Proyecto de Segundo Parcial
Métodos Numéricos para ingeniería

Victor Uriel Mendoza Salazar

October 24, 2021

1 Introduction

En este segundo parcial se estudiaron diferentes métodos y técnicas que son útiles para resolver sistemas de ecuaciones que cuentatan con más de una incógnita o variable desconocida. De igual manera se revisó el método de Newton-Raphson que es útil cuando se cuenta con un sistema de ecuaciones no lineales. Para lograr encontrar la respuesta correcta a cualquier sistema planteado es necesario verificar con el tipo de ecuación o ecuaciones se cuentan y posteriormente elegir un método que facilite la resolución del mismo (ya sea lineal o no lineal). Una de las ventajas de los métodos estudaidos es que estos pueden resolverse a través de hojas de cálculo, con procedimiento a mano o mediante un programa computacional como matlab, mathematica, etc.

Para este proyecto se plantean dos problemas, uno de ellos puede ser descrito mediante ecuaciones lineales y el otro mediante ecuaciones no lineales. Por lo que para la resolución de los problemas planteados será necesario utilizar métodos de resolución de ecuaciones lineales y no lineales también.

2 Descripción del problema

Como se mencionó anteriormente en este proyecto se van a plantear dos problemas. En este reporte se comenzará a tratar con el problema que necesita la resolución de ecuaciones lineales de primer órden.

Ley de Kirchhoff de voltajes. Una de las principales herramientas para entender y modelar el comportamiento de infiinda de circuitos eléctricos y electrónicos ha sido la ley de kirchhoff de voltajes conocida como LKV. Esta ley menciona que el voltaje que circula dentro de un circuito es igual al consumido por cada uno de los componentes del circuito, es decir la suma de los voltajes en los componentes es exactamente igual al voltaje suministrado.Se basa en la ley de la conservación de la energía donde esta se conserva y transforma, no se destruye o se crea.

De igual manera cabe destacar que el valor de las resistencias mostradas está en kilOhms por lo que esperamos que el resultado de las corrientes se encuentre en el orden de los miliAmperes.

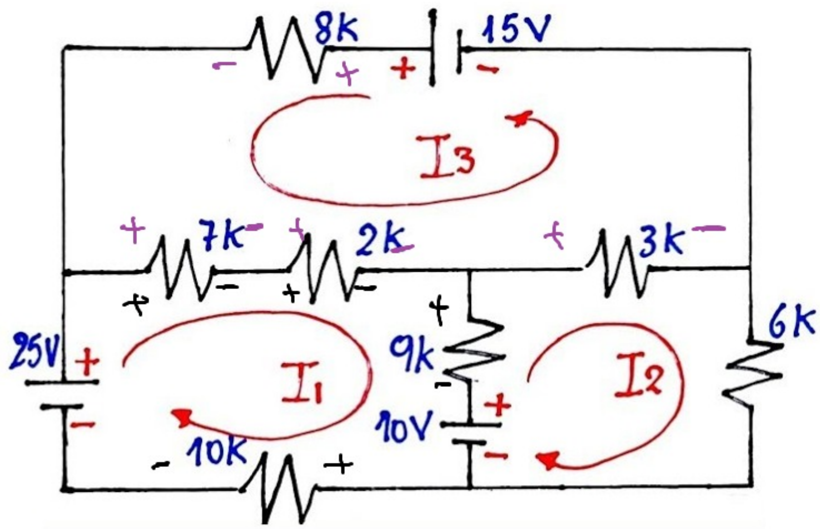


Figure 1: Circuito de 3 mallas

La imagen nos muestra un circuito que cuenta con tres "mallas" y que puede resolverse planteando tres ecuaciones mediante un análisis LKV.

Las ecuaciones que describen al circuito son las siguientes:

- **M**alla 1: 15=28x-9y+9z
- **M**alla 2: 10=-9x18y+3z
- **M**alla 3: 15=9x+3y+20z

Donde "x" es la corriente I1 que circula por la malla 1, "y" es la corriente I2 que circula por la segunda malla y finalmente "z" es la corriente I3 que circula por la tercer malla.

Para la resolución de este sistema de ecuacioens lineal se utilizaron tres diferentes métodos: Cramer, Gauss y Siedel. Para los tres métodos se realizó el procedimiento en hojas de cálculo de excel y también se ingresó a un programa de MATLAB para obtener la respuesta.

A continuación se muestran el desarrollo de cada método así como el resultado obtenido.

Cramer Para realizar exitosamente el método de cramer, y debido a que la matriz del sistema es de 4X3, para calcular los determinantes de cada variable fue necesario repetir las dos primeras filas en el cálculo de cada determinante.

La siguiente imágen muestra el desarrollo en excel así como el programa de MATLAB con el resultado encontrado.

	x	y	z	Término Idp		28x-9y+9=15				
Ec1	28	-9	9	15		18y-9x+3z=10				
Ec2	-9	18	3	10		9x+3y+20z=15				
Ec3	9	3	20	15						
	Determinante X						Determinante S			
15	-9	9				28	-9	9		
10	18	3				-9	18	3		
15	3	20	Det x	4500		9	3	20	Det s	6264
15	-9	9				28	-9	9		
10	18	3				-9	18	3		
	Determinante Y									
28	15	9				x	0.7183908	mA		
-9	10	3								
9	15	20	Det y	5420		y	0.86526181	mA		
28	15	9								
-9	10	3				z	0.29693487	mA		
	Determinante Z									
28	-9	15								
-9	18	10								
9	3	15	Det z	1860						
28	-9	15								
-9	18	10								

Figure 2: Desarrollo del método de Cramer en Excel.

Como se puede observar el resultado encontrado para los resultados encontrados para las corrientes son los siguientes:

- **x**=I1=0.7183 mA
- **y**=I2=0.8652 mA
- **z**=I3=0.2969 mA

De igual manera se observa que en MATLAB los resultados son muy similares. Los resultados se muestran en la esquina superior derecha enmarcados en un rectángulo rojo.

Se observan los siguientes resultados:

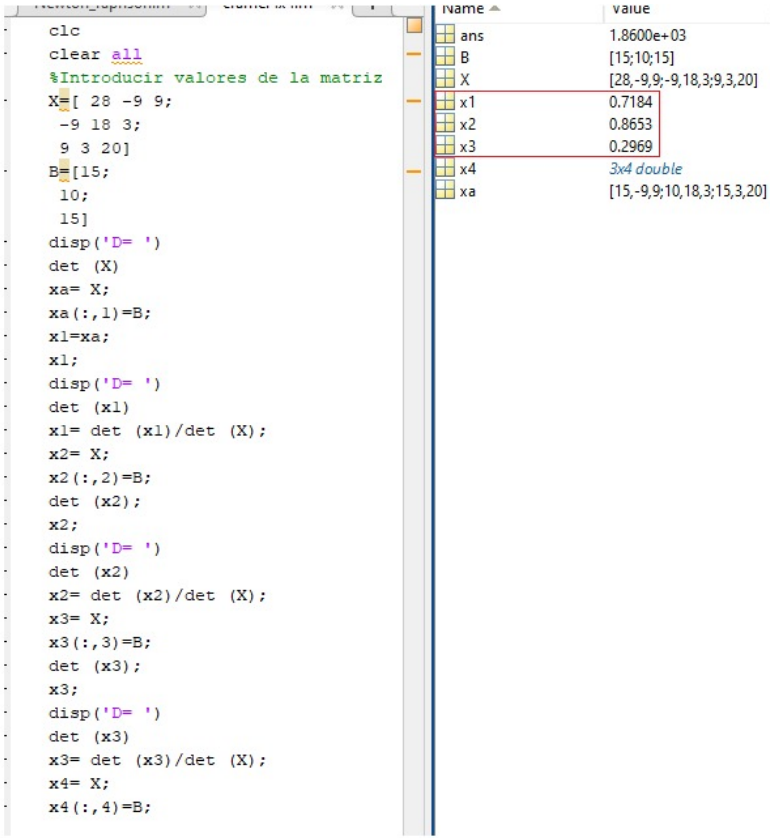


Figure 3: Desarrollo del método de Cramer en MATLAB

- **x**=I1=0.7184 mA
- **y**=I2=0.8653 mA
- **z**=I3=0.2969 mA

Gauss El resolver el sistema de ecuaciones mediante el método Gaussiano era algo naturalmente adecuado debido a las características del sistema y a que este realmente no cuenta con demasiadas variables desconocidas y puede resolverse en pocas iteraciones u operaciones. En este caso mediante excel se resolvió con tan solo 4 operaciones básicas. Se muestra el desarrollo en excel a continuación.

x	y	z	Término Idp		28x-9y+9=15
28	-9	9	15		18y-9x+3z=10
-9	18	3	10		9x+3y+20z=15
9	3	20	15		
Iteración 1					
28	-9	9	15		
0	423/28	165/28	415/28	x	0.718390805
9	3	20	15		
Iteración 2					
28	-9	9	15		
0	423/28	165/28	415/28	y	0.865261814
0	165/28	479/28	285/28		
Iteración 3					
28	-9	9	15		
0	423/28	165/28	415/28		
0	0	696/47	620/141		
	z	0.29693487			

Figure 4: Excel Gauss

Se observa en rojo los resultados obtenidos para las corrientes del circuito.

- **x**=I1=0.7183 mA
- **y**=I2=0.8652 mA
- **z**=I3=0.2969 mA

El desarrollo del programa para resolver el sistema de ecuaciones por el método Gaussiano es un poco más complicado debido a lo complejo que puede ser indicarle al programa cómo y cuándo debe de realizar las operaciones.

El resultado obtenido para el programa de Gauss en MATLAB es el siguiente mostrado a continuación.

```
clear
clc
%x = [2  1  0; 6 5 -2];

x = [28 -9 9 15; -9 18 3 10;
     9 3 20 15];

%x = [1 3 1 10; 1 -2 -1 -6; 2 1 2 10];
% x = [2 -1 4 1 -1 7;
%      -1 3 -2 -1 2 1;
%      5 1 3 -4 1 33;
%      3 -2 -2 -2 3 24;
%      -4 -1 -5 3 -4 -49];

for n = 1:(length(x)-1)
        % Step 1: make the row N's Nth term 1 by dividing
        % the whole row by it
        A = x(n,:);
        A = A/A(n);
        x(n,:) = A;

        % Step 2: for every other row add to it -1 * that rows Nth term *
        % the Nth row
        for k = 1:(length(x)-1)
            if n~=k
                x(k,:) = A*(-1*x(k,n)) + x(k,:);
            end
        end
end

y = x(:,length(x))';
y
```

Command Window

```
y =

    0.7184    0.8653    0.2969
```

Figure 5: Desarrollo del método de Gauss en MATLAB

Se observa en rojo los resultados obtenidos para las corrientes del circuito calculadas mediante el programa de Matlab son las siguientes.

- **x**=I1=0.7184 mA
- **y**=I2=0.8653 mA

- **z**=I3=0.2969 mA

Seidel El desarrollo de este método es un poco más complejo comparado con el desarrollo de los métodos anteriores, sin embargo se logra encontrar un resultado preciso. Como se observará a continuación para llegar a un resultado y que el resultado sea exactamente 0 o lo más aproximado al cero fue necesario realizar 41 iteraciones, lo que no resulta práctico si se realiza a mano y tampoco resulta muy eficiente si se realiza a computadora debido a que consumiría más recursos comparados con otros métodos como el Gaussiano.

28x-9y+9=15		x	y	z	Término Idp	
18y-9x+3z=10		28	-9	9	15	
9x+3y+20z=15		-9	18	3	10	
		9	3	20	15	
Iteración	x	y	z	Error x	Error y	Error z
0	0	0	0	0	0	0
1	0.53571429	0.8234127	0.38541667	1	1	1
2	0.67649872	0.82956881	0.32114025	0.20810747	0.00742085	0.2001506
3	0.69913775	0.85160105	0.30764785	0.03238135	0.02587156	0.04385663
4	0.71055639	0.85955911	0.30131576	0.01606999	0.00925829	0.02101481
5	0.71514965	0.86291109	0.298746	0.0064228	0.0038845	0.00860184
6	0.71705306	0.86429109	0.29768246	0.0026545	0.00159669	0.00357273
7	0.71783849	0.86486106	0.29724352	0.00109415	0.00065903	0.00147669
8	0.71816278	0.86509636	0.2970623	0.00045156	0.00027199	0.00061006
9	0.71829666	0.8651935	0.29698748	0.00018639	0.00011228	0.00025193
10	0.71835194	0.86523361	0.29695659	7.6947E-05	4.6354E-05	0.00010402
11	0.71837476	0.86525017	0.29694383	3.1767E-05	1.9137E-05	4.2948E-05
12	0.71838418	0.86525701	0.29693857	1.3115E-05	7.9009E-06	1.7731E-05
13	0.71838807	0.86525983	0.29693639	5.4145E-06	3.2619E-06	7.3206E-06
14	0.71838968	0.86526099	0.2969355	2.2354E-06	1.3467E-06	3.0223E-06
15	0.71839034	0.86526148	0.29693513	9.229E-07	5.5599E-07	1.2478E-06
16	0.71839061	0.86526167	0.29693497	3.8103E-07	2.2954E-07	5.1516E-07
17	0.71839073	0.86526176	0.29693491	1.5731E-07	9.4768E-08	2.1269E-07
18	0.71839077	0.86526179	0.29693488	6.4946E-08	3.9126E-08	8.7809E-08
19	0.71839079	0.8652618	0.29693487	2.6813E-08	1.6153E-08	3.6252E-08
20	0.7183908	0.86526181	0.29693487	1.107E-08	6.6689E-09	1.4967E-08
21	0.7183908	0.86526181	0.29693487	4.5703E-09	2.7533E-09	6.1792E-09
22	0.7183908	0.86526181	0.29693487	1.8869E-09	1.1367E-09	2.5511E-09
23	0.7183908	0.86526181	0.29693487	7.7901E-10	4.693E-10	1.0532E-09
24	0.7183908	0.86526181	0.29693487	3.2162E-10	1.9375E-10	4.3484E-10
25	0.7183908	0.86526181	0.29693487	1.3278E-10	7.9992E-11	1.7952E-10
26	0.7183908	0.86526181	0.29693487	5.4819E-11	3.3025E-11	7.4117E-11
27	0.7183908	0.86526181	0.29693487	2.2633E-11	1.3635E-11	3.06E-11
28	0.7183908	0.86526181	0.29693487	9.344E-12	5.6291E-12	1.2633E-11
29	0.7183908	0.86526181	0.29693487	3.8577E-12	2.3241E-12	5.2156E-12
30	0.7183908	0.86526181	0.29693487	1.5927E-12	9.5938E-13	2.1534E-12
31	0.7183908	0.86526181	0.29693487	6.5743E-13	3.9609E-13	8.8893E-13
32	0.7183908	0.86526181	0.29693487	2.7153E-13	1.6372E-13	3.6735E-13
33	0.7183908	0.86526181	0.29693487	1.122E-13	6.7491E-14	1.5161E-13
34	0.7183908	0.86526181	0.29693487	4.6208E-14	2.7843E-14	6.244E-14
35	0.7183908	0.86526181	0.29693487	1.9163E-14	1.1548E-14	2.5986E-14
36	0.7183908	0.86526181	0.29693487	7.7272E-15	4.6192E-15	1.0282E-14
37	0.7183908	0.86526181	0.29693487	3.3999E-15	2.053E-15	4.4867E-15
38	0.7183908	0.86526181	0.29693487	1.3909E-15	1.0265E-15	2.4303E-15
39	0.7183908	0.86526181	0.29693487	4.6363E-16	2.5662E-16	1.8695E-16
40	0.7183908	0.86526181	0.29693487	3.0909E-16	0	3.7389E-16
41	0.7183908	0.86526181	0.29693487	0	0	0

Figure 6: Desarrollo del método de Seidel en Excel

Se observa en rojo los resultados obtenidos para las corrientes del circuito calculadas mediante el excel son las siguientes.

- **x**=I1=0.7183 mA
- **y**=I2=0.8652 mA
- **z**=I3=0.2969 mA

El resultado obtenido para el programa de Gauss en MATLAB es el siguiente mostrado a continuación.

```matlab
    %% Gauss Seidel Method
A= input('A =');
b= input('b = ');
x= input('x = ');
n=size(x,1);
normVal=Inf;
tol=1e-5; itr=0;
while normVal>tol
    x_old=x;

    for i=1:n

        sigma=0;

        for j=1:i-1
                sigma=sigma+A(i,j)*x(j);
        end

        for j=i+1:n
                sigma=sigma+A(i,j)*x_old(j);
        end

        x(i)=(1/A(i,i))*(b(i)-sigma);
    end

    itr=itr+1;
    normVal=norm(x_old-x);
end
fprintf('Solution of the system is : \n%f\n%f\n%f\n%f in %d iterations',x,itr);
```

Command Window

```
>> Gauss_siedel
A =[28 -9 9; -9 18 3; 9 3 20]
b = [ 15; 10; 15]
x = [0 0 0]'
Solution of the system is :
0.718388
0.865260
0.296936
13.000000 in >>
```

Figure 7: Desarrollo del método de Seidel en MATLAB

Se observa en rojo los resultados obtenidos para las corrientes del circuito calculadas mediante el programa de Matlab son las siguientes.

- **x**=I1=0.7183 mA
- **y**=I2=0.8652 mA
- **z**=I3=0.2969 mA

Como se observa el programa de matlab tuvo que realizar muchas menos iteraciones (solamente realizó 13 iteraciones), sin embargo considero que este método sigue siendo el menos eficiente de los tres revisados.

Segundo problema: Newton-Raphson El problema nos plantea una reacción química de descomposición, es necesario encontrar el número de moles al equilibrio de cada elemento químico presente en la misma, pero para ello primero es necesario encontrar el grado de avance de la misma. El problema plantea un par de reacciones reversibles que cuentan con un valor de Kp1=0.0004 y Kp2= 0.037. Lo que nos indica que esta reacción puede llevarse a cabo. A continuación se muestra el enunciado del problema a mano, así como el planteamiento del sistema de ecuaciones que se utilizó.

$$2H_2 + OH \rightleftarrows H_2O \qquad K_1 = \frac{H_2O}{2H_2^2\,(OH)}$$

$$H_2 + O \rightleftarrows H_2O$$

$$K_2 = \frac{H_2O}{H_2\,(O)} \qquad K_1 = 4\times10^{-4} \quad K_2 = 3.7\times10^{-2}$$

$$nH_2 = 50 \quad nOH = 20 \quad nH_2O = 5 \quad nO = 10$$

$$H_2 = 50 - 2x - y \qquad H_2O = 5 + x + y$$

$$OH = 20 - x \qquad O = 10 - y$$

$$F_1(x,y) = \frac{5+x+y}{(50-2x-y)^2(20-x)} - 4\times10^{-4}$$

$$F_2(x,y) = \frac{(5+x+y)}{(50-2x-y)(10-y)} - 3.7\times10^{-2}$$

Figure 8: Desarrollo del segundo problema a mano

Como se observa el sistema de ecuaciones que resulta es no lineal por lo que para poder encontrar el resultado fue necesario desarrollar el procedimiento en las hojas de cálculo de excel y en MATLAB mediante el método de Newton-Raphson.

Cabe destacar que la creación de una hoja de cálculo en excel para este sistema de ecuaciones en específico significó una tara un poco tediosa, debido a la naturaleza de las ecuaciones del sistema. El cálculo y evaluación realizadas para excel resultaron largas pero efectivas, puesto que se llegó a un resultado tan solo en la sexta iteración, mientras que como veremos en el programa de MATLAB requirió al rededor de 7000 iteraciones. A continuación se muestra el método de Newton-Raphson desarrollado en excel, así como los resultados encontrados resaltados en rojo.

	Valor inicial	3,3						
f1	(5 + x(1) + x(2)) / (50 - 2 * x(1) - x(2)) ^ 2 / (20 - x(1)) - 0.0004		x	y	f1	f2	h1	h2
f2	(5 + x(1) + x(2)) / (50 - 2 * x(1) - x(2)) / (10 - x(2)) - 0.037		3	3	-1.50751E-05	0.23129268	-0.09897966	0.45558641
			2.901020337	3.45558641	1.15847E-07	0.24174191	-0.07559264	-0.1327745
			2.825427695	3.3228119	-1.43486E-05	0.23473376	-0.07351135	0.13123631
derivada f1 (x)	(8*x(1)^3+16*(x(1)^2)*x(2)-220*(x(1)^2)+8*x(1)*(x(2)^2)-520*x(1)*x(2)-2800*x(1)+(x(2)^3)+3600*x(2)+82500-155*(x(2)^2))/(((50-2*x(1)-x(2))^2))*(20-x(1))		2.751916341	3.45404821	-1.43007E-05	0.23603572	-0.07139249	0.12750942
derivada f1 (y)	(x(2)+60)/((20-x(1))*(50-2*x(1)-x(2))^3)		2.680523846	3.58155763	-1.42543E-05	0.23730093	-0.06937096	-0.1239443
			2.611152889	3.45761333	-2.71741E-05	0.23087862		
derivada f2 (x)	((-2*x(1)^2)-4*x(1)*x(2)-20*x(1)+55*x(2)+850-x(2)^2)/(((50-2*x(1)-x(2))^2)*(10-x(1))^2)							
derivada f2 (y)	(-x(1)+55)/((10-x(1))*(50-2*x(1)-x(2))^2)							
			Iteración					
			x	y	B			
		2.68052385	9.70178E-05	5.4311E-05	1.42543E-05			
		3.84710664	3.853417825	0.17523063	-0.237300928			
			inversa			Inversa *B		
			-911.321971	0.28245431		-0.08001689		
			20040.47043	-0.50456067		0.40539481		

Figure 9: Desarrollo del sistema en Excel y resultados.

En la tabla de abajo se muestran los resultados de los moles al equilibrio que se calcularon mediante los valores de grado de avance encontrados a través de Excel.

Elemento Químico	Número de moles al equilibrio
H_2	41.33
OH	17.39
H_2O	11.06
O	6.55

Figure 10: Resultados de moles al equilibrio con los valores de grado de avance encontrados con Excel.

El programa desarrollado en MATLAB es mucho más amable, para este solamente es necesario ingresar las ecuaciones del sistema así como el jacobiano del sistema, lo cual resulta no ser una tarea tan difícil o extensa como lo es programar una hoja de cálculo en Excel. Se muestra a continuación el programa que se utilizó para resolver el sistema en MATLAB, así como los resultados obtenidos.

```
clc
%% x(1) --> x
%% x(2) --> y
F= @(x) [(5 + x(1) + x(2)) / (50 - 2 * x(1) - x(2)) ^ 2 / (20 - x(1)) - 0.0004;
     (5 + x(1) + x(2)) / (50 - 2 * x(1) - x(2)) / (10 - x(2)) - 0.037];
J= @(x) [
(8*x(1)^3+16*(x(1)^2)*x(2)-220*(x(1)^2)+8*x(1)*(x(2)^2)-520*x(1)*x(2)-2800*x(1)+(x(2)^3)+3600*x(2)+82500-155*(x(2)^2))/(((50-2*x(1)-x(2))^2)*(20-x(1)), (x(2)+60)/((20-
((-2*x(1)^2)-4*x(1)*x(2)-20*x(1)+55*x(2)+850-x(2)^2)/(((50-2*x(1)-x(2))^2)*(10-x(1))^2),(-x(1)+55)/((10-x(1))*(50-2*x(1)-x(2))^2)];
x = [3; 3];
error = 1e-2;
tol = 1e-5;
n = 0;
while error > tol
 dx = -J(x)\F(x);
 error = norm(dx)/norm(x);
 x = x + dx;
 n = n+1;
end
fprintf("Iteraciones : %d \n",n);
fprintf("Respuesta : %f \n", x.');
```

Command Window

```
Iteraciones : 7025970
Respuesta : 3.310730
Respuesta : 2.691570
```

Figure 11: Desarrollo del sistema en MATLAB y resultados encontrados.

Como se puede apreciar en la imágen el resultado encontrado para la primer respuesta (valor de grado de avance de la reacción 1) es de 3.31 y el valor encontrado para la segunda respuesta es de 2.69. Por lo que el número de moles de cada elemento serían los mostrados en la siguiente tabla.

Elemento químico	Número de moles al equilibrio
H_2 (hidrógeno)	40.68
OH (ion-hidroxilo)	16.69
H_2O (agua)	11.001
O (oxígeno)	7.30

Figure 12: Resultados de moles al equilibrio con los valores de grado de avance encontrados con MATLAB

3 Resulados

Para el problema que plantea el circuito de tres mallas podemos observar que los resultados obtenidos mediante los tres diferentes métodos son muy similares. En la siguiente tabla se muestran un resumen de los resultados.

Resumen de resultados sistema lineal de ecuaciones						
	Cramer		Gauss		Seidel	
	Excel	Matlab	Excel	Matlab	Excel	Matlab
X (I1)	0.7183	0.7184	0.7183	0.7184	0.7183	0.7184
Y (I2)	0.8652	0.8653	0.8652	0.8653	0.8652	0.8653
Z (I3)	0.2969	0.2969	0.2969	0.2969	0.2969	0.2969

Figure 13: Resumen de resultados problema 1.

En la sección de conclusiones se dara con más detalle una explicación o argumentación sobre cuál es el método que se considera el más eficiente o adecuado para este sistema de ecuaciones en particular y para casos similares.

4 Conclusiones

En cuanto al primer sitema de ecuaciones y las soluciones encontradas mediante los diferentes métodos de resolución de ecuaciones lineales puedo decir con seguridad que es mucho más eficiente realizar el método de Gauss o Cramer,debido a que si se hace mediante un programa como MATLAB se requerirán menos recursos del sistema, además de que mediante estos métodos son más fáciles de ejecutar a mano y en excel, por otro lado el método de Seidel resulta ser preciso sin embargo es más tardado y puede llegar a ser un poco más confuso comparándolo con el método de Gauss que puede ser resuelto mediante sencillas operaciones algebráicas.

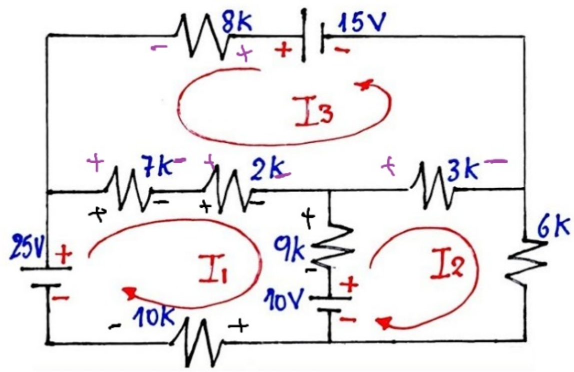
5 Bibligrafía y Apéndices

Bibliografía

- **C**hapra, S., Canale, R., Garcia Ruiz, R., Ibarra Mercado, V., Munoz Diaz, E. and Evangelista Benites, G., 2015. Metodos numericos. 7th ed. Lima, Perú: McGraw Hill, pp.320-325.

Apéndices

- **S**e muestra el desarrollo a mano para obtener las ecuaciones del circuito eléctrico.



Malla 1:

$$25 - 7I_1 - 2I_1 - 9I_1 - 10 - 10I_1 + 9I_2 - (7+2)I_3 = \emptyset$$

$$15 = 28I_1 - 9I_2 + 9I_3$$

Malla 2:

$$10 = (9+3+6)I_2 - 9I_1 + 3I_3$$

$$10 = 18I_2 - 9I_1 + 3I_3$$

Malla 3:

$$(8+7+2+3)I_3 + (7+2)I_1 + 3I_2 = 15$$

$$15 = 9I_1 + 3I_2 + 20I_3$$

Figure 14: Desarrollo del método de Seidel en MATLAB