Facultad de Ingeniería | Universidad de Buenos Aires

95.10 | Modelación numérica

75.12 | Análisis numérico I A

95.13 | Métodos matemáticos y numéricos

Trabajo Práctico 2– Cuatrimestre 1 2022

Ley de Kirchoff: resolución numérica del sistema de ecuaciones lineales

Grupo Nº5	Marcelo Ariel Rondán	105703
	Ariana Magalí Salese D'Assaro	105558

Fecha	Correcciones / Observaciones	Docente

Calificación Final	Docente	Fecha

1. Introducción

En este trabajo se recurrirá a la utilización de métodos numéricos de resolución de sistemas de ecuaciones lineales para resolver el problema planteado en el enunciado. En este caso, el problema consiste en encontrar las corrientes de cada malla del circuito propuesto (***Figura 1***). En una primera instancia se utilizó la segunda ley de Kirchhoff, también conocida como “Ley de mallas” o “Ley de tensiones” para encontrar las seis ecuaciones que conforman el sistema que resolveremos.

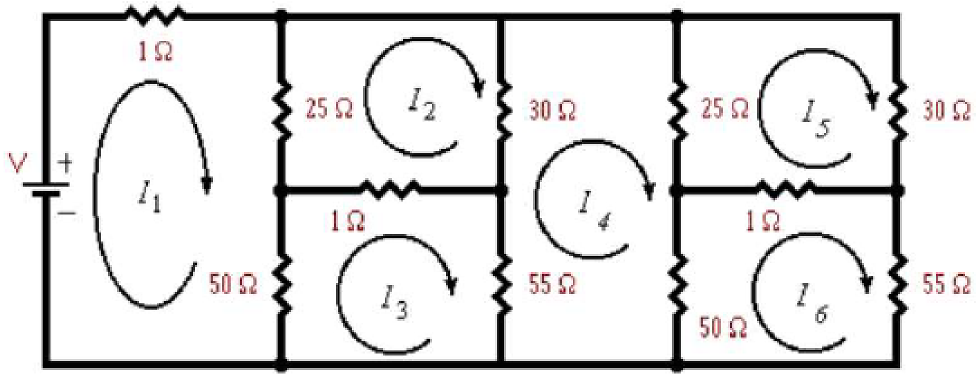

Figura 1. *Circuito propuesto en el enunciado*

El sistema de ecuaciones que resolveremos es el siguiente. Se trata de un sistema de ecuaciones linealmente independiente, con seis ecuaciones y seis incógnitas.

$$
\begin{aligned}
&1i_1 + 25(i_1 - i_2) + 50(i_1 - i_3) = V \\
&25(i_2 - i_1) + 30(i_2 - i_4) + 1(i_2 - i_3) = 0 \\
&50(i_3 - i_1) + 1(i_3 - i_2) + 55(i_3 - i_4) = 0 \\
&55(i_4 - i_3) + 30(i_4 - i_2) + 25(i_4 - i_5) + 50(i_4 - i_6) = 0 \\
&25(i_5 - i_4) + 30i_5 + 1(i_5 - i_6) = 0 \\
&50(i_6 - i_4) + 1(i_6 - i_5) + 55i_6 = 0
\end{aligned}
$$

El sistema puede ser expresado de forma matricial:

$$
A.x = b \quad \Rightarrow \quad \begin{bmatrix} 76 & -25 & -50 & 0 & 0 & 0 \\ -25 & 56 & -1 & -30 & 0 & 0 \\ -50 & -1 & 106 & -55 & 0 & 0 \\ 0 & -30 & -55 & 160 & -25 & -50 \\ 0 & 0 & 0 & -25 & 56 & -1 \\ 0 & 0 & 0 & -50 & -1 & 106 \end{bmatrix} . \begin{bmatrix} i1 \\ i2 \\ i3 \\ i4 \\ i5 \\ i6 \end{bmatrix} = \begin{bmatrix} V \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}
$$

2. Metodología

Para el desarrollo del trabajo se recurrirá a diversos métodos de resolución de ecuaciones lineales, los cuales explicaremos a continuación. También se implementó un método de aproximación de funciones que será desarrollado.

Métodos Directos para la resolución de sistemas de ecuaciones lineales

Eliminación de Gauss

Este método consiste de dos etapas, la primera siendo la triangulación de la matriz, y la segunda, la obtención del vector resultado realizando sustitución inversa sobre la matriz triangular.

La fórmula generalizada para la triangulación es:

$$m_{ik} = \frac{a_{ik}^{(k)}}{a_{kk}^{(k)}} \quad \begin{cases} 2 \leq k \leq n \\ k+1 \leq i \leq n \end{cases} \quad \begin{cases} a_{ij}^{(k+1)} = a_{ij}^{(k)} - m_{ik}\, a_{kj}^{(k)} & k+1 \leq i, j \leq n \\ b_i^{(k+1)} = b_i^{(k)} - m_{ik}\, b_k^{(k)} & k+1 \leq i \leq n \end{cases}$$

El resultado es un sistema matricial tal que

$$U = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ 0 & a_{22}^{(2)} & \dots & a_{2n}^{(2)} \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & a_{nn}^{(n)} \end{bmatrix} \quad \tilde{b} = \begin{bmatrix} b_1 \\ b_2^{(2)} \\ \dots \\ b_n^{(n)} \end{bmatrix}$$

La fórmula generalizada para la sustitución inversa es:

$$X_n = \frac{b_n}{U_{nn}} \qquad X_i = \frac{\widetilde{b_i} - \sum_{k=i+1}^{n} U_{ik} X_k}{U_{ii}} \;\; ; \;\; i = n, n-1, \dots, 1$$

El esfuerzo de cálculo necesario al llevar a cabo este método es aproximadamente:

$$\text{Triangulación} \approx \frac{n^3}{3}$$

$$\text{Sustitución inversa} \approx \frac{n^2}{2}$$

Como podemos observar, la triangulación requiere de un esfuerzo de cálculo muy superior.

En algunos casos particulares, la eliminación de Gauss puede amplificar errores, esto sucede cuando los multiplicadores conseguidos al realizar la triangulación son valores grandes. Una posible solución es el pivoteo. Que consiste en reordenar las filas (pivoteo parcial) o las filas y columnas (pivoteo completo) para obtener multiplicadores con valores bajos, y reducir el error.

Descomposición LU de Doolittle

Este método consiste en la descomposición de la matriz en otras dos matrices, una triangular superior y otra triangular inferior tal que:

$$A = L.U$$

Para luego resolver el sistema planteado utilizando sustitución directa y sustitución inversa.

La matriz U es conseguida a partir de la triangulación antes explicada, mientras que la matriz L se obtiene a partir de los multiplicadores de la misma.

$$U = \begin{bmatrix} U_{11} & U_{12} & \dots & U_{1n} \\ 0 & U_{22} & \dots & U_{2n} \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & U_{nn} \end{bmatrix} \quad L = \begin{bmatrix} 1 & 0 & \dots & 0 \\ m_{21} & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ m_{n1} & m_{n2} & \dots & 1 \end{bmatrix}$$

Las ecuaciones que se plantean para resolver el sistema consisten en

$$A.x = b \quad \Rightarrow \quad L.U.x = b \quad \begin{cases} U.x = y \\ L.y = b \end{cases}$$

Una vez planteados los sistemas de ecuaciones, solo resta realizar sustitución directa para obtener el vector y, para luego obtener el vector x realizando sustitución inversa. Es por esto que, si no tenemos en cuenta el esfuerzo necesario para obtener las matrices L y U, podemos inferir que el esfuerzo de cálculo es equivalente al de realizar ambas sustituciones.

$$\text{Sustituciones} \approx 2.\left(\frac{n^2}{2}\right) = n^2$$

Este método es conveniente cuando debemos resolver sistemas con varios términos independientes, pero con determinación encadenada a soluciones previas. Por lo cual no puede trabajarse con la matriz ampliada.

Métodos Indirectos para la resolución de sistemas de ecuaciones lineales

A partir de una estimación inicial, a la que se la denomina semilla, se aplica un algoritmo recursivo para mejorar progresivamente esa estimación.

Método de Gauss-Seidel

El método de Gauss-Seidel consiste en el despeje de una incógnita de cada ecuación y para resolver se utilizan los valores más actualizados posibles de todas las incógnitas.

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

$$1 \leq i \leq n, k = 0,1,2..$$

Siendo n la cantidad de incógnitas (cantidad de filas de la matriz) y k el número de iteración.

Si la matriz A es estrictamente diagonal dominante, es decir, que el módulo del valor en la diagonal es el mayor de cada fila, se puede asegurar la convergencia de este método.

Aproximación de funciones

Interpolación lineal

A partir de un conjunto de datos de la forma $(x_i, f(x_i))$ se determina el valor de un x tal que:

$$x_i \leq x \leq x_{i+1}$$

que pertenece al segmento de recta que pasa por $f(x_i)$ y $f(x_{i+1})$:

$$f^*(x) = \left(\frac{x_{i+1} - x}{x_{i+1} - x_i}\right).f(x_i) + \left(\frac{x - x_i}{x_{i+1} - x_i}\right).f(x_{i+1})$$

3. Resolución

Todo el código que escribimos se encuentra en el anexo para su consulta.

a) Programar la resolución del sistema de ecuaciones lineales (A.x=b) por el método directo de Eliminación Gaussiana. Asumir que el voltaje V de la batería es igual a 10.

El primer paso es realizar la triangulación, cuyo resultado es:

$$\widetilde{A} = \begin{bmatrix} 76 & -25 & -50 & 0 & 0 & 0 \\ 0 & 47.7763 & -17.4474 & -30 & 0 & 0 \\ 0 & 0 & 66.7336 & -65.9557 & 0 & 0 \\ 0 & 0 & 0 & 75.9751 & -25 & -50 \\ 0 & 0 & 0 & 0 & 47.7736 & -17.4527 \\ 0 & 0 & 0 & 0 & 0 & 66.7187 \end{bmatrix} \quad \tilde{b} = \begin{bmatrix} 10 \\ 3.28947 \\ 7.78023 \\ 9.75508 \\ 3.20996 \\ 7.59259 \end{bmatrix}$$

Luego, planteando el siguiente sistema, y realizando el método de sustitución inversa, conseguimos el resultado.

$$\widetilde{A}.x = \tilde{b} \quad \Rightarrow \quad x = \begin{bmatrix} 0.478161 \\ 0.347845 \\ 0.352881 \\ 0.239081 \\ 0.108765 \\ 0.113800 \end{bmatrix}$$

b) Programar la resolución del sistema de ecuaciones lineales con el método de factorización LU de Doolitle. Asumir que el voltaje V de la batería es igual a 10. Resolver con 3 y con 6 decimales de precisión y comparar resultados. Especificar diferencias entre los métodos.

Como se mencionó con anterioridad, la matriz U es la misma que obtuvimos mediante la triangulación de la matriz A.

$$U = \begin{bmatrix} 76 & -25 & -50 & 0 & 0 & 0 \\ 0 & 47.7763 & -17.4474 & -30 & 0 & 0 \\ 0 & 0 & 66.7336 & -65.9557 & 0 & 0 \\ 0 & 0 & 0 & 75.9751 & -25 & -50 \\ 0 & 0 & 0 & 0 & 47.7736 & -17.4527 \\ 0 & 0 & 0 & 0 & 0 & 66.7187 \end{bmatrix}$$

Por otra parte, al haber guardado los multiplicadores, podemos reconstruir la matriz L.

$$L = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ -0.328947 & 1 & 0 & 0 & 0 & 0 \\ -0.657895 & -0.365189 & 1 & 0 & 0 & 0 \\ 0 & -0.627926 & -0.988343 & 1 & 0 & 0 \\ 0 & 0 & 0 & -0.329055 & 1 & 0 \\ 0 & 0 & 0 & -0.658110 & -0.365321 & 1 \end{bmatrix}$$

Mediante una función de redondeo utilizada en cada cálculo, obtuvimos los resultados utilizando tres y seis decimales de precisión.

	3 decimales	6 decimales	Valor esperado*
i_1	0.480 A	0.487161 A	0.484532 A
i_2	0.349 A	0.347845 A	0.352480 A
i_3	0.355 A	0.352881 A	0.357582 A
i_4	0.241 A	0.239081 A	0.242266 A
i_5	0.109 A	0.108765 A	0.110214 A
i_6	0.115 A	0.113800 A	0.115316 A

*obtenido a partir de software

Podemos observar que las diferencias entre ambos resultados no son únicamente que se obtienen más decimales de precisión en el segundo cálculo, sino que además, los resultados son más exactos. Esto se debe a que no se realizó un redondeo en la etapa final, sino que todos los cálculos intermedios fueron realizados con una mayor cantidad de decimales. Por lo tanto, el error de redondeo es menor en el segundo caso.

Como ya se mencionó cuando se explicaron las metodologías, el método por descomposición LU es preferible cuando debemos resolver sistemas con varios términos independientes, pero con determinación encadenada a soluciones previas. En este caso, no nos encontramos en esa situación. Sin embargo, podemos aprovechar que ya contamos con la matriz triangular y los multiplicadores que obtuvimos en el inciso anterior.

c) Obtener mediante Eliminación Gaussiana la inversa de la matriz A del sistema (A.x=b). Asumir que el voltaje V de la batería es igual a 10. Calcular el número de condición de la matriz. Establecer conclusiones en función de los resultados anteriores.

Para obtener la inversa de la matriz A, planteamos el siguiente sistema de ecuaciones

$$A.A^{-1} = I$$

Y utilizamos la estrategia de varios términos independientes, dividiendo a la matriz I en varios vectores b. Y encontrando una columna de la matriz A^{-1} como resultado de cada cálculo.

Utilizando este método obtenemos que la matriz inversa de A es equivalente a:

$$A^{-1} = \begin{bmatrix} 0.0478161 & 0.0347846 & 0.0352880 & 0.0239080 & 0.0108764 & 0.0113800 \\ 0.0347845 & 0.0468946 & 0.0294250 & 0.0242352 & 0.0110253 & 0.0115357 \\ 0.0352881 & 0.0294252 & 0.0389252 & 0.0242226 & 0.0110195 & 0.0115297 \\ 0.0239081 & 0.0242353 & 0.0242226 & 0.0245083 & 0.0111495 & 0.0116657 \\ 0.0108765 & 0.0110253 & 0.0110195 & 0.0111495 & 0.0229324 & 0.0054755 \\ 0.0113800 & 0.0115357 & 0.0115297 & 0.0116657 & 0.0054755 & 0.0149883 \end{bmatrix}$$

Aprovechando que calculamos la inversa, el número de condición de la matriz lo calculamos como:

$$k(A) \equiv \| A \| . \| A^{-1} \| \approx 52.5$$

El número de condición de la matriz está entre uno y dos órdenes de magnitud por encima de 1, por lo que concluimos que el sistema está mal condicionado.

Hacemos este comentario, ya que, si bien el número de condición de la matriz no es bueno, observamos resultados aceptables en cuanto a exactitud utilizando métodos directos.

d) Programar la resolución del sistema de ecuaciones lineales por el método directo de Gauss Seidel. Asumir que el voltaje V de la batería es igual a 10. Proponer un criterio de corte adecuado. No utilizar la matriz de iteración para realizar los cálculos. Verificar el funcionamiento de esa metodología. Establecer conclusiones.

Como se mencionó anteriormente, si la matriz A es estrictamente diagonal dominante, el método de Gauss-Seidel convergerá a una solución. Al comienzo del trabajo práctico planteamos la matriz A con un valor de la diagonal erróneo, lo que generó que esta no sea estrictamente diagonal dominante y como consecuencia, el método divergía. Luego de encontrar y corregir el error se logró la convergencia. El resultado fue obtenido con todos los valores iniciales en 0 (semilla):

$$x = \begin{bmatrix} 0.477680 \\ 0.347435 \\ 0.352469 \\ 0.238785 \\ 0.108630 \\ 0.113659 \end{bmatrix}$$

Para la obtención del resultado se planteó que el error de truncamiento sea menor a 1.10^{-4}:

$$||x^{(k)} - x^{(k+1)}|| < 1.10^{-4}$$

Para diferentes semillas, la cantidad de iteraciones necesaria para cumplir con la condición de corte establecido varía:

Semilla	Iteraciones necesarias
[0 0 0 0 0]	40
[10 10 10 10 10]	62
[50 50 50 50 50]	71
[100 100 100 100 100]	76

A partir de los resultados podemos observar que el método converge bastante rápido, pero cuanto más alejada está la semilla del resultado final, más iteraciones se necesitarán.

e) Considerar el caso de la variación diaria del voltaje V.

Hora	*Voltaje*	*Hora*	*Voltaje*	*Hora*	*Voltaje*
0	10.00	9	10.05	18	9.80
3	10.20	12	9.95	21	10.05
6	10.30	15	9.85	24	10.15

Utilizando interpolación lineal, determinar el voltaje V según cada hora del día. Utilizar el algoritmo desarrollado para resolver el sistema con el método de Doolitle para calcular el problema para cada hora del día.

En primer lugar, se realizó la interpolación lineal, como se explicó previamente, para encontrar los valores faltantes en la tabla:

Hora	*Voltaje*	*Hora*	*Voltaje*	*Hora*	*Voltaje*	*Hora*	*Volaje*	*Hora*	*Voltaje*
0	10.00	**5**	**10.27**	**10**	**10.02**	15	9.86	**20**	**9.97**
1	**10.07**	6	10.30	**11**	**9.98**	**16**	**9.83**	21	10.05
2	**10.13**	**7**	**10.22**	12	9.95	**17**	**9.82**	**22**	**10.08**
3	10.20	**8**	**10.13**	**13**	**9.92**	18	9.80	**23**	**10.12**
4	**10.23**	9	10.05	14	9.88	**19**	**9.88**	24	10.15

Los resultados obtenidos y plasmados en la tabla podemos observarlos en el siguiente gráfico:

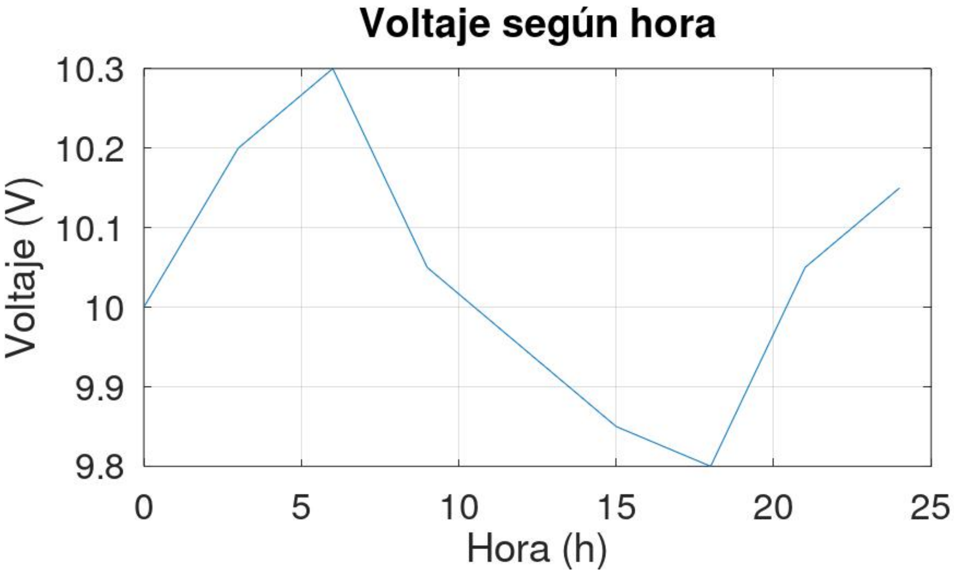

Figura 2. *Gráfico de voltaje según la hora del día*

Como se pide en el enunciado, el sistema fue resuelto para cada V a través del método LU. Este presenta una ventaja frente a eliminación de Gauss, ya que la factorización de A solo se realiza una vez y luego se realiza sustitución inversa y directa para cada uno de los términos independientes. Los resultados para cada hora fueron:

		Valor de las corrientes en amperes (A)					
		i_1	i_2	i_3	i_4	i_5	i_6
Hora (h)	0	0.478161	0.478161	0.478161	0.478161	0.478161	0.478161
	1	0.481349	0.481349	0.481349	0.481349	0.481349	0.481349
	2	0.484536	0.484536	0.484536	0.484536	0.484536	0.484536
	3	0.487722	0.487722	0.487722	0.487722	0.487722	0.487722
	4	0.489316	0.489316	0.489316	0.489316	0.489316	0.489316
	5	0.490913	0.490913	0.490913	0.490913	0.490913	0.490913
	6	0.492504	0.492504	0.492504	0.492504	0.492504	0.492504
	7	0.488522	0.488522	0.488522	0.488522	0.488522	0.488522
	8	0.484536	0.484536	0.484536	0.484536	0.484536	0.484536
	9	0.480551	0.480551	0.480551	0.480551	0.480551	0.480551
	10	0.478958	0.478958	0.478958	0.478958	0.478958	0.478958
	11	0.477362	0.477362	0.477362	0.477362	0.477362	0.477362
	12	0.475768	0.475768	0.475768	0.475768	0.475768	0.475768
	13	0.474177	0.474177	0.474177	0.474177	0.474177	0.474177
	14	0.472579	0.472579	0.472579	0.472579	0.472579	0.472579
	15	0.470988	0.470988	0.470988	0.470988	0.470988	0.470988
	16	0.470190	0.470190	0.470190	0.470190	0.470190	0.470190
	17	0.469394	0.469394	0.469394	0.469394	0.469394	0.469394
	18	0.468596	0.468596	0.468596	0.468596	0.468596	0.468596
	19	0.472579	0.472579	0.472579	0.472579	0.472579	0.472579
	20	0.476565	0.476565	0.476565	0.476565	0.476565	0.476565
	21	0.480551	0.480551	0.480551	0.480551	0.480551	0.480551
	22	0.482143	0.482143	0.482143	0.482143	0.482143	0.482143
	23	0.483741	0.483741	0.483741	0.483741	0.483741	0.483741
	24	0.485333	0.485333	0.485333	0.485333	0.485333	0.485333

4. Conclusiones

A través de la resolución del trabajo práctico concluimos que los métodos iterativos son útiles principalmente cuando tenemos una aproximación del resultado y podemos definir la semilla de manera tal que se reduzcan la cantidad de iteraciones. Por otro lado, al trabajar con métodos directos conocemos la cantidad de pasos con anterioridad. Todos los métodos utilizados nos sirvieron para obtener resultados cercanos al esperado.

Referencias

Convergencia del método Gauss-Seidel:

- http://mathforcollege.com/nm/simulations/nbm/04sle/nbm_sle_sim_convgaussseidel.pdf

ANEXO: Códigos

Todo el código que se presenta a continuación se encuentra en:

https://github.com/ariana-salese/TP2---AN1

Eliminación de Gauss

```
function vector_resultado = eliminacion_de_gauss(A, b, ds)
       [matriz_triangular_a, vector_b, _] = triangulacion(A, b, ds);
       vector_resultado = sustitucion_inversa(matriz_triangular_a, vector_b, ds);
endfunction
```

Triangulacion

```
function [matriz_triangular_a, vector_b, m] = triangulacion(A, b, ds)
  matriz_ampliada = A(:,:);
  matriz_ampliada = [matriz_ampliada, b(:,:)];
  n = columns(A);
  m = [];
  for i = 1:n - 1
    for j = i+1:n
      multiplicador = redondear(matriz_ampliada(j, i)/matriz_ampliada(i,i), ds);
      m = vertcat(m, [multiplicador]);
      for k = 1:n + 1
        matriz_ampliada(j,k) = redondear(matriz_ampliada(j,k) -
redondear(multiplicador*matriz_ampliada(i,k), ds), ds);
      endfor
    endfor
  endfor
  matriz_triangular_a = matriz_ampliada(:,1:n);
  vector_b = matriz_ampliada(:,end);
endfunction
```

Método LU

```
function matriz_resultado = metodo_LU(A, b_matriz, ds)
  col_b = columns(b_matriz);
  fil_b = rows(b_matriz);
  matriz_resultado = zeros(fil_b, col_b);
  [U, _, m] = triangulacion(A, zeros(fil_b, 1), ds);
  L = matriz_L(m, columns(A));
  for i = 1:col_b
    y = sustitucion_directa(L, b_matriz(:, i), ds);
    matriz_resultado(:,i) = sustitucion_inversa(U, y, ds);
  endfor
endfunction


#contruimos la matriz L
function matriz = matriz_L(mult, n)
  matriz = zeros(n, n);
  cont = 1;
  for columna = 1:n
    for fila = columna + 1:n
      matriz(fila, columna) = mult(cont, 1);
      cont += 1;
    endfor
  endfor
  for i = 1:n
    matriz(i, i) = 1;
  endfor
endfunction
```

Sustitución inversa

```
function solucion = sustitucion_inversa(A, b, ds)
  n = columns(A);
  solucion = zeros(n,1);
  solucion(n,1) = redondear(b(n) / A(n,n), ds);
  suma = 0;
  for i = n-1:-1:1
       suma = 0;
    for j = i+1:n
       suma = redondear(suma + redondear(A(i,j) * solucion(j,1), ds), ds);
    endfor
    solucion(i,1) = redondear((b(i) - suma)/A(i,i), ds);
  endfor
endfunction
```

Sustitución directa

```
function solucion = sustitucion_directa(A, b, ds)
  n = columns(A);
  solucion = zeros(n,1);
  solucion(1,1) = redondear(b(1) / A(1,1), ds);
  suma = 0;
  for i = 2:n
       suma = 0;
    for j = 1:i
       suma = redondear(suma + redondear(A(i,j) * solucion(j,1), ds), ds);
    endfor
    solucion(i,1) = redondear(redondear(b(i) - suma, ds)/A(i,i), ds);
  endfor
endfunction
```

Implementacion de función de redondeo según dígitos significativos

```
function resultado = redondear(x, ds)
  if x != 0
    D = 10^(ds-ceil(log10(abs(x))));
    resultado = round(x*D)/D;
  else
    resultado = 0;
  endif
endfunction
```

Inversa de matriz

```
function matriz_inversa = inversa(A, ds)
  n = columns(A);
  identidad = eye(n);
  matriz_inversa = zeros(n, n);
  for i = 1:n
    matriz_inversa(:, i) = eliminacion_de_gauss(A, identidad(:,i), ds);
  endfor
endfunction
```

Calculo de número de condición de la matriz

```
function numero_condicion = numero_condicion_matriz(A, ds)
  numero_condicion = norm(A, 'inf')*norm(inversa(A, ds), 'inf');
 endfunction
```

Método Gauss-Seidel

```
function vector_resultado = metodo_gauss_seidel(A, b)
  res = [0; 0; 0; 0; 0; 0]; #semilla
  res_inicial = res;
  n = rows(A);
  do
    for j = 1:n
      res(j,1)=(b(j)-(A(j,1:j-1)*res(1:j-1))-(A(j,j+1:n)*res(j+1:n)))/A(j,j);
    endfor
  error = norm(res - res_inicial, inf);
  res_inicial = res;
  until (error < 0.0001)
  vector_resultado = res;
endfunction
```

Interpolacion lineal

```
#Retorna una matriz con los valores naturales faltantes
function  matriz_resultado = interpolacionLineal(matriz)
  n = rows(matriz);
  matriz_resultado = zeros(matriz(n, 1) + 1, 2);

  for i = 1:n - 1
    x_i = matriz(i, 1);
    x_f = matriz(i + 1, 1);

    matriz_resultado(x_i + 1,:) = matriz(i,:);
    matriz_resultado(x_f + 1,:) = matriz(i + 1,:);

      for j = x_i + 1:x_f - 1
        valor_en_j = ((x_f - j)/(x_f - x_i))*matriz(i, 2)+((j - x_i)/(x_f -
x_i))*matriz(i + 1, 2);
        matriz_resultado(j + 1,:) = [j valor_en_j];
      endfor
  endfor
endfunction
```

Gráfico volaje según hora

```
function graficar_voltajes(horas, voltajes)

  plot(horas, voltajes);

  title("Voltaje según hora");

  xlabel ("Hora (h)");

  ylabel ("Voltaje (V)");

  grid on

  set(0, "defaulttextfontsize", 24);

  set(0, "defaultaxesfontsize", 16);

  print -djpg voltajes_segun_hora.jpg

endfunction
```