

ANÁLISIS DE REDES ELÉCTRICAS

Por E. Lora-Tamayo D'Ocon (*).

Introducción y resumen

El objeto del siguiente trabajo es poner a punto un programa Fortran destinado a analizar una red eléctrica en su caso más general es decir, que aparte de tener cargas resistivas, admita cargas capacitivas, e inductivas (pudiendo estas últimas estar situadas en distintas mallas y acopladas entre sí), utilizando un método rápido basado en la especial estructura de las matrices eléctricas.

Pretendemos que una vez descrita la red o el sistema eléctrico a analizar, a la computadora (mediante un algoritmo matricial), ésta nos dé como resultado las tensiones e intensidades en cada rama, así como las tensiones de cada nodo respecto a otro cualquiera de ellos elegido como nodo de referencia o de tierra.

Con estos resultados sería relativamente fácil elaborar una serie de programas de una utilidad práctica mucho más concreta, como puede ser el cálculo de la potencia consumida en líneas de transmisión, el reparto óptimo de potencia desde un generador a una serie de cargas, la distribución de potencia entre varios generadores que suministran energía a una carga grande, etc...

El proceso consiste en traducir el problema eléctrico a un problema de tipo matricial, y es mediante el álgebra de matrices y de sistemas lineales como se resuelve.

En cuanto a esto último, el método más exacto y más rápido para obtener la solución de un sistema de ecuaciones lineales es el de triangularización de Gauss-Seidel, y es el que

(*) Trabajo dirigido por R. García de la Rosa, del Instituto de Electricidad y Automática del C.S.I.C.

usaremos en algunos ejemplos. Ahora bien, en el análisis de redes eléctricas siempre se llega a la necesidad de resolver sistemas de la forma:

$$A \cdot x = b$$

donde A en nuestro caso es la matriz de impedancias más o menos reformada por los condicionamientos de la red, x es el vector corrientes o incógnita y b representa las fuentes de tensión. La utilidad y eficacia del método de Gauss-Seidal queda bien demostrada si lo que queremos es despejar x y continuar adelante con el problema. Pero en la realidad, no ya a nivel industrial, sino a nivel de investigación misma, a veces se requieren diversas soluciones del sistema para distintos valores del vector fuentes b. Es evidente que la técnica de resolver el sistema para cada b es larga y engorrosa, resultando más sencilla invertir la matriz A y multiplicar el resultado por cada uno de los vectores b distintos, obteniendo el correspondiente vector x. Lógicamente este método es tanto más rápido y útil, cuanto mayor sea el número de cambios a que el vector fuentes esté sujeto.

Este es el motivo que nos ha llevado a buscar un método matemático de inversión de matrices, para adaptarlo a nuestro programa y comprobar su utilidad al resolver la ecuación de que hablábamos, ahora ya como el producto $x = A^{-1}b$. En la programación de este método hemos seguido las bases matemáticas publicadas por E. Marchand y C.D. Sparrow (cfr. Bibliografía) sobre la inversión de matrices aprovechando el hecho de que puedan tener un gran número de elementos nulos. Esto es muy importante porque los métodos convencionales no tienen en cuenta la ventaja de que las matrices a invertir esten más o menos vacías y éste es el caso que más frecuentemente ocurre en los problemas que envuelven matrices eléctricas.

Las matrices usadas en este programa son casi todas de tipo complejo por lo que realmente ocupan el doble de la capacidad especificada en las dimensiones del programa, haciendo que el presente programa haya sido confeccionado para un sistema de menos de 50 nodos y ramas.

No hay ningún inconveniente en ampliar este número si la computadora utilizada lo admite.

Podemos entonces dividir el presente trabajo en varias partes:

- I) Planteamiento de la información eléctrica y su traducción a lenguaje matricial.
- II) Resolución de sistemas de ecuaciones (eléctricas)
- III) Subrutina SIMCV
- IV) Conclusiones aplicaciones de la subrutina SIMCV y del programa en general.

I).- PLANTEAMIENTO DEL PROBLEMA.- Matrices topológicas.- Métodos posibles.- Elección del método.- Sistema final.

En la exposición de este apartado seguiremos a FRANK BRANIN (I.B.M.) (cfr. Bibl.)

El problema es el mismo ya se trate de trabajar en AC, transitorios o DC. Las propiedades físicas de una red están completamente determinadas por los elementos que la integran y la manera como están conectados entre sí.

El primer paso consiste en convertir la red en un sistema de líneas respetando la posición de los nodos y dotando a cada rama de una dirección (de circulación de la corriente) totalmente arbitraria.

Para pasar esta red a notación matricial se pueden designar cuatro matrices según el método de resolución que empleemos. Una vez conseguida la notación matricial, no se trata más que de aplicar las leyes de Kirchhof a nuestro problema.

Los métodos de resolución que se pueden emplear son los cuatro siguientes:

- Mallas (matriz C)
- NODOS (matriz A). Este será el que utilizaremos.
- Conjuntos conectados (matriz D)
- Método Mixto (matrices C y D)

A partir de una tabla de conexiones de elementos, han de poder generarse las matrices que se utilizarán en las ecuaciones de funcionamiento del sistema (según sea el método de análisis utilizado se formarán unas matrices u otras)

MATRIZ \bar{A} : A cada fila se asocia una rama del circuito y a cada columna un nudo; todos los elementos de la fila son nulos excepto dos, uno de ellos vale 1 (aquél del que parte la rama) y otro -1 (aquél al que llega).

MATRIZ A^E : En la matriz \bar{A} , cualquier columna es la suma cambiada de signo de las demás; como esta matriz posee información redundante se suprime una columna (la correspondiente al nudo de tierra) y de esta manera se obtiene la matriz llamada A.

ARBOL DE UN GRAFICO, MATRICES A_a y A_e : Un árbol es un conjunto de ramas conectadas entre sí que incluye todos los nodos y que no forman ninguna malla; las ramas restantes son llamadas eslabones. Siempre que se añada un eslabón al árbol se formará una malla.

Si la matriz A se parte en dos, de modo que una de las dos resultantes contenga todas las ramas del árbol y la otra todos los eslabones, se obtienen las matrices A_a y A_e .

MATRIZ B_a : Esta matriz se llama matriz de caminos hacia tierra. Cada fila está asociada a una rama del árbol y cada columna a un nudo (excepto al de tierra). Cada columna indica en qué sentido se encontraría cada rama en un hipotético camino desde el nudo en cuestión hasta tierra. Si la dirección de la rama no coincide con la del camino, el término vale -1; si coincide vale 1, y si la rama no está en el camino su valor es 0. Esta matriz está relacionada con A_a según: $A_a = B_a$.

MATRIZ C, C_a y C_e : Es la matriz de mallas. Una malla se forma añadiendo un eslabón al árbol; por lo tanto a cada eslabón le corresponde una malla. A cada malla se le hace corresponder el sentido de recorrido del eslabón correspondiente. La matriz C tiene tantas filas como ramas del circuito y columnas como eslabones. Si una rama pertenece a la malla con el mismo sentido de recorrido que el que da el eslabón, el elemento correspondiente tiene un valor de 1, si tiene sentido distinto -1 y si no pertenece a la malla 0.

Las ramas del circuito se dividen en dos grupos, primero ramas del árbol y segundo eslabones, de este modo la matriz se subdivide en los C_a y C_e (C_e será la matriz unidad I)

Esta matriz C_a también está relacionada con las anteriores. Para un gráfico lineal se tiene:

$$A^t * C = 0 \qquad C^t * A = 0$$

De la primera de estas expresiones y como $A_a^{-1} = B_a^t$, la expresión para deducir C_a es :

$$C_a = - B_a * A_e^t$$

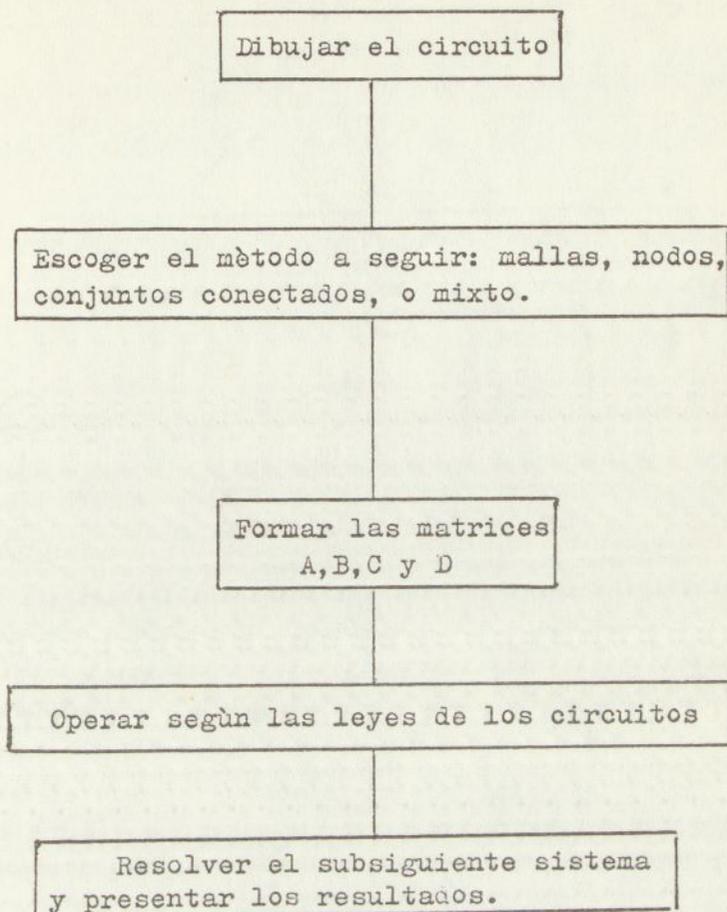
MATRICES D , D_a y D_e : Son las necesarias para el estudio por conjuntos conectados. A la matriz D se le llama matriz básica de conexión. Un conjunto desconectado es un conjunto de ramas, que, siendo suprimidas, dividen el circuito en dos desconectados. Cada conjunto desconectado básico está asociado con una rama del árbol.

La matriz D tiene asociado a cada fila una rama y a cada columna un conjunto desconectado. A cada conjunto se le asigna un sentido positivo de entrar o salir, según que su rama de árbol entre o salga; los elementos de cada columna se forman de acuerdo con el siguiente criterio: se coloca 1 si su sentido de entrada o salida del corte coincide con el término de uno positivo, en -1 si es opuesto y en 0 si no tuviese el corte.

La matriz D se divide en dos, una que agrupa a todas las ramas del árbol y otra a los eslabones; la primera D_a es igual a la matriz unidad - $D_a = I$ y la segunda se llama D_e .

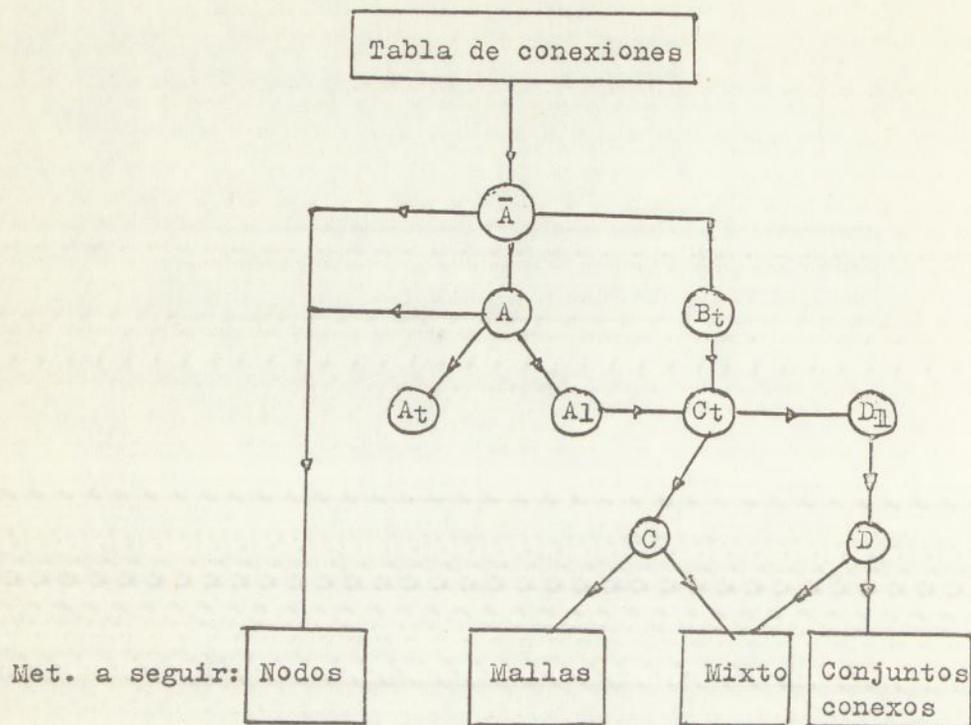
Se verifica $D_e = - C^t a$

Una vez explicadas las matrices topológicas, analicemos los sucesivos pasos a seguir. Esto lo podemos entender mejor mediante el siguiente diagrama.



De todo lo expuesto anteriormente es fácil deducir que para analizar una red eléctrica hay que comenzar "trauciéndosela" a la máquina en forma de una tabla de conexiones totalmente especificadas (ver ejemplo más adelante).

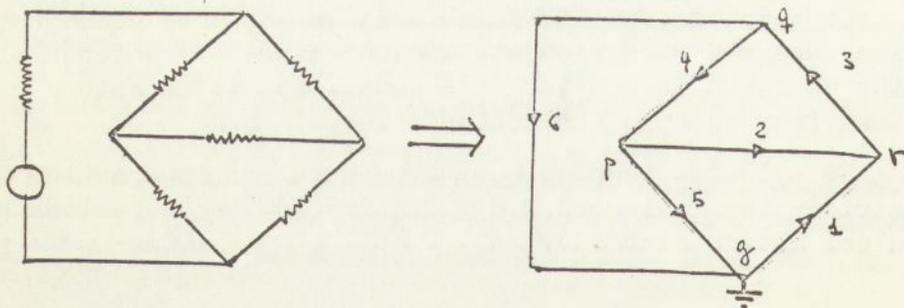
Es evidente también que no habremos de generar todas las matrices topológicas A, B, C y D, sino los que necesitamos a la vista del método escogido.



En el presente programa hemos seguido el Método de NODOS que explicaremos mediante un ejemplo. (Para los otros métodos ver Frank Branin I.B.M.- Computer-aided design.)

Sea el clásico circuito de Wheasestone.

El primer paso es pasar la red dada a otra de tipo lineal en la que no figuren más que las conexiones y los sentidos de la corriente en cada rama, que pueden ser arbitrarias.



De aquí se puede obtener la tabla de conexiones que es la forma de "contarle" el circuito a la computadora.

TABLA DE CONEXIONES

Rama	l	Nodo inicial	g	Nodo final	r	Impedancia asociada (etc..)
"	2	"	p	"	r
"	3	"	r	"	q
"	4	"	q	"	p
"	5	"	p	"	g
"	6	"	q	"	g

Con esta tabla de conexiones recibida como dato; (En realidad viene completada con las impedancias asociadas e inductancias mutuas entre las ramas. Cada rama y su descripción figuran en una tarjeta de datos.) La computadora puede generar la matriz \bar{A} y a partir de ella sin más que suprimir una fila cualquiera, lo que equivale a elegir un nudo cualquiera como referencia o nudo de tierra.

MATRIZ \bar{A}

MATRIZ A

nodos	MATRIZ \bar{A}				nodos	MATRIZ A			
	p	q	r	g		p	q	r	
ramas					ramas				
1	0	0	-1	1	1	0	0	-1	} A_t
2	1	0	-1	0	2	1	0	-1	
3	0	-1	1	0	3	0	-1	1	
4	-1	1	0	0	4	-1	1	0	} A_l
5	1	0	0	-1	5	1	0	0	
6	0	1	0	-1	6	0	1	0	

Una vez explicada la generación de la matriz A pasemos a intentar expresar las leyes fundamentales de la Teoría de Redes en forma asequible a nuestra computadora. Esta va a ser una forma matricial que nos va a hacer entender la utilidad exacta de la matriz A.

Es evidente la necesidad de ponernos de acuerdo en la notación que seguiremos en cuanto a la nomenclatura y sentidos de las fuerzas electromotrices, fuentes de tensión y corriente, etc...

Adoptaremos la más extendida.

Llamaremos:

i_r = Corriente en la rama

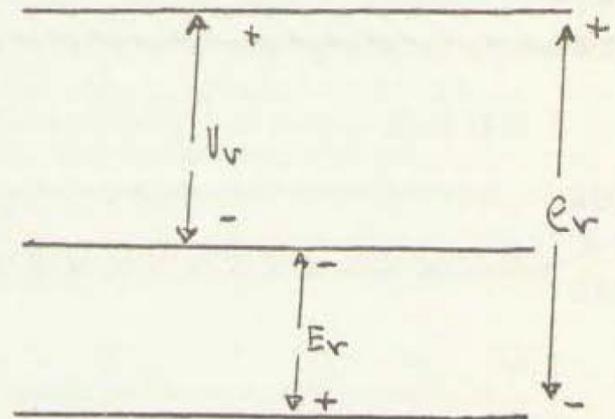
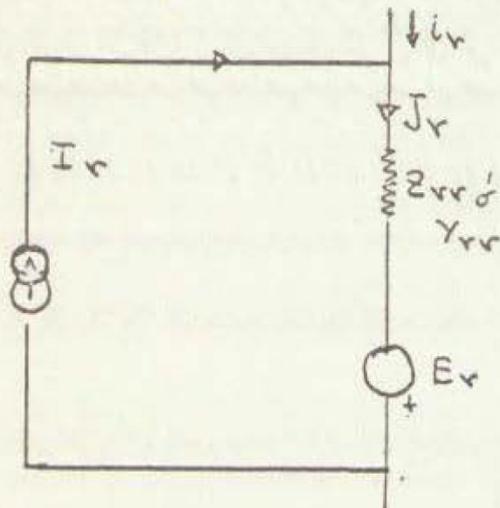
I_r = Corriente asociada (producida por una fuente en paralelo)

J_r = Corriente en la impedancia

E_r = F.é.m. de una fuente de tensión en serie con la rama.

V_r = Caída de tensión en la impedancia

e_r = Caída total en la rama



Es evidente que se verifica

$$J_r = I_r + i_r$$

$$V_r = E_r + e_r$$

De aquí se puede deducir que para una rama la ley de Ohm es

$$V = Z J$$

o

$$J = Y V$$

Para todas las ramas de nuestra red la ley de Ohm adopta una forma matricial.

Es obvio que igual que hablamos de una matriz de impedancias podemos hacer otro tanto con la matriz de admitancias sin más que tener en cuenta los significados de ambas y la relación que entre ellos existe.

Se trata ahora de llegar a una expresión matricial de los Lemas de Kirchhoff.

1º La Suma de corrientes que concurren en un nodo ha de ser 0.

2º La suma de las tensiones en una malla ha de ser 0.

Si nos atenemos al ejemplo antes expuesto vemos que se ha de verificar:

$$i_2 - i_4 + i_5 = 0 \text{ en el nodo p}$$

$$-i_3 + i_4 + i_6 = 0 \text{ en el nodo q}$$

$$-i_1 - i_2 + i_3 = 0 \text{ en el nodo r}$$

Estas relaciones pueden ser combinadas en forma matricial

$$\begin{pmatrix} 0 & 1 & 0 & -1 & 1 & 0 \\ 0 & 0 & -1 & 1 & 0 & 1 \\ -1 & -1 & 1 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} i_1 \\ i_2 \\ i_3 \\ i_4 \\ i_5 \\ i_6 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

donde la matriz de los coeficientes no es más que la traspuesta de la matriz A. Luego esta primera ley de Kirchhoff se puede expresar como:

$$A^t \cdot i = 0$$

De forma análoga la segunda ley (para tensiones) se puede expresar en forma matricial

$$\begin{pmatrix} 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} e_1 \\ e_2 \\ e_3 \\ e_4 \\ e_5 \\ e_6 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

donde la matriz de los coeficientes es la traspuesta de la matriz C. con lo que queda:

$$C^t \times e = 0$$

No hacemos ningún hincapié en esta expresión porque en el método que seguiremos no va a ser usada.

Resolución del problema.-

Para el desarrollo de nuestro problema utilizaremos las variables auxiliares i' y e'

i' = vector de corrientes de mallas

e' = vector de tensiones en los nodos

Estas variables están relacionadas con las i y e ya conocidas mediante las expresiones:

$$i = C \cdot i' \qquad e = A \cdot e'$$

Como dijimos en un principio el método a seguir será el de nodos, que utiliza la matriz A y la 1ª ley de Kirchhoff como principales herramientas para la resolución del problema.

El desarrollo operacional se puede justificar de la forma siguiente:

Aplicando las expresiones $e = A \cdot e'$ y $A^t \cdot i = 0$ a la ley de Ohm (expresada en forma de admitancias por tensiones = J) obtenemos:

Ley de Ohm $J = YV$

Como $J = I + i$ y $V = E + e$ Sustituyendo

$$I + i = Y (E + e) \quad \text{o sea} \quad Ye = I + i - YE \quad \text{de donde}$$

$$Ye = i + (I - YE)$$

Multiplicando en los 2 miembros por A^t

$$A^t \cdot Ye = A^t \cdot i + A^t \cdot (I - YE)$$

Como $A^t \cdot i = 0$ y $e = A \cdot e'$

$$A^t (I - YE) = A^t Y A e'$$

de donde multiplicando en ambos miembros por $(A^t Y A)^{-1}$ obtenemos

$$e' = (A^t Y A)^{-1} A^t (I - YE) \quad \text{y deshaciendo el cambio } e = A \cdot e'$$

$$\text{queda } e = A \left[(A^t Y A)^{-1} A^t (I - YE) \right]$$

Este es el proceso que se trata de programar.

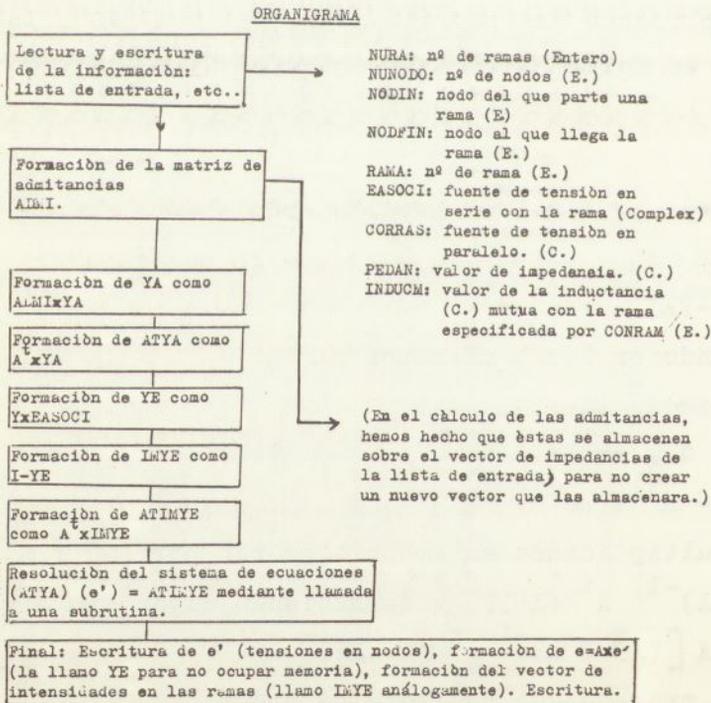
Representación.-

A la vista de los datos que nos describen la red tenemos que ir formando las matrices A (madre), Y (admi), YA (ya), A YA (atya), YE (ye), I-YE (imye), A (I-YE) (atimye), etc... para llegar al planteamiento de un sistema de n ecuaciones (n = número de nodos ÷ nunodo) con n incógnitas.

$$[e'] [Atya] = [Atimye]$$

Con objeto de ahorrar posiciones de memoria y siempre que se pueda hacer sin perjuicio del programa, almacenaremos unas matrices sobre otras. Con este mismo objeto la matriz A no se forma nunca sino que se trabaja directamente con la lista de entrada y con la variable auxiliar no dimensionada BALA, formándose directamente las matrices YA, ATYA, y ATIMYE.

La solución del sistema de ecuaciones es el vector e' (tensiones en nodos) al que multiplicamos por A (siempre sin existencia real) para obtener el vector de tensiones en las ramas e , y a partir de éste obtener mediante un sencillo cálculo el valor de la intensidad que circula por cada una de las ramas.



$$X_i = a_{i,n+1} - \sum_{j=i+1}^n a_{i,j} \cdot X_j$$

Ha de hacerse notar que se ha tenido en cuenta la posibilidad de que un término a_{kk} sea cero en algún momento, (entonces no se podría dividir por él). Cuando ocurra esto, basta simplemente cambiar la fila en cuestión por una de las siguientes, de modo que el nuevo a_{kk} sea no nulo (Al referirnos a un elemento nulo queremos decir un cero o un elemento menor que un cierto error dado.) Si no se encuentra ningún elemento por el que dividir (PIVOTE) no nulo el determinante es nulo y el cálculo ha de pararse. En nuestro caso el hecho de que un a_{kk} sea 0 (cero) quiere decir que una rama tiene impedancia nula.

Esto en la práctica no se va a producir porque aunque una rama no tenga impedancia propia siempre tiene una resistencia formada por el propio conductor. De cualquier forma si se produjera el caso, la subrutina tiene prevista una salida y un mensaje de error advirtiéndolo al mismo tiempo que devuelve el control al programa principal.

En esencia al llamar el programa al método de Gauss (llamada a la subrutina RSL (Matriz de coeficientes, vector independiente, número de nodos-1, error máximo permitido en la solución, vector solución e')), le es transferida la matriz inicial a esta subrutina y opera sobre ella misma para no ocupar nuevas posiciones de memoria. Lo primero que hace es adjuntar a la matriz $n \times n$ una nueva columna: el vector independiente, de modo que la matriz queda de dimensión $n \times n+1$; después se elige un elemento o pivote superior al error (haciendo cambio de filas si el pivote elegido fuera nulo) por el cual se van dividiendo los otros elementos, etc.... Después de operar sobre la matriz aparece el vector resultado como la columna $n+1$ de la susodicha matriz pudiendo ser entregada en esta forma al programa principal.

Con esta subrutina como método de resolución de sistemas, el programa de análisis eléctrico fue realizado y funcionó correctamente y regularmente en distintos casos.

III).- SUBROUTINA SIMCV (Subrutina de Inversión de Matrices Casi Vacías. MARCHAND.- SPARROW.

Para el desarrollo de este punto nos hemos basado en los trabajos de MARCHAND y SPARROW (cfr. Blbl.)

Ya hemos hablado en la introducción sobre la conveniencia de resolver los sistemas del tipo $A \cdot x = b$ como $A^{-1} \cdot b = x$ dado que en los problemas eléctricos la matriz A (y por ello la A^{-1}) permanece constante. De esta manera en el caso frecuente de tener que resolver el sistema muchas veces con un vector b , de fuentes distinto en cada ocasión, la única operación a realizar es el producto de la matriz A^{-1} por cada uno de los vectores b empleados, en vez de volver a resolver el sistema con cada cambio del vector fuentes a que aludimos.

Es este el motivo que nos obliga a programar un método matemático tal que nos permita obtener a partir de la matriz de coeficientes A , su inversa A^{-1} .

En problemas eléctricos las matrices a tratar tienen una gran cantidad de ceros. Utilizando métodos clásicos de inversión, después de un pequeño número de operaciones, han desaparecido todos los ceros y no se saca ningún provecho de esta característica de la matriz.

El método consiste en aprovechar la relación existente entre la inversa de una matriz y la que se obtiene añadiendo a la matriz inicial unas nuevas filas y columnas. La ganancia de tiempo proviene del hecho de que cuando se añade una columna de ceros, puede ahorrarse un número determinado de operaciones.

Para invertir la matriz A_{m+1} supondremos está formada por las A_m , c_a , t_a y el elemento, a , y supondremos también que conocemos la inversa de A_m . Sea esta A_m^{-1} .

$$A_{m+1} = \left(\begin{array}{c|c} A_m & c_a \\ \hline t_a & a \end{array} \right) \quad B_{m+1} = A_{m+1}^{-1} = \left(\begin{array}{c|c} B_m & c_b \\ \hline t_b & b \end{array} \right)$$

$$\text{de } B_{m+1} \cdot A_{m+1} = I_{m+1} \Rightarrow \left(\begin{array}{c|c} B_m & c_b \\ \hline t_b & b \end{array} \right) \left(\begin{array}{c|c} A_m & c_a \\ \hline t_a & a \end{array} \right) = \left(\begin{array}{c|c} I_m & 0 \\ \hline 0 & 1 \end{array} \right).$$

$$\begin{cases} t_a \cdot A_m + b \cdot t_a = 0 \\ t_b \cdot c_a + b \cdot a = 1 \end{cases} \quad \begin{cases} t_b = -b \cdot t_a \cdot A_m^{-1} \\ b \cdot (a - t_a \cdot A_m^{-1} \cdot c_a) = 1 \end{cases}$$

$$\text{de } A_{m+1} \cdot B_{m+1} = I_{m+1} \Rightarrow \left(\begin{array}{c|c} A_m & c_a \\ \hline t_a & a \end{array} \right) \left(\begin{array}{c|c} B_m & c_b \\ \hline t_b & b \end{array} \right) = \left(\begin{array}{c|c} I_m & 0 \\ \hline 0 & 1 \end{array} \right)$$

$$A_m \cdot c_b + c_a \cdot b = 0$$

$$c_b = -A_m^{-1} \cdot c_a \cdot b$$

$$A_m \cdot B_m + c_a \cdot t_b = I_m$$

$$B_m = A_m^{-1} - A_m^{-1} \cdot c_a \cdot t_b$$

Luego el método a seguir será el siguiente:

$$q = t_a \cdot A_m^{-1}$$

$$b = 1/(a - q \cdot c_a)$$

$$t_b = -b \cdot q$$

$$P = A_m^{-1} \cdot c_a$$

$$c_b = -P \cdot b$$

$$B_m = A_m^{-1} - P \cdot t_b$$

Para utilizar este método se parte de la matriz $A_m = a_{11}$, para $m=1$, y se obtiene la inversa de A_2 , y el proceso se continúa hasta invertir totalmente la matriz.

Si la fila a añadir, al aumentar la matriz, es una fila de ceros $t_a=0$, se verifica:

$$\begin{aligned} t_a=0 & \quad q=0 \\ & \quad b=1/a \\ & \quad t_b=0 \\ & \quad P = A_m^{-1} \cdot c_a \\ & \quad c_b = -P \cdot b \\ & \quad B_m = A_m^{-1} \end{aligned}$$

es decir podemos omitir el cálculo de q, t_b y B_m .

Si el término a añadir es la columna c_a , se simplifica igualmente el resultado

$$\begin{aligned}
 c_a &= 0 & q &= t_a \cdot A_m^{-1} \\
 & & b &= 1/a \\
 & & t_b &= -b \cdot q \\
 & & P &= 0 \\
 & & c_b &= 0 \\
 & & B_m &= A_m^{-1}
 \end{aligned}$$

en este caso son P, c_b y B_m los elementos no necesitados de cálculo.

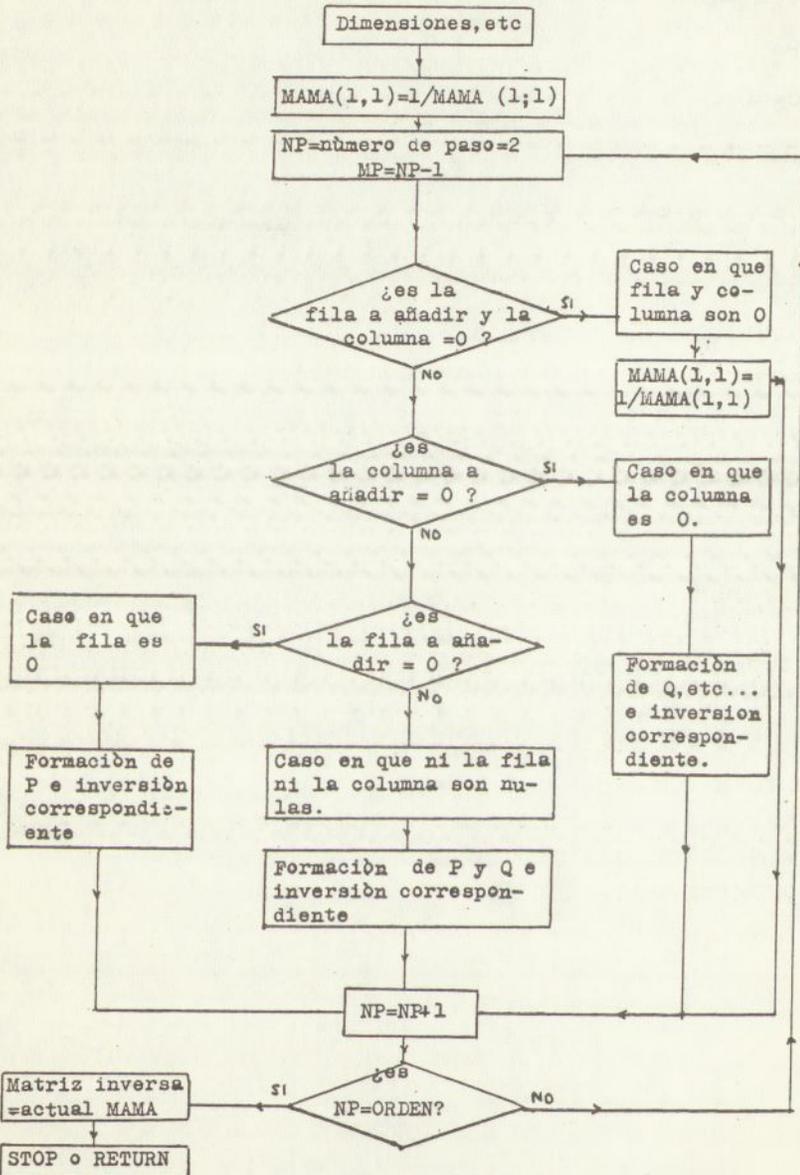
En realidad para el programa no se trabaja con los vectores a, B_m, t_b, c_b , etc.... que supondrían una pérdida de memoria apreciable, sino que lo haremos con la matriz a invertir o matriz madre (MAMA) y dos vectores auxiliares P y Q de dimensiones m (mama es $m \times m$), almacenándose las matrices intermedias (de órdenes 1, 2, 3 m) sobre la primitiva MAMA.

El hecho de que la matriz A sea vacía, no implica necesariamente que su inversa A^{-1} la sea también, Es importante recordar esto porque se nos podría objetar el haber usado palabras de memoria inútilmente al almacenar A , siendo como es casi vacía y simétrica, y pudiendo almacenar únicamente el valor numérico de los elementos no nulos y los dos subíndices que indican su posición en la matriz.

A^{-1} La respuesta a esta cuestión está ya indicada al decir que A^{-1} no tiene que ser necesariamente vacía, necesitando por tanto los $m \times m$ palabras de memoria que usó su matriz original A y que a al efecto están programados.

ORGANIGRAMA

Subrutina SIMCV(MAMA(matriz madre o matriz a invertir), ORDEN
(orden de la matriz a invertir))



Segùn vemos en el organigrama, en los tres posibles casos que considera la subrutina SIMCV (ni la fila ni la columna a añadir sean nulas, la columna sea nula, o la fila sea nula) se forman las matrices auxiliares P o Q o P y Q realizándose a continuación una serie de operaciones, casi todas productos matriciales de distinta conformación (matriz cuadrada, vector, etc...)

A este efecto se han programado tres subrutinas que los realizan y que son llamadas durante la ejecución de la SIMCV

Estas son:

Sub. FAM (matriz entera, orden de la matriz cuadrada, resultado)
 Multiplica la matriz cuadrada constituida por las n primeras filas y columnas de la matriz dada por la fila $n+1$.

Sub. AMC (matriz entera, orden de la matriz cuadrada, resultado)
 Hace lo mismo que la anterior sólo que el producto es ahora por la columna $n+1$.

Sub. PQC (fila $n+1$, orden n, elemento resultado, matriz entera)
 Multiplica la fila dada por la columna $n+1$ de la matriz dada, dando como resultado un elemento.

Estas tres subrutinas no tienen ninguna dificultad al no ser más que sencillos programas de productos matriciales entre matrices complejas. El único rasgo notable en ellas es que en los productos a realizar entre los distintos elementos, se pregunta antes que nada si uno de los factores a multiplicar es nulo (cosa muy frecuente) en cuyo caso se prescinde de la operación en que dicho elemento interviene ahorrándose con ello un tiempo de ejecución considerable. Además, esta pregunta, mediante un cambio de índices en las dos sentencias DO que se utilizan en la multiplicación; pasa por uno sólo de ellos (de los DO) con el consiguiente ahorro de tiempo.

IV) CONCLUSIONES Y APLICACIONES DE LA SUBROUTINA SIMCV Y DEL PROGRAMA EN GENERAL.-

La subrutina SIMCV así expuesta fue probada en programa aparte, en la computadora, llegándose a resultados satisfactorios, tanto en el campo real como en el complejo sin más que cambiar las sentencias iniciales (dimensiones, formatos, datos, etc, ...).

Para establecer la utilidad de la nueva subrutina se procedió a compararla con una subrutina de inversión de matrices empleada normalmente en problemas eléctricos basada también en el método de Gauss. Se trata de la subrutina SINVMA (matriz a invertir, orden de la misma, error permitido). De esta comparación se obtuvieron las siguientes conclusiones:

a) En matrices de pequeño orden (menor que 10) no es posible apreciar diferencias en cuanto al tiempo consumido que viene a ser del orden de un sesentaavo de segundo.

b) En matrices de mayor orden y cuando sean casi todos los términos nulos excepto los de la diagonal principal tarda menos la nueva SIMCV que la convencional SINVMA. En el ejemplo adjunto siendo la matriz a invertir de orden 12 y distinta de cero sólo en la diagonal principal, la subrutina SIMCV tardó sólo 2 sesentaavos de segundo mientras que la SINVMA hizo el mismo problema en 34 sesentaavos de segundo.

Parece pues evidente que la subrutina construida SIMV es útil para invertir matrices, y tanto más mientras éstas estén más vacías, caso que es el nuestro.

El paso siguiente fue acoplar esta subrutina al programa principal de análisis de redes (quitando claro está la RSL), para que llegado el caso la matriz de coeficientes del sistema fuera invertida y el resultado multiplicado por el vector independiente para obtener la solución e' .

Realizado esto y funcionando el programa a la perfección (aunque naturalmente en la resolución del único sistema que entraña nuestro ejemplo, tarda más la subrutina de inversión SIMCV que la de resolución de sistemas RSL) se procedió a constatar la utilidad del método en la resolución de problemas eléctricos.

Para ponerla de manifiesto y esto es quizá la parte más importante del presente trabajo, se probó el programa principal de análisis mediante la subrutina SINVMA con idéntica estructura y datos que la última prueba de la SIMCV. De la comparación

(ambas subrutinas para resolver el sistema, la invierten y multiplican por el vector independiente) resultò mas ràpida y menos laboriosa en cuanto a número de operaciones (y por tanto con menor error) la nueva SIMCV, que la convencional SINVMA.

Concretamente en los ejemplos adjuntos (Programas D - 1 y D - 2) para una misma red, los resultados obtenidos son:

	<u>Programa con SIMCV</u>	<u>Programa con SINVMA</u>
Nº de operaciones	40328	110544
Tiempo de inversión (en 1/60 de segundo)	71	305
Tiempo de resolución del sistema (1/60 de segundo)	74	380

Con ello queda demostrada la ventaja que aporta esta nueva subrutina en cuanto a prevision y economía del tiempo se refiere. en la inversión de matrices eléctricas.

Es preciso resaltar el papel que juega el método que hemos desarrollado en el amplio mosaico que forman los métodos usuales para invertir matrices. En general éstos pueden ser de dos clases:

Métodos directos.-

Son los más rápidos. En ellos el número de operaciones al realizar es proporcional a n^3 . Tienen grandes pegas; acumula errores con la consiguiente desviación de la matriz inversa, y si el orden de la matriz a invertir es demasiado grande, el método es absolutamente inexacto.

Métodos iterativos.-

Son mucho más lentos que los directos y en ellos el número de operaciones a realizar es mucho mayor (del orden de $2n^2+n$ por iteración). Con ellos pueden ocurrir dos cosas:

- Que no converjan, separándose a cada iteración más de la solución.
- Caso de que converjan lo normal es que sea lentamente.

Nuestro método podría encuadrarse entre los que hemos llamado directos.

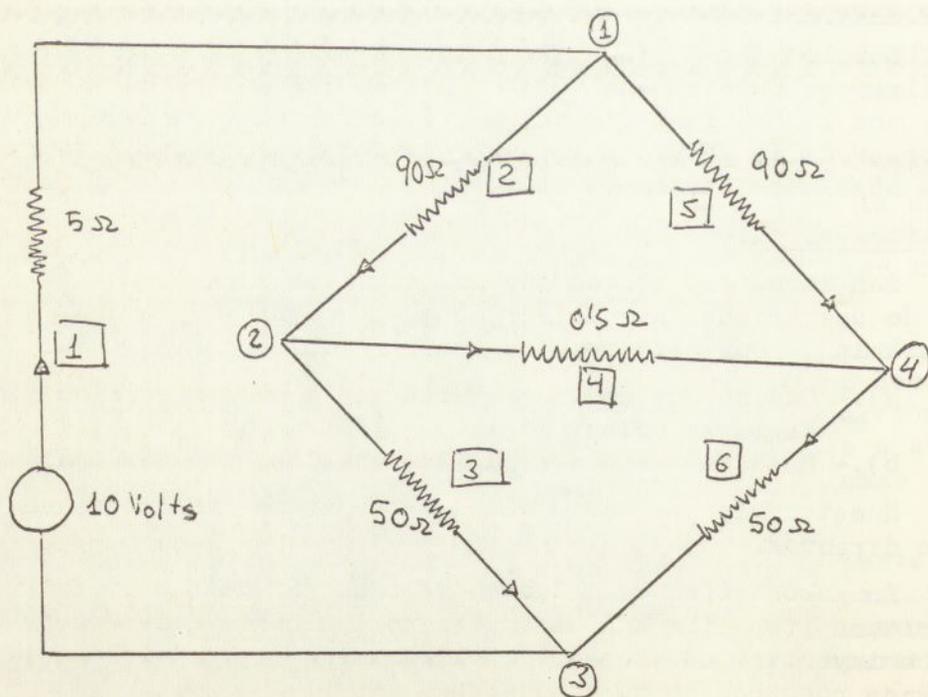
En ambos tipos de métodos (ya sean directos o iterativos) a la primera iteración han desaparecido los ceros que tuviera la matriz a invertir, no sacando de ello ningún provecho, cosa que sí hacemos nosotros en nuestro método.

En cuanto a la novedad que puede representar el programa general de análisis eléctrico, es necesario destacar que se ha planteado de la forma más general posible. Nos referimos al hecho de que la red que nuestro programa admite puede tener elementos cualesquiera, bien sean inductivos, resistivos o capacitivos, reales o complejos e incluso como hemos dicho en otros apartados, está prevista la posible existencia de inductancias mutuas entre las distintas ramas de la red.

PROGRAMA A-1.-

Es un programa de análisis de redes (con Gauss-Seidel) aplicado a la simulación del puente de Wheastone que a continuación se detalla. Como se apreciará los resultados son exactos y reflejan el comportamiento de un puente de Wheassestone convencional.

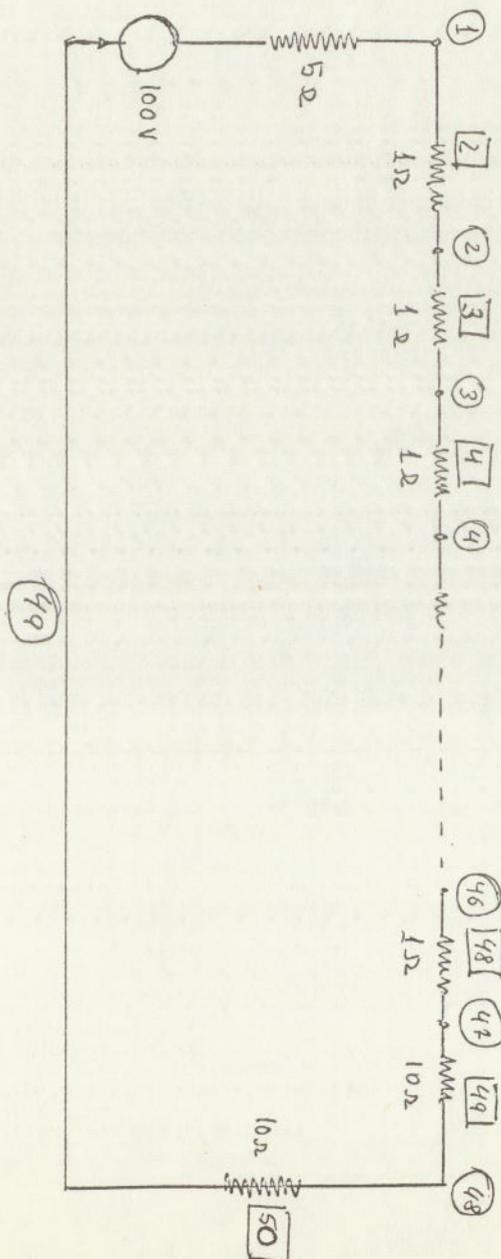
Aunque el programa está preparado para trabajar con cualquier tipo de impedancia (incluso inductancias mutuas), aquí se presenta únicamente con resistencias puras (valores reales), para facilitar los cálculos de comprobación.



PROGRAMA B-1 y B-2.-

Son dos programas cargados con la misma serie de datos que simulan el circuito de la figura. El B-1 lo resuelve mediante la subrutina RSL y el B-2 mediante la SIMCV.

Hemos vuelto a tomar valores reales en las impedancias para facilitar la comprobación del programa.



PROGRAMA C-1 y C-2.-

Se trata de invertir una matriz casi vacía 12×12 . El C-1 lo hace mediante una subrutina convencional (SINVMA) tardando más tiempo que el C-2 que utiliza la SIMCV.

	SIMCV		SINVMA
Tiempo de inversión	2	34

De nuevo por simplificar la comprobación, hemos utilizado únicamente valores reales y de cálculo sencillo.

PROGRAMAS D-1 y D-2.-

Analizan da misma red que los programas B-1 y B-2 pero resuelven el sistema por inversión de la matriz de coeficientes mediante SIMCV (Emilio-SIMCV. Análisis redes D-1) y SINVMA (Emilio Galo- Análisis redes D-2) respectivamente.

Mediante contadores de operaciones y llamadas al reloj permiten constatar la utilidad que la nueva subrutina SIMCV tiene.

APENDICE I

```

SUBROUTINE SIMCV(MAMA,ORDEN)
C
C * * * * *
DIMENSION MAMA(50,50),P(50),Q(50)
INTEGER ORDEN
COMPLEX MAMA,P,Q
C
C * * * * *
C PASO GENERAL EN QUE SE VA TOMANDO UNA FILA Y UNA COLUMNA INCREMENTA
C DAS EN UN FLEMENTO PARA APLICARLES UNA DE LAS PARTES DEL METODD
C NP=NUMERO DEL PASO
C
C * * * * *
MAMA(1,1)=1./MAMA(1,1)
DO 20 NP=2,ORDEN
MP=NP-1
C
C PREGUNTAS QUE SIRVEN PARA DIRECCIONAR EL SISTEMA
C
C * * * * *
DO 81 I=1,MP
IF(CABS(MAMA(I,NP)).EQ.0.0.AND.CABS(MAMA(NP,I)).EQ.0.0) GO TO 81
GO TO 82
81 CONTINUE
GO TO 10
82 DO 3 I=1,MP
IF(CABS(MAMA(I,NP)).NE.0.) GO TO 4
3 CONTINUE
GO TO 11
4 DO 5 I=1,MP
IF(CABS(MAMA(NP,I)).NE.0.) GO TO 12
5 CONTINUE
GO TO 13
C
C * * * * *
C PARTES DEL METODO A LAS QUE VIENE EL SISTEMA SELECCIONADO
C
C * * * * *
CC 1- CASO EN QUE LA FILA Y LA COLUMNA SEAN CERO
C * * * * *
C 10 MAMA(NP,NP)=1./MAMA(NP,NP)
GO TO 99
C
C * * * * *
C 2-CASO EN QUE LA COLUMNA SEA CERO
C
C * * * * *
C 11 CALL FAM(MAMA,MP,Q)
MAMA(NP,NP)=1./MAMA(NP,NP)
DO 30 I=1,MP
MAMA(NP,I)=-MAMA(NP,NP)*Q(I)
30 MAMA(I,NP)=0.
GO TO 99
C
C * * * * *
C 3-CASO GENERAL.LA COLUMNA Y LA FILA NO SON CERO
C
C * * * * *
C 12 CALL FAM(MAMA,MP,Q)
CALL AMC(MAMA,MP,P)
CALL POC(Q,MP,ELEAUX,MAMA)
MAMA(NP,NP)=1./((MAMA(NP,NP)-ELEAUX)
DO 31 I=1,MP
MAMA(I,NP)=-MAMA(NP,NP)*P(I)

```

.../...

```

31 MAMA(NP,I)=-MAMA(NP,MP)*Q(I)
   DO 32 I=1,MP
   DO 32 J=1,MP
32 MAMA(I,J)=MAMA(I,J)-P(I)*MAMA(NP,J)
   GO TO 99

```

```

C
C * * * * *
C 4-CASO EN QUE LA FILA SEA CERO

```

```

CC
C * * * * *
C 13 CALL AMC(MAMA,MP,P)
   MAMA(NP,MP)=1./MAMA(NP,MP)
   DO 33 I=1,MP
   MAMA(I,MP)=-MAMA(NP,MP)*P(I)
33 MAMA(NP,I)=0.
   GO TO 99

```

```

C
C * * * * *

```

```

C ESCRITURA DE LOS PASOS SUCESIVOS Y DE LA MATRIZ INVERSA

```

```

C
99 IF(CABS(MAMA(NP,MP)).EQ.0.) GO TO 85
   GO TO 20
85 WRITE(6,86)
86 FORMAT(/1H1/,5X,24HNO EXISTE MATRIZ INVERSA,/,5X,22HSE SUPRIME EL
   * PROGRAMA,2X)
   GO TO 89
20 CONTINUE
89 RETURN
   END

```

APENDICE II

```

C
C ESTA SUBROUTINA MULTIPLICA UNA FILA POR UNA MATRIZ CUADRADA
C

```

```

SUBROUTINE FAM(A,M,RES)
DIMENSION A(50,50),RES(50)
COMPLEX A,RES
DO 41 I=1,M
41 RES(I)=0.
   DO 51 J=1,M
   IF(CABS(A(M+1,J)).EQ.0.) GO TO 51
   DO 21 I=1,M
   RES(I)=RES(I)+A(M+1,J)*A(J,I)
21 CONTINUE
51 CONTINUE
RETURN
END

```

APENDICE III

```

C
C ESTA SUBROUTINA MULTIPLICA UNA MATRIZ POR UNA COLUMNA
C

```

```

SUBROUTINE AMC(A,M,RES)
DIMENSION A(50,50),RES(50)
COMPLEX A,RES
DO 42 I=1,M
42 RES(I)=0.
   DO 52 J=1,M
   IF(CABS(A(J,M+1)).EQ.0.) GO TO 52
   DO 22 I=1,M
   RES(I)=RES(I)+A(I,J)*A(J,M+1)
22 CONTINUE
52 CONTINUE
RETURN
END

```

APENDICE IVC
C
C
C

ESTA SUBROUTINA MULTIPLICA FILA POR COLUMNA Y DA UN NUMERO X

```

SUBROUTINE PQC(FILA,M,X,A)
DIMENSION FILA(50),A(50,50)
COMPLEX FILA,X,A
X=0.
DO 23 I=1,M
IF(CABS(FILA(I)).EQ.0.0.OR.CABS(A(I,M+1)).EQ.0.0) GO TO 23
X=X+FILA(I)*A(I,M+1)
23 CONTINUE
RETURN
END

```

RELACION DE APENDICES

Apèndice I .- Subrutina SIMCV, preparada para matrices complejas de 50x50 elementos.

Apèndice II .- Subrutina FAM, llamada por la SIMCV. Multiplica una matriz fila por una matriz cuadrada.

Apèndice III .- Subrutina AMC, llamada por la SIMCV. Multiplica una matriz cuadrada por una matriz columna.

Apèndice IV .- Subrutina PQC, llamada por la SIMCV. Multiplica una matriz fila por una matriz columna.

Bibliografía

- FRANK BRANIN. (I.B.M.) Computer-Aided design.
- DURAND, E. Solutions numériques des équations algébriques. (Tomo II. Systèmes de plusieurs équations, valeurs propres des matrices). Masson et Cie. 1961.
- BECKETT AND HURT. Numerical calculations and Algorithms. Mc.Graw Hill. 1967.
- MARCHAND, E. + SPARROW, C.D. Fast method for obtaining the inverse of sparse matrices. IEEE vol. 115, n° 9, sept. 1968.
- SCHAUM'S OUTLINE SERIES. Electric Circuits.