ANALISIS MATRICIAL DE CIRCUITOS DE CORRIENTE CONTINUA USANDO MATLAB

Carlos Enrique Henao Osorio
John Alexander Londoño Ramírez

Universidad Tecnológica de Pereira
Facultad de Tecnología
Tecnología Eléctrica
Pereira
2007

ANALISIS MATRICIAL DE CIRCUITOS DE CORRIENTE CONTINÚA USANDO MATLAB

Carlos Enrique Henao Osorio
John Alexander Londoño Ramírez

TRABAJO DE GRADO
PARA OPTAR AL TÍTULO DE TECNÓLOGO EN ELECTRICIDAD

DIRECTOR
Pompilio Tabares E.
Ingeniero Electricista

Universidad Tecnológica de Pereira
Facultad de Tecnología
Tecnología Eléctrica
Pereira
2007

Nota de aceptación:

Firma del

presidente del jurado

Firma del jurado

Firma del jurado

Pereira, Noviembre de 2007

A mi padre y en especial a mi madre por el apoyo y sacrificio que demostraron durante el transcurso de la carrera, además a mis hermanas y sobrinos por su motivación en mi formación como profesional.

Carlos Enrique

Dedico este trabajo a mi familia por su apoyo y acompañamiento durante el transcurso de la carrera, en especial a mis padres, aunque uno de ellos no alcanzo a ver este logro, porque una de sus metas era que fuera un profesional

John Alexánder

.

AGRADECIMIENTOS

Agradecemos al ingeniero Pompilio Tabares Espinosa por su acompañamiento como director, además por su respaldo y sus oportunas aclaraciones para lograr la culminación del presente trabajo. Al ingeniero Ricardo Henao por su asesoría y desinteresada orientación al inicio de este trabajo. Finalmente al ingeniero William Jaramillo por sus oportunos aportes y observaciones para la culminación de ésta meta.

CONTENIDO

pág.

LISTA DE TABLAS

pág.

LISTA DE FIGURAS

pág.

INTRODUCCIÓN

Actualmente los cursos de teoría de circuitos en Tecnología Eléctrica hacen poco uso de las herramientas computacionales y uno de los objetivos fundamentales es incluirlas en los cursos correspondientes de circuitos de dicho programa. También utilizar las ventajas de estos programas de computación en los cursos posteriores como: Electrónica, Máquinas Eléctricas, Sistemas de Transmisión y Distribución de Energía, Teoría de Control entre otras asignaturas, para la formación del tecnólogo en electricidad.

El propósito fundamental del proyecto es complementar los conceptos teóricos con el uso del programa MATLAB, un programa para realizar cálculos numéricos con vectores y matrices, es fácil de usar y trabaja con una serie de comandos de línea para manejar programas de computador. Este programa será de gran utilidad para analizar circuitos de corriente continua y ayudar a programar los métodos convencionales tales como análisis nodal y análisis de mallas.

La importancia del presente proyecto de grado radica en que establecerá ideas fundamentales acerca de matrices y determinantes, y sus métodos de solución como la regla de Crámer y/o la eliminación de Gauss entre otros, así como, algunos conceptos básicos acerca del manejo del paquete computacional MATLAB; se introducen conceptos de análisis nodal y el análisis de mallas y sus variaciones dependiendo de las características del circuito a analizar; también se crean unas funciones en MATLAB que serán usadas para el desarrollo de ejercicios propuestos, cuyos resultados son comparados con los métodos tradicionales para la solución de circuitos y también con los programas comerciales de simulación.

Así pues, el propósito de este trabajo es proponer una nueva herramienta para el análisis de circuitos lineales, la cual puede ir de la mano con la teoría de circuitos de la asignatura Circuitos Eléctricos I, y maneja un lenguaje sencillo para la comprensión del estudiante.

1. MATRICES Y DETERMINANTES

Este capítulo describe las propiedades de los determinantes y matrices que se necesitan para entender el uso del MATLAB en forma eficiente. Los determinantes y matrices son herramientas poderosas para el análisis de ecuaciones lineales. Debido a que el análisis de circuitos lineales involucra ecuaciones lineales, el conocimiento de cómo usar determinantes y matrices en forma eficiente es esencial para la solución de los circuitos lineales.

1.1. MATRICES [12]

1.1.1 Definiciones y Notación.

Una ordenación de números dispuestos en filas y columnas, recibe el nombre de matriz; también, una matriz se puede definir como un arreglo de números (o funciones) encerrados por corchetes. Estos números (o funciones) son llamados entradas o elementos de la matriz.

Las matrices se denotan por letras mayúsculas en negritas **A, B, C,…,** o escribiendo la entrada general entre corchetes; así, **A** = $[a_{ij}]_{(m,n)}$ donde a_{ij} representa la componente de la fila i-ésima y la columna j-ésima, y (m,n) indica que la matriz tiene m filas y n columnas o que es una matriz de m x n. Los elementos de las matrices se pueden ubicar dentro de paréntesis ( ), corchetes [ ] ó barras ‖ ‖.

Una matriz de m x n, tiene la forma general:

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & a_{m3} & \cdots & a_{mn} \end{bmatrix}$$

Si el número de filas de una matriz es igual al número de columnas (m=n), la matriz es cuadrada y se dice que tiene orden n. Su diagonal contiene los elementos a_{11}, a_{22}, a_{33},…, a_{nn} y se llama diagonal principal o central. Una matriz

que no es cuadrada se denomina rectangular. Una matriz se llama real si todos sus elementos son números reales.
Un vector es una matriz que tiene una sola fila y se denomina vector fila ó una sola columna y se denomina vector columna. En ambos casos los elementos de los vectores se representan con letras minúsculas, colocándose en negrita la que representa el vector, así: **a**= [a_j]. Las filas y columnas de una matriz **A** de m x n algunas veces reciben los nombres de vectores fila y vectores columna de **A**.

1.1.2 Algebra de Matrices

Esta sección describe las propiedades algebraicas de las matrices. Estas incluyen negación, adición, y substracción y su comportamiento asociativo y conmutativo. Se aprenderá cómo multiplicar y factorizar matrices, a particionar matrices, y a realizar división de una matriz por la derecha y por la izquierda.

- **Negación**

Debido a que cada ecuación en un conjunto de ecuaciones puede ser multiplicada por -1 sin cambiar la solución, se puede negar una matriz cambiando el signo de todos los elementos de la matriz, lo cual da:

$$-\mathbf{A} = -\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} = \begin{bmatrix} a_{11} & -a_{12} & \cdots & -a_{1n} \\ a_{21} & -a_{22} & \cdots & -a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & -a_{n2} & \cdots & -a_{nn} \end{bmatrix}$$

- **Suma**

Sumar dos matrices requiere que las matrices tengan el mismo número de filas y columnas. Esta afirmación es la regla de conformidad de la suma. Si **A**=[a_{ij}] y **B**=[b_{ij}] son matrices m x n, entonces la suma entre ellas **A**+**B** se define como la matriz m x n: **A**+**B**=**C**=[c_{ij}] donde $c_{ij}=a_{ij}+b_{ij}$.
En la suma de matrices, los elementos correspondientes se suman, así:

$$\mathbf{A}+\mathbf{B} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} + \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1n} \\ b_{21} & b_{22} & \cdots & b_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ b_{m1} & b_{m2} & \cdots & b_{mn} \end{bmatrix} = \begin{bmatrix} a_{11}+b_{11} & a_{12}+b_{12} & \cdots & a_{1n}+b_{1n} \\ a_{21}+b_{21} & a_{22}+b_{21} & \cdots & a_{2n}+b_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1}+b_{m1} & a_{m2}+b_{m2} & \cdots & a_{mn}+b_{mn} \end{bmatrix}$$

Ejemplo 1.1.

$$\begin{bmatrix} 3 & -5 & 7 \\ 4 & 8 & 2 \end{bmatrix} + \begin{bmatrix} 6 & 2 & -4 \\ 3 & 1 & 2 \end{bmatrix} = \begin{bmatrix} 9 & -3 & 3 \\ 7 & 9 & 4 \end{bmatrix}$$

Substracción

La substracción sigue la forma de la negación y la adición, así

$$\mathbf{A} - \mathbf{B} = \mathbf{A} + (-\mathbf{B}) = \begin{bmatrix} a_{11} - b_{11} & a_{12} - b_{12} & \cdots & a_{1n} - b_{1n} \\ a_{21} - b_{21} & a_{22} - b_{21} & \dots & a_{2n} - b_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} - b_{m1} & a_{m2} - b_{m2} & \cdots & a_{mn} - b_{mn} \end{bmatrix}$$

a) Conmutación

La suma de matrices es conmutativa, así

$$\mathbf{A} + \mathbf{B} = \mathbf{B} + \mathbf{A} \qquad (1.1.)$$

b) Asociación

La adición o substracción de matrices es asociativa, así

$$\mathbf{A} \pm (\mathbf{B} \pm \mathbf{C}) = (\mathbf{A} \pm \mathbf{B}) \pm \mathbf{C} = \mathbf{A} \pm \mathbf{B} \pm \mathbf{C} \qquad (1.2.)$$

- **Multiplicación y Factorización [3]**

Para conformar la regla que permita multiplicar matrices se requiere que el número de columnas de la primera matriz sea el mismo que el número de filas de la segunda matriz. La definición usual del producto de una matriz:

$$\mathbf{C} = \mathbf{A} * \mathbf{B} \qquad (1.3.)$$

define cada elemento de la matriz C como la suma de los productos de los elementos sucesivos de la fila de la matriz A con los elementos correspondientes de la columna de la matriz B, que tiene la forma matemática:

$$C_{ij} = a_{i1}b_{1j} + a_{i2}b_{2j} + \dots a_{i.ca}b_{ca.j} = \sum_{k=1}^{ca=fb} a_{ik}b_{kj} \tag{1.4.}$$

cuando ca es el número de columnas de A y fb es el número de filas de B. El número de filas fc de la matriz C es igual al número de filas fa de A, y el número de columnas cc de la matriz C es igual al número de columnas cb de B.

a) Conmutación

Con excepción para casos especiales, la multiplicación de matrices no es conmutable:

$$\mathbf{A} * \mathbf{B} \neq \mathbf{B} * \mathbf{A} \tag{1.5.}$$

b) Asociación

La multiplicación de matrices es asociativa.

$$\mathbf{A} * (\mathbf{B} * \mathbf{C}) = (\mathbf{A} * \mathbf{B}) * \mathbf{C} = \mathbf{A} * \mathbf{B} * \mathbf{C} \tag{1.6.}$$

El producto tiene fa filas y cc columnas, ca es igual a fb, y cb igual a fc.

c) Distribución

Es distributiva sobre la adición ó substracción, así

$$\mathbf{A} *(\mathbf{B} \pm \mathbf{C}) = \mathbf{A} * \mathbf{B} \pm \mathbf{A} * \mathbf{C} \tag{1.7.}$$

y

$$(\mathbf{B} \pm \mathbf{C}) * \mathbf{A} = \mathbf{B} * \mathbf{A} \pm \mathbf{C} * \mathbf{A} \tag{1.8.}$$

Las reglas de prioridad se aplican, así

$$\mathbf{A} * \mathbf{B} + \mathbf{C} = (\mathbf{A} * \mathbf{B}) + \mathbf{C} \tag{1.9.}$$

- **Partición[3]**

Una matriz puede ser particionada interiormente en submatrices. Por ejemplo:

$$\boldsymbol{A} = \left[\begin{array}{cc|c|cc} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ \hline a_{21} & a_{22} & a_{23} & a_{24} & a_{25} \\ a_{31} & a_{32} & a_{33} & a_{34} & a_{35} \\ \hline a_{41} & a_{42} & a_{43} & a_{44} & a_{45} \\ a_{51} & a_{52} & a_{53} & a_{54} & a_{55} \end{array}\right] = \left[\begin{array}{c|c|c} A_{11} & A_{12} & A_{13} \\ \hline A_{21} & A_{22} & A_{23} \\ \hline A_{31} & A_{32} & A_{33} \end{array}\right]$$

donde, por ejemplo, la matriz A_{12} es

$A_{12} = [a_{13}] = a_{13}$

Una matriz o una matriz particionada puede no tener filas o columnas. En cualquier evento la matriz es una matriz nula. Si dos matrices particionadas son adicionadas el número de filas y columnas de cada parte correspondiente a la partición debe ser el mismo. Para multiplicar dos matrices particionadas, el número de columnas de cada partición sucesiva de la primera matriz debe ser igual al número de filas en cada partición sucesiva de la segunda matriz. Por supuesto, el número de particiones de la columna en la primera matriz tiene que ser igual al número de particiones de fila de la segunda matriz.

- **División de Matrices [3]**

Aunque la división de matrices permanece a menudo sin definir, MATLAB define las operaciones división por la izquierda y división por la derecha. La división por la izquierda (\) significa:

$$\boldsymbol{Y} = \boldsymbol{A} \setminus \boldsymbol{X} \quad Þ \quad \boldsymbol{Y} = \boldsymbol{A}^{-1} * \boldsymbol{X} \qquad (1.10.)$$

y la división por la derecha (/) significa:

$$\mathbf{Y} = \mathbf{X} / \mathbf{A} \quad Þ \quad \mathbf{Y} = \mathbf{X} * \mathbf{A}^{-1} \qquad (1.11.)$$

Estos operadores tienen las mismas prioridades que la multiplicación, con el orden usual de izquierda a derecha de las operaciones de la misma prioridad, así:

$$\boldsymbol{A} + \boldsymbol{B} * \boldsymbol{C} \setminus \boldsymbol{D} = \boldsymbol{A} + ((\boldsymbol{B} * \boldsymbol{C}) \setminus \boldsymbol{D}) \tag{1.12.}$$

Por supuesto, las matrices en cada operación deben ser conformes.

1.1.3 Inversión de matrices [9]

Debido a que la división entre matrices no está definida, se debe premultiplicar por la matriz inversa de la matriz que se desea dividir. Se dice que una matriz cuadrada ***A*** es invertible, si existe una matriz ***B*** que cumpla la siguiente propiedad:

$$\boldsymbol{AB} = \boldsymbol{BA} = \boldsymbol{I}$$

siendo ***I*** la matriz identidad. Denominamos a la matriz ***B*** la inversa de ***A*** y la denotamos por $\boldsymbol{A}^{-1}$.

A continuación se presentará una forma para encontrar la matriz inversa de una matriz.

Sea $\boldsymbol{A} = (a_{ij})$ una matriz cuadrada de orden *n*. Para calcular la matriz inversa de ***A***, que denotaremos como $\boldsymbol{A}^{-1}$, seguiremos los siguientes pasos:

Paso 1. Construir la matriz *n x 2n*, $\boldsymbol{M} = (\boldsymbol{A} \vdots \boldsymbol{I})$ esto es, ***A*** está en la mitad izquierda de ***M*** y la matriz identidad ***I*** en la derecha.

Paso 2. Se deja tal y como está la primera fila de *M*, y debajo del primer término de la diagonal principal, a_{11}, que llamaremos *pivote*, ponemos ceros. Luego se opera como se indica en el siguiente ejemplo.

Consideremos una matriz 3 x 3 arbitraria

$$\boldsymbol{A} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{13} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}$$

Paso 1.

$$\boldsymbol{M = (A \vdots I)} = \left(\begin{array}{ccc|ccc} a_{11} & a_{12} & a_{13} & 1 & 0 & 0 \\ a_{21} & a_{22} & a_{23} & 0 & 1 & 0 \\ a_{31} & a_{32} & a_{33} & 0 & 0 & 1 \end{array}\right)$$

Paso 2.

$$\left(\begin{array}{ccc|ccc} a_{11} & a_{12} & a_{13} & 1 & 0 & 0 \\ 0 & a_{11}a_{22} - a_{21}a_{12} & a_{11}a_{23} - a_{21}a_{13} & a_{11}0 - a_{21}1 & a_{11}1 - a_{21}0 & a_{11}0 - a_{21}0 \\ 0 & a_{11}a_{32} - a_{31}a_{12} & a_{11}a_{33} - a_{31}a_{13} & a_{11}a_{22} - a_{21}a_{12} & a_{11}0 - a_{21}0 & a_{11}1 - a_{21}0 \end{array}\right)$$

El siguiente paso es igual que el anterior, pero esta vez se toma como pivote el segundo término de la diagonal principal.

Al llegar al último término de la diagonal, se procede igual que antes, pero poniendo los ceros encima del nuevo pivote. Se observa que al coger como pivote el último término de la diagonal, la matriz ***A*** se transforma en una matriz triangular. Una vez realizados todos los pasos, la mitad izquierda de la matriz ***M*** se convierte en una matriz diagonal. En este momento hay que proceder a transformar, si es que no lo está, la mitad izquierda en la matriz identidad, dividiendo si fuera necesario las filas de ***M*** por un escalar.

Ejemplo 1.4.

Supongamos que queremos encontrar la inversa de

$$\boldsymbol{A} = \begin{pmatrix} 1 & 0 & 2 \\ 2 & -1 & 3 \\ 4 & 1 & 8 \end{pmatrix}$$

Primero construimos la matriz ***M = (A ⋮ I)***,

$$\boldsymbol{M} = \left(\begin{array}{ccc|ccc} 1 & 0 & 2 & 1 & 0 & 0 \\ 2 & -1 & 3 & 0 & 1 & 0 \\ 4 & 1 & 8 & 0 & 0 & 1 \end{array}\right) \sim \left(\begin{array}{ccc|ccc} 1 & 0 & 2 & 1 & 0 & 0 \\ 0 & -1-2\cdot 0 & 3-2\cdot 2 & 0-2 & 1-2\cdot 0 & 0 \\ 0 & 1-4\cdot 0 & 8-4\cdot 2 & 0-4 & 0 & 1-0 \end{array}\right)$$

$$\sim\begin{pmatrix} 1 & 0 & 2 & \vdots & 1 & 0 & 0 \\ 0 & -1 & -1 & \vdots & -2 & 1 & 0 \\ 0 & 1 & 0 & \vdots & -4 & 0 & 1 \end{pmatrix},$$

luego se toma como pivote $a_{22} = -1$,

$$\sim\begin{pmatrix} 1 & 0 & 2 & \vdots & 1 & 0 & 0 \\ 0 & -1 & -1 & \vdots & -2 & 1 & 0 \\ 0 & 0 & 0-(-1) & \vdots & 4-(-2) & 0-1 & -1-0 \end{pmatrix}$$

$$\sim\begin{pmatrix} 1 & 0 & 2 & \vdots & 1 & 0 & 0 \\ 0 & -1 & -1 & \vdots & -2 & 1 & 0 \\ 0 & 0 & 1 & \vdots & 6 & -1 & -1 \end{pmatrix}\sim$$

La mitad izquierda de ***M*** está en forma triangular, por consiguiente, ***A*** es invertible. Si hubiera quedado toda una fila con ceros en la mitad ***A*** de ***M***, la operación habría terminado (***A*** no es invertible).

A continuación, tomamos como pivote a_{33}, ponemos ceros encima de éste y seguimos operando hasta que nos quede una matriz diagonal.

$$\sim\begin{pmatrix} 1 & 0 & 0 & \vdots & -11 & 2 & 2 \\ 0 & -1 & 0 & \vdots & 4 & 0 & -1 \\ 0 & 0 & 1 & \vdots & 6 & -1 & -1 \end{pmatrix}.$$

Ya que la matriz colocada en la mitad izquierda es diagonal, no hay que operar más. Transformamos la matriz diagonal en una matriz identidad; para ello hay que dividir la segunda fila entre -1:

$$\sim\begin{pmatrix} 1 & 0 & 0 & \vdots & -11 & 2 & 2 \\ 0 & 1 & 0 & \vdots & -4 & 0 & 1 \\ 0 & 0 & 1 & \vdots & 6 & -1 & -1 \end{pmatrix}.$$

La matriz que ha quedado en la mitad derecha de ***M*** es precisamente la matriz inversa de ***A***:

$$A^{-1} = \begin{pmatrix} -11 & 2 & 2 \\ -4 & 0 & 1 \\ 6 & -1 & -1 \end{pmatrix}$$

Para comprobar si el resultado es correcto, se procede a multiplicar $\boldsymbol{AA^{-1}}$, teniendo que dar como resultado la matriz identidad ***I.***

Comprobación:

AA⁻¹ = I

$$\begin{pmatrix} 1 & 0 & 2 \\ 2 & -1 & 3 \\ 4 & 1 & 8 \end{pmatrix} \begin{pmatrix} -11 & 2 & 2 \\ -4 & 0 & 1 \\ 6 & -1 & -1 \end{pmatrix} = \begin{pmatrix} -11+0+12 & 2+0-2 & 2+0-2 \\ -22+4+18 & 4+0-3 & 4-1-3 \\ -44-4+48 & 8+0-8 & 8+1-8 \end{pmatrix} =$$

$$= \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = I.$$

1.2. DETERMINANTES. DEFINICIÓN Y PROPIEDADES

1.2.1 Definición

El determinante es una función que le asigna a una matriz de orden *n*, un único número real (o complejo) llamado el determinante de la matriz. Si **A** es una matriz de orden *n*, el determinante de la matriz **A** lo denotaremos por *det(A)* o también por |A| (las barras no significan valor absoluto). Un determinante A = A (a_{ij})**,** donde i, j = 1…n, se escribe de la forma

$$A = A(a_{ij}) = \begin{vmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{vmatrix}$$

Tienen las siguientes propiedades de definición:

1. El valor del determinante A no cambia si la suma de los elementos de una fila (columna) y los elementos correspondientes de alguna otra fila (columna) reemplazan los elementos de esta fila (columna).

2. El valor del determinante A cambia por un factor k si todos los elementos de alguna fila (columna) cambian por el factor k.

3. El valor del determinante A es igual a uno si todos los elementos sobre la diagonal principal (a_{kk}, donde k=1…n) son de valor uno mientras que los elementos fuera de la diagonal son cero.

Estas tres propiedades fundamentales llevan a las siguientes propiedades adicionales:

4. El valor del determinante A no cambia si la suma de los elementos de una fila (columna) y el producto de un factor arbitrario por los elementos correspondientes de cualquier otra fila (columna) reemplazan los elementos de cualquier fila (columna).

5. El signo algebraico del determinante A cambia si los elementos de dos Filas (columnas) son intercambiadas.

6. El valor del determinante A es cero si todos los elementos de alguna fila (columna) son cero, o si los elementos correspondientes de dos filas (columnas) son iguales o tienen una relación en común.

1.2.2 Menores y Cofactores

Si A es un determinante, el menor del elemento a_{ij} se denota por M_{ij} y se define como el determinante que queda después de quitar el i-ésimo renglón y la j-ésima columna de A. El número $(-1)^{i+j} M_{ij}$ se denota por C_{ij} y se denomina cofactor del elemento a_{ij}.

1.3. SISTEMAS DE ECUACIONES LINEALES [12]

Una ecuación de la forma $ax + by = c$ se llama una ecuación lineal en x e y. Así mismo, una ecuación lineal en tres variables x, y, z, es una ecuación de la forma $ax + by + cz = d$, donde los coeficientes son números reales. En forma similar se definen las ecuaciones lineales de cualquier número de variables. Si x_1, x_2, $x_3,\ldots,x_n$ (donde n es un número entero positivo) denotan variables, entonces una expresión de la forma

$$a_1x_1 + a_2x_2 + \ldots + a_nx_n = b \tag{1.13.}$$

en la cual $a_1, a_2,\ldots,a_n$ y b son números reales, es una ecuación lineal en n variables con coeficientes reales.

Una solución de una ecuación lineal es una sucesión de n números $s_1, s_2,\ldots,s_n$ tales, que satisface la ecuación al sustituir $x_1= s_1$, $x_2= s_2$,…, $x_n= s_n$. Al conjunto formado por todas las soluciones de la ecuación se llama conjunto solución.

Un conjunto finito de ecuaciones lineales en las variables $x_1, x_2, \ldots, x_n$ recibe el nombre de sistema de ecuaciones lineales. Una sucesión de números $s_1, s_2, \ldots, s_n$, es solución del sistema si las substituciones $x_1 = s_1$, $x_2 = s_2, \ldots, x_n = s_n$, son una solución para cada una de las ecuaciones en el sistema.

En un sistema de ecuaciones lineales se cumple uno solo de los siguientes enunciados:

- El sistema tiene un número infinito de soluciones.
- El sistema no tiene soluciones.
- El sistema tiene exactamente una solución.

Un sistema arbitrario de **m** ecuaciones lineales con **n** incógnitas tiene la siguiente configuración:

$$\begin{array}{ccccccccc} a_{11}x_1 & + & a_{12}x_2 & + & \ldots & + & a_{1n}x_n & = & b_1 \\ a_{21}x_1 & + & a_{22}x_2 & + & \ldots & + & a_{2n}x_n & = & b_2 \\ \vdots & & \vdots & & & & \vdots & & \vdots \\ a_{m1}x_1 & + & a_{m2}x_2 & + & \ldots & + & a_{mn}x_n & = & b_m \end{array} \tag{1.14.}$$

Los a_{ij} son números dados llamados coeficientes del sistema. Los b_i son también números dados. Si todos los b_i son cero, entonces el sistema anterior es homogéneo. Si al menos un b_i es diferente de cero, entonces el sistema es llamado sistema no homogéneo.

El empleo de dos subíndices para los coeficientes de las incógnitas es una notación muy útil que se adopta para determinar la colocación de los coeficientes en el sistema. El primer subíndice del coeficiente a_{ij} indica la ecuación en la cual aparece el coeficiente y el segundo subíndice, qué incógnita multiplica.

Si el sistema de ecuaciones es homogéneo él tiene, al menos, una solución denominada solución trivial:

$$x_1 = 0, \ldots, x_n = 0.$$

1.3.1 Forma Matricial de un Sistema Lineal.

Un sistema de n ecuaciones puede ser escrito como una sola ecuación vectorial:

$$\mathbf{A\ x = b} \tag{1.15.}$$

donde la **matriz de coeficientes A= [a_{ij}],** es la matriz de **m x n** y **x** y **b** son vectores columna:

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} \qquad x = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} \qquad b = \begin{bmatrix} b_1 \\ \vdots \\ b_m \end{bmatrix}$$

Se asume que los coeficientes a_{ij} no son todos cero, por lo tanto **A** no es la matriz cero. Debe notarse que **x** tiene n componentes, mientras que **b** tiene m componentes.

1.3.2 Regla de Crámer para un Sistema Lineal de n Ecuaciones.

Sea un sistema de n ecuaciones lineales con n incógnitas, $x_1, x_2, \ldots, x_n$,

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n &= b_2 \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ a_{n1}x_1 + a_{n2}x_2 + \ldots + a_{nn}x_n &= b_n \end{aligned} \qquad (1.16.)$$

Llamando D al determinante de los coeficientes de $x_1, x_2, \ldots, x_n$, es decir,

$$D = \begin{vmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & a_{n2} & \cdots & a_{nn} \end{vmatrix}$$

y si se representa por D_k el determinante D en el que la columna k (que corresponde a los coeficientes de la incógnita x_k) se ha reemplazado por la columna de los términos independientes, pasados al segundo miembro, se tiene:

$$x_1 = \frac{D_1}{D}, \quad x_2 = \frac{D_2}{D}, \ldots, \quad x_n = \frac{D_n}{D}, \ldots D \neq 0 \qquad (1.17.)$$

Si D = 0, el sistema de ecuaciones no tiene solución ó es incompatible.

Si el sistema de ecuaciones (1.15.) es homogéneo y $D \neq 0$ el sistema sólo tiene la solución trivial $x_1=0, x_2=0, \ldots x_n=0$.

Ejemplo 1.2.

Obtener la solución del siguiente sistema de ecuaciones, empleando la regla Crámer.

$$-I_1 - 4I_2 + 2I_3 + I_4 = -32$$

$$2I_1 - I_2 + 7I_3 + 9I_4 = 14$$

$$-I_1 + I_2 + 3I_3 + I_4 = 11$$

$$I_1 - 2I_2 + I_3 - 4I_4 = -4$$

Se definen, inicialmente los cinco determinantes con base en los coeficientes y los términos independientes del sistema por resolver.

$$D_1 = \begin{vmatrix} -32 & -4 & 2 & 1 \\ 14 & -1 & 7 & 9 \\ 11 & 1 & 3 & 1 \\ -4 & -2 & 1 & -4 \end{vmatrix} = -2115 \;;\; D_2 = \begin{vmatrix} -1 & -32 & 2 & 1 \\ 2 & 14 & 7 & 9 \\ -1 & 11 & 3 & 1 \\ 1 & -4 & 1 & -4 \end{vmatrix} = -3384$$

$$D_3 = \begin{vmatrix} -1 & -4 & -32 & 1 \\ 2 & -1 & 14 & 9 \\ -1 & 1 & 11 & 1 \\ 1 & -2 & -4 & -4 \end{vmatrix} = -1269 \;;\; D_4 = \begin{vmatrix} -1 & -4 & 2 & -32 \\ 2 & -1 & 7 & 14 \\ -1 & 1 & 3 & 11 \\ 1 & -2 & 1 & -4 \end{vmatrix} = 423$$

$$D = \begin{vmatrix} -1 & -4 & 2 & 1 \\ 2 & -1 & 7 & 9 \\ -1 & 1 & 3 & 1 \\ 1 & -2 & 1 & -4 \end{vmatrix} = -423$$

Se despejan las incógnitas, así:

$$I_1 = \frac{D_1}{D} = \frac{-2115}{-423} = 5, \qquad I_2 = \frac{D_2}{D} = \frac{-3384}{-423} = 8$$

$$I_3 = \frac{D_3}{D} = \frac{-1269}{-423} = 3, \qquad I_4 = \frac{D_4}{D} = \frac{423}{-423} = -1$$

1.3.3 Eliminación de Gauss.

La eliminación de Gauss es un método estándar para resolver sistemas lineales. Es un proceso de eliminación sistemática, un método de gran importancia que funciona eficientemente en la práctica y es razonable con respecto al tiempo de computación.
El método básico para resolver un sistema de ecuaciones lineales consiste en reemplazar el sistema dado por un nuevo sistema que tenga el mismo conjunto solución, pero que sea más fácil de resolver. Por lo general, este nuevo sistema se obtiene en una serie de etapas, aplicando las tres operaciones elementales para las ecuaciones:

- Multiplicar una ecuación por una constante diferente de cero.
- Intercambiar dos ecuaciones.
- Sumar un múltiplo de una ecuación a otra.

A estas tres operaciones elementales para ecuaciones les corresponden las siguientes tres operaciones elementales entre filas para matrices:

- Multiplicar una fila por una constante diferente de cero.
- Intercambiar dos filas.
- Sumar un múltiplo de una fila a otra.

Un sistema lineal $\mathbf{S_1}$ es equivalente por filas a un sistema lineal $\mathbf{S_2}$**,** si $\mathbf{S_1}$ se puede obtener de $\mathbf{S_2}$ por operaciones elementales entre filas.

Teorema (sistemas equivalentes por filas). Los sistemas equivalentes por filas tienen los mismos conjuntos de soluciones.

Un sistema lineal de ecuaciones se denomina **sobredeterminado** si tiene más ecuaciones que incógnitas, **determinado** si el número de ecuaciones es igual al número de incógnitas y **subdeterminado** si tiene menos ecuaciones que incógnitas.

Un sistema lineal se denomina **consistente** si tiene al menos una solución, es **inconsistente** si no tiene solución.

Ejemplo 1.3 Eliminación de Gauss si existen infinitas soluciones.

Resolver el siguiente sistema de ecuaciones lineales:

$$3.0I_1 + 2.0I_2 + 2.0I_3 - 5.0I_4 = 8.0$$

$$0.6I_1 + 1.5I_2 + 1.5I_3 - 5.4I_4 = 2.7$$

$$1.2I_1 - 0.3I_2 - 0.3I_3 + 2.4I_4 = 2.1$$

La matriz ampliada del sistema anterior es la siguiente:

$$\begin{bmatrix} 3.0 & 2.0 & 2.0 & -5.0 & \vdots & 8.0 \\ 0.6 & 1.5 & 1.5 & -5.4 & \vdots & 2.7 \\ 1.2 & -0.3 & -0.3 & 2.4 & \vdots & 2.1 \end{bmatrix}$$

Escogiendo como pivote el coeficiente de I_1 en la primera fila, es decir 3.0, se dividen los coeficientes de I_1 de la segunda y tercera filas por este pivote, anteponiéndole el signo negativo y se obtiene los valores -0.2 y -0.4. La prime fila se multiplica por cada uno de estos valores y se le adiciona a la segunda y tercera filas respectivamente, resultando la matriz:

$$\begin{bmatrix} 3.0 & 2.0 & 2.0 & -5.0 & \vdots & 8.0 \\ 0 & 1.1 & 1.1 & -4.4 & \vdots & 1.1 \\ 0 & -1.1 & -1.1 & 4.4 & \vdots & -1.1 \end{bmatrix}$$

Para eliminar a I_2 de la tercera fila se multiplica su coeficiente (-1.1) por el inverso del coeficiente de I_2 en la segunda fila, obteniéndose el factor por el cual se multiplica la segunda, producto éste que se le adiciona a la tercera fila, anteponiéndole el signo menos:

$$\begin{bmatrix} 3.0 & 2.0 & 2.0 & -5.0 & \vdots & 8.0 \\ 0 & 1.1 & 1.1 & -4.4 & \vdots & 1.1 \\ 0 & 0 & 0 & 0 & \vdots & 0 \end{bmatrix}$$

El sistema de ecuaciones correspondiente a esta última matriz ampliada es:

$$\begin{aligned} 3.0I_1 + 2.0I_2 + 2.0I_3 - 5.0I_4 &= 8 \\ 1.1I_2 + 1.1I_3 - 4.4I_4 &= 1.1 \\ 0 &= 0 \end{aligned}$$

Del sistema anterior la solución es, $I_2 = 1 - I_3 + 4I_4$, $I_1 = 2 - I_4$. Este sistema se redujo a uno de dos ecuaciones con cuatro incógnitas y por lo tanto se hace necesario obtener la solución de dos de las incógnitas (I_1 e I_2) como función de las otras dos (I_3 e I_4).

2. INTRODUCCIÓN DEL MATLAB

MATLAB responde a cada uno de sus comandos. Cuando se digita un carácter o una combinación de caracteres que él entiende, ejecuta la tarea apropiada, para este comando. De otro modo, MATLAB suministra una frase de diagnostico que permite saber por qué el comando es inapropiado.

Al teclear “help” hace que MATLAB liste todas las palabras y operadores que se pueden usar. Adicionalmente una operación o el nombre de una función después de “help” entrega información sobre esta operación o función. Por ejemplo:

```
>> help sin(seno)
```

SIN   SIN (x) is the sine of the elements of x (es el seno de los elementos de x)

La variable x está en radianes. Al digitar **Help** MATLAB lo verá como un caso sensible. La documentación de ayuda aquí es engañosa, porque sin (no SIN) es la función seno de MATLAB.

Tabla 2.1. Operadores generales [3]

Palabra	Operación
help	Describe operadores y funciones del MATLAB
who	Lista nombres de variables
whos	Lista nombres de variadles y dice su tamaño
what	Lista los archivos M-Files en el espacio de trabajo
size	Entrega las dimensiones del argumento
length	Regresa la dimensión máxima del argumento
clear	Remueve todas las variables del espacio de trabajo
quit	Finaliza la sesión en MATLAB
exit	Finaliza la sesión en MATLAB
save	Guarda un archivo en una carpeta de MATLAB
load	Carga un archivo guardado en una carpeta de MATLAB

Cuando se digita una expresión sin crear una asignación, MATLAB asigna el resultado a la variable **ans**. Los nombres de las variables empiezan con algún carácter del alfabeto. Después del primer carácter en un nombre, se puede usar cualquier otro carácter alfanumérico o carácter subrayado (_) como se desee, pero MATLAB recuerda sólo los primeros diecinueve caracteres.

El uso de **who** o **whos** lista los nombres de todas las variables utilizadas en una sesión de trabajo. **whos** entrega mas detalles que **who**, el cual lista justamente los nombres. **size** retorna un arreglo cuyos elementos son el número de filas y columnas en el argumento de la matriz. **length** retorna un valor escalar igual a la longitud de su vector argumento. Digitando **clear** remueve todas las variables del espacio de trabajo. Usando **save** almacena todas las variables en un archivo con el nombre “MATLAB.MAT.” Digitando load se llama el contenido de lo más reciente guardado en MATLAB.MAT.

2.1. VARIABLES Y CARACTERES ESPECIALES [3]

MATLAB tiene caracteres especiales que se pueden usar para controlar MATLAB o para formar expresiones. La tabla 2.2. entrega un resumen de estos caracteres. El símbolo de porcentaje (%) estipula una manera para documentar el trabajo al ser digitado. Se digita:

```
>> %  esto es un comentario
```

Tabla 2.2. Caracteres del MATLAB

Carácter	Función
[ ]	Delinea vectores y matrices
( )	Controla operadores de precedencia
´	Separa una función y suscribe argumentos o separa declaraciones en un comando de línea
;	Termina filas en una matriz o suprime el contenido precedente
%	Denota un comentario
:	Vector o generador de arreglo
!	Ejecuta una función del sistema solo en (MS-DOS)

Tabla 2.3. Variables de MATLAB

variable	Valor
ans	Variable que se asigna por omisión de una
eps	Valor de precisión de punto flotante
pi	Valor de p
i o j	$\sqrt{-1}$
Inf	Infinito (¥)
NaN	Definición IEEE de No es un Número
flops	Operación de conteo en punto flotante
nargin	Número de función de argumentos de entrada
nargout	Número de función de argumentos de salida

MATLAB ignora la línea y muestra otro símbolo (>>). El símbolo de porcentaje puede aparecer en la línea siguiendo una declaración. Por ejemplo:

>> 123 % Este es un comentario

La omisión del símbolo de porcentaje produce un mensaje de error. Dos o más declaraciones pueden aparecer en una sola línea, si se separa cada declaración de las que están al lado con una coma o un punto y coma, como se muestra:

```
>>123;  456,  789;
ans  = 456
```

Este ejemplo muestra el punto y coma, en adición como un separador de declaraciones, suprime la impresión de la declaración precedente. Los corchetes ([ ]), la coma (,) y los dos puntos (:) aplican para una generación de vectores o matrices.

MATLAB define variables especiales que aparecen en la tabla 2.3. Ya se conoce **ans** la cual almacena el resultado de una expresión que no tiene un nombre asignado. La variable aleatoria **ans** puede aparecer en alguna expresión, justo como una variable cualquiera que se define. La constante **eps** contiene el valor de precisión de un punto flotante de MATLAB. La variable **pi** almacena el valor numérico de **π**. Las letras **i** ó **j** representan $\sqrt{-1}$. **inf** representa infinito, y **NaN** representa un resultado no definido de una operación según la IEEE como (0.0/0.0).

La variable **flops** acumula el número de operaciones de punto flotante. **Flops** (0) inhabilita el conteo. Las variables **nargin** y **nargout** contienen el actual número de argumentos de entrada y salida de una función. Usando **clear** remueve todas las variables del espacio de trabajo, excepto para **eps**, inhabilita las definiciones de las variables especiales. Si se trabaja con números complejos se debe tener cuidado de no usar i ó j como índices.

2.2. GUARDAR Y LLAMAR UNA SESIÓN [3]

Cuando se trabaja con MATLAB en el computador, éste se puede apagar y volver después para resumir la actual sesión. Para cumplir este fin, sin tener que teclear todas las variables que se han usado, antes de apagar el computador se debe digitar:

```
>> save
```

Este comando crea o sobrescribe un archivo de MATLAB (MAT-file) en el actual subdirectorio o en la carpeta de MATLAB con el nombre “MATLAB.MAT.” Después se podrá llamar al espacio de trabajo debido a que existe en “MATLAB.MAT”.

Para llamar el archivo se digita:

```
>> load
```

2.3. ESCALARES [3]

La unidad básica en MATLAB es una matriz que tiene un número diferente de filas y columnas. Cuando se tiene una fila y una columna la matriz es un escalar.

2.3.1 Creación de una variable escalar

La creación de un escalar simple implica escribir una asignación. Por ejemplo, digitando

```
>>k=1.38e-23
```

define la variable k, la cual se puede usar después en cálculos que envuelvan la constante de Boltzmann. En notación exponencial se puede usar la letra minúscula e ó la letra mayúscula E. MATLAB no distingue entre un punto flotante y un numero entero. Por ejemplo, cuando se digita:

```
>>b=6.00
```

MATLAB responde con

```
b=6
```

El comando **format** indica algún control de la apariencia de las respuestas de MATLAB. Estos comandos aparecen en la tabla 2.4. Algunos de éstos se muestran:

```
>>format short
>>6.2
ans = 6.2000
>>format short e
>>6.2
ans = 6.2000e+00
>>format long
>>6.2
ans = 6.20000000000000
```

Tabla 2.4. Tipos de apariencia para un resultado

Comandos	Efecto
format short	Punto fijo, cinco dígitos
format long	Punto fijo, 15 dígitos
format short e	Punto flotante, 5 dígitos
format long e	Punto flotante, 15 dígitos
format hex	Hexadecimal
format +	+,-, y blanco para positivo, negativo y cero
format bank	Punto fijo, dos dígitos
format compact	Suprimir una alimentación de extralínea
format loose	Incluye una alimentación de extralínea

2.3.2 Operaciones aritméticas

Además de las operaciones aritméticas básicas +, -, *, / y paréntesis, MATLAB define la división por la izquierda (\ ). Por ejemplo

```
>> 4\1
ans= 0.2500
```

lo que muestra la división por la izquierda es que divide el operador de la derecha por el de la izquierda.

2.3.3 Relaciones y operadores lógicos

En la tabla 2.5. se muestran los operadores de relación usados en MATLAB. Para los valores entregados la relación puede ser verdadera (1) o falsa (0). Por ejemplo:

```
>> 4<5
ans = 1

>> 5<4
ans = 0

>>0|1
ans = 1
```

```
4<=4
ans = 1
>>~ ans
ans = 0

>>0&1
ans = 0
```

Cualquier número diferente de cero, positivo o negativo es verdadero. Los operadores aritméticos toman prelación sobre operadores de relación.

```
>> 5+3>2
ans = 1
```

```
>> 5+ (3>2)
ans = 6
```

Los operadores de relación toman prelación sobre operadores lógicos, por ejemplo:

```
>> 2&0.5>-0.4
ans = 1
```

Tabla 2.5. Operadores de relación y lógicos

Operador	Consecuencia
<	Menor que
<=	Menor o igual a
>	Mayor que
>=	Mayor o igual a
==	Igual
~=	Diferente
&	Operador lógico AND
|	Operador lógico OR
~	Negador

Tabla 2.6. Funciones matemáticas básicas

Función	Comentario
abs	Valor absoluto o magnitud del número complejo
angle	Fase de un número complejo en radianes
sqrt	Raíz cuadrada
real	Parte real del número complejo
imag	Parte imaginaria del número complejo
conj	Complejo conjugado del número complejo
round	Redondea lo más cerca del número entero
fix	Redondea la parte decimal a cero
floor	Redondea al número entero inferior
ceil	Redondea al número entero superior
sign	Signo(devuelve 1, 0 o -1 como argumento si es positivo, cero o negativo)
rem	Resto
exp	Base exponencial (e = 2.7183)
log	Logaritmo natural
log10	Logaritmo en base 10

2.3.4 Funciones básicas

La tabla 2.6. muestra funciones matemáticas básicas disponibles. Por ejemplo:

```
rem (12,7)
ans = 5
```

```
>>exp (1)
ans = 2.7183
```

```
>>fix (3.5)
ans = 3
```

```
>>floor (3.5)
ans = 3
```

```
>>ceil (3.5)
ans = 4
```

```
>>round (3.5)
ans = 4
```

2.3.5 Funciones hiperbólicas y trigonométricas

MATLAB tiene las funciones trigonométricas e hiperbólicas usuales. Estas funciones se enumeran en la tabla 2.7. El argumento de las funciones trigonométricas y el resultado de sus inversas está en radianes.

Tabla 2.7. Funciones trigonométricas e hiperbólicas

Función	Comentario
sin	Seno
cos	Coseno
tan	Tangente
asin	Seno inverso
acos	Coseno inverso
atan	Tangente inverso
atan2	Tangente inverso (y,x) en el cuarto cuadrante
sinh	Seno hiperbólico
cosh	Coseno hiperbólico
tanh	Tangente hiperbólica
asinh	Seno hiperbólico inverso
acosh	Coseno hiperbólico inverso
atanh	Tangente hiperbólica inverso

Note que el primer argumento de la función atan2 es la componente **y**, el segundo es la componente **x**.

2.4. NÚMEROS COMPLEJOS

MATLAB trabaja con números complejos. En realidad, todas las operaciones aritméticas, relaciones y operadores lógicos, funciones básicas, trigonométricas e hiperbólicas se aplican al cálculo de números complejos. Los operadores lógicos consideran los números complejos con su parte real o imaginaria diferente de cero como un verdadero lógico. Excepto por igual (==) y desigual (~=), los operadores de relación utilizan la parte real de sus operadores complejos. Los operadores igualdad y desigualdad requieren que los números complejos sean iguales o diferentes. MATLAB define **i** o **j** como $\sqrt{-1}$. Se debe tener cuidado al redefinir ambos si se quiere continuar usando su representación. Usando el comando **clear** borra el espacio de trabajo y restituye la identidad de **i** o **j**. MATLAB usa i o j algebraicamente, no como etiqueta. Por ejemplo

```
>> 1+3*j
ans = 1.0000 + 3.0000i

>> 2+5i
??? 2+5i
|
Falta de operador, coma, o punto y coma

Y

>>y= sqrt ((4 + j * 5) * (2 + 6 * j) / (3 – i * 6))
y= 0.1359 – 2.4533i

>> abs (y)  % Magnitud
ans = 2.4570

>>180*angle (y) / pi   % fase en grados
ans = - 86.8299
```

Sin embargo se puede colocar la parte imaginaria con i o j seguido inmediatamente después del valor imaginario. Repitiendo el ejemplo anterior se muestra

```
>> 1+3j                          >> 2 + 5i
ans = 1.0000 + 3.0000i           ans = 2.0000 + 5.0000i


>> y = sqrt ((4 + 5j) *(2 + 6j) / (3 – 6j))
Y =0.1359 – 2.4533i
```

```
>> abs (y)     % magnitud
ans = 2.4570

>> 180*angle(y) / pi          % fase en grados
ans = -86.8299
```

2.5. CREACIÓN Y MANIPULACION DE VECTORES Y MATRICES

Como en casi todos los lenguajes de programación, en MATLAB las matrices y vectores son variables que tienen nombre. Se sugiere que se utilicen letras mayúsculas para matrices y letras minúsculas para vectores y escalares (MATLAB no exige esto, pero puede resultar útil).
Para definir una matriz no hace falta declararlas o establecer su tamaño (de hecho, se puede definir un tamaño y cambiarlo posteriormente). MATLAB determina el numero de filas y de columnas en función del número de elementos que se proporcionan (o se utilizan). Las matrices se definen o se introducen por medio de filas; los elementos de una fila están separados por blancos o comas, mientras que las filas están separadas por pulsaciones **intro** o por caracteres punto y coma (;). Por ejemplo, el siguiente comando define una matriz **A** de dimensión (3x3): [2]

```
>> A = [ 1 2 3; 4 5 6; 7 8 9]
```

La respuesta del programa es la siguiente

```
A =
        1       2       3
        4       5       6
        7       8       9
```

A partir de este momento la matriz **A** está disponible para hacer cualquier tipo de operación con ella (además de valores numéricos, en la definición de una matriz o vector se pueden utilizar expresiones y funciones matemáticas), por ejemplo, una sencilla operación con **A** es hallar su matriz transpuesta. En MATLAB el apóstrofo (‘) es el símbolo de transposición matricial. Para calcular **A’** (transpuesta de **A**) basta teclear lo siguiente:

```
>> A’
ans =
        1       4       7
        2       5       8
        3       6       9
```

Como el resultado de la operación no ha sido asignado a ninguna otra matriz, MATLAB utiliza un nombre de variable por defecto (**ans,** de answer), que contiene el resultado de la última operación. La variable **ans** puede ser utilizada como operador en la siguiente expresión que se introduzca. También podría haberse asignado el resultado a otra matriz llamada **B**:

```
>> B = A'
B =
        1       4       7
        2       5       8
        3       6       9
```

Ahora ya están definidas las matrices **A** y **B**, y es posible seguir con ellas. Por ejemplo, se puede hacer el producto **B*A** (deberá resultar una matriz simétrica):

```
>> B*A
ans =
        66      78      90
        78      93      108
        90      108     126
```

En MATLAB se accede a los elementos de un vector poniendo el índice entre paréntesis (por ejemplo **x(3)** ó **x(i)**). Los elementos de las matrices se acceden poniendo los dos índices entre paréntesis, separados por una coma (por ejemplo **A(1,2)** ó **A(i,j)**). Las matrices se almacenan por columnas (aunque se introduzcan por filas, como se ha dicho antes), y teniendo en cuenta ésto puede accederse a cualquier elemento de una matriz con un solo subíndice. Por ejemplo, si **A** es una matriz (3x3) se obtiene el mismo valor escribiendo **A(1,2)** que escribiendo **A(4)**.

Invertir una matriz es casi tan fácil como transponerla. A continuación se va a definir una nueva matriz **A** – no singular – en la forma:

```
>> A = [ 1  4  -3; 2  1  5; -2  5  3]
A =
        1       4       -3
        2       1       5
        -2      5       3
```

Ahora se va a calcular la inversa de **A** y el resultado se asignará a **B**. Para ello basta hacer uso de la función **inv( )** (la precisión o numero de cifras con que se muestra el resultado se puede cambiar con el menú **File/Preferentes/General**):

```
B = inv (A)
B =
        0.1803      0.2213      -0.1885
        0.1311      0.0246      0.0902
        -0.0984     0.1076      0.0574
```

Para comprobar que este resultado es correcto basta pre-multiplicar **A** por **B**:

```
>> B*A
ans =
        1.0000      0.0000      0.0000
        0.0000      1.0000      0.0000
        0.0000      0.0000      1.0000
```

De forma análoga a las matrices, es posible definir un **vector fila x** en la forma siguiente (si los tres números están separados por blancos o comas, el resultado será un vector fila):

```
>> x = [10  20  30]        % vector fila

x =
        10      20      30
```

Por el contrario, si los números están separados por **intros** o **puntos y coma (;)** se obtendrá un **vector columna**:

```
>> y = [11; 12; 13]        % vector columna
y =
        11
        12
        13
```

MATLAB tiene en cuenta la diferencia entre **vectores fila** y **vectores columna**. Por ejemplo, si se intenta sumar los vectores **x** e **y** se obtendrá el siguiente mensaje de error:

```
>> x+y
??? Error using ==> +
Matrix dimensions must agree.
```

Estas dificultades desaparecen si se suma **x** con el vector transpuesto de **y**:

```
>> x+y’
ans =
        21      32      43
```

MATLAB considera vectores fila por defecto, como se ve en el ejemplo siguiente:

```
>> x(1) = 1 ,  x(2) = 2
x =
       1
x =
       1       2
```

2.5.1 Operaciones con matrices

- **Operadores aritméticos**

MATLAB puede operar con matrices por medio de **operadores** y por medio de **funciones.** Se han visto ya los operadores suma (+), producto (*) y transpuesta (‘), así como la función invertir **inv( ).** Los operadores matriciales del MATLAB son mostrados en la tabla 2.8.

Estos operadores se aplican también a las variables o valores escalares, aunque con algunas diferencias. Todos estos operadores son coherentes con las correspondientes operaciones matriciales: no se puede por ejemplo sumar matrices que no sean del mismo tamaño. Si los operadores no se usan de modo correcto se obtiene un mensaje de error.

Tabla 2.8. Operadores aritméticos de matrices[2]

operador	comentario
+	Adición o suma
-	Sustracción o resta
*	Multiplicación
‘	Transpuesta
^	Potenciación
\	División - izquierda
/	División - derecha
.*	Producto elemento a elemento
./ y .\	División elemento a elemento
.^	Elevar a una potencia elemento a elemento

Los operadores anteriores se pueden aplicar también de modo **mixto,** es decir con un operador escalar y otro matricial. En este caso la operación con el escalar se aplica a cada uno de los elementos de la matriz. Considere el siguiente ejemplo

```
>> A = [1  2; 3  4]
A =
        1        2
        3        4

        >> A*2
        ans =
                2        4
                6        8

        >> A – 4
        ans =
                -3       -2
                -1        0
```

- **Operadores para la resolución de sistemas de ecuaciones lineales**

MATLAB utiliza **operadores de división** para la resolución de sistemas de ecuaciones lineales. Por su gran importancia, estos operadores requieren una explicación detenida. Considere el siguiente sistema de ecuaciones lineales,

Ax = **b**

Donde **x** y **b** son vectores columna, y **A** una matriz cuadrada invertible. La resolución de este sistema de ecuaciones se puede escribir en las dos formas siguientes (¡atención a la 2ª forma, basada en la **barra invertida (\),** que puede resultar un poco extraña!):

x= **inv(A)*b**
x= **A\b**

Así pues, el operador **división por la izquierda** (barra invertida \) equivalente a pre-multiplicar por la inversa de esa matriz. En realidad este operador es **más general** y **más inteligente** de lo que aparece en el ejemplo anterior: el operador división-izquierda es aplicable aunque la matriz no tenga inversa e incluso no sea cuadrada, en cuyo caso la solución que se obtiene (por lo general) es la que proporciona el método de los mínimos cuadrados.

- **Operadores elemento a elemento**

En MATLAB existe también la posibilidad de aplicar **elemento a elemento** los operadores matriciales (*,^, \,y /). Para ello basta precederlos por un punto (.). Por ejemplo:

```
>> [1  2  3  4] ^2
???Error using   ==>   ^
Matrix must be square.

>> [1  2  3  4] .^2
ans =
        1        4        9        16

>> [1  2  3  4] * [1  -1  1  -1]
??? Error using   ==>   *
Inner matrix dimensions must agree.

ans =
        1       -2        3        -4
```

2.5.2 Tipos de datos

Ya se ha dicho que MATLAB es un programa preparado para trabajar con vectores y matrices. Como caso particular también trabaja con variables escalares (matrices de dimensión 1). MATLAB trabaja siempre en **doble precisión**, es decir guardando cada dato en 8 bytes, con unas 15 cifras decimales exactas. También con otros tipos de datos: **matrices de más de dos dimensiones, matrices dispersas, vectores y matrices de celdas, estructuras y clases y objetos**.

- **Números reales de doble precisión**

Los elementos constitutivos de vectores y matrices son números reales almacenados en 8 Bytes (53 bites para la mantisa y 11 para el exponente de 2; entre 15 y 16 cifras decimales equivalentes). Es importante saber cómo trabaja MATLAB con estos números y los casos especiales que presentan.

MATLAB mantiene una forma especial para los números muy grandes (más grandes que los que es capaz de representar), que son considerados como **infinito**. Por ejemplo, obsérvese cómo responde el programa al ejecutar el siguiente comando:

```
>> 1.0 / 0.0
Warning:  Divide by zero
ans =
        inf
```

Así pues, para MATLAB el **infinito** se representa como **inf** o **Inf**. MATLAB tiene también una representación especial para los resultados que no están definidos como números. Por ejemplo, ejecútense los siguientes comandos y obsérvense las respuestas obtenidas:

```
>> 0.0 / 0.0
Warning: Divide by zero
ans =
        NaN
>> inf / inf
ans =
        NaN
```

En ambos casos la respuesta es **NaN**, que es la abreviatura de **Not a Number**. Este tipo de respuesta, así como la de **Inf**, son enormemente importantes en MATLAB, pues permiten controlar la fiabilidad de los resultados de los cálculos matriciales. Los **NaN** se propagan al realizar con ellos cualquier operación aritmética, en el sentido de que, por ejemplo, cualquier número sumado a un **NaN** da otro **NaN**. MATLAB tiene esto en cuenta. Algo parecido sucede con los **Inf**.

MATLAB dispone de tres funciones útiles relacionadas con las operaciones de coma flotante. Estas funciones, que no tienen argumentos, son las siguientes:

eps devuelve la diferencia entre 1.0 y el número de coma flotante inmediatamente superior. Da una idea de la precisión o número de cifras almacenadas. En un **PC**, **eps** vale 2.2204e-016.

realmin devuelve el numero más pequeño con que se puede trabajar (2.2251e-308).

realmax devuelve el número más grande con que se puede trabajar (1.7977e+308).

- **Otros tipos de variables: INTIGER, FLOAT Y LOGICAL**

Como ya se ha comentado, por defecto MATLAB trabaja con variables de punto flotante o doble precisión (**double**). Con estas variables pueden resolverse casi todos los problemas prácticos y con frecuencia no es necesario complicarse la vida declarando variables de tipos distintos, como se hace con cualquier otro lenguaje de programación. Sin embargo, en algunos casos es conveniente declarar variables de otros tipos porque puede ahorrarse mucha memoria y pueden hacerse los cálculos mucho más rápidamente.

MATLAB permite crear variables enteras con 1, 2, 4 y 8 Bytes (8, 16, 32 y 64 bits). A su vez, estas variables pueden tener signo o no tenerlo. Las variables con signo representan números en intervalos "casi" simétricos respecto al 0; las variables sin signo representan números no negativos, desde el 0 al número máximo.

Los tipos de los enteros con signo son **int8, int16, int32** e **int64**, y sin signo **uint8, uint16, uint32** y **uint64**. Para crear una variable entera de un tipo determinado se utilizan sentencias como las siguientes:

```
>> i = int32(100) ;      % se crea un entero de 4 Bytes de valor 100
>> j = zeros(100);  i = int32(j);      % se crea un entero i a partir de j
>> i = zeros(1000, 1000, 'int32')   % se crea una matriz 1000x1000 de enteros
```

Las funciones **intmin ('int64')** e **intmax ('int64')** permiten por ejemplo saber el valor del entero más pequeño y más grade (en valor algebraico) que puede formarse con variables enteras de 64 bits:

```
>> disk ([intmin('int64'), intmax('int64')])
-9223372036854775808   9223372036854775807
```

La función **isinteger(i)** devuelve 1 si la variable i es entera y 0 en otro caso. La función **class(i)** devuelve el tipo de variable que es **i** (**int8, int16, …**), mientras que la función **isa(i, 'int16')** permite saber exactamente si la variable **i** corresponde a un entero de 16 bits.

MATLAB dispone de dos tipos de variables reales o **float**: **single** y **double**, que ocupan respectivamente 4 y 8 Bytes. Por defecto se utilizan **doubles**. Las funciones **single(x)** y **double(y)** permiten realizar conversiones entre ambos tipos de variables.

Las funciones **realmin** y **realmax** permiten saber los números **double** más pequeño y más grande (en valor absoluto) que aprueba el computador. Para los correspondientes números de simple precisión habría que utilizar **realmin(‘single’)** y **realmax(‘single’)**. La función **isfloat(x)** permite saber si **x** es una variable real, de simple o de doble precisión. Para saber exactamente de qué tipo de variable se trata se puede utilizar las funciones **isa(x, ‘single’)** ó **isa(x, ‘double’)**. Obsérvese el ejemplo siguiente, en el que se ve cómo con variables **single** se reduce el tiempo de CPU y la memoria:

```
>> n = 1000;    AA = rand (n);   A = single (AA);
>> tic,   Bs = inv(A);   toc
Elapsed time is 1.985000 seconds.
>> tic, Bd = inv(AA);   toc
Elapsed time is 4.296000 seconds.
```

Quizás las variables más interesantes –aparte de las variables por defecto, las **double**- sean las variables lógicas, que sólo pueden tomar valores **true(1)** y **false(0)**. Las variables lógicas surgen como resultado de los operadores relacionales (==, <, <=, >, >=, ~=) y de muchas funciones lógicas como **any** y **all** que se aplican a vectores y matrices.

La función **logical(A)** produce una variable lógica, con el mismo número de elementos que **A**, con valores 1 ó 0 según el correspondiente elemento de **A** sea distinto de cero o igual a cero.

Una de las aplicaciones más importantes de las variables lógicas es para separar o extraer los elementos de una matriz o vector que cumplen cierta condición, y operan luego selectivamente sobre dichos elementos. Obsérvese, el siguiente ejemplo:

```
>> A = magic (4)
A =
        16    2      3     13
        5     11    10      8
        9      7      6     12
        4     14     15      1

>> j = A>10
j =
        1       0       0       1
        0       1       0       0
        0       0       0       1
        0       1       1       0
```

```
>> isa(j, 'logical')
ans =
        1

>> A(j) = -10
A =
        -10      2      3    -10
        5      -10     10      8
        9        7      6    -10
        4      -10    -10      1
```

2.5.3 Números complejos: FUNCIÓN COMPLEX

En muchos cálculos matriciales los datos y/o los resultados no son reales sino complejos, con parte real y parte imaginaria. MATLAB trabaja sin ninguna dificultad con números complejos. Para ver como se representan por defecto los números complejos, ejecútense los siguientes comandos:

```
>> a = sqrt (-4)
a =
        0 + 2.0000i

>> 3 + 4j
ans =
        3.0000 + 4.0000i
```

En la entrada de datos de MATLAB se pueden utilizar indistintamente la i y la j para representar el número imaginario unidad (en la salida, sin embargo, puede verse que siempre aparece la i). Si la i y la j no están definidas como variables, puede intercalarse el signo (*). Esto no es posible en el caso de que sí estén definidas, porque entonces se utiliza el valor de la variable. En general, cuando se está trabajando con números complejos, conviene no utilizar la i como variable ordinaria, pues puede dar lugar a errores y confusiones. Por ejemplo, obsérvense los siguientes resultados:

```
>> i = 2
i =
        2

>> 2 +3i
ans =
        2.0000 + 3.0000i
```

```
>> 2 + 3*i
ans =
        8

>> 2 + 3*j
ans = 2.0000 + 3.0000i
```

cuando **i** y **j** son variables utilizadas para otras finalidades, como unidad imaginaria puede utilizarse también la función **sqrt(-1)**, o una variable a la que se haya asignado el resultado de esta función.

La asignación de valores complejos a vectores y matrices desde teclado puede hacerse de las dos formas, que se muestran en el ejemplo siguiente (conviene hacer antes **clear i**, para que i no esté definida como variable):

```
>> A = [1+2i  2+3i;  -1+i  2-3i ]
A =
        1.0000 + 2.0000i     2.0000 + 3.0000i
        -1.0000 + 1.0000i     2.0000 + 3.0000i

>> A = [ 1  2;  -1  2 ]  +  [ 2  3;  1  -3 ]*I    % En este caso el * es necesario

A =
        1.0000 + 2.0000i     2.0000 + 3.0000i
        -1.0000 + 1.0000      2.0000 – 3.0000i
```

Puede verse que es posible definir las partes reales e imaginarias por separado. En este caso sí es necesario utilizar el operador (*), según se muestra en el ejemplo anterior.

MATLAB dispone también de la función **complex**, que crea un número complejo a partir de dos argumentos que representan la parte real e imaginaria, como en el ejemplo siguiente:

```
>> complex (1,2)
ans =
        1.0     + 2.0000i
```

Es importante advertir que el operador de matriz transpuesta (‘), aplicado a matrices complejas, produce la matriz conjugada y transpuesta. Existe una función que permite hallar la matriz conjugada (conj ()) y el operador punto y apóstrofo (.’) que calcula simplemente la matriz transpuesta.

2.5.4 Variables y expresiones matriciales

Ya han aparecido algunos ejemplos de **variables** y **expresiones** matriciales. Ahora se va a tratar de generalizar un poco lo visto hasta ahora.

Una **variable** es un nombre que se le da a una entidad numérica, que puede ser una matriz, un vector o un escalar. El valor de esa variable, e incluso el tipo de entidad numérica que representa, pueden cambiar a lo largo de una sesión de MATLAB o a lo largo de la ejecución de un programa. La forma más normal de cambiar el valor de una variable es colocándola a la izquierda del **operador de asignación** (=).

Una expresión de MATLAB puede tener las dos formas siguientes: primero, asignando su resultado a una variable

variable = expresión

y segundo evaluando simplemente el resultado del siguiente modo,

expresión

en cuyo caso el resultado se asigna automáticamente a una variable interna del MATLAB llamada **ans**(de answer) que almacena el último resultado obtenido. Se considera por defecto que una expresión termina cuando se presiona **intro**. Si se desea que una expresión continué en la línea siguiente, hay que introducir **tres puntos** (…) antes de pulsar **intro**. También se pueden incluir varias expresiones en una misma línea separándolas por comas (,) o puntos y comas (;).

Si una expresión **termina en punto y coma** (;) su resultado se calcula, pero no se escribe en la pantalla. Esta posibilidad es muy interesante, tanto para evitar la escritura de resultados intermedios, como para evitar la impresión de grandes cantidades de números cuando se trabaja con matrices de gran tamaño.

A semejanza del lenguaje de programación **C**, **MATLAB distingue entre mayúsculas y minúsculas** en los nombres de las variables. Los **nombres de las variables** deben empezar siempre por una letra y pueden constar de hasta 63 letras y números. La función **namelengthmax** permite preguntar al programa por este número máximo de caracteres. El carácter guión bajo (_) se considera como una letra. A diferencia del lenguaje **C**, no hace falta declarar las variables que se vayan a utilizar. Esto hace que se deba tener especial cuidado con no utilizar nombres erróneos en las variables, porque no se recibirá ningún aviso del ordenador.

Cuando se quiere tener una relación de las variables que se han utilizado en la sesión de trabajo se puede utilizar el comando **who**. Existe otro comando llamado **whos** que proporciona además información sobre el tamaño, la cantidad de memoria ocupada y el carácter real o complejo de cada variable. Se sugiere utilizar de vez en cuando estos comandos en la sesión de MATLAB que se tiene abierta. Esta misma información se puede obtener gráficamente con el **Workspace Browser**, que aparece en el comando **View/Workspace** o activando la ventana correspondiente si estaba abierto.

El comando **clear** tiene varias formas posibles:

clear	sin argumentos, clear elimina todas las variables creadas previamente (excepto las variables globales).
clear A,b	borra las variables indicadas.
clear global	borra las variables globales.
clear functions	borra las funciones.
clear all	borra todas las variables, incluyendo las globales, y las funciones.

2.5.5 Otras formas de definir matrices

MATLAB dispone de varias formas de definir matrices. El introducirlas por el teclado sólo es práctico en casos de pequeño tamaño y cuando no hay que repetir esa operación muchas veces. Recuérdese que en MATLAB no hace falta definir el tamaño de una matriz. Las matrices toman tamaño al ser definidas y este tamaño puede ser modificado por el usuario mediante adición y/o borrado de filas y columnas. A continuación se van a ver otras formas más potentes y generales de definir y modificar matrices.

- **Tipos predefinidos de matrices**

Existen en MATLAB varias funciones orientadas a definir con gran facilidad matrices particulares. Algunas de estas funciones son las siguientes:

eye(4)	forma la matriz unidad de tamaño (4×4)
zeros (3,5)	forma una matriz de ceros de tamaño (3×5)
zeros(4)	ídem de tamaño (4×4)
ones(3)	forma una matriz de unos de tamaño (3×3)

ones(2,4)	idem de tamaño (2×4)
linspace(x1,x2,n)	genera un vector con n valores igualmente espaciados entre x1 y x2
logspace(d1,d2,n)	genera un vector con n valores espaciados logarítmicamente entre 10^d1 y 10^d2. Si d2 es pi, los puntos se generan entre 10^d1 y pi
rand(3)	forma una matriz de números aleatorios entre 0 y 1, con distribución uniforme, de tamaño (3×3)
rand(2,5)	idem de tamaño (2×5)
randn(4)	forma una matriz de números aleatorios de tamaño (4×4), con distribución normal, de valor medio 0 y varianza 1.
magic(4)	crea una matriz (4×4) con los números 1, 2, ... 4*4, con la propiedad de que todas las filas y columnas suman lo mismo
hilb(5)	crea una matriz de Hilbert de tamaño (5×5). La matriz de Hilbert es una matriz cuyos elementos (i,j) responden a la expresión (1/(i+j-1)). Esta es una matriz especialmente difícil de manejar por los grandes errores numéricos a los que conduce
invhilb(5)	crea directamente la inversa de la matriz de Hilbert
kron(x,y)	produce una matriz con todos los productos de los elementos del vector x por los elementos del vector y. Equivalente a x'*y, donde x e y son vectores fila
compan(pol)	construye una matriz cuyo polinomio característico tiene como coeficientes los elementos del vector pol (ordenados de mayor grado a menor)
vander(v)	construye la matriz de Vandermonde a partir del vector v (las columnas son las potencias de los elementos de dicho vector)

Existen otras funciones para crear matrices particulares. Con Help/ MATLAB **Help** se puede obtener información sobre todas las funciones disponibles en MATLAB, que aparecen agrupadas por categorías o por orden alfabético. En la categoría **Mathematics** aparecen la mayor parte de las funciones estudiadas en este apartado.

2.5.6 Formación de una matriz a partir de otras

MATLAB ofrece también la posibilidad de crear una matriz a partir de matrices previas ya definidas, por varios caminos posibles:

– recibiendo alguna de sus propiedades (como por ejemplo el tamaño),
– por composición de varias submatrices más pequeñas,
– modificándola de alguna forma.

A continuación se describen algunas de las funciones que crean una nueva matriz a partir de otra o de otras, comenzando por dos funciones auxiliares:

[m,n] = size(A)	devuelve el número de filas y de columnas de la matriz A. Si la matriz es cuadrada basta recoger el primer valor de retorno
N = length(x)	calcula el número de elementos de un vector x
zeros(size(A))	forma una matriz de ceros del mismo tamaño que una matriz A previamente creada
ones(size(A))	ídem con unos
A=diag(x)	forma una matriz diagonal A cuyos elementos diagonales son los elementos de un vector ya existente x
x=diag(A)	forma un vector x a partir de los elementos de la diagonal de una matriz ya existente A
diag(diag(A))	crea una matriz diagonal a partir de la diagonal de la matriz A
blkdiag(A,B)	crea una matriz diagonal de submatrices a partir de las matrices que se le pasan como argumentos

triu(A)	forma una matriz triangular superior a partir de una matriz A (no tiene por qué ser cuadrada). Con un segundo argumento puede controlarse que se mantengan o eliminen más diagonales por encima o debajo de la diagonal principal.
tril(A)	ídem con una matriz triangular inferior
rot90(A,k)	Gira k*90 grados la matriz rectangular A en sentido antihorario. k es un entero que puede ser negativo. Si se omite, se supone k=1
flipud(A)	halla la matriz simétrica de A respecto de un eje horizontal
fliplr(A)	halla la matriz simétrica de A respecto de un eje vertical
reshape(A,m,n)	Cambia el tamaño de la matriz A devolviendo una matriz de tamaño m×n cuyas columnas se obtienen a partir de un vector formado por las columnas de A puestas una a continuación de otra. Si la matriz A tiene menos de m×n elementos se produce un error.

Un caso especialmente interesante es el de crear una nueva matriz componiendo como submatrices otras matrices definidas previamente. A modo de ejemplo, ejecútense las siguientes líneas de comandos y obsérvense los resultados obtenidos:

```
>> A=rand(3)
>> B=diag(diag(A))
>> C=[A, eye(3); zeros(3), B]
```

En el ejemplo anterior, la matriz C de tamaño (6×6) se forma por composición de cuatro matrices de tamaño (3×3). Al igual que con simples escalares, las submatrices que forman una fila se separan con blancos o comas, mientras que las diferentes filas se separan entre sí con intros o puntos y comas. Los tamaños de las submatrices deben ser coherentes.

3. ANALISIS NODAL Y DE MALLAS

Este capítulo describe dos maneras para formular un conjunto de ecuaciones linealmente independientes, de un circuito lineal. El primer método, el análisis nodal, entrega la solución del circuito en términos de un conjunto de variables de voltajes de nodo linealmente independientes. El segundo método es el análisis de mallas, el cual halla la solución para un circuito plano usando un conjunto de variables de corriente de malla linealmente independiente.

3.1. ANÁLISIS NODAL [3]

El análisis nodal formula una solución de circuito usando un conjunto de ecuaciones linealmente independientes según la ley de Kirchhoff de corrientes, escritas en términos de voltajes de nodo. Usando los valores de estos voltajes de nodo y los valores de las fuentes independientes, se puede determinar el valor de cualquier variable restante del circuito. Se asume que el circuito es propio. Un circuito propio no tiene fuentes de voltaje o inductancias en serie y no tiene fuentes de corriente o conjuntos de condensadores en paralelo. El problema con una fuente de voltaje o inductancias en serie es que la corriente en el lazo no tiene un único valor. Las fuentes de corriente o conjunto de condensadores en paralelo crean circuitos separados que no tienen un único voltaje entre un nodo en un punto y otro nodo en otro punto.

3.1.1 Análisis nodal con fuentes de corriente independientes

El análisis nodal de un circuito lineal que tiene n+1 nodos esenciales implica seleccionar un nodo de referencia y escribir un conjunto de n ecuaciones independientes mediante la ley de Kirchhoff de corrientes, en cada uno de los nodos seleccionados en términos de los voltajes de nodos. Estos voltajes de nodo constituyen un conjunto linealmente independiente de variables de solución. Si el circuito solo tiene conductancias y fuentes de corriente, se puede escribir el conjunto de ecuaciones independientes de forma matricial directamente inspeccionando el circuito. Este hecho proviene directamente de la forma de la ley de Kirchhoff de corrientes. Cada conductancia en el circuito se conecta entre dos nodos y transporta una corriente proporcional a ambos: la diferencia entre los voltajes de nodo asociados y el valor de conductancia. Si la secuencia de los voltajes de nodo en el vector solución del voltaje de nodo es la misma como para las ecuaciones de la ley de Kirchhoff de corrientes, entonces las ecuaciones de nodo tienen una diagonal simétrica. Por ejemplo, si una conductancia $\mathbf{G_A}$ está conectada desde el nodo **i** al nodo **j** entonces los términos mostrados en la ecuación 3.1 aparecen en las ecuaciones de la ley de Kirchhoff de corrientes.

$$
\begin{array}{ll}
\vdots & \\
i & \cdots + G_A\left(V_{Ni} - V_{Nj}\right) + \cdots = \cdots \\
\vdots & \\
j & \cdots + G_A\left(V_{Nj} - V_{Ni}\right) + \cdots = \cdots \\
\vdots &
\end{array}
\tag{3.1}
$$

El coeficiente G_A aparece como una entrada positiva en la ley de Kirchhoff de corrientes (LKC) en las posiciones i-ésima y j-ésima de la diagonal y como una entrada negativa en la i-ésima fila y j-ésima columna, y en la j-ésima fila e i-ésima columna. Si una conductancia está conectada entre el nodo **i** y el nodo de referencia, entonces esta conductancia aparece solamente como un término positivo en la posición i-ésima de la diagonal.

Ejemplo 3.1. Análisis nodal con fuentes de corriente independientes

Para el circuito mostrado en la figura 3.1:

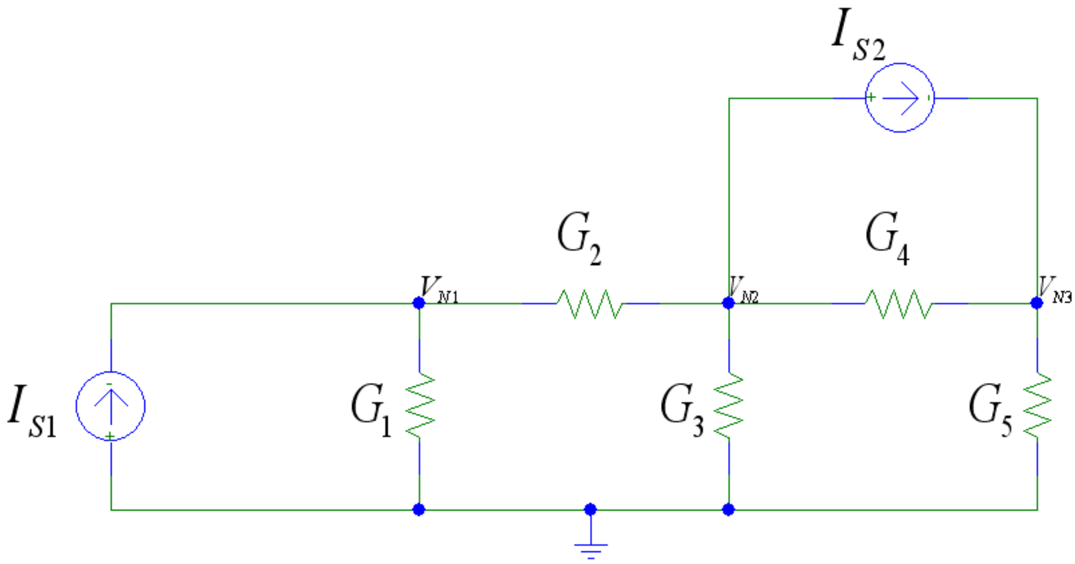
a) Escribir las ecuaciones de voltaje de nodo según la LKC en términos de las conductancia individuales, corrientes y voltajes de nodo V_{N1}, V_{N2} y V_{N3}.

b) Reescribir las ecuaciones agrupando los coeficientes comunes para cada voltaje de nodo.

c) Expresar las tres ecuaciones de voltaje de nodo según la LKC en forma matricial. Note la simetría de diagonal de la matriz de conductancia y la posición de las entradas diferentes a cero en la matriz de fuentes de corriente.

d) Escribir la matriz de conductancias como una sumatoria de cinco matrices separadas, cada una involucrando solamente una conductancia, y la matriz de fuentes de corriente como la sumatoria de dos matrices, cada una involucrando solamente una de las fuentes de corriente.

Figura 3.1. Circuito para el ejemplo 3.1



Solución

a) Las tres ecuaciones según la LKC son

$$G_1 V_{N1} + G_2 \left(V_{N1} - V_{N2}\right) = I_{S1} \tag{3.2}$$

$$G_2 \left(V_{N2} - V_{N1}\right) + G_3 V_{N2} + G_4 \left(V_{N2} - V_{N3}\right) = -I_{S2} \tag{3.3}$$

$$G_4 \left(V_{N3} - V_{N2}\right) + G_5 V_{N3} = I_{S2} \tag{3.4}$$

b) Agrupando los coeficientes de cada nodo de voltaje

$$\left(G_1 + G_2\right) V_{N1} - G_2 V_{N2} = I_{S1} \tag{3.5}$$

$$-G_2 V_{N1} + \left(G_2 + G_3 + G_4\right) V_{N2} - G_4 V_{N3} = -I_{S2} \tag{3.6}$$

$$-G_4 V_{N2} + \left(G_4 + G_5\right) V_{N3} = I_{S2} \tag{3.7}$$

c) Escribiendo estas ecuaciones en forma matricial

$$\begin{bmatrix} G_1 + G_2 & -G_2 & 0 \\ -G_2 & G_2 + G_3 + G_4 & -G_4 \\ 0 & -G_4 & G_4 + G_5 \end{bmatrix} \begin{bmatrix} V_{N1} \\ V_{N2} \\ V_{N3} \end{bmatrix} = \begin{bmatrix} I_{S1} \\ I_{S2} \\ -I_{S2} \end{bmatrix} \tag{3.8}$$

Se puede escribir esta matriz solución directamente inspeccionando el circuito. Cada término sobre la diagonal resulta de la sumatoria de las conductancias que conectan a cada nodo. Estos términos son expresiones de autoconductancia. Cada término de conductancia mutua fuera de la diagonal es la suma negativa de las conductancias que se conectan entre pares de nodos.

d) Rescribiendo la matriz de conductancias y la matriz de fuentes de corriente como términos separados se tiene.

$$\left\{ \begin{bmatrix} G_1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} G_2 & -G_2 & 0 \\ -G_2 & G_2 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & G_3 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & G_4 & -G_4 \\ 0 & -G_4 & G_4 \end{bmatrix} \right.$$
$$\left. + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & G_5 \end{bmatrix} \right\} \begin{bmatrix} V_{N1} \\ V_{N2} \\ V_{N3} \end{bmatrix} = \left\{ \begin{bmatrix} I_{S1} \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ I_{S2} \\ -I_{S2} \end{bmatrix} \right\} \tag{3.9}$$

Esta forma final de la ecuación matricial de la LKC muestra como cada elemento del circuito afecta la estructura de la ecuación matricial del análisis nodal. Una conductancia que se conecte entre un nodo y el nodo de referencia aparece solamente en la posición de la diagonal para ese nodo. Una conductancia que se conecte entre dos nodos, ninguno de los cuales es el nodo de referencia, aparece en cuatro posiciones. Los dos términos de la diagonal son positivos, y los dos términos fuera de la diagonal son negativos. Los términos fuera de la diagonal se presentan en la fila de un nodo y la columna de otro nodo.

Para una fuente de corriente, se ingresa el valor de la fuente en una o dos filas de la matriz de excitación al lado derecho de la ecuación. La entrada es positiva en la fila correspondiente al nodo hacia donde señala la dirección de la corriente. Cuando la corriente está en sentido opuesto la entrada es negativa. Se omite la entrada si el nodo es el nodo de referencia.

3.1.2 Análisis nodal con supernodos

La existencia de una fuente de voltaje en un circuito simplifica y al mismo tiempo complica el método de análisis nodal. Cada fuente de voltaje reduce el número de voltajes de nodo desconocidos en uno. El número de ecuaciones independientes de acuerdo a la LKC se reduce a

$$\text{No. de ecuaciones independientes (LKC)} = n - n_{fv} \qquad (3.10)$$

donde n_{fv} es el número de fuentes de voltajes del circuito. Una fuente de voltaje que está conectada entre un nodo y el nodo de referencia identifica ese voltaje de nodo.

Una fuente de voltaje que conecta dos nodos diferentes al nodo de referencia identifica uno de los voltajes de nodo en términos del otro, eliminando uno de los voltajes de nodo como una incógnita en la solución. Al mismo tiempo, cada fuente de voltaje en el circuito tiene una corriente que en algunos casos se puede determinar. Dicha corriente se determina inmediatamente. Cada fuente de voltaje elimina una ecuación LKC del conjunto de ecuaciones LKC entregando la solución del voltaje de nodo. Debido a que cada una de las ecuaciones restantes es esencial para completar el conjunto de ecuaciones linealmente independientes necesarias para determinar los voltajes de nodo, dichas ecuaciones se denominan ecuaciones esenciales de la ley de Kirchhoff de corrientes (LKC).

Para una fuente de voltaje conectada entre un nodo y el nodo de referencia, la ecuación para el nodo que no es referencia no es una ecuación esencial, pero se puede usar para calcular las corrientes en las fuentes de voltaje, posteriormente. Para una fuente de voltaje que no está conectada al nodo de referencia, la ecuación para cualquiera de los nodos incluye la corriente de la fuente de voltaje, ninguna de estas ecuaciones es esencial. Para la LKC sin embargo, la suma de estas dos ecuaciones no depende de la corriente de la fuente de voltaje, porque esta corriente se da como un término positivo en una de las ecuaciones y como un término negativo en la otra. La adición de dos ecuaciones entrega una ecuación que no incluye la corriente de la fuente de voltaje. Si no hay más fuentes de voltaje conectadas a este par de nodos, este par de nodos conforman un supernodo.

Generalmente, un supernodo es un conjunto de nodos que contienen unos voltajes de nodo que dependen solamente de un voltaje de nodo y un conjunto de fuentes de voltaje. La ecuación del supernodo consiste de una sumatoria de corrientes que salen de una superficie que encierra el grupo de nodos. Esta ecuación de supernodo iguala la suma de ecuaciones individuales para cada nodo en el conjunto de supernodo. La sumatoria de las ecuaciones de nodo formando la ecuación del supernodo elimina las corrientes individuales de las fuentes de voltaje, ya que cada corriente de la fuente de voltaje aparece dos veces, primero como un término positivo y luego como un término negativo.

Ejemplo 3.2 Circuito con fuentes de voltaje

Para el circuito mostrado en la figura 3.2:

a) Seleccione un conjunto independiente de nodos o supernodos. Defina todos los voltajes de nodo y supernodo. Dibuje la superficie que representa el supernodo.

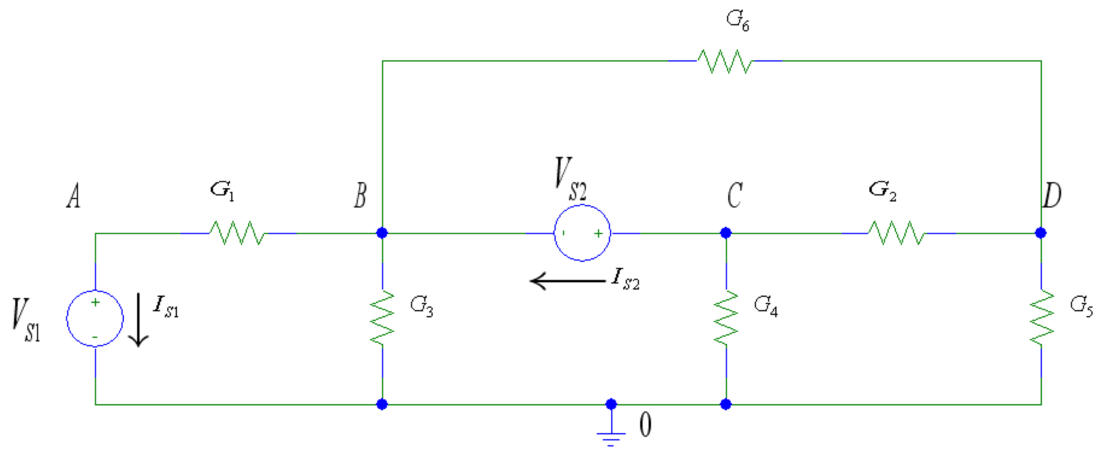
b) Escriba las ecuaciones para cada nodo en el supernodo o sume estas ecuaciones para obtener la ecuación del supernodo.

c) Escriba la(s) ecuación(es) restante(s) para el circuito.

d) Cambie el conjunto de ecuaciones esenciales, agrupando coeficientes.

e) Exprese este conjunto de ecuaciones como una ecuación matricial única.

Figura 3.2. Ejemplo de circuito para ilustrar el análisis nodal con fuentes de voltaje.



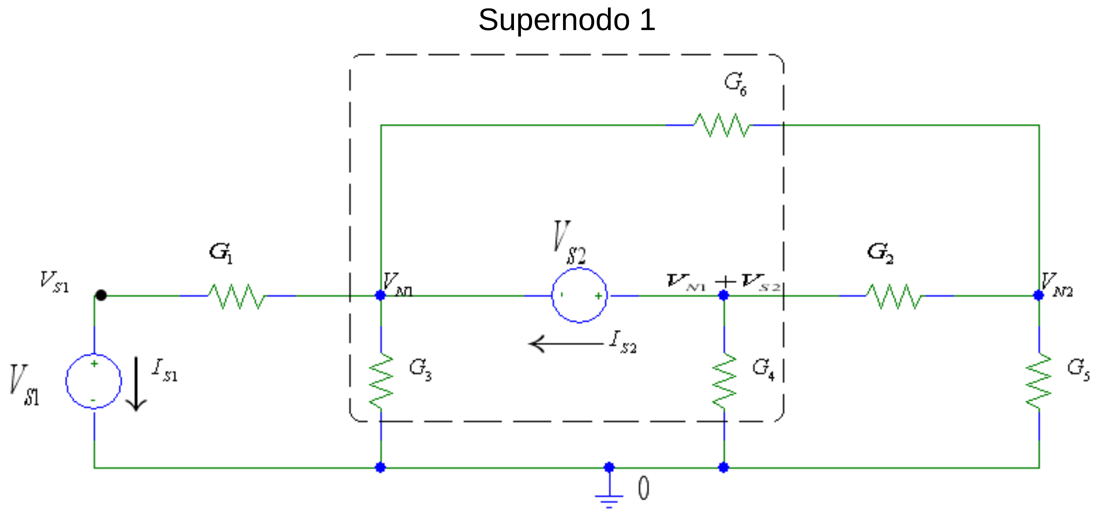
Solución

a) El circuito tiene n = 4 y n_{fv} = 2, por lo tanto el número de nodos independientes es

Número de ecuaciones = n – n_{fv} = 4 – 2 = 2

El circuito de la figura 3.2 se redibuja en la figura 3.3, mostrando una selección de nodos independientes, se define los voltajes de nodo y de supernodo, y se muestra la superficie del supernodo. Debido a que la fuente de voltaje V_{S1} está conectada al nodo A, el voltaje en éste nodo es igual al voltaje de la fuente. Se selecciona el voltaje de nodo para el nodo B el cual es V_{N1}. Entonces, por LKV, el voltaje de nodo para el nodo C es igual a $V_{N1} + V_{S2}$. Si se escoge el voltaje para el nodo C, en lugar del nodo B, para que este sea V_{N1} entonces el voltaje para el nodo B será $V_{N1} - V_{S2}$. La elección es arbitraria pero debe permanecer por el resto del análisis. El nodo D es el segundo nodo independiente y es nombrado V_{N2}.

Figura 3.3. Circuito de la figura 3.2 mostrando nodos y supernodos



b) Las dos ecuaciones (según LKC) son:

$$G_1\left(V_{N1} - V_{S1}\right) + G_3 V_{N1} + G_6\left(V_{N1} - V_{N2}\right) - I_{S2} = 0 \tag{3.11}$$

$$G_2\left[\left(V_{N1} + V_{S2}\right) - V_{N2}\right] + G_4\left(V_{N1} + V_{S2}\right) + I_{S2} = 0 \tag{3.12}$$

Al sumar estas ecuaciones se obtiene:

$$G_1\left(V_{N1} - V_{S1}\right) + G_3 V_{N1} + G_6\left(V_{N1} - V_{N2}\right) + G_2\left[\left(V_{N1} + V_{S2}\right) - V_{N2}\right] + G_4\left(V_{N1} + V_{S2}\right) = 0 \tag{3.13}$$

Esta ecuación es la ecuación de la ley de Kirchhoff de corriente para el supernodo de la figura 3.3.

c) La ecuación esencial restante para el nodo 2 es:

$$G_2\left[V_{N2} - \left(V_{N1} + V_{S2}\right)\right] + G_5 V_{N2} + G_6\left(V_{N2} - V_{N1}\right) = 0 \tag{3.14}$$

d) Agrupando términos para las ecuaciones del nodo y del supernodo se obtiene:

$$
\begin{aligned}
&\left(G_1 + G_2 + G_3 + G_4 + G_6\right)V_{N1} - \left(G_2 + G_6\right)V_{N2} = G_1 V_{S1} - \left(G_2 + G_4\right)V_{S2} \\
&- \left(G_2 + G_6\right)V_{N1} + \left(G_2 + G_5 + G_6\right)V_{N2} = G_2 V_{S2}
\end{aligned}
\tag{3.15}
$$

e) La forma matricial de estas dos ecuaciones es:

$$
\begin{aligned}
&\begin{bmatrix} G_1 + G_2 + G_3 + G_4 + G_6 & -\left(G_2 + G_6\right) \\ -\left(G_2 + G_6\right) & G_2 + G_5 + G_6 \end{bmatrix}
\begin{bmatrix} V_{N1} \\ V_{N2} \end{bmatrix}
= \begin{bmatrix} G_1 V_{S1} - \left(G_2 + G_4\right)V_{S2} \\ G_2 V_{S2} \end{bmatrix} \\
&= \begin{bmatrix} G_1 \\ 0 \end{bmatrix} V_{S1} + \begin{bmatrix} \left(G_2 + G_4\right) \\ G_2 \end{bmatrix} V_{S2}
= \begin{bmatrix} G_1 & -\left(G_2 + G_4\right) \\ 0 & G_2 \end{bmatrix} \begin{bmatrix} V_{S1} \\ V_{S2} \end{bmatrix}
\end{aligned}
\tag{3.16}
$$

En el ejemplo 3.2, las conductancias G_1, G_2, G_3, G_4 y G_6 son las conductancias propias para el supernodo 1, porque cada conductancia está conectada al supernodo.
Las conductancias G_2, G_5 y G_6 están conectadas al nodo 2, por lo tanto son conductancias propias del nodo 2. G_2 y G_6 son conductancias mutuas entre el supernodo 1 y el nodo 2, estos se acumulan negativamente en la fila 1, columna 2 y en la fila 2, columna 1 de la matriz de conductancias. También, G_2 y G_6 están conectados entre dos nodos que no son de referencia y aparecen como una trayectoria típica de cuatro, con dos términos positivos en la diagonal y dos términos negativos fuera de la diagonal. Las conductancias restantes se conectan desde un nodo al nodo de referencia, éstas solamente aparecen como términos aditivos en las posiciones apropiadas de la diagonal.

Otro punto de vista útil es pensar en las componentes de la corriente debido sólo a una fuente o voltaje de nodo a la vez. Por ejemplo, con el conjunto V_{S1}, V_{S2} y V_{N2} iguales a cero, la sumatoria de corrientes asociada al supernodo es:

$$\left(G_1 + G_2 + G_3 + G_4 + G_6\right)V_{N1} \tag{3.17}$$

Este término cuenta para la entrada en la primera fila y la primera columna de la matriz de conductancias. La corriente en el nodo 2 debida a V_{N1} solo es:

$$(G_2 + G_6)V_{N1} \tag{3.18}$$

La cual cuenta para la entrada en la fila 1, columna 2 de la matriz de conductancias.

Los términos de la fuente independiente aparecen en el lado derecho de la ecuación matricial. Con V_{S1} solo diferente de cero, hay una corriente G_1V_{S1} en la conductancia G_1 con dirección hacia el supernodo 1. Este hecho se cuenta para la entrada diferente de cero en la fila 1 del voltaje V_{S1} de la matriz de coeficientes. Con V_{S2} solo tomando un valor diferente de cero, la conductancia G_2 tiene una componente de corriente (G_2V_{S2}) desde el supernodo 1 y en dirección hacia el nodo 2. Puesto que una corriente con la dirección hacia un nodo es positiva en el lado derecho de las ecuaciones (según LKC), este hecho cuenta para la entrada positiva de G_2 en la fila 2 y la entrada negativa en la fila 1 de la matriz de coeficientes de V_{S2}.

El punto de estas observaciones es la entrada de cada elemento en la formulación matricial de las ecuaciones esenciales que pueden ser hechas por inspección directa del diagrama de circuito. Escribiendo la forma matricial de la solución directamente por inspección del circuito se ahorra tiempo y se reduce la posibilidad de cometer errores al momento de transcribir las ecuaciones.

3.1.3 Análisis nodal con fuentes de corriente controladas por voltaje (FCCV)

La entrada de una fuente de corriente controlada por voltaje (FCCV) en la matriz de conductancias de un circuito es simple y directa. Por ejemplo, una FCCV con un valor g_m y conectada desde el nodo **i** hasta el nodo **j** y bajo el control de la diferencia de voltajes entre el nodo **k** y el nodo **l** da los siguientes términos en las ecuaciones de voltaje de nodo

$$\begin{array}{ll}
\vdots & \\
i & \ldots + g_m (V_{Nk} - V_{Nl}) + \ldots = \ldots \\
\vdots & \\
j & \ldots - g_m (V_{Nk} - V_{Nl}) + \ldots = \ldots \\
\vdots &
\end{array} \tag{3.19}$$

Se puede observar un modelo de cuatro valores de g_m. Las entradas positivas aparecen en la fila **i** y columna **k** (el “de” la fila y el “+” controlando la columna) y en la fila **j**, y la columna **l** (controlando la columna el signo “-“). Este modelo es el mismo en cuanto a conductancia donde **i** es igual a **k** y **j** es igual a **l**. Este hecho conduce a la observación de que una FCCV es igual a una conductancia cuando los nodos de control son los mismos nodos a los que está conectada la fuente. En general, una FCCV destruye la simetría de la matriz de conductancias cuando **i** y **k** no son iguales o **j** y **l** son diferentes. Si **i** o **j** llegan a ser el nodo de referencia, entonces esta fila queda fuera de la matriz y solamente los dos términos de la fila restante aparecen en la matriz. Si **k** o **l** es el nodo de referencia, entonces esta columna queda fuera de la matriz de conductancias y solamente aparecen los dos términos de la columna restante. Si una fila y una columna corresponden al nodo de referencia, entonces solamente un término aparece en la matriz. Aunque teniendo ambas filas y ambas columnas en el nodo de referencia es posible, pero esto es una situación trivial.

3.1.4 Ecuaciones nodales con fuentes dependientes

Una forma general de la ecuación matricial del análisis nodal (según la LKC) es:

$$G_{nn} V_n + A_{nd} X_d = A_{ns} X_s \qquad (3.20)$$

Donde V_n es el vector solución del voltaje de nodo, X_d es un vector de fuentes dependientes y X_s es un vector de fuentes independientes. Para tener una matriz de conductancias G_{nn} que muestre simetría, se debe escribir las ecuaciones de nodo en un orden como el de las componentes de voltaje de nodo en el vector solución, además agregar las corrientes de cada nodo e incluyendo las FCCV en A_{nd}. Sin embargo, se pueden acoplar las fuentes de corriente controladas por voltaje en G_{nn}.
De la ecuación 3.20 pueden resultar algunas ecuaciones, las cuales pueden ser ecuaciones de supernodo. La matriz G_{nn} se tiene en cuenta para los términos de corriente en la ecuación nodal debido a las conductancias del circuito. La matriz A_{nd} se tiene en cuenta para las contribuciones a las ecuaciones debido a las fuentes dependientes y la matriz A_{ns} inserta componentes de corriente debido al vector de fuente X_s.

Asumiendo una dependencia lineal del vector de fuentes dependientes X_d en el vector de variables controladoras X_c, se tiene:

$$X_d = A_{dc} X_c \qquad (3.21)$$

Usualmente la matriz A_{dc} de fuente controlada es cuadrada y diagonal, pero esta condición no es necesaria. Una fuente dependiente puede depender de dos o más variables controladoras ó dos o más fuentes dependientes pueden tener la misma variable controladora.

Finalmente, asumiendo que el vector de variable controladora X_c depende linealmente del vector de fuente dependiente, del vector voltaje de nodo, y del vector fuente independiente, se tiene:

$$X_c = A_{cd} X_d + A_{cn} V_n + A_{cs} X_s \tag{3.22}$$

Las ecuaciones individuales en ecuación 3.22 son las ecuaciones de la LKC o LKV que identifican las variables controlando en términos de varios voltajes de nodo y fuentes dependientes o independientes.

Sustituyendo la ecuación 3.22 en la ecuación 3.21 y resolviendo para X_d se tiene:

$$X_d = \left(I_d - A_{dc} A_{cd}\right)^{-1} A_{dc} \left(A_{cn} V_n + A_{cs} X_s\right) \tag{3.23}$$

Donde I_d es la matriz identidad con un orden igual al número de fuentes dependientes en X_d. Alternativamente, la sustitución de la ecuación 3.21 en la ecuación 3.22, resolviendo para X_c y la sustitución de regreso en la ecuación 3.20 da

$$X_d = A_{dc} \left(I_c - A_{cd} A_{dc}\right)^{-1} \left(A_{cn} V_n + A_{cs} X_s\right) \tag{3.24}$$

Donde I_c es la matriz identidad con orden igual al número de variables controladoras en X_c. Las ecuaciones 3.23 y 3.24 son equivalentes, por lo tanto:

$$\left(I_d - A_{dc} A_{cd}\right)^{-1} A_{dc} = A_{dc} \left(I_c - A_{cd} A_{dc}\right)^{-1} \tag{3.25}$$

Finalmente, sustituyendo de la ecuación 3.23 en la ecuación 3.20 y agrupando coeficientes da una ecuación general de voltajes de nodo:

$$\left\{G_{nn} + A_{nd}\left(I_d - A_{dc}A_{cd}\right)^{-1} A_{dc}A_{cn}\right\}V_n = \left\{A_{ns} - A_{nd}\left(I_d - A_{dc}A_{cd}\right)^{-1} A_{dc}A_{cs}\right\}X_s \tag{3.26}$$

Si el circuito contiene sólo una fuente dependiente entonces el factor de la matriz $I_d - A_{dc}A_{cd}$ se convierte en un escalar y es conocido en la teoría de realimentación como la diferencia de retorno. Se le llama a esta cantidad la matriz diferencia de retorno. La matriz diferencia de retorno juega un papel menos significativo aquí que en la teoría de realimentación, porque A_{cd} es nula siempre que la variable controladora no es directamente dependiente de cualquiera de las fuentes dependientes. Las variables solución de voltajes de nodo, frecuentemente intervienen entre las fuentes dependientes y sus variables controladoras. Si un circuito tiene una o más fuentes dependientes, entonces la inversa de la matriz diferencia de retorno tiene que existir para que el circuito tenga una solución.

Los voltajes de nodo pueden no ser las variables solución de interés. Las variables de salida como un grupo constituyen un vector de salida X_o. El vector de salida es una función lineal del vector de fuentes dependientes X_d, del vector de voltaje de nodo V_n, y del vector de fuentes X_s, así:

$$X_o = A_{od} X_d + A_{on} V_n + A_{os} X_s \tag{3.27}$$

La sustitución de la ecuación 3.23 en la ecuación 3.27 entrega:

$$X_o = \left[A_{on} + A_{od}\left(I - A_{dc}A_{cd}\right)^{-1} A_{dc}A_{cn}\right]V_n + \left[A_{os} + A_{od}\left(I - A_{dc}A_{cd}\right)^{-1} A_{dc}A_{cs}\right]X_s \tag{3.28}$$

La cual expresa el vector de salida en términos del vector voltaje de nodo y del vector de fuentes independientes.

Otra manera para resolver las ecuaciones 3.20, 3.21, 3.22 y 3.27 es reescribirlas como:

$$G_{nn} V_n + A_{nd} X_d + 0 X_c + 0 X_o = A_{ns} X_s \tag{3.29}$$

$$0 V_n + I X_d - A_{dc} X_c + 0 X_o = 0 \tag{3.30}$$

$$- A_{cn} V_n - A_{cd} X_d + I X_c + 0 X_o = A_{cs} X_s \tag{3.31}$$

$$- A_{on} V_n - A_{od} X_d + 0 X_c + I X_o = A_{os} X_s \tag{3.32}$$

Ahora, escribiendo estas ecuaciones como una ecuación matricial:

$$\begin{bmatrix} G_{nn} & A_{nd} & 0 & 0 \\ 0 & I & -A_{dc} & 0 \\ -A_{cn} & -A_{cd} & I & 0 \\ -A_{on} & -A_{od} & 0 & I \end{bmatrix} \begin{bmatrix} V_n \\ X_d \\ X_c \\ X_o \end{bmatrix} = \begin{bmatrix} A_{ns} \\ 0 \\ A_{cs} \\ A_{os} \end{bmatrix} X_s \qquad (3.33)$$

Esta ecuación de super-matriz muestra los parámetros del circuito en su forma más simple, evitando la complejidad de la solución de la ecuación 3.28. Y permite usar el método del sistema propuesto en el ejercicio 5.2. del capítulo 5.

3.2. ANÁLISIS DE MALLAS [3]

El análisis de mallas expresa una solución de circuitos planares usando un conjunto linealmente independiente de corrientes de malla. Una corriente de malla es una componente de corriente que circula a través de todos los elementos en el perímetro de una ventana de un diagrama esquemático de circuito. Por convención todas las corrientes de malla circulan en la misma dirección, usualmente lo hacen en sentido de las manecillas del reloj. Una corriente de malla físicamente es medible si las corrientes de los elementos del circuito tienen solamente una componente de la corriente de malla. La solución del circuito consiste de un conjunto de ecuaciones linealmente independiente de corrientes de malla según la ley de Kirchhoff de voltaje (LKV). Se asume que el circuito es propio.

3.2.1 Análisis de malla con fuentes de voltaje independientes

El análisis de corrientes de malla de un circuito que sólo tiene fuentes de voltaje independientes involucra identificar un conjunto de corrientes de malla y escribir un número de ecuaciones independientes (de la LKV) dadas por

$$\text{No. de ecuaciones independientes} = b - n \qquad (3.34)$$

donde b es el número de ramas esenciales y n es el número nodos esenciales en el circuito.

Se puede escribir el conjunto de independiente de ecuaciones en forma matricial directamente por inspección del circuito. Este hecho se presenta directamente de la forma de sumatoria de la LKV. Cada resistencia en el circuito tiene una corriente igual a la diferencia entre dos corrientes de malla y tiene un voltaje proporcional a la diferencia de las corrientes de malla y al valor de la resistencia. Si la secuencia de las corrientes de malla en el vector solución de corrientes de malla es la misma como para las ecuaciones de la LKV, entonces las ecuaciones de corrientes de malla tienen simetría diagonal. Por ejemplo, si la resistencia R_A está conectada entre la malla **i** y la malla **j**, los términos aparecen en las siguientes ecuaciones según LKV

$$
\begin{array}{ll}
 & \vdots \\
i & \ldots + R_A\left(I_{Mi} - I_{Mj}\right) + \ldots = \ldots \\
 & \vdots \\
j & \ldots - R_A\left(I_{Mi} - I_{Mj}\right) + \ldots = \ldots \\
 & \vdots
\end{array}
\qquad (3.35)
$$

El término R_A aparece como un término positivo en la ecuación de la LKV en la i-ésima y j-ésima posiciones de la diagonal y como un término negativo en la i-ésima fila y la j-ésima columna y, en la j-ésima fila y la en i-ésima columna. Si una resistencia está conectada en el perímetro externo del diagrama del circuito, entonces esta resistencia aparece sólo como un término aditivo en la i-ésima posición de la diagonal.

Ejemplo 3.3. Análisis de mallas con fuentes de voltaje independientes

Para el circuito mostrado en la figura 3.4.

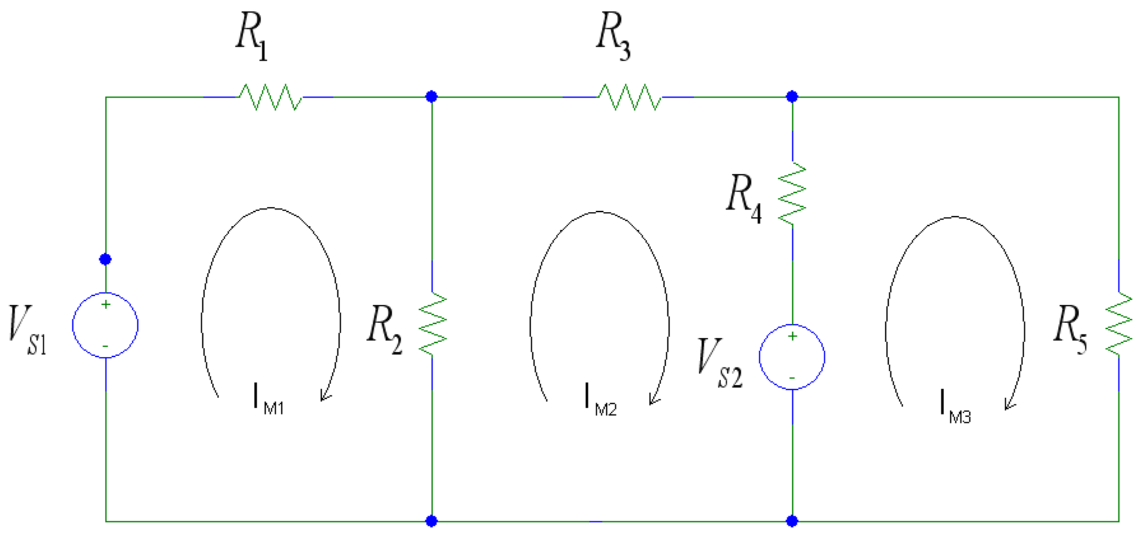
a) Escriba las ecuaciones de corrientes de malla en términos de los voltajes individuales de las fuentes y de las corrientes de malla I_{M1}, I_{M2} e I_{M3}.

b) Reescriba las ecuaciones agrupando los coeficientes comunes de cada corriente de malla.

c) Exprese las tres ecuaciones de corriente de malla (LKV) en forma matricial.
Nótese la simetría diagonal de la matriz de resistencias y la localización de las entradas diferentes de cero en la matriz de fuentes de voltaje.

d) Escriba la matriz de resistencias como una sumatoria de cinco matrices separadas, cada una participe envolviendo sólo una resistencia y la matriz de fuentes de voltaje como la suma de dos matrices donde cada una contiene una de las fuentes de voltaje.

Figura 3.4. Circuito para el ejemplo 3.3



Solución

a) Las tres ecuaciones son:

$$R_1 I_{M1} + R_2 \left(I_{M1} - I_{M2}\right) = V_{S1} \tag{3.36}$$

$$R_2 \left(I_{M2} - I_{M1}\right) + R_3 I_{M2} + R_4 \left(I_{M2} - I_{M3}\right) = - V_{S2} \tag{3.37}$$

$$R_4 \left(I_{M3} - I_{M2}\right) + R_5 I_{M3} = V_{S2} \tag{3.38}$$

b) Agrupando coeficientes de cada corriente de malla:

$$\left(R_1 + R_2\right) I_{M1} - R_2 I_{M2} = V_{S1} \tag{3.39}$$

$$- R_2 I_{M1} + \left(R_2 + R_3 + R_4\right) I_{M2} - R_4 I_{M3} = - V_{S2} \tag{3.40}$$

$$- R_4 I_{M2} + \left(R_4 + R_5\right) I_{M3} = V_{S2} \tag{3.41}$$

c) Escribiéndola en forma matricial

$$\begin{bmatrix} R_1 + R_2 & - R_2 & 0 \\ - R_2 & R_2 + R_3 + R_4 & - R_4 \\ 0 & - R_4 & R_4 + R_5 \end{bmatrix} \begin{bmatrix} I_{M1} \\ I_{M2} \\ I_{M3} \end{bmatrix} = \begin{bmatrix} V_{S1} \\ V_{S2} \\ V_{S2} \end{bmatrix} \tag{3.42}$$

Se puede escribir la solución matricial directamente por inspección del circuito. Cada término de la diagonal consiste de la sumatoria de resistencias de cada corriente de malla que circula a través de ellas. Estas sumas son términos de la resistencia propia. Los términos fuera de la diagonal, conocidos como términos de la resistencia mutua, son la suma negativa de resistencias que cada una de las mallas comparte con la malla adyacente.

d) Reescribiendo las matrices de resistencias y fuentes de voltaje en términos separados

$$\left\{\begin{bmatrix} R_1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} R_2 & -R_2 & 0 \\ R_2 & R_2 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & R_3 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & R_4 & -R_4 \\ 0 & -R_4 & R_4 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & R_5 \end{bmatrix}\right\} \begin{bmatrix} I_{M1} \\ I_{M2} \\ I_{M3} \end{bmatrix} = \left\{\begin{bmatrix} V_{S1} \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ V_{S2} \\ V_{S2} \end{bmatrix}\right\} \qquad (3.43)$$

Esta forma matricial de la ecuación de malla muestra cómo cada elemento del circuito contribuye a la estructura de la solución de la ecuación matricial. Una resistencia que existe fuera del perímetro del circuito aparece sólo en la posición de la diagonal para esa malla. Una resistencia que existe entre dos mallas aparece en cuatro posiciones. Dos de éstas en la diagonal son positivas, y los dos restantes fuera de la diagonal son términos negativos. Las entradas fuera de la diagonal se presentan en la fila de una de las mallas y la columna de la otra malla.

Contando con una fuente de voltaje, se introduce el valor de ésta en una o dos filas al lado derecho de la matriz de excitación. La entrada es positiva en la fila de una malla si la fuente de voltaje está de menos a más y es negativa en la fila de una de las mallas cuando la fuente de voltaje está de más a menos.

3.2.2 Análisis de mallas con supermallas

Se ha podido observar cómo manipular circuitos con fuentes de voltaje independientes y dependientes, éstas obligan la diferencia entre dos voltajes de nodo en el análisis nodal. Ahora, se puede aprender que hacer en el análisis de mallas cuando un circuito tiene fuentes de corriente común a dos mallas. Para ilustrar esto consideramos el circuito de la figura 3.5 que tiene dos fuentes de corriente.

Ejemplo 3.5. Ejercicio con supermallas

Para el circuito mostrado en la figura 3.5:

a) Seleccionar el conjunto de mallas o supermallas existentes en el circuito. Definir todas las corrientes y dibujar la trayectoria de la(s) supermalla(s).

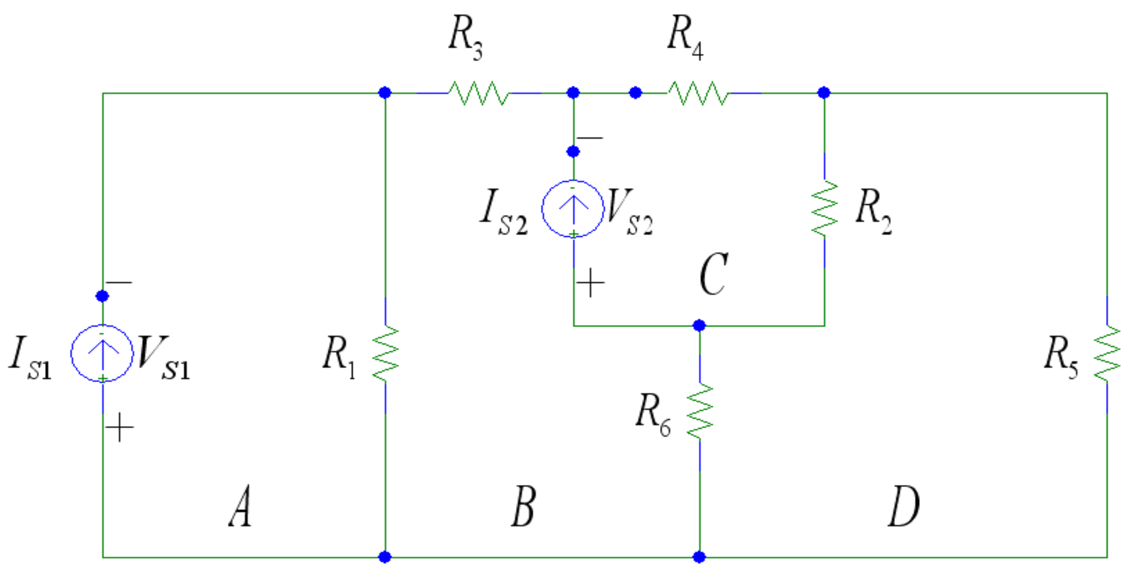
b) Escribir las ecuaciones de la LKV para cada malla en la supermalla. Sume estas ecuaciones para obtener la ecuación de la supermalla.

c) Escribir las ecuaciones da la LKV de voltaje de kirchhoff restantes para el circuito.

d) Organizar el conjunto de ecuaciones esenciales reagrupando los coeficientes.

e) Expresar este conjunto de ecuaciones en forma matricial.

Figura 3.5. Circuito con fuentes de corriente



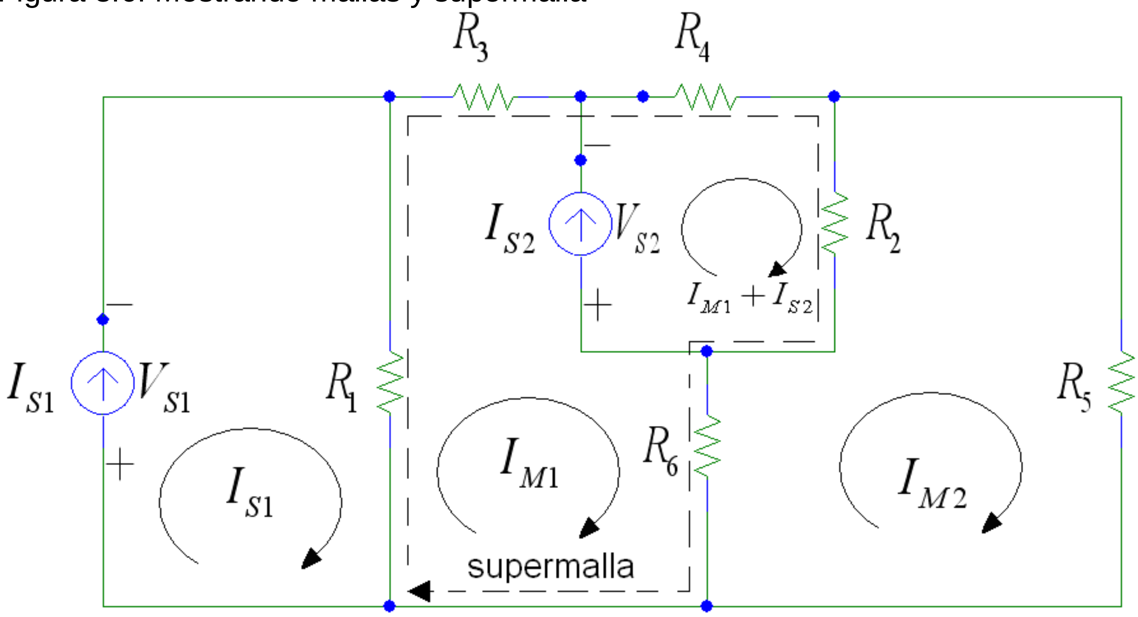
Solución

a) El circuito tiene b = 8, n = 4 y n_{fc} = 2 (n_{fc} representa el número de fuentes de corriente), por lo tanto el numero de mallas independientes son

Número de ecuaciones de LKV = b – n – n_{fc} = 8 – 4 – 2 = 2

El circuito de la figura 3.5 está redibujado en la figura 3.6, mostrando una selección de mallas, definiendo todas las corrientes de malla y supermalla y la trayectoria de la supermalla. Se puede observar que la corriente de la malla A es igual a la corriente de la fuente I_{S1}. La malla B se convierte en la malla 1 y tiene una corriente de malla I_{M1}. Por la LKC, la corriente de malla en la malla C es igual a I_{M1} + I_{S2}. Eligiendo la malla B en vez de la malla C para que tenga una corriente de malla I_{M1} es una escogencia conveniente pero arbitraria que genera una segunda corriente de malla en el sistema de la supermalla que depende de la primera malla. La malla D es la segunda malla independiente.

Figura 3.6. Mostrando mallas y supermalla



b) Las ecuaciones para las mallas B y C son:

$$R_1 \left(I_{M1} - I_{S1}\right) + R_3 I_{M1} + R_6 \left(I_{M1} - I_{M2}\right) - V_{S2} = 0 \tag{3.44}$$

$$R_2 \left[\left(I_{M1} + I_{S2}\right) - I_{M2}\right] + R_4 \left(I_{M1} + I_{S2}\right) + V_{S2} = 0 \tag{3.45}$$

Adicionando éstas se tiene:

$$\begin{aligned} & R_1\left(I_{M1} - I_{S1}\right) + R_3 I_{M1} + R_6\left(I_{M1} - I_{M2}\right) + R_2\left[\left(I_{M1} + I_{S2}\right) - I_{M2}\right] \\ & \qquad + R_4\left(I_{M1} + I_{S2}\right) = 0 \end{aligned} \tag{3.46}$$

Esta ecuación es la LKV aplicada a la supermalla en la figura 3.6.

c) La ecuación de malla restante pertenece a la malla D y es:

$$R_2\left[I_{M2} - \left(I_{M1} + I_{S2}\right)\right] + R_5 I_{M2} + R_6\left(I_{M2} - I_{M1}\right) = 0 \tag{3.47}$$

d) Agrupando términos en las dos últimas ecuaciones se tiene:

$$\begin{aligned} & \left(R_1 + R_2 + R_3 + R_4 + R_6\right) I_{M1} - \left(R_2 + R_6\right) I_{M2} = R_1 I_{S1} - \left(R_2 + R_4\right) I_{S2} \\ & -\left(R_2 + R_6\right) I_{M1} + \left(R_2 + R_5 + R_6\right) I_{M2} = R_2 I_{S2} \end{aligned} \tag{3.48}$$

e) Escribiendo estas dos ecuaciones en forma matricial se tiene:

$$\begin{aligned} & \begin{bmatrix} R_1 + R_2 + R_3 + R_4 + R_6 & -\left(R_2 + R_6\right) \\ -\left(R_2 + R_6\right) & R_2 + R_5 + R_6 \end{bmatrix} \begin{bmatrix} I_{M1} \\ I_{M2} \end{bmatrix} = \begin{bmatrix} R_1 I_{S1} - \left(R_2 + R_4\right) I_{S2} \\ R_2 I_{S2} \end{bmatrix} \\ & \qquad = \begin{bmatrix} R_1 \\ 0 \end{bmatrix} I_{S1} + \begin{bmatrix} \left(R_2 + R_4\right) \\ R_2 \end{bmatrix} I_{S2} = \begin{bmatrix} R_1 & -\left(R_2 + R_4\right) \\ 0 & R_2 \end{bmatrix} \begin{bmatrix} I_{S1} \\ I_{S2} \end{bmatrix} \end{aligned} \tag{3.49}$$

En este ejemplo R_1, R_2, R_3, R_4 y R_6 son resistencias propias de la supermalla, debido a que se encuentran en la trayectoria de ésta. R_2, R_5 y R_6 pertenecen a la malla D. R_2 y R_6 son resistencias mutuas entre la supermalla y la malla D por esto los valores de éstas resistencias se acumulan negativamente en la fila 1, columna 2 y en la fila 2, columna 1 de la matriz de resistencias. A su vez R_1 y R_6 se encuentran entre dos mallas, por esto aparecen en un típico patrón de cuatro entradas, con dos entradas positivas en la diagonal y dos entradas negativas fuera de la diagonal. Las resistencias restantes están en la trayectoria de sólo una de las mallas o de la supermalla, por esto sólo aparecen cómo términos aditivos en las posiciones apropiadas de la diagonal.

4. DESCRIPCIÓN DEL MANEJO DE LOS PROGRAMAS A DESARROLLAR

En este capítulo se muestran una serie de funciones necesarias para la solución de los ejercicios en el MATLAB.

4.1. FUNCIÓN nmAcc

Función **nmAcc,** sirve para la agrupación de términos de cada conductancia(o resistencia) en un problema. Esta función tiene cuatro argumentos. Los primeros dos son **i** y **j**, los cuales identifican los nodos de las conductancias. El tercer argumento es **val**, el cual identifica el valor de la conductancia. El último argumento es **bmat**, que identifica la matriz de conductancias sobre la cual acumula el valor de la respectiva conductancia. Si **i** ó **j** es cero entonces la acumulación del valor de la conductancia ocurre solamente sobre la posición diagonal del nodo distinto de la referencia. Cuando **i** y **j** son diferentes de cero, la acumulación positiva del valor de la conductancia se da sólo en cada posición diagonal y la acumulación negativa se presenta en las posiciones fuera de la diagonal. No se presenta acumulación si **i** y **j** son cero, lo cual es un error.

```
Function amat = nmAcc(i, j, val, bmat)
if (i<0)|(j<0)
        disp('los índices fila/columna no deben ser negativos')
elseif  (i == 0) & (j == 0)
        disp('los índices son iguales a cero. Trivial!')
else
        if (i > 0)
        bmat(i,i)=bmat(i,i)+val;
        end
        if (j > 0)
        bmat(j,j)=bmat(j,j)+val;
        end
        if (i > 0) & (j > 0)
                bmat(i,j)=bmat(i,j)-val;
                bmat(j,i)=bmat(j,i)-val;
        end
end
amat=bmat;
```

- nmAcc(i, j, val, bmat)
- Acumula los valores de G o R en la matriz **val** (bmat)
- i y j son los índices de nodo o malla

Las fuentes de corriente aparecen en las ecuaciones como términos al lado derecho dentro de la matriz de excitación. Una corriente I_S con dirección del nodo i al nodo j acumula positivamente en la j-ésima fila de nodo y negativamente en la i-ésima fila de nodo. Si cualquiera de los nodos es el nodo de referencia entonces la acumulación se presenta solamente en el nodo que no es referencia.

4.2. FUNCIÓN srcAcc

Función **srcAcc**, acumula una fuente de corriente sobre una matriz de fuentes de corriente. El primer argumento de esta función es el nodo i, desde donde fluye la fuente. El segundo argumento es el nodo j, hacia donde fluye la fuente. El tercer argumento **val** entrega el valor de la fuente de corriente, y el cuarto argumento **bmat** identifica la matriz fuente. En el ejemplo 4.1 se usará esta función.

```
Function amat = srcAcc (i, j, val, bmat)
if (i<0)|(j<0)
	disp('los índices desde/hacia no deben ser negativos ')
elseif (i == 0) & (j == 0)
	disp('los índices son iguales a cero. Trivial!')
else
	if (j > 0)
		bmat(j)=bmat(j)+val;
	end
	if (i > 0)
		bmat(i)=bmat(i)-val;
	end
end
amat=bmat;
```

- srcAcc (i, j, val, bmat)
- Acumula los valores de Is o Vs en la matriz fuente (bmat)
- i: de donde sale la fuente
- j: a donde llega la fuente

Ejemplo 4.1 Cubo de resistencias de 1 Ω

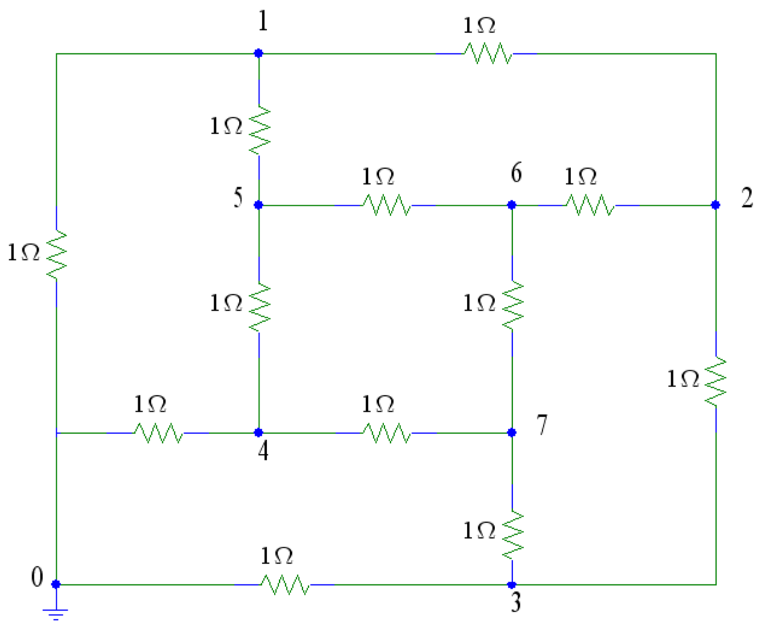
El circuito mostrado en la figura 4.1 es un cubo conformado por resistencias de 1-Ω. Evaluar:

a) El valor de la resistencia entre el nodo 6 y el nodo 0.

b) El valor de la resistencia entre el nodo 2 y el nodo 0.

c) El valor de la resistencia entre el nodo 1 y el nodo 0.

Figura 4.1. Cubo de 1-Ω



Solución

Se aplica una fuente de 1 Amp entre cada par de nodos apropiadamente y se calcula el valor del voltaje entre el par de nodos. El archivo .M de MATLAB en la figura 4.2 encuentra la solución utilizando una matriz de excitación de tres columnas. La función **nmAcc** acumula los valores de conductancia en la matriz de conductancias **gMat**. La función **srcAcc** acumula las fuentes en la matriz I_S. Cada columna de Is es un vector fuente de corriente para a), b) o c). La solución de voltaje de nodo V es una matriz 7x3, donde cada columna de V es el vector solución de voltaje de nodo para la correspondiente columna de I_S. Por ejemplo, la componente de voltaje V (6,1) es el voltaje en el nodo 6 cuando una fuente de corriente de 1A se conecta desde el nodo 0 al nodo 6. Este voltaje es igual numéricamente al valor de la resistencia entre los mismos nodos.

Figura 4.2. Programa para solucionar ejercicio 4.1

```
% Ejemplo 4.1 - Cubo de resistencias de 1 ohm
clear
% Se define la matriz de conductancias
gMat=nmAcc(1,0,1,zeros(7));      gMat=nmAcc(2,1,1,gMat);
gMat=nmAcc(3,2,1,gMat);          gMat=nmAcc(0,3,1,gMat);
gMat=nmAcc(5,4,1,gMat);          gMat=nmAcc(6,5,1,gMat);
gMat=nmAcc(7,6,1,gMat);          gMat=nmAcc(4,7,1,gMat);
gMat=nmAcc(4,0,1,gMat);          gMat=nmAcc(5,1,1,gMat);
gMat=nmAcc(6,2,1,gMat);          gMat=nmAcc(7,3,1,gMat)
Is(:,1)=srcAcc(0,6,1,zeros(7,1));
Is(:,2)=srcAcc(0,2,1,zeros(7,1));
Is(:,3)=srcAcc(0,1,1,zeros(7,1));
V=gMat\Is
disp('El valor de la resistencia del nodo 6 a tierra es:')
fprintf('R_61 = %f\n', V(6,1))
disp('El valor de la resistencia del nodo 2 a tierra es:')
fprintf('R_22 = %f\n', V(2,2))
disp('El valor de la resistencia del nodo 1 a tierra es:')
fprintf('R_13 = %f\n', V(1,3))
```

a) Se aplica una fuente de corriente de 1A del nodo 0 al nodo 6 y se mide el voltaje en el nodo 6.

b) Se aplica una fuente de corriente de 1A del nodo 0 al nodo 2 y se mide el voltaje en el nodo 2.

c) Se aplica una fuente de corriente de 1A del nodo 0 al nodo 1 y se mide el voltaje en el nodo 1.

Los resultados numéricos para las matrices de conductancias y de voltajes de nodo están en la figura 4.3. Los valores de las resistencias son 0.8333Ω, 0.7500Ω y 0.5833Ω para las conexiones entre los nodos 0 y 6, 0 y 2, y 0 y 1 respectivamente.

Figura 4.3. Solución para ejercicio 4.1

```
gMat =

     3    -1     0     0    -1     0     0
    -1     3    -1     0     0    -1     0
     0    -1     3     0     0     0    -1
     0     0     0     3    -1     0    -1
    -1     0     0    -1     3    -1     0
     0    -1     0     0    -1     3    -1
     0     0    -1    -1     0    -1     3


V =

    0.3333    0.3750    0.5833
    0.5000    0.7500    0.3750
    0.3333    0.3750    0.2083
    0.3333    0.2500    0.2083
    0.5000    0.3750    0.3750
    0.8333    0.5000    0.3333
    0.5000    0.3750    0.2500

El valor de la resistencia del nodo 6 a tierra es:
R_61 = 0.833333
El valor de la resistencia del nodo 2 a tierra es:
R_22 = 0.750000
El valor de la resistencia del nodo 1 a tierra es:
R_13 = 0.583333
```

4.3. FUNCIÓN gmAcc

Función **gmAcc**, sirve para acumular los términos de las fuentes dependientes en la matriz de conductancias. Esta función posee seis argumentos. Los primeros dos son **j** y **jp**, identifican los nodos del elemento que controla la fuente dependiente. Los dos siguientes son **k** y **kp**, identifican los nodos donde se encuentra conectada la fuente dependiente. El quinto argumento es **val**, identifica el valor de la ganancia de la fuente. El último argumento es **bmat**, identifica la matriz de conductancias sobre la cual acumula el valor de la fuente.

```
Function amat = gmAcc (j, jp, k, kp, val, bmat)
if (j<0)|(jp<0)|(k<0)|(kp<0)
	disp('Los indices no pueden ser negativos')
elseif ((j==0) & (jp==0)) | ((k==0) & (kp==0))
	disp('Un elemento trivial en el circuito')
else
	if (k > 0) & (j > 0)
		bmat (k, j)=bmat (k, j)+val;
	end
	if (kp > 0) & (jp > 0)
		bmat (kp, jp)=bmat (kp, jp)+val;
	end
	if (k > 0) & (jp > 0)
		bmat (k, jp)=bmat (k, jp)-val;
	end
	if (kp > 0) & (j > 0)
		bmat (kp, j)=bmat (kp, j)-val;
	end
end
amat = bmat;
```

- gmAcc (j, jp, k, kp, val, bmat)
- Acumula los valores de gm en la matriz b (bmat)
- j y jp son los nodos de control positivo y negativo
- k y kp son desde y hacia los nodos donde se encuentra conectada la fuente dependiente

4.4. FUNCIÓN genAnal

Función **genAnal**, permite calcular el vector voltaje de nodo o el vector corriente de malla. Consta de ocho argumentos. El primer argumento es **xs** que representa el vector fuente independiente, el segundo es **axx** que identifica la matriz de conductancias o resistencias, el tercer argumento es **axd**, el cual identifica la matriz de fuente dependiente, el cuarto es **axs**, éste argumento representa las componentes de las corrientes debido al vector fuente independiente, el quinto es **adc**, este relaciona las fuentes dependientes con las variables controladoras, el sexto es **acd**, este relaciona las variables controladoras con las fuentes dependientes, el séptimo argumento es **acx**, el cual relaciona las variables controladoras con los voltajes de nodo o las corrientes de malla y el último argumento es **acs**, este relaciona las variables controladoras con las fuentes independientes.

```
Function xx = genAnal (xs, axx, axd, axs, adc, acd, acx, acs)

[m,n]=size (adc);
b=axd*((eye (m)-adc*acd)\adc);
xx=(axx+b*acx)\(axs-b*acs)*xs;
```

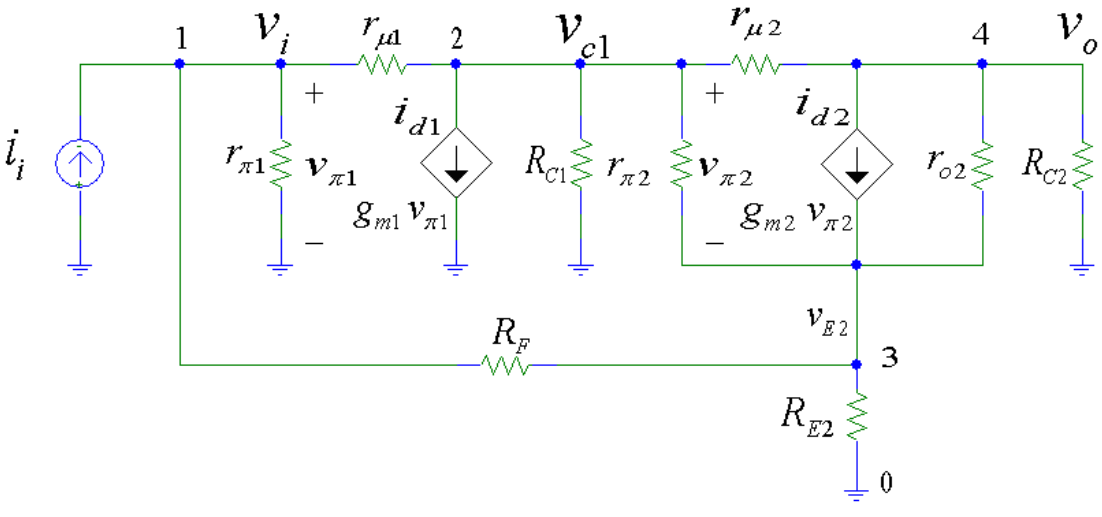
5. DESARROLLO DE LOS PROGRAMAS Y COMPARACIÓN DE RESULTADOS CON LOS ANÁLISIS TEÓRICOS Y LOS PROGRAMAS COMERCIALES DE CIRCUITOS

En el presente capítulo se desarrollarán una serie de ejercicios en los que se aplicarán las funciones vistas en el capítulo cuarto; los resultados serán confrontados con las soluciones dadas por los programas de simulación CircuitMaker y Spice, además de la solución realizada manualmente.

Ejercicio 5.1 Circuito con fuentes de corriente controladas por voltaje (FCCV).

Para el circuito mostrado en la figura 5.1, realizar un programa para evaluar la matriz de conductancias G_{nn} y la matriz fuente de corriente I_s, y hallar el vector voltaje de nodo Vn.

Figura 5.1. Ejercicio del circuito de realimentación del transistor ($g_{m1} = g_{m2} = 40$ S, $r_{\pi1} = r_{\pi2} = 2.5\ \Omega$, $r_{o2} = 100\ \Omega$, $r_{\mu1} = r_{\mu2} = 10\text{k}\Omega$, $R_{C1} = 10\Omega$, $R_{C2} = 8\ \Omega$, $R_{E2} = 1.3\ \Omega$, $R_F = 10\ \Omega$, $I_i = 1$A).



5.1.1. Solución manual aplicando análisis nodal

Al momento de realizar las operaciones se debe tener en cuenta que: $V_{\pi1} = V_1$, $V_{\pi2} = V_2 - V_3$ y que se trabaja con conductancias en lugar de resistencias.

Las ecuaciones empleando LKC en los nodos 1, 2, 3 y 4 son:

$$\left(V_1 - V_2\right)G_{m1} + \left(V_1 - V_3\right)G_F + V_1 G_{\pi 1} = I_i \tag{5.1}$$

$$\left(V_2 - V_1\right)G_{m1} + i_{d1} + V_2 G_{C1} + \left(V_2 - V_3\right)G_{\pi 2} + \left(V_2 - V_4\right)G_{m2} = 0 \tag{5.2}$$

$$\left(V_3 - V_1\right)G_F + \left(V_3 - V_2\right)G_{\pi 2} + \left(V_3 - V_4\right)G_{O2} + V_3 G_{E2} = i_{d2} \tag{5.3}$$

$$\left(V_4 - V_2\right)G_{m2} + \left(V_4 - V_3\right)G_{O2} + i_{d2} + V_4 G_{C2} = 0 \tag{5.4}$$

Agrupando los coeficientes de cada voltaje de nodo y teniendo en cuenta que la corriente i_{d1} es igual a $g_{m1}v_{\pi 1}$ y que i_{d2} es igual a $g_{m2}v_{\pi 2}$, se tiene:

$$\left(G_{m1} + G_F + G_{\pi 1}\right)V_1 - G_{m1}V_2 - G_F V_3 = I_i \tag{5.5}$$

$$-\left(G_{m1} - G_{m1}\right)V_1 + \left(G_{m1} + G_{C1} + G_{\pi 2} + G_{m2}\right)V_2 - G_{\pi 2}V_3 - G_{m2}V_4 = 0 \tag{5.6}$$

$$-G_F V_1 - \left(G_{\pi 2} + G_{m2}\right)V_2 + \left(G_{m2} + G_F + G_{\pi 2} + G_{O2} + G_{E2}\right)V_3 - G_{O2}V_4 = 0 \tag{5.7}$$

$$-\left(G_{m2} - G_{m2}\right)V_2 - \left(G_{O2} + G_{m2}\right)V_3 + \left(G_{m2} + G_{O2} + G_{C2}\right)V_4 = 0 \tag{5.8}$$

Escribiendo éstas en forma matricial tenemos:

$$\begin{bmatrix} G_{m1} + G_F + G_{\pi 1} & -G_{m1} & -G_F & 0 \\ -G_{m1} + G_{m1} & G_{m1} + G_{C1} + G_{\pi 2} + G_{m2} & -G_{\pi 2} & -G_{m2} \\ -G_F & -G_{\pi 2} - G_{m2} & G_{m2} + G_F + G_{\pi 2} + G_{O2} + G_{E2} & -G_{O2} \\ 0 & -G_{m2} + G_{m2} & -G_{O2} - G_{m2} & G_{m2} + G_{O2} + G_{C2} \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \end{bmatrix} = \begin{bmatrix} I_i \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{5.9}$$

Reorganizando las matrices para hallar los voltajes, se tiene:

$$\begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \end{bmatrix} = \begin{bmatrix} G_{m1} + G_F + G_{\pi 1} & -G_{m1} & -G_F & 0 \\ -G_{m1} + G_{m1} & G_{m1} + G_{C1} + G_{\pi 2} + G_{m2} & -G_{\pi 2} & -G_{m2} \\ -G_F & -G_{\pi 2} - G_{m2} & G_{m2} + G_F + G_{\pi 2} + G_{O2} + G_{E2} & -G_{O2} \\ 0 & -G_{m2} + G_{m2} & -G_{O2} - G_{m2} & G_{m2} + G_{O2} + G_{C2} \end{bmatrix}^{-1} \begin{bmatrix} I_i \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{5.10}$$

Remplazando los valores dados en el problema y teniendo en cuenta que se está trabajando con conductancias, se tiene:

$$\begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \end{bmatrix} = \begin{bmatrix} 0.5001 & -0.0001 & -0.1 & 0 \\ 39.9999 & 0.5002 & -0.4 & -0.0001 \\ -0.1 & -40.4 & 41.2792 & -0.01 \\ 0 & 39.9999 & -40.01 & 0.1351 \end{bmatrix}^{-1} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Al invertir la matriz de conductancias se obtiene:

$$\begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \end{bmatrix} = \begin{bmatrix} 0.0277 & 0.0247 & 0.0003 & 0.0000 \\ -10.0824 & 0.1260 & -0.0249 & -0.0017 \\ -9.8512 & 0.1232 & 0.0018 & 0.0002 \\ 67.7214 & -0.8269 & 7.8963 & 7.9858 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Los voltajes obtenidos son

$$\begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \end{bmatrix} = \begin{bmatrix} 0.0277 \\ -10.0824 \\ -9.8512 \\ 67.7214 \end{bmatrix}$$

5.1.2 Solución empleando MATLAB

El programa para resolver este circuito aparece en la figura 5.2. La solución aparece en la figura 5.3. Las unidades son V, mA y kΩ para voltaje, corriente y resistencia respectivamente. El programa almacena los valores de las conductancias y los acumula en una matriz de conductancias de 4 x 4 G_{nn} usando la función nmAcc, además utiliza la función srcAcc, para ingresar una fuente de corriente I_i de 1mA en la matriz fuente I_s.

Figura 5.2. Programa para resolver el ejercicio 5.1.

```
% Ejemplo 5.1 - circuito con fuentes de corriente controladas por voltaje
gm1=40;
gm2=40;
gpi=1/2.5;
go=1/100;
gmu=1/1E4;
GC1=1/10;
GC2=1/8;
GE2=1/1.3;
GF=1/10;
% Se define la matriz de conductancias Gnn
Gnn=nmAcc(1,0,gpi,zeros(4));
Gnn=nmAcc(1,2,gmu,Gnn);
Gnn=nmAcc(2,0,GC1,Gnn);
Gnn=nmAcc(2,3,gpi,Gnn);
Gnn=nmAcc(2,4,gmu,Gnn);
Gnn=nmAcc(4,3,go,Gnn);
Gnn=nmAcc(4,0,GC2,Gnn);
Gnn=nmAcc(3,0,GE2,Gnn);
Gnn=nmAcc(1,3,GF,Gnn);
Gnn=gmAcc(1,0,2,0,gm1,Gnn);
Gnn=gmAcc(2,3,4,3,gm2,Gnn)
Is=srcAcc(0,1,1,zeros(4,1));
Vn=Gnn\Is
```

Figura 5.3. Solución para ejercicio 5.1

```
Gnn =

    0.5001   -0.0001   -0.1000         0
   39.9999    0.5002   -0.4000   -0.0001
   -0.1000  -40.4000   41.2792   -0.0100
         0   39.9999  -40.0100    0.1351


Vn =

    0.0277
  -10.0824
   -9.8512
   67.7238
```

5.1.3 Solución empleando CircuitMaker

En este programa de simulación se crea el circuito dado en el ejercicio 5.1. Se simula el circuito y luego se procede a medir el voltaje en los nodos solicitados (V_1, V_2, V_3 y V_4).

Figura 5.4. Solución para el nodo 1 del ejercicio 5.1

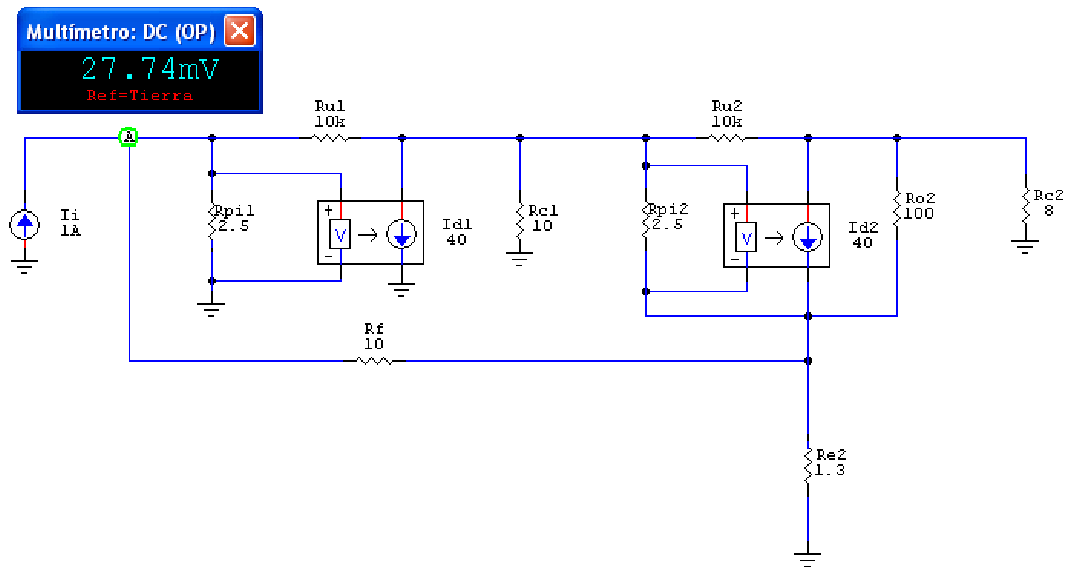


Figura 5.5. Solución para el nodo 2 del ejercicio 5.1

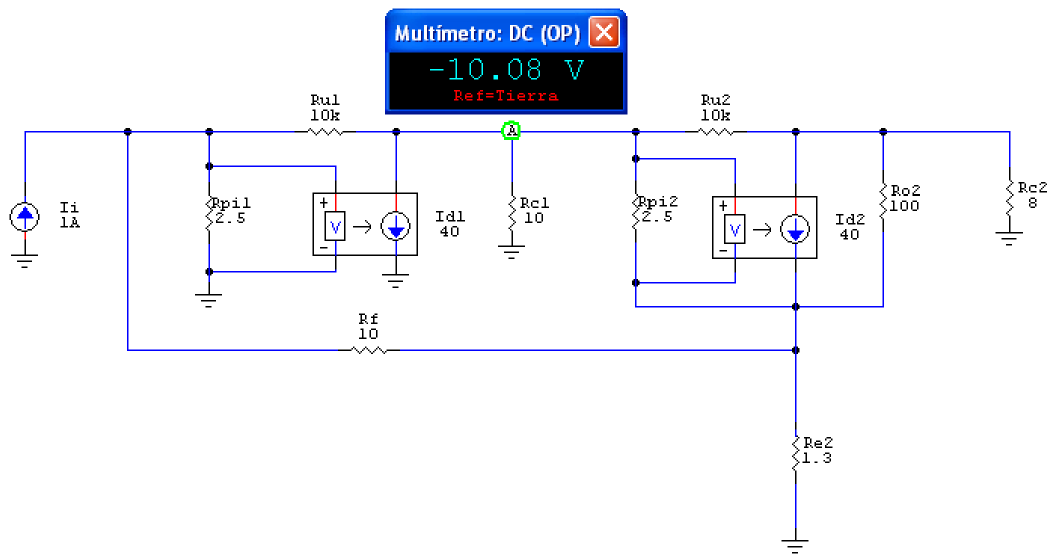


Figura 5.6. Solución para el nodo 3 del ejercicio 5.1

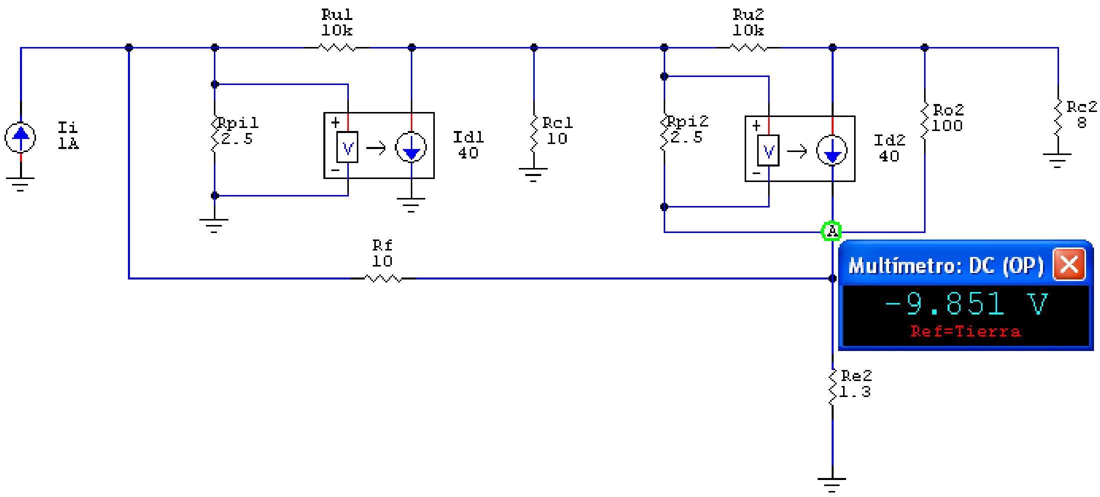
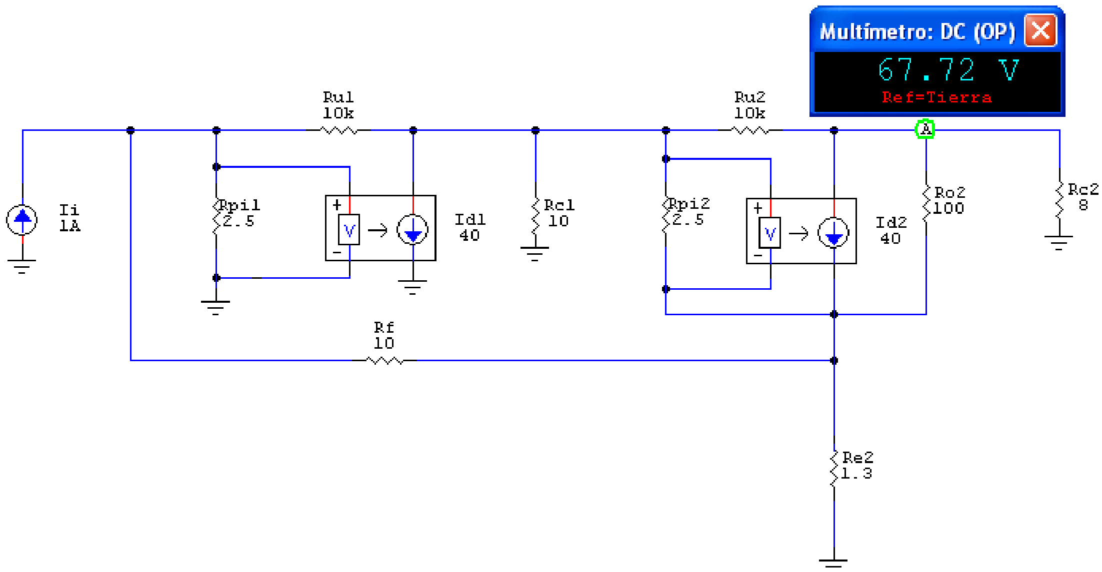


Figura 5.7. Solución para el nodo 4 del ejercicio 5.1



5.1.4 Solución empleando Spice

Se crea el circuito en el programa de simulación Spice y se solicita el valor de los voltajes de nodo por medio de una función de librería de dicho programa.

Figura 5.8. Simulación para el ejercicio 5.1.

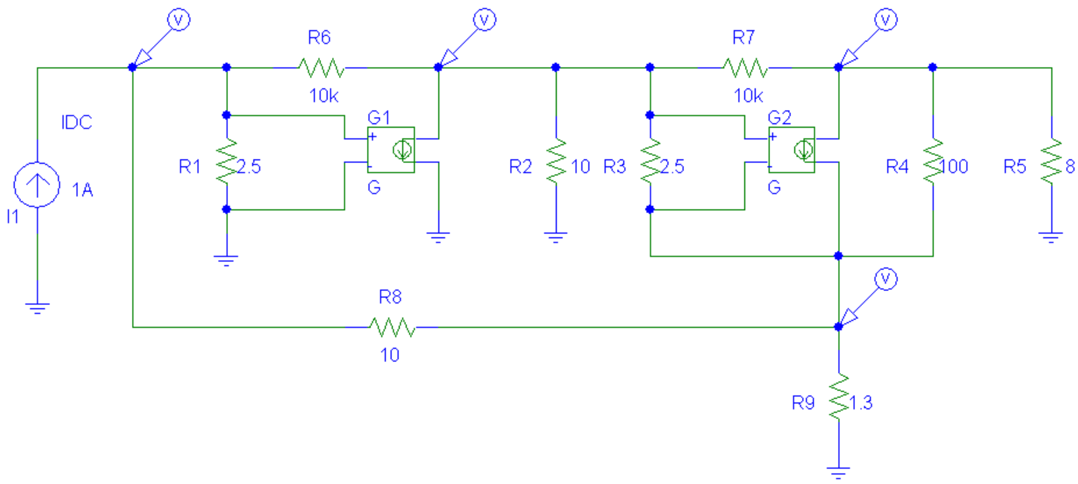


Figura 5.9. Solución aplicando Spice para el ejercicio 5.1

NODE	VOLTAGE	NODE	VOLTAGE
(N_0001)	.0277	(N_0002)	-10.0820
(N_0003)	-9.8512	(N_0004)	67.7240

Ejercicio 5.2 Circuito con fuentes de voltaje controladas por corriente (FVCC) y fuentes de corriente controladas por corriente (FCCC)

El circuito mostrado en la figura 5.10 donde se tiene una FVCC y una FCCC

La solución del circuito se logra siguiendo los pasos a), b), c) y d), teniendo en cuenta la información y ecuaciones suministradas en el capítulo tres.

a) Establecer los nodos y supernodos con los voltajes apropiados e indicar la(s) superficie(s) del supernodo o supernodos encontrados.

b) Escribir las ecuaciones de nodo y/o supernodo en términos de los voltajes de nodo, fuentes independientes y dependientes. Reescribir dichas ecuaciones, agrupando coeficientes y colocando los términos independientes al lado derecho de cada ecuación. Escribir estas ecuaciones en forma matricial.

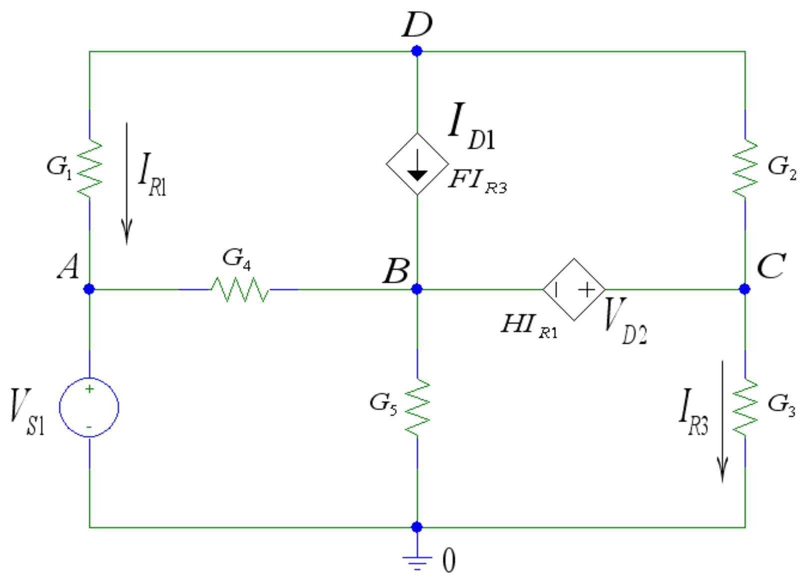
c) Escribir las relaciones de dependencia para I_{D1} y V_{D2} en la forma de la ecuación 3.21.

d) Escribir las relaciones para las variables controladoras I_{R1} e I_{R3} en términos de las fuentes independientes, los voltajes de nodo y las fuentes dependientes

De acuerdo con los pasos a), b), c) y d), calcular:

e) Los voltajes de nodo, los valores de las fuentes dependientes (I_{D1} y V_{D2}) y las corrientes controladoras (I_{R1} e I_{R3}).

Figura 5.10. Circuito para el ejercicio con FVCC y FCCC (V_{S1} = 10V, $G_1 = G_2 = G_3 = G_4 = G_5$ = 2S, F = 10 y H = 5Ω).



5.2.1 Solución manual aplicando análisis nodal

a) En la figura 5.11 se muestra el diagrama del circuito con los voltajes de nodo y la superficie del supernodo. Para este circuito n+1 es igual a 5 y n_{fv} igual a 2, por lo tanto el número de voltajes de nodo independientes es

Número de ecuaciones = $n - n_{fv} = 4 - 2 = 2$

El diagrama del circuito muestra que el nodo del centro es el nodo 1 y el nodo de la parte superior es el nodo 2. La fuente V_{S1} entrega el voltaje al nodo A y la fuente dependiente V_{D2} relaciona los voltajes de los nodos C y B (nodo 1). Los nodos C y D forman el supernodo.

b) Las ecuaciones del supernodo y el nodo 2 son:

$$G_2\left[\left(V_{N1}+V_{D2}\right)-V_{N2}\right]+G_3\left(V_{N1}+V_{D2}\right)+G_4\left(V_{N1}-V_{S1}\right)+G_5V_{N1}-I_{D1}=0 \qquad (5.11)$$

$$G_1\left(V_{N2}-V_{S1}\right)+G_2\left[V_{N2}-\left(V_{N1}+V_{D2}\right)\right]+I_{D1}=0 \qquad (5.12)$$

Agrupando coeficientes tenemos:

$$\left(G_2+G_3+G_4+G_5\right)V_{N1}-G_2V_{N2}-I_{D1}+\left(G_2+G_3\right)V_{D2}=G_4V_{S1} \qquad (5.13)$$

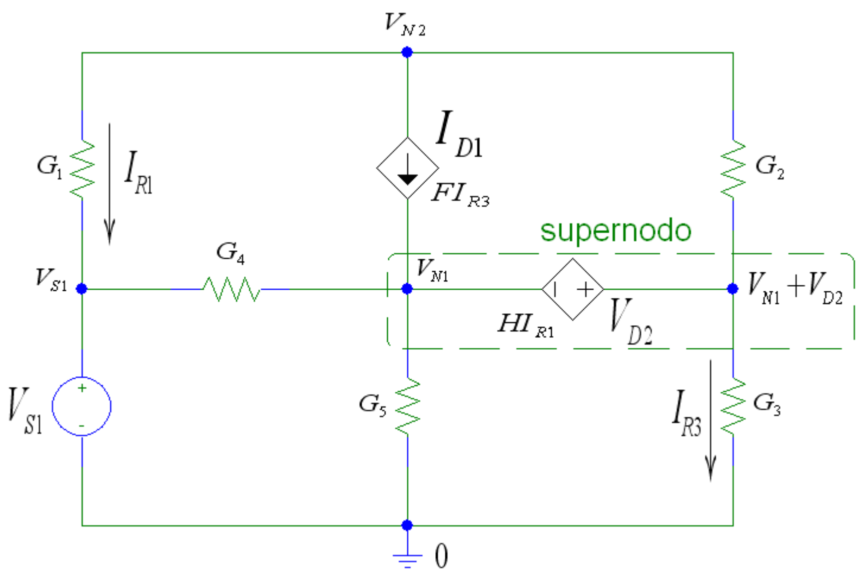
$$-G_2V_{N1}+\left(G_1+G_2\right)V_{N2}+I_{D1}-G_2V_{D2}=G_1V_{S1} \qquad (5.14)$$

Escribiendo las ecuaciones 5.13 y 5.14 en forma matricial, se tiene:

$$\begin{bmatrix} G_2+G_3+G_4+G_5 & -G_2 \\ -G_2 & G_1+G_2 \end{bmatrix}\begin{bmatrix} V_{N1} \\ V_{N2} \end{bmatrix}+\begin{bmatrix} 1 & G_2+G_3 \\ 1 & -G_2 \end{bmatrix}\begin{bmatrix} I_{D1} \\ V_{D2} \end{bmatrix}=\begin{bmatrix} G_4 \\ G_1 \end{bmatrix}V_{S1} \qquad (5.15)$$

Inspeccionando el circuito es evidente que G_2, G_3, G_4 y G_5 son conductancias propias del supernodo, y G_1 y G_2 son conductancias propias del nodo 2.

Figura 5.11. Circuito mostrando los nodos y supernodos



c) Para este circuito las relaciones de dependencia para I_{D1} y V_{D2} según la ecuación 3.21 son:

$$I_{D1} = F\, I_{R3} \tag{5.16}$$

$$V_{D2} = H\, I_{R1} \tag{5.17}$$

Escribiendo estas ecuaciones en forma matricial se tiene:

$$\begin{bmatrix} I_{D1} \\ V_{D2} \end{bmatrix} = \begin{bmatrix} F & 0 \\ 0 & H \end{bmatrix} \begin{bmatrix} I_{R3} \\ I_{R1} \end{bmatrix} \tag{5.18}$$

d) Las relaciones de las variables controladoras son:

$$I_{R3} = G_3 \left(V_{N1} + V_{D2} \right) \tag{5.19}$$

$$I_{R1} = G_1 \left(V_{N2} - V_{S1} \right) \tag{5.20}$$

En forma matricial y según la ecuación 3.22 se tiene

$$\begin{bmatrix} I_{R3} \\ I_{R1} \end{bmatrix} = \begin{bmatrix} 0 & G_3 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} I_{D1} \\ V_{D2} \end{bmatrix} + \begin{bmatrix} G_3 & 0 \\ 0 & G_1 \end{bmatrix} \begin{bmatrix} V_{N1} \\ V_{N2} \end{bmatrix} + \begin{bmatrix} 0 \\ G_1 \end{bmatrix} V_{S1} \tag{5.21}$$

e) Para hallar los voltajes de nodo (V_{N1} y V_{N2}), los valores de las fuentes dependientes (I_{D1} y V_{D2}) y las corrientes controladoras (I_{R3} e I_{R1}) se crea una matriz según la ecuación matricial 3.33.:

$$\left[\begin{array}{cc:cc:cc:c} G_2+G_3+G_4+G_5 & -G_2 & -1 & G_2+G_3 & 0 & 0 & 0 \\ -G_2 & G_1+G_2 & 1 & -G_2 & 0 & 0 & \\ \hdashline 0 & 0 & 1 & 0 & -F & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & -H & \\ \hdashline -G_3 & 0 & 0 & -G_3 & 1 & 0 & 0 \\ 0 & -G_1 & 0 & 0 & 0 & 1 & \\ \hdashline \multicolumn{2}{c:}{-A_{on}} & \multicolumn{2}{c:}{-A_{od}} & \multicolumn{2}{c:}{0} & I \end{array}\right] \begin{bmatrix} V_N \\ X_d \\ X_c \\ X_o \end{bmatrix} = \left[\begin{array}{c} G_4 \\ G_1 \\ \hdashline 0 \\ 0 \\ \hdashline 0 \\ -G_1 \\ \hdashline A_{os} \end{array}\right] V_{S1} \tag{5.22}$$

V_N representa los voltajes de nodo V_{N1} y V_{N2}, X_d representa los valores de las fuentes dependientes I_{D1} y V_{D2}, y X_c representa las corrientes controladoras I_{R3} e I_{R1}. También se elimina la cuarta fila debido a que el vector X_o de la salida no es necesario determinarlo en el ejercicio. Por lo tanto al despejar el vector solución se da lo siguiente:

$$\begin{bmatrix} V_N \\ X_d \\ X_c \end{bmatrix} = \begin{bmatrix} G_2+G_3+G_4+G_5 & -G_2 & -1 & G_2+G_3 & 0 & 0 \\ -G_2 & G_1+G_2 & 1 & -G_2 & 0 & 0 \\ 0 & 0 & 1 & 0 & -F & 0 \\ 0 & 0 & 0 & 1 & 0 & -H \\ -G_3 & 0 & 0 & -G_3 & 1 & 0 \\ 0 & -G_1 & 0 & 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} G_4 \\ G_1 \\ 0 \\ 0 \\ 0 \\ G_1 \end{bmatrix} V_{S1} \tag{5.23}$$

Al reemplazar por los valores dados en el ejercicio se obtiene:

$$\begin{bmatrix} V_N \\ X_d \\ X_c \end{bmatrix} = \begin{bmatrix} 8 & -2 & -1 & 4 & 0 & 0 \\ -2 & 4 & 1 & -2 & 0 & 0 \\ 0 & 0 & 1 & 0 & -10 & 0 \\ 0 & 0 & 0 & 1 & 0 & -5 \\ -2 & 0 & 0 & -2 & 1 & 0 \\ 0 & -2 & 0 & 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 2 \\ 2 \\ 0 \\ 0 \\ 0 \\ -2 \end{bmatrix} 10$$

Al invertir la matriz se obtiene:

$$\begin{bmatrix} V_N \\ X_d \\ X_c \end{bmatrix} = \begin{bmatrix} 0.2599 & 0.2288 & 0.0311 & 0.0395 & 0.3107 & 0.1977 \\ -0.0254 & -0.0169 & -0.0085 & -0.1017 & -0.0847 & -0.5085 \\ 0.1130 & 1.1864 & -0.0734 & 0.4520 & -0.7345 & 2.2599 \\ -0.2542 & -0.1695 & -0.0847 & -0.0169 & -0.8475 & -0.0847 \\ 0.0113 & 0.1186 & -0.1073 & 0.0452 & -0.0734 & 0.2260 \\ -0.0508 & -0.0339 & -0.0169 & -0.2034 & -0.1695 & -0.0169 \end{bmatrix} \begin{bmatrix} 20 \\ 20 \\ 0 \\ 0 \\ 0 \\ 20 \end{bmatrix}$$

El vector solución es:

$$\begin{bmatrix} V_N \\ X_d \\ X_c \end{bmatrix} = \begin{bmatrix} 5.8192 \\ 9.3220 \\ -19.2090 \\ -6.7797 \\ -1.9209 \\ -1.3559 \end{bmatrix}$$

5.2.2 Solución empleando MATLAB

En la figura 5.12 se puede observar el programa para la solución numérica del circuito que se muestra en la figura 5.11. La información sobre la función genAnal, la cual usa la ecuación 3.26 se encuentra en el capítulo cuarto.

Figura 5.12. Programa para la solución del ejercicio 5.2

```
% Ejercicio 5.2 - Análisis Nodal con FVCC y FCCC
clear
format compact
G1=2;   G2=2;   G3=2;   G4=2;   G5=2;
% Se define la matriz de conductancias.
Gnn=nmAcc(1,2,G2,zeros(2,2));   Gnn=nmAcc(1,0,G3+G4+G5,Gnn);
Gnn=nmAcc(2,0,G1,Gnn);
% Se define And                 % Se define Ans
And=[-1 G2+G3;1 -G2];           Ans=[G4;G1];
% Se define Adc                 % Se define Acd
Adc=[10 0;0 5];                 Acd=[0 G3;0 0];
% Se define Acn                 % Se define Acs
Acn=[G3 0;0 G1];                Acs=[0;-G1];
% Se define el vector fuente
Vs1=10;
disp('Vector voltage de nodo Vn empleando genAnal:')
Vn=genAnal(Vs1,Gnn,And,Ans,Adc,Acd,Acn,Acs);
disp(Vn)
disp('Super matriz Gsuper:')
[rGnn,cGnn]=size(Gnn);          [rAdc,cAdc]=size(Adc);
[rAcd,cAcd]=size(Acd);
Znc=zeros(rGnn,cAdc);           Zdn=zeros(rAdc,cGnn);
Ud=eye(rAdc);                   Uc=eye(rAcd);
Gsuper=[Gnn And Znc;Zdn Ud -Adc;-Acn -Acd Uc];
disp(Gsuper)
XSuper=[Ans;zeros(rAdc,1);Acs];
disp('Vn, Xd, and Xc')
Ysuper=Gsuper\XSuper*Vs1;
disp(Ysuper)
```

La matriz solución usando la ecuación 3.33 aparece en la figura 5.13. Las dos primeras componentes son los voltajes de nodo, las dos siguientes son los valores de las fuentes dependientes y las últimas dos son los valores de las corrientes controladoras.

Figura 5.13. Solución del ejercicio 5.2

```
Vector voltage de nodo Vn empleando genAnal:
    5.8192
    9.3220
Super matriz Gsuper:
     8    -2    -1     4     0     0
    -2     4     1    -2     0     0
     0     0     1     0   -10     0
     0     0     0     1     0    -5
    -2     0     0    -2     1     0
     0    -2     0     0     0     1
Vn, Xd, and Xc
    5.8192
    9.3220
  -19.2090
   -6.7797
   -1.9209
   -1.3559
```

5.2.3 Solución empleando CircuitMaker

En las figuras 5.14 y 5.15 se muestra la solución para los voltajes de nodo V_{N1} y V_{N2} al simular los circuitos en CircuitMaker.

Figura 5.14. Solución para V_{N1} del ejercicio 5.2

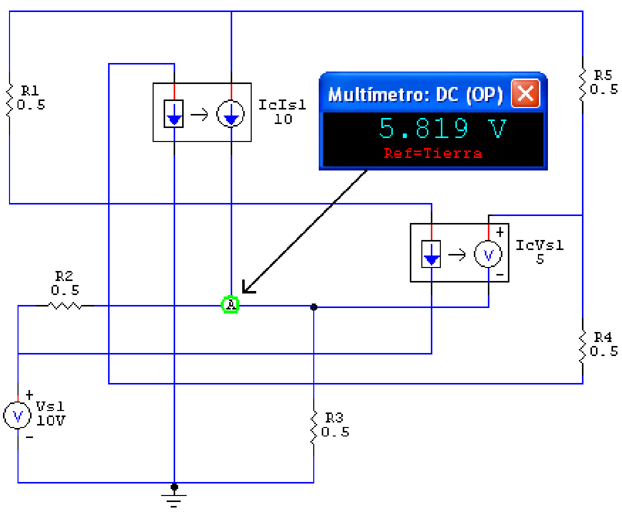
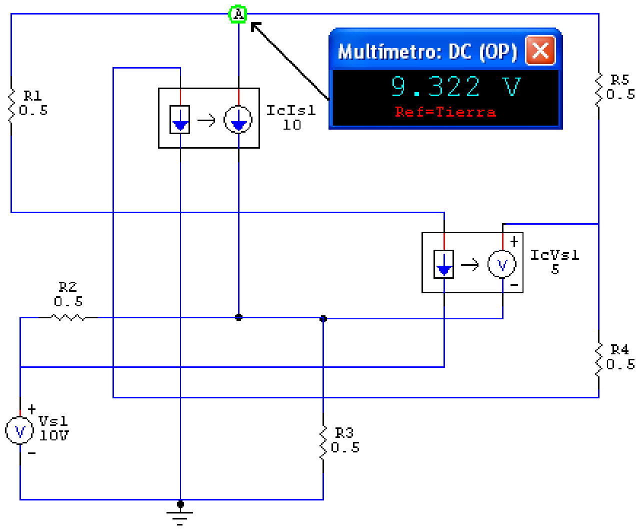


Figura 5.15. Solución para V_{N2} del ejercicio 5.2



5.2.4 Solución empleando Spice

El las figuras 5.17, 5.18 y 5.19 se muestra los resultados obtenidos para los voltajes de nodo, las fuentes dependientes y las corrientes controladoras al simular el circuito en Spice.

Figura 5.16. Simulación para el ejercicio con FVCC y FCCC

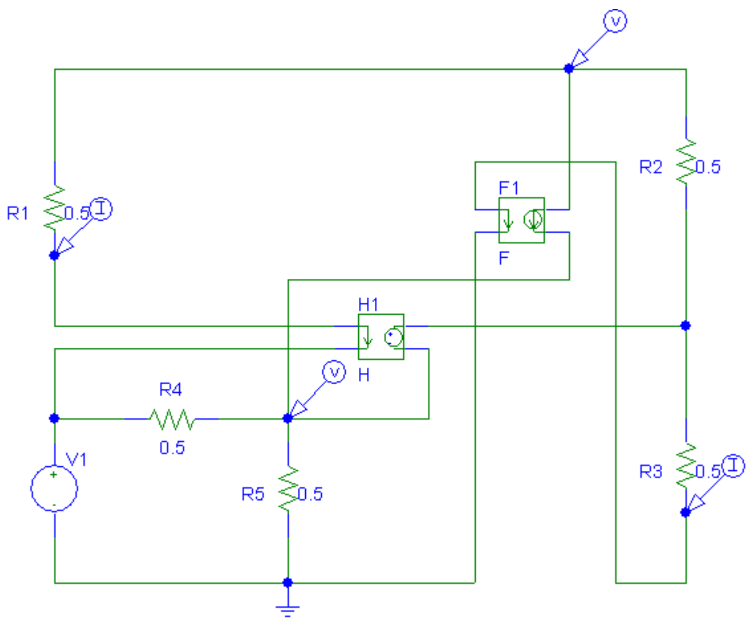


Figura 5.17. Solución para V_{N1} y V_{N2}.

```
 NODE   VOLTAGE


(N_0001)    5.8192

(N_0002)    9.3220
```

Figura 5.18. Valores de las fuentes dependientes

```
**** CURRENT-CONTROLLED CURRENT SOURCES

NAME          F_F1
I-SOURCE    -1.921E+01

**** CURRENT-CONTROLLED VOLTAGE SOURCES

NAME          H_H1
V-SOURCE    -6.780E+00
```

Figura 5.19. Valores de las corrientes controladoras

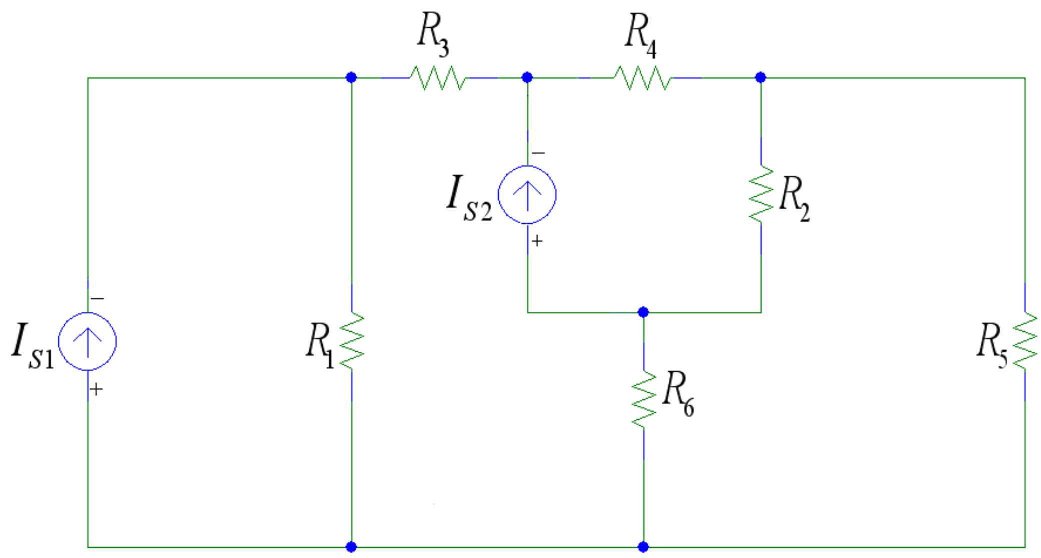
```
  CONTROLLING CURRENTS
  NAME          CURRENT

  VF_F1       -1.921E+00
  VH_H1       -1.356E+00
```

Ejercicio 5.3 Circuito con fuentes independientes para su solución con análisis de mallas y creando una supermalla

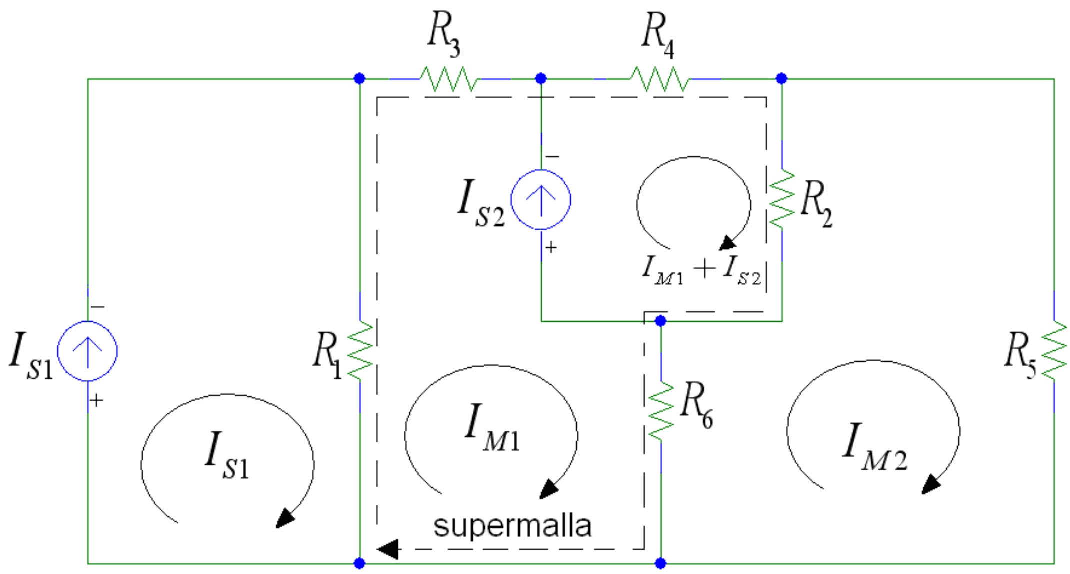
Para el circuito de la figura 5.20 encontrar el vector solución de corrientes si I_{S1} es 1A e I_{S2} es 0, y si I_{S1} es 0 e I_{S2} es 1A.

Figura 5.20. Circuito para el ejercicio 5.3.



5.3.1 Solución manual aplicando análisis de mallas

Figura 5.21. Circuito mostrando las mallas y la supermalla



En la supermalla se aplica LKV:

$$R_1\left(I_{M1}-I_{S1}\right)+R_3 I_{M1}+R_6\left(I_{M1}-I_{M2}\right)+R_2\left[\left(I_{M1}+I_{S2}\right)-I_{M2}\right]+R_4\left(I_{M1}+I_{S2}\right)=0 \qquad (5.24)$$

La ecuación para la malla 2 es:

$$R_2\left[I_{M2}-\left(I_{M1}+I_{S2}\right)\right]+R_5 I_{M2}+R_6\left[I_{M2}-I_{M1}\right]=0 \qquad (5.25)$$

Agrupando términos se tiene:

$$\left(R_1+R_2+R_3+R_4+R_6\right)I_{M1}-\left(R_2+R_6\right)I_{M2}=R_1 I_{S1}-\left(R_2+R_4\right)I_{S2} \qquad (5.26)$$

$$-\left(R_2+R_6\right)I_{M1}+\left(R_2+R_5+R_6\right)I_{M2}=R_2 I_{S2} \qquad (5.27)$$

Escribiendo las ecuaciones 5.26 y 5.27 en forma matricial se tiene:

$$\begin{bmatrix} R_1+R_2+R_3+R_4+R_6 & -\left(R_2+R_6\right) \\ -\left(R_2+R_6\right) & R_2+R_5+R_6 \end{bmatrix}\begin{bmatrix} I_{M1} \\ I_{M2} \end{bmatrix}=\begin{bmatrix} R_1 & -\left(R_2+R_4\right) \\ 0 & R_2 \end{bmatrix}\begin{bmatrix} I_{S1} \\ I_{S2} \end{bmatrix} \qquad (5.28)$$

Despejando el vector solución de corrientes:

$$\begin{bmatrix} I_{M1} \\ I_{M2} \end{bmatrix}=\begin{bmatrix} R_1+R_2+R_3+R_4+R_6 & -\left(R_2+R_6\right) \\ -\left(R_2+R_6\right) & R_2+R_5+R_6 \end{bmatrix}^{-1}\left\{\begin{bmatrix} R_1 & -\left(R_2+R_4\right) \\ 0 & R_2 \end{bmatrix}\begin{bmatrix} I_{S1} \\ I_{S2} \end{bmatrix}\right\} \qquad (5.29)$$

Remplazando términos por los valores dados en el problema y con I_{S1}=1A e I_{S2}=0

$$\begin{bmatrix} I_{M1} \\ I_{M2} \end{bmatrix}=\begin{bmatrix} 5 & -2 \\ 2 & 3 \end{bmatrix}^{-1}\left\{\begin{bmatrix} 1 & -2 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} 1 \\ 0 \end{bmatrix}\right\}$$

Esto es igual a:

$$\begin{bmatrix} I_{M1} \\ I_{M2} \end{bmatrix} = \begin{bmatrix} 0.2727 & 0.1818 \\ 0.1818 & 0.4545 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

El resultado obtenido es:

$$\begin{bmatrix} I_{M1} \\ I_{M2} \end{bmatrix} = \begin{bmatrix} 0.2727 \\ 0.1818 \end{bmatrix}$$

Ahora realizando el mismo procedimiento pero con $I_{S1}= 0$ e $I_{S2}=1A$ el vector solución de corrientes es

$$\begin{bmatrix} I_{M1} \\ I_{M2} \end{bmatrix} = \begin{bmatrix} 5 & -2 \\ -2 & 3 \end{bmatrix}^{-1} \left\{ \begin{bmatrix} 1 & -2 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right\}$$

Esto es igual a:

$$\begin{bmatrix} I_{M1} \\ I_{M2} \end{bmatrix} = \begin{bmatrix} 0.2727 & 0.1818 \\ 0.1818 & 0.4545 \end{bmatrix} \begin{bmatrix} -2 \\ 1 \end{bmatrix}$$

El resultado obtenido es:

$$\begin{bmatrix} I_{M1} \\ I_{M2} \end{bmatrix} = \begin{bmatrix} -0.3636 \\ 0.0909 \end{bmatrix}$$

5.3.2 Solución empleando MATLAB

En la figura 5.22 se muestra el programa para resolver este ejercicio y en la figura 5.23 se encuentra la solución numérica del mismo. Cuando se usa la función srcAcc para construir la matriz fuente, al ingresar un valor cuando una malla señala una dirección y la otra malla señala en dirección opuesta como sucede en la resistencia R_2, los argumentos de la función srcAcc son 1, 2, R_4 y $A_S(:,2)$ para acumular R_4 en la segunda columna de la matriz A_S. Para obtener ambas soluciones en el mismo programa se utiliza una matriz fuente de dos columnas.

Figura 5.22. Programa para la solución del ejercicio 5.3

```
% Ejercicio 5.3 - empleando una supermalla
clear
R1=1;
R2=1;
R3=1;
R4=1;
R5=1;
R6=1;
Rmat=nmAcc(0,1,R1,zeros(2,2));
Rmat=nmAcc(1,2,R2,Rmat);
Rmat=nmAcc(0,1,R3,Rmat);
Rmat=nmAcc(0,1,R4,Rmat);
Rmat=nmAcc(0,2,R5,Rmat);
Rmat=nmAcc(1,2,R6,Rmat)
As=zeros(2,2);
As(:,1)=srcAcc(0,1,R1,As(:,1));
As(:,2)=srcAcc(1,2,R2,As(:,2));
As(:,2)=srcAcc(1,0,R4,As(:,2))
Is=[1 0;0 1]
Im=Rmat\(As*Is)
```

Figura 5.23. Solución numérica para el ejercicio 5.3

```
Rmat =

     5    -2
    -2     3


As =

     1    -2
     0     1


Is =

     1     0
     0     1


Im =

    0.2727   -0.3636
    0.1818    0.0909
```

5.3.3 Solución empleando CircuitMaker

Se debe tener en cuenta que el CicuitMaker al momento de simular toma el sentido de las corrientes con respecto a la tierra. Por esto algunos resultados aparecen con signo contrario.

Figura 5.24. Solución para la supermalla (I_{M1})

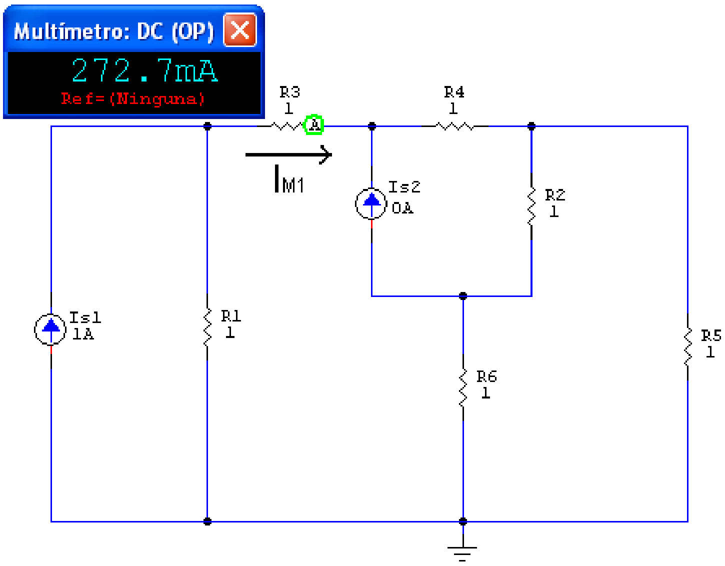
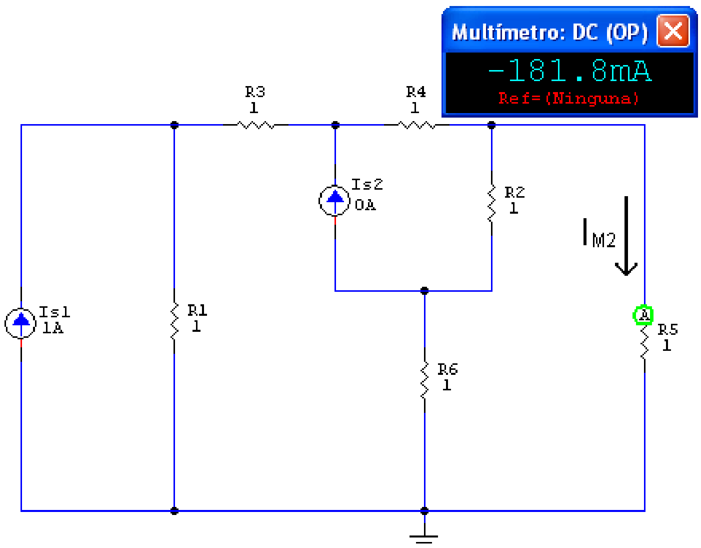


Figura 5.25. Solución para la malla 2 (I_{M2})



Para $I_{S1} = 0$ e $I_{S2} = 1A$, la solución es la siguiente:

Figura 5.26. Solución para la supermalla (I_{M1})

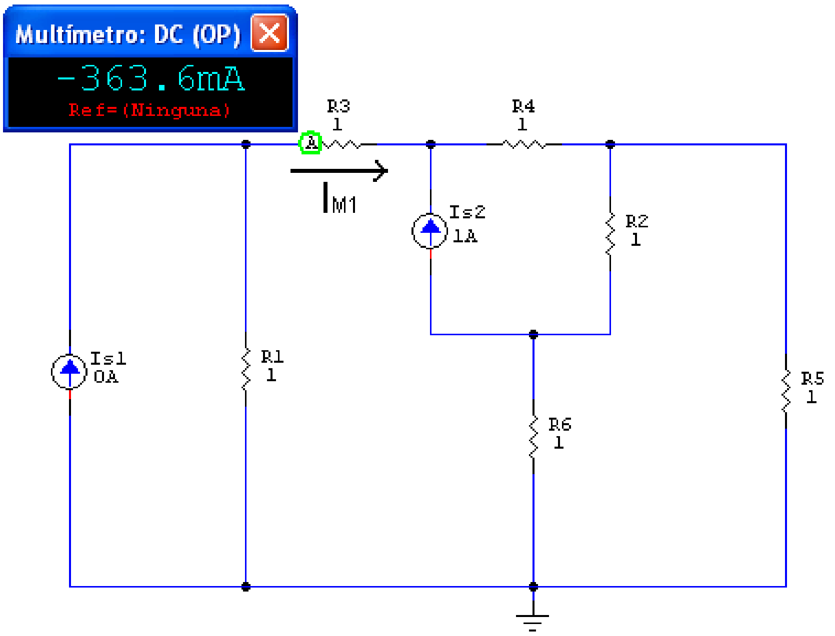
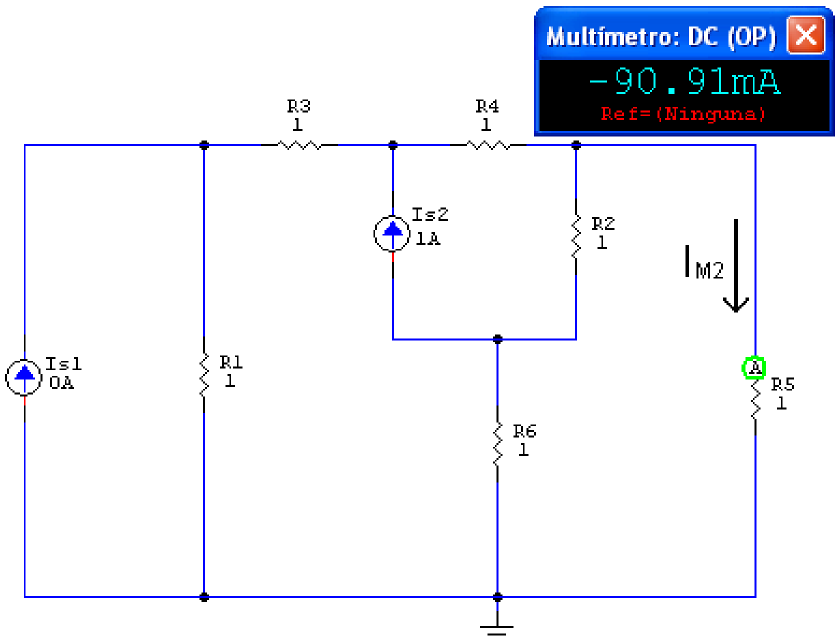


Figura 5.27. Solución para la malla 2 (I_{M2})



5.3.4 Solución empleando Spice

Figura 5.28. Solución para I_{M1} e I_{M2} cuando I_{S1} = 1A e I_{S2} = 0A

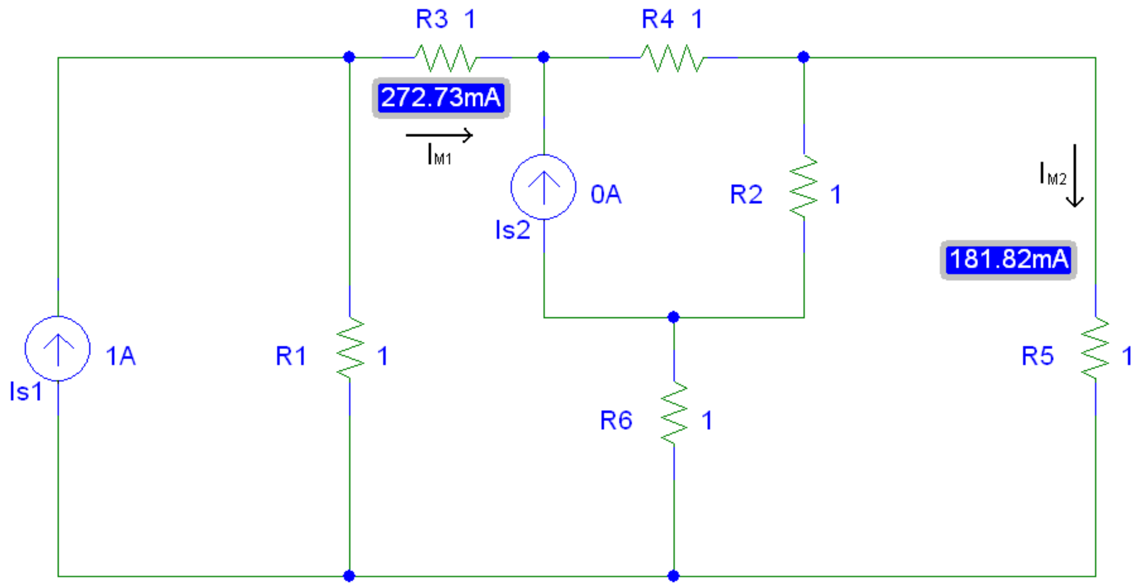
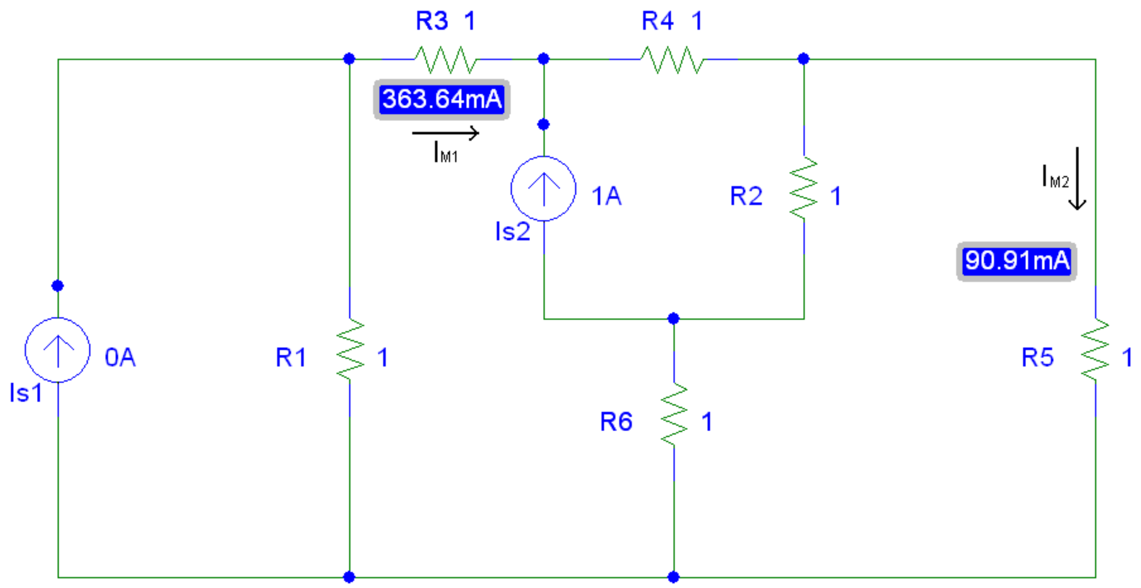


Figura 5.29. Solución para I_{M1} e I_{M2} cuando I_{S1} = 0A e I_{S2} = 1A



6. CONCLUSIONES

Al desarrollar los ejercicios planteados se aplicaron las funciones **nmAcc**, **srcAcc**, **gmAcc** y **genAnal**. La función **nmAcc** permite la agrupación de conductancias (o resistencias) en una sola matriz; la función **srcAcc** permite agrupar los valores de las fuentes de corriente en una matriz; la función **gmAcc** permite agrupar los términos de las fuentes dependientes en una matriz de conductancias y la función **genAnal** permite calcular el vector voltaje de nodo o el vector corriente de malla en el análisis de circuitos.

La apropiada utilización de las funciones internas y programadas del MATLAB depende del debido conocimiento de la teoría del análisis nodal y del análisis de mallas para obtener los resultados deseados.

En cada uno de los ejercicios del presente trabajo se evidencia el uso del análisis matricial en los circuitos de corriente continua y la aplicación de métodos como el análisis de nodos y el análisis de mallas.

La confrontación de los resultados obtenidos por los programas de simulación CircuitMaker y Spice, y por los métodos teóricos, fué una gran ventaja para verificar la certeza y exactitud de los programas y soluciones desarrollados y entregados por MATLAB respectivamente.

En el desarrollo del presente trabajo se utilizo la versión de MATLAB 7.0 la cual es una herramienta sencilla y rápida para realizar cálculos numéricos con vectores y matrices, y además es una gran ventaja al permitir una fácil interacción con el estudiante.

Al simular los circuitos en CircuitMaker o Spice se observó que en algunos casos el valor tomado y/o suministrado por el circuito aparecía con signo contrario, esto debido a que los programas de simulación toman un nodo como el nodo de referencia y con respecto a dicho nodo proyectan el sentido de las corrientes y por consiguiente el signo de los valores suministrados.

Al trabajar con MATLAB se puede concluir que los pasos a seguir para la solución de un circuito lineal son menos que los que hay que seguir en un proceso convencional (manual) para la solución del mismo problema.

Sería de gran importancia acompañar la parte teórica de la asignatura circuitos eléctricos I con los programas computacionales referenciados aquí. Justamente éste es uno de los propósitos principales de este trabajo.

BIBLIOGRAFIA

[1] ALEXANDER, Charles K. y SADIKU, Matthew N.O. Fundamentals of Electric circuits. USA. McGraw-Hill, 2000.

[2] GARCÍA DE JALÓN, Javier, RODRIGUEZ, José Ignacio y VIDAL, Jesús. Aprenda MATLAB 7.0 como si estuviera de primero. Extraído el 04 Abril, 2006 del sitio Web de la Universidad Politécnica de Madrid. Disponible en: http://www.tayuda.com/ayudainf/index.htm.

[3] GOTTLING, James G. Matrix Analysis of Circuits Using MATLAB. USA. Prentice-Hall. 1995.

[4] HANSELMAN, Duane; LITTLEFIELD, Bruce. Mastering MATLAB. A comprehensive tutorial and reference. USA. Prentice-Hall. 1996.

[5] HAYT, William H. y KEMMERLY, Jack E. Análisis de circuitos en Ingeniería. (Quinta Edición). México. McGraw-Hill, 1993.

[6] INVERSIÓN DE MATRICES, PAGINA WEB. Disponible en: http://luda.uam.mx/curso2/tema3/sistem03.html#inv

[7] JOHNSON, David E; HILBURN, John; JOHNSON, Johnny R. Análisis Básico de circuitos Eléctricos. México. Prentice-Hall Hispanoamericana S.A. 1991.

[8] MARULANDA OLARTE, Milton Cesar y RUIZ CRUZ, Uriel. Libro Guía de Circuitos Eléctricos II con Aplicaciones en MATLAB. Pereira, Año 2005. Tesis de grado (Ingeniero Electricista). Universidad Tecnológica de Pereira. Facultad de Ingeniería Eléctrica.

[9] MÉTODO DE GAUSS, PAGINA WEB. Disponible en: http://www.investigacion-operaciones.com/Calculo%20matricial.htm

[10] NILSSON, James W. y RIEDEL, Susan A. Circuitos eléctricos. México. Pearson Educación de México. 2001.

[11] PAUL, Clayton R. Fundamentals of Electric Circuit Analysis. USA. John Wiley & Sons, Inc.2001.

[12] TABARES, P. Y ESCOBAR, F. Sistemas de Ecuaciones Lineales y Matrices. Universidad Tecnológica de Pereira. 2001.