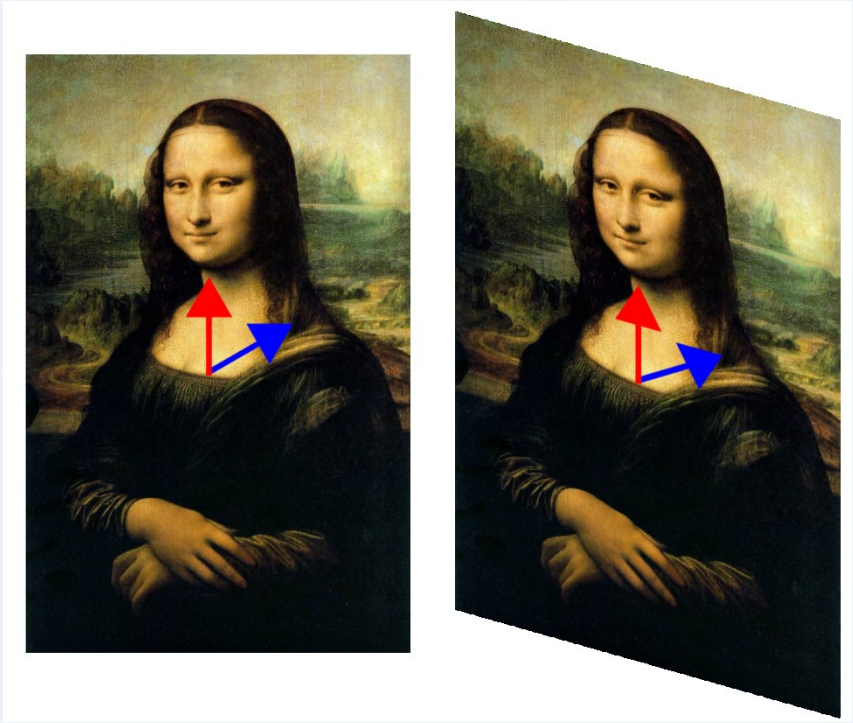
A es la matriz cuadrada, I la matriz identidad y Landa el valor característico.

Solución numérica de sistemas de ecuaciones lineales

Polinomio característico

Ejemplo de aplicación en la transformación de una imagen.





Solución numérica de sistemas de ecuaciones lineales

Polinomio característico

Ejemplo: Utilizando el método de Krylov obtener en forma manual el polinomio característico para la siguiente matriz.

$$\left|A - \lambda I\right|$$

$$A = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$

Solución numérica de sistemas de ecuaciones lineales

Polinomio característico

Aplicando el método:

$$|A-\lambda I| = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} - \lambda \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1-\lambda & 1 & 0 \\ 1 & 0-\lambda & 1 \\ 0 & 1 & 1-\lambda \end{bmatrix}$$

Solución numérica de sistemas de ecuaciones lineales

Polinomio característico

Simplificando:

$$(1-\lambda)\left[(-\lambda)(1-\lambda)-(1)(1)\right]-(1)\left[(1)(1-\lambda)-(0)(1)\right]+(0)\left[(1)(1)-(0)(-\lambda)\right]$$

$$(1-\lambda)\left(-\lambda+\lambda^2-1\right)-(1)(1-\lambda)=-\lambda+\lambda^2-1+\lambda^2-\lambda^3+\lambda-1+\lambda$$

$$p(\lambda)=-\lambda^3+2\lambda^2+\lambda-2=0$$

Solución numérica de sistemas de ecuaciones lineales

Polinomio característico

Si desarrolláramos el determinante para una matriz de 3x3:

$$\begin{aligned}\left|A-\lambda I\right| &= (a_{11}-\lambda)\left(a_{23}(a_{22}-\lambda)-a_{32}(a_{33}-\lambda)\right)\\ &\text{-}a_{12}\left(a_{21}(a_{33}-\lambda)-a_{31}a_{23}\right)\\ &+a_{13}\left(a_{21}a_{32}-a_{31}(a_{22}-\lambda)\right)\end{aligned}$$

Para simplificar podemos usar la siguiente ecuación (teniendo como incógnitas a las ***bi***).

$$\lambda^n + b_1\lambda^{n-1} + b_2\lambda^{n-2} + \ldots + b_{n-1}\lambda^{n-(n-1)} + b_n = 0$$

Solución numérica de sistemas de ecuaciones lineales

Polinomio característico

Si aplicamos el teorema de ***Cayley-Hamilton*** que establece que "Toda matriz cuadrada A satisface su ecuación característica expresada como una ecuación matricial".

$$A^n + b_1 A^{n-1} + b_2 A^{n-2} + \ldots + b_{n-1} A + b_n I = 0$$

Que es equivalente a:

$$A^n \vec{y} + b_1 A \vec{y} + b_2 A \vec{y} + \ldots + b_{n-1} A \vec{y} + b_n \vec{y} = 0$$

$\vec{y} \rightarrow$ vector cualesquiera diferente de vector nulo

Solución numérica de sistemas de ecuaciones lineales

Polinomio característico

Polinomio característico con Matlab

```
n=3;
syms x;
L=eye(n)*x;
A=magic(n);
f=det(A-L)
r=solve(f)
ezplot(f,-5,16)
```

Solución numérica de sistemas de ecuaciones lineales

Polinomio característico

Ejemplo: Si la matriz 3x3 tendríamos

$$A = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} \Longrightarrow A^3\vec{y} + \upsilon_1 A^2\vec{y} + \upsilon_2 A\vec{y} + \upsilon_3 \vec{y} = 0$$

Donde $\vec{y} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$

Solución numérica de sistemas de ecuaciones lineales

Polinomio característico

Desarrollando las matrices A elevadas a la n:

$$A=\begin{bmatrix}1 & 1 & 0\\ 1 & 0 & 1\\ 0 & 1 & 1\end{bmatrix} \quad A^2=\begin{bmatrix}2 & 1 & 1\\ 1 & 2 & 1\\ 1 & 1 & 2\end{bmatrix} \quad A^3=\begin{bmatrix}3 & 3 & 2\\ 3 & 2 & 3\\ 2 & 3 & 3\end{bmatrix}$$

Solución numérica de sistemas de ecuaciones lineales

Polinomio característico

Desarrollando la ecuación:

$$|A-\lambda I| = A^3\vec{y} + b_1 A^2\vec{y} + b_2 A\vec{y} + b_3\vec{y} = 0$$

$$|A-\lambda I| = \begin{bmatrix} 3 & 3 & 2 \\ 3 & 2 & 3 \\ 2 & 3 & 3 \end{bmatrix}\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + b_1 \begin{bmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{bmatrix}\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + b_2 = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + b_3 \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = 0$$

$$|A-\lambda I| = \begin{bmatrix} 3 \\ 3 \\ 2 \end{bmatrix} + b_1 \begin{bmatrix} 2 \\ 1 \\ 1 \end{bmatrix} + b_2 \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} + b_3 \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Solución numérica de sistemas de ecuaciones lineales

Polinomio característico

Entonces nos quedaría el siguiente sistema, que a partir de este momento puede resolverse con los métodos previos (Gauss_Jordan, Pivoteo, etc.):

$$\begin{aligned}(2)b_1+(1)b_2+(1)b_3&=-3\\(1)b_1+(1)b_2+(0)b_3&=-3\\(1)b_1+(0)b_2+(0)b_3&=-2\end{aligned}$$

$$\Rightarrow \begin{bmatrix}2&1&1\\1&1&0\\1&0&0\end{bmatrix}\begin{bmatrix}b_1\\b_2\\b_3\end{bmatrix}=\begin{bmatrix}-3\\-3\\-2\end{bmatrix}$$

$$\Downarrow$$

$$b_1=2 \qquad b_2=-1 \qquad b_3=-2$$

$$\Downarrow$$

$$\lambda^3-2\lambda^2-1\lambda^1+2=0$$

Solución numérica de sistemas de ecuaciones lineales

Polinomio característico

```
function [X,B] = metodo_potencias(Ma,y )
        n=size(Ma,1);
        X=y;
        for i=1:n-1
          A{i}=Ma^(i);
          X=[ X  A{i}*y ];
        end
        B=Ma^(n)*y*(-1);
end
```

Solución numérica de sistemas de ecuaciones lineales

Método de las potencias

Este método nos sirve para calcular los eigenvalores máximos y mínimos de una matriz real. Sólo es válido (es decir converge) cuando dicho eigenvalor es real y es único.

$$
\begin{array}{ccccccccccc}
(a_{11}-\lambda)X_1 & + & a_{12}X_2 & + & a_{13}X_3 & + & \ldots & + & a_{1n}X_n & = & b_1 \\
a_{21}X_1 & + & (a_{22}-\lambda)X_2 & + & a_{23}X_3 & + & \ldots & + & a_{2n}X_n & = & b_2 \\
a_{31}X_1 & + & a_{32}X_2 & + & (a_{33}-\lambda)X_3 & + & \ldots & + & a_{3n}X_n & = & b_3 \\
\vdots & & \vdots & & \vdots & & \vdots & & \vdots & & \vdots \\
a_{n1}X_1 & + & a_{n2}X_2 & + & a_{n3}X_3 & + & \ldots & + & (a_{nn}-\lambda)X_n & = & b_n
\end{array}
$$

$$\left|A-\lambda I\right| X = 0$$

Solución numérica de sistemas de ecuaciones lineales

Método de las potencias – Mayor Valor característico

El mayor valor característico se puede obtener al acomodar la ecuación anterior de la siguiente forma:

$$AX - \lambda X = 0 \Rightarrow AX = \lambda X$$

La formula anterior puede ser utilizada para emplear un método de aproximaciones sucesivas al presentarse de la siguiente manera:

$$AX_{(k)} = \lambda_{(k+1)} X_{(k+1)}$$

El proceso se repetirá hasta que la diferencia entre dos aproximaciones cumpla con la tolerancia preestablecida.

Solución numérica de sistemas de ecuaciones lineales

Método de las potencias – Menor Valor característico

Para este caso multiplicaremos la ecuación por la inversa:

$$AX = \lambda X \Rightarrow A^{-1}AX = A^{-1}\lambda X$$

Lo que resulta: $X = A^{-1}\lambda X$

Si dividimos entre λ : $\left(\frac{1}{\lambda}\right)X = A^{-1}\lambda\left(\frac{1}{\lambda}\right)X = A^{-1}X$

Poniendo en modo de aproximaciones sucesivas nos queda:

$$A^{-1}X_{(k)} = \left(\frac{1}{\lambda_{(k+1)}}\right)X_{(k+1)}$$

Solución numérica de sistemas de ecuaciones lineales

Método de las potencias – Mayor Valor Característico

Ejemplo: Calcular el mayor y menor valor característico (lambda) de la siguiente matriz.

$$A = \begin{bmatrix} 2 & 2 & 1 \\ 1 & 3 & 1 \\ 1 & 2 & 2 \end{bmatrix}$$

Solución numérica de sistemas de ecuaciones lineales

Método de las potencias – Mayor Valor Característico

El primer paso es definir un vector inicial diferente del vector nulo: $Xo = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$

$$AX_{(k)} = \lambda_{(k+1)} X_{(k+1)}$$

Al realizar la primera iteración para buscar el autovalor máximo:

$$AX_{(0)} = \begin{bmatrix} 2 & 2 & 1 \\ 1 & 3 & 1 \\ 1 & 2 & 2 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 2 \\ 1 \\ 1 \end{bmatrix}$$

Pivote Landa (máximo en valor absoluto)

Solución numérica de sistemas de ecuaciones lineales

Método de las potencias – Mayor Valor Característico

El elemento de mayor valor representa la primera iteración: $\lambda_{(1)} = 2$

$$AX_{(k)} = \lambda_{(k+1)} X_{(k+1)}$$

Obteniendo el vector X1 con base AXo:

$$AX_{(0)} = \begin{bmatrix} 2 \\ 1 \\ 1 \end{bmatrix} = \lambda_{(1)} X_1 \Rightarrow X_1 = \begin{bmatrix} 2 \\ 1 \\ 1 \end{bmatrix} / 2 = \begin{bmatrix} 1 \\ 0.5 \\ 0.5 \end{bmatrix} \leftarrow$$

Nuevo vector Xi

Solución numérica de sistemas de ecuaciones lineales

Método de las potencias – Mayor Valor Característico

Realizando una segunda iteración:

$$Xo = \begin{bmatrix} 1 \\ 0.5 \\ 0.5 \end{bmatrix}$$

$$AX_{(1)} = \begin{bmatrix} 2 & 2 & 1 \\ 1 & 3 & 1 \\ 1 & 2 & 2 \end{bmatrix} \begin{bmatrix} 1 \\ 0.5 \\ 0.5 \end{bmatrix} = \begin{bmatrix} 3.5 \\ 3 \\ 3 \end{bmatrix} \Rightarrow \lambda_{(2)} = 3.5$$

$$X_2 = \begin{bmatrix} 3.5 \\ 3 \\ 3 \end{bmatrix} / 3.5 = \begin{bmatrix} 1 \\ 0.8571 \\ 0.8571 \end{bmatrix}$$

$$|\lambda_2 - \lambda_1| = |3.5 - 2| = 1.5 > 0.01$$

Solución numérica de sistemas de ecuaciones lineales

Método de las potencias – Mayor Valor Característico

Este proceso se puede seguir repitiendo hasta que la diferencia entre las lambdas sea menor a la tolerancia:

	X_0		X_1		X_2		X_3		X_4		X_5
	1,00000	2,00000	1,00000	3,50000	1,00000	4,57143	1,00000	4,90625	1,00000	4,98089	1,00000
	0,00000	1,00000	0,50000	3,00000	0,85714	4,42857	0,96875	4,87500	0,99363	4,97452	0,99872
	0,00000	1,00000	0,50000	3,00000	0,85714	4,42857	0,96875	4,87500	0,99363	4,97452	0,99872
$\lambda_{(k)}$		2,00000		3,50000		4,57143		4,90625		4,98089	
Tol				1,50000		1,07143		0,33482		0,07464	

	X_6		X_7		X_8		X_9	
	4,99616	1,00000	4,99923	1,00000	4,99985	1,00000	4,99997	1,00000
	4,99488	0,99974	4,99898	0,99995	4,99980	0,99999	4,99996	1,00000
	4,99488	0,99974	4,99898	0,99995	4,99980	0,99999	4,99996	1,00000
	4,99616		4,99923		4,99985		4,99997	
$\lambda_{(k)}$	0,01527		0,00307		0,00061		0,00012	
Tol								

Solución numérica de sistemas de ecuaciones lineales

Método de las potencias – Mayor Valor Característico

En resumen tenemos: $\lambda_{Máximo} = 4.99997 \approx 5$

Su vector Asociado: $\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$

Solución numérica de sistemas de ecuaciones lineales

Método de las potencias – Mayor Valor Característico

```
function [X,Landa] = algoritmo_potencias_maximas(A,X )
        AX=A*X;
        maxValor=max(AX);
        minValor=min(AX);
        if abs(maxValor) >= abs(miniValor)
          Landa=maxValor;
        else
          Landa=miniValor;
        end
        X=AX/Landa;
end
```

Solución numérica de sistemas de ecuaciones lineales

Método de las potencias para el Menor valor Característico

Para el lambda mínimo el proceso sería similar pero cambiando de ecuación de recurrencia.

$$A^{-1}X_{(k)} = \left(\frac{1}{\lambda_{(k+1)}}\right)X_{(k+1)}$$

Con: $Xo = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$

$$A^{-1}X_{(0)} = \begin{bmatrix} 0.8 & -0.4 & -0.2 \\ -0.2 & 0.6 & -0.2 \\ -0.2 & -0.4 & 0.8 \end{bmatrix}\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0.8 \\ -0.2 \\ -0.2 \end{bmatrix}$$

Pivote Landa (máximo en valor absoluto)

Solución numérica de sistemas de ecuaciones lineales

Método de las potencias

Calculando la otra parte de la ecuación

$$A^{-1}X_{(k)} = \left(\frac{1}{\lambda_{(k+1)}}\right)X_{(k+1)}$$

$$\left(\frac{1}{\lambda_{(1)}}\right)X_1 = 0.8X_1 \Rightarrow \lambda = \left(\frac{1}{0.8}\right) = \frac{5}{4}$$

$$X_{(1)} = \begin{bmatrix} 0.8 \\ -0.2 \\ -0.2 \end{bmatrix} / 0.8 = \begin{bmatrix} 1 \\ -0.25 \\ -0.25 \end{bmatrix}$$

Solución numérica de sistemas de ecuaciones lineales

Método de las potencias

Los valores finales después de 7 iteraciones serian:

En resumen tenemos: $\lambda_{Mínimo} = 1.00002 \approx 1$

Su vector Asociado:

$$\begin{bmatrix} 0.333 \\ -0.333 \\ -0.333 \end{bmatrix} \approx \begin{bmatrix} 1/3 \\ -1/3 \\ -1/3 \end{bmatrix}$$

Solución numérica de sistemas de ecuaciones lineales

Método de las potencias – Mayor Valor Característico

```
function [X,Landa] = algoritmo_potencias_minimas(A,X )
        AX=inv(A)*X;
        maxValor=max(AX);
        minValor=min(AX);
        if abs(maxValor) >= abs(miniValor)
          Landa=1/maxValor;
        else
          Landa=1/miniValor;
        end
        X=AX*Landa;
end
```

Solución numérica de sistemas de ecuaciones lineales

Método de las potencias

Si aplicáramos el método de Krylov obtendríamos el siguiente polinomio característico y si lo graficáramos observaríamos que las raíces mínimas y máximas caen en 5 y 1 que fueron los valores obtenidos en este ejemplo:

$$-\lambda^3 + 7\lambda^2 - 11\lambda + 5 = 0$$