

Distr.
RESTRINGIDA

YC/MEX/R.198
23 de enero de 1990

ORIGINAL: ESPAÑOL

CATALOGADO

C E P A L

Comisión Económica para América Latina y el Caribe

BIBLIOTECA NACIONES UNIDAS MEXICO

PROGRAMA DE FLUJOS DE POTENCIA

Volumen II. Manual de programa y código fuente

Documento elaborado por el consultor Ricardo Mota Palomino. Las opiniones en él expresadas son de la exclusiva responsabilidad del autor y pueden no coincidir con las de la Organización.

90-1-25

INDICE

	<u>Página</u>
Presentación	v
I. Manual de Programa	1
Introducción	3
Programa principal	4
Lectura de datos	5
Formación de modelos	9
Formación de la matriz de admitancias NODAL	10
Solución del estudio	10
Subrutina AJUSQ	10
Subrutina FORM	17
Subrutina FACTI	21
Referencias	24
Anexo: Biblioteca para la solución de estudios de flujos convencionales por el método desacoplado rápido	25
II. Código Fuente	89

PRESENTACION

Con el propósito de atender, en la medida de lo posible, la solicitud de las empresas eléctricas del Istmo Centroamericano de promover el desarrollo de metodologías para el análisis de redes eléctricas en las áreas de operación y de planificación, la Comisión Económica para América Latina y el Caribe (CEPAL) contrató a un consultor con el fin de que desarrollara, en una etapa inicial, programas digitales para efectuar estudios de flujos de potencia y cálculo de fallas.

Para el desarrollo de estos programas, se plantearon los siguientes requisitos: a) que permitieran analizar las redes eléctricas del Istmo Centroamericano, incluyendo aspectos dimensionales y características técnicas; b) que se pudieran utilizar en microcomputadoras tipo XT o AT, equipo del que disponen todas las empresas eléctricas de la subregión, y c) que su entrega a los organismos nacionales de electrificación de América Central también significara, si bien modesta, una auténtica transferencia tecnológica. Los primeros dos requerimientos se lograron mediante la selección de la metodología y una programación adecuadas. Para satisfacer el tercero, se optó por entregar a las empresas eléctricas del Istmo, además de los manuales de usuario y técnico, el manual de programa y código fuente de los programas y subrutinas.

Se considera que el uso y estudio de estas herramientas digitales permitirá a los técnicos de las empresas eléctricas: a) efectuar análisis de redes en régimen permanente para la solución de casos y estudios puntuales, evitando costosas erogaciones de divisas para consultorías; b) depurar la información y representar en forma cada vez más realista los sistemas eléctricos nacionales e interconectados, y c) capacitarse en las bases técnicas y simulación digital de redes eléctricas, así como contar con una preparación más adecuada para aprovechar mejor los resultados del PARSEICA.

La documentación de los dos programas está organizada en dos volúmenes para cada uno; el primer volumen contiene los manuales del usuario y técnico, y el segundo, el manual de programa y el código fuente. Ambos programas se desarrollaron en lenguaje FORTRAN. El manual del usuario explica, mediante un ejemplo, la utilización del modelo, mientras que el técnico presenta las bases conceptuales y expresiones algorítmicas del método empleado para la solución del problema. El manual de programa está orientado al personal

responsable del mantenimiento de los programas y explica la organización del programa principal y las subrutinas. Se incluye el diagrama de flujo para cada caso. Los requerimientos mínimos de computación para ambos modelos consisten en una microcomputadora (PC) compatible con IBM XT o AT con 640 KB de memoria RAM.

El programa de flujos de potencia (CEPFL1) se basa en el método de Newton desacoplado rápido. El procedimiento para utilizarlo se explica en el manual del usuario (volumen I), mediante un sistema eléctrico de prueba con 14 nodos. En el manual técnico (volumen I) se desarrollan las ecuaciones que sirven para formular el estudio de flujos, se expone el desacoplamiento de las ecuaciones y se deducen las utilizadas en el algoritmo computacional del método. En dicho manual también se abordan brevemente algunos aspectos de la implantación del método para explotar la estructura de los modelos desacoplados. El manual de programa, contenido en el volumen II, describe de manera detallada el programa digital desarrollado. Las rutinas que forman este programa pueden utilizarse en otros modelos, por lo que se incorporaron como un apéndice de dicho manual. En el volumen II también se incluye el código fuente del programa.

Sin duda que estas herramientas son susceptibles de mejorarse y ampliarse. Ya se han identificado algunas mejoras. Sin embargo, como no se dispone al presente de recursos para incorporarlas al programa, se optó por entregar estas versiones para que las empresas empiecen a utilizarlas de inmediato. Posteriormente, y de contar con financiamiento, se incorporarán mejoras y ampliaciones en nuevas versiones. Se agradecerán los comentarios y sugerencias con tal fin.

I. MANUAL DEL PROGRAMA

INTRODUCCION

A continuación se describe el programa CEPFL1. Este programa está escrito en lenguaje FORTRAN y en él, el proceso de lectura de archivo de datos, formación y solución de modelos y generación de reportes se realizan en forma secuencial. En esta versión solo se interactúa con el programa para darle información del archivo de datos de entrada, parámetros propios del estudio de flujos y solicitar reportes.

El detalle del archivo de datos requerido para resolver un estudio de flujos de potencia se describe en el manual de usuarios de CEPFL1 [1]. Las subrutinas utilizadas en la formación de modelos desacoplados de potencia activa y reactiva del método desacoplado rápido se describen en detalle en la referencia [2] y que se anexa como apéndice a este documento.

Programa Principal

Todo el proceso involucrado en la solución de un estudio de flujos es controlado en CEPFL1 mediante un programa principal. Las actividades más importantes de este programa se muestran esquemáticamente en la Figura 1.

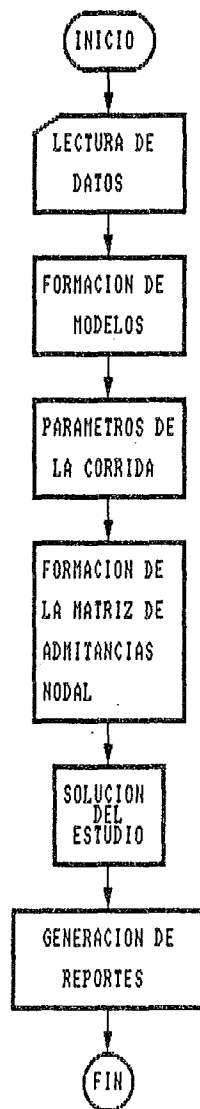


FIGURA 1. ESTRUCTURA DEL PROGRAMA PRINCIPAL DE CEPFL1.

A continuación se describen brevemente las actividades y subrutinas componentes de cada bloque mostrado en la Figura 1.

1. Lectura de datos

La entrada de datos al programa se hace desde un archivo previamente constituido y descrito en detalle en [1].

La lectura se hace mediante la subrutina siguiente:

1.1 Subrutina LECTUR.

La subrutina de lectura es llamada mediante la instrucción

```
CALL LECTUR (ARCH, DNOMB, LETR1, LETR2, NOMB1, NOMB2, IFIN, NE, NR,
2 NCT, NST, NU, NAU, , NSH, R, X, B, PC, QC, T1, PG, VG, QMA, QMI, VLS, QG,
3 NNOD, NLIN, XNTUG, NTR, NSHU)
```

Los argumentos de la subrutina son:

Variable	Entrada ó Salida	Tipo	Descripción
ARCH	E/S	Character * 20	Nombre del archivo de datos de entrada al programa. Se introduce interactivamente.
LETR1	S	Character * 70	Primera línea de descripción del archivo de datos.

Variable	Entrada ó Salida	Tipo	Descripción
LETR2	S	Character * 70	Segunda línea de descripción del archivo de datos.
DNOMB(NNOD)	S	Character * 8	Lista de nombres de nodos de la red a analizar
NOMB1	S	Character * 8	Variable auxiliar para procesos de nombres de nodos.
NOMB2	S	Character * 8	Variable auxiliar para proceso de nodos
IFIN	E	Character * 8	Variable auxiliar para detectar fin de grupo de datos en archivo de entrada. Se inicializa con "0".
NE(NNOD)	S	Entera	Lista de nodos de envío de elementos de transmisión.
NR(NNOD)	S	Entera	Lista de nodos de recepción de elementos de transmisión
NCT(NTR)	S	Entera	Lista de nodos con cambiador de derivaciones en transformadores.

Variable	Entrada ó Salida	Tipo	Descripción
NST(NTR)	S	Entera	Lista de nodos sin cambiador de derivaciones en transformador.
NU(NTUG)	S	Entera	Lista de nodos de voltaje controlado.
NAU(NTUG)	S	Entera	Vector auxiliar asociado a nodos de voltaje controlado.
NSH(NSHU)	S	Entera	Lista de nodos con compensación en derivación.
R(NLIN)	S	Real	Resistencia de elementos de transmisión.
X(NLIN)	S	Real	Reactancia serie de elementos de transmisión.
B(NLIN)	S	Real	Semi-suceptancia en derivación de elementos de transmisión.
PC(NNOD)	S	Real	Lista de potencia real de carga en nodos.
QC(NNOD)	S	Real	Lista de potencia reactiva de carga en nodos.
TI(NTR)	S	Real	Lista de relaciones de transformación en pu en banco.

Variable	Entrada ó Salida	Tipo	Descripción
PG(NTUG)	S	Real	Lista de potencias reales generadas en nodos de voltaje controlado.
VG(NTUG)	S	Real	Lista de voltajes en nodos de voltaje controlado.
QMA(NTUG)	S	Real	Lista de límites superiores en potencia reactiva generada en nodos de voltaje controlado.
QMI(NTUG)	S	Real	Lista de límites inferiores en potencia reactiva generada en nodos de voltaje controlado.
VLS(NSHU)	S	Real	Lista de susceptancias en derivación en nodos con compensación.
QG(NTUG)	S	Real	Lista de potencias reactivas generadas en nodos de voltaje controlado, se inicializa en ceros.
NNOD	S	Entera	Número de nodos en la red estudiada.
NLIN	S	Entera	Número de elementos de transmisión en la red estudiada.

Variable	Entrada ó Salida	Tipo	Descripción
NTUG	S	Entera	Número de nodos de voltaje controlado en la red estudiada.
NTR	S	Entera	Número de bancos de transformación en la red estudiada.
NSHU	S	Entera	Número de elementos en derivación en la red estudiada.

En el archivo de datos, cada elemento generador, de transmisión ó compensador se identifica por el nodo ó nodos entre los que se conecta. La subrutina LECTUR utiliza una subrutina auxiliar llamada VERIF que se utiliza para verificar que los nodos en que se conecta algún elemento se encuentre en la lista de nombres del archivo de datos. En caso contrario el elemento correspondiente es omitido en el resto del estudio.

FORMACIÓN DE MODELOS

A continuación CEPFL1 forma y factoriza los modelos de potencia real y reactiva necesarios para aplicar el método desacoplado rápido. En la referencia [1] se describen en detalle las subrutinas utilizadas en la formación de modelos. Estas subrutinas son GRAPH, PARTS, ACTIV, PREPA, ORDER, FORMA Y FACTA.

FORMACIÓN DE LA MATRIZ DE ADMITANCIAS NODAL

Para facilitar el cálculo de inyecciones nodales requeridos en el algoritmo del método, se construye, antes de iniciar el proceso iterativo, la matriz de admitancias nodal de la red analizada. Para esto se ejecuta la subrutina YBUSC descrita en la referencia [1].

SOLUCIÓN DEL ESTUDIO

Las subrutinas que se ejecutan al implantar el método desacoplado rápido se describen con detalle en la referencia [1]. La única adición en la implementación de CEPFL1 es la de la subrutina AJUSQ que permite verificar el ajuste dentro del límites de generación de potencia reactiva en nodos de voltaje controlado, así como de las subrutinas FORM Y FACTI que son llamadas por AJUSQ para reconstruir y triangularizar el modelo de potencia reactiva al cambiar nodos de voltaje controlado a nodos de carga cuando aquellos llegan a un límite en la generación de potencia reactiva. A continuación se describen estas subrutinas.

Subrutina AJUSQ

Esta subrutina verifica la generación de potencia reactiva en nodos de voltaje controlado si la máxima desviación de potencia reactiva en un nodo de carga es menor a la tolerancia definida para empezar a verificar este ajuste. Si en algún nodo de voltaje controlado se exceden los límites de potencia reactiva especificados, el nodo se cambia a nodo de carga y se reconstruye el modelo de potencia reactiva y se retriangulariza.

La subrutina es ejecutada mediante la instrucción:

CALL AJUSQ (NNOD, NACT, NTUG, NISLA, NISA, TINIA, L9, D2, ISCOM, NU,
2 M, N, IFIL, NB1, NB1C, NORD, NORD1, B1, B2, QMA, QMI, Q2, E, E3, F3,
3 YR, YI, IE2, IE3, IE4, DNOMB, QC, NL2, BASE)

La lista de argumentos de AJUSQ se describen a continuación

Variable	Entrada o Salida	Tipo	Descripción
NNOD	E	Entera	Número total de nodos en la red.
NACT	E	Entera	Número de nodos activos en la red.
NTUG	E	Entera	Número de nodos de voltaje controlado.
NISLA	E	Entera	Número de islas, eléctricas en la red
NISA	E	Entera	Número de islas activas en la red.
TINIA	E	Entera	Tolerancia a partir de la cual se inicia verificación de límites de potencia reactiva generada (pu).
L9	E/S	Entera	Número de nodo con máxima desviación de potencia reactiva.

Variable	Entrada o Salida	Tipo	Descripción
D2	E/S	Entera	Máxima desviación de potencia reactiva en algún nodo.
ISCOM(NISLA)	E	Entera	Lista de nodos compensadores, uno por isla eléctrica.
NU(NTUG)	E	Entera	Lista de nodos de voltaje controlado.
M(NNOD)	E	Entera	Lista de apuntadores de primera posición por nodo en Ybus.
N(NL2)	E	Entera	Vector de columnas de Ybus por nodo. N(M(I)) es el número de elementos en la fila I.
IFIL	E	Entera	Extensión máxima de arreglos triangularizados.
NB1(NACT)	E	Entera	Vector de primeras posiciones por nodo activo en modelos triangularizados.

Variable	Entrada o Salida	Tipo	Descripción
NBC1(IFIL)	E	Entera	Lista de columnas de arreglos triangularizados. NB1C(NB1(I)) dá el número de elementos no cero en la fila I.
NORD(NNOD)	E	Entera	Nuevo orden de nodos.
NORD1(NNOD)	E	Entera	Lista de orden de eliminación de nodos
B1(IFIL)	E	Real	Modelo de potencia real triangularizado. B1(NB1(I)) dá el elemento diagonal de la fila I.
B2(IFIL)	E	Real	Modelo de potencia reactiva triangularizado.
QMA(NTUG)	E	Real	Límites superiores de generación de potencia reactiva en nodos de voltaje controlado.
QMI(NTUG)	E	Real	Límites inferiores de generación de potencia reactiva en nodos de voltaje controlado.

Variable	Entrada o Salida	Tipo	Descripción
Q2(NNOD)	E/S	Real	Vector auxiliar se almacena Q neta en el nodo.
E(NNOD)	E/S	Real	Vector auxiliar, se almacena en él $\Delta Q/V$.
E3(NNOD)	E	Real	Magnitud de voltajes de nodales(pu).
F3(NNOD)	E	Real	Angulos de voltajes nodales (rad).
YR(NL2)	E	Real	Parte real de Ybus. YR (M(I)) es el valor diagonal de la fila I.
YI(NL2)	E	Real	Parte imaginaria de Ybus.
IE2	S	Entera	Variable auxiliar.
IE3	S	Entera	Variable auxiliar.
IE4	S	Entera	Variable auxiliar.
DNOMB(NNOD)	E	Character * 8	Lista de nombres de nodos.
Q(NNOD)	E	Real	Potencia reactiva de carga en nodos.

Variable	Entrada o Salida	Tipo	Descripción
NL2	E	Entera	$2 * NLIN + NNOD$
BASE	E	Real	Base de potencia usada.

La figura 2 muestra un diagrama de bloques de la subrutina (AJUSQ)

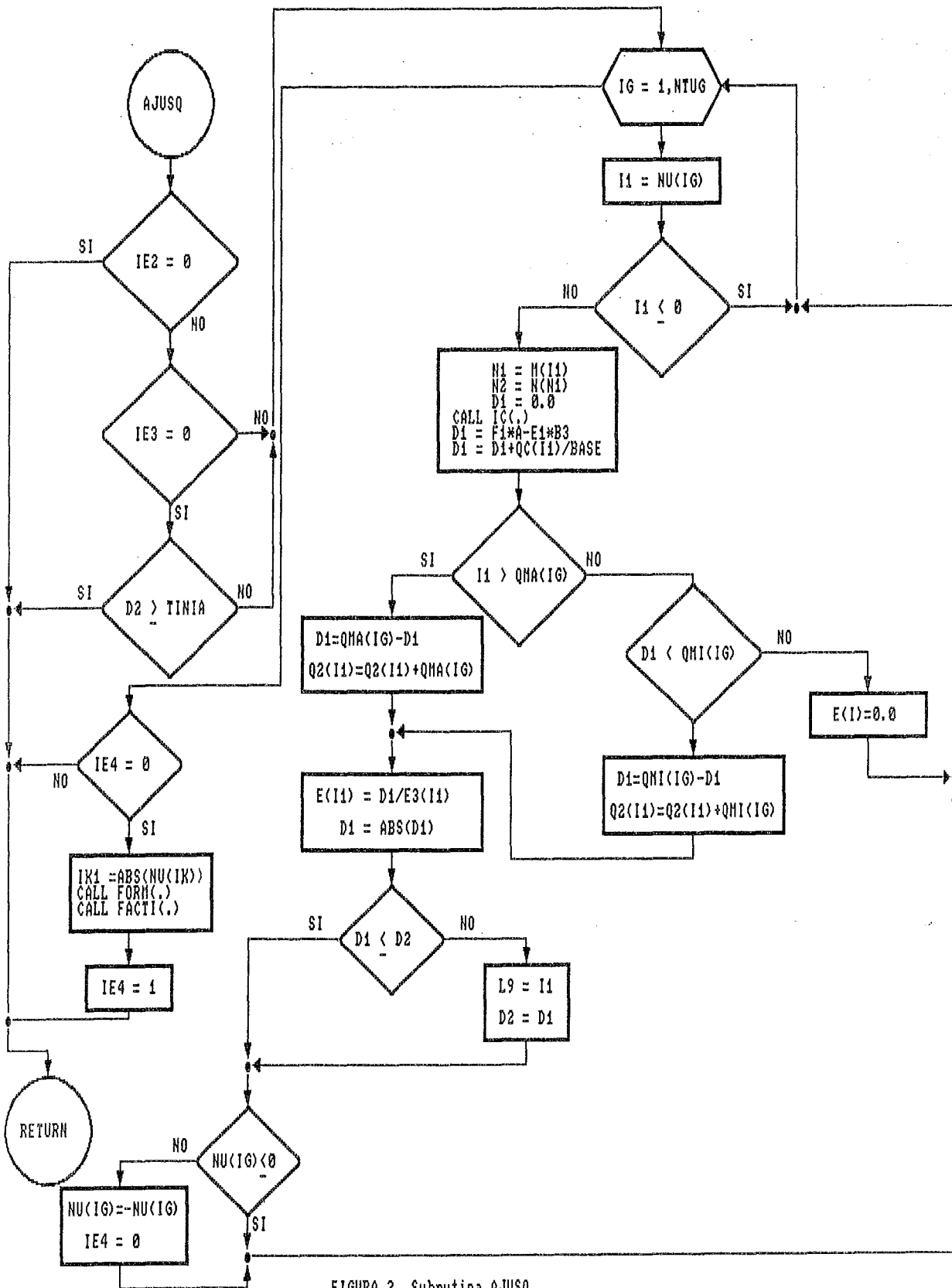


FIGURA 2 Subrutina AJUSQ

SUBROUTINA FORM

Esta subrutina es llamada por la subrutina AJUSQ para rehacer el modelo de potencia reactiva, usando la parte imaginaria de la matriz de admitancia nodal cuando en una iteración del método desacoplado existen nodos de voltaje controlado que se convierten en nodos de carga.

La subrutina se ejecuta mediante la instrucción:

CALL FORM (NNOD, NACT, NISLA, NISA, NTUG, ISCOM, NORD, NORD1, NU,
2 M, N, NB1, NB1C, B2, YI)

La lista de argumentos de FORM es la siguiente:

Variable	Entrada ó Salida	Tipo	Descripción
NNOD	E	Entera	Número de nodos en la red.
NACT	E	Entera	Número de nodos activos en la red.
NISLA	E	Entera	Número de islas eléctricas en la red.
NTUG	E	Entera	Número de nodo de voltaje controlado.
ISCOM(NISA)	E	Entera	Lista de nodos compensadores.

Variable	Entrada o Salida	Tipo	Descripción
NORD(NNOD)	E	Entera	Lista de nuevo orden de eliminación nodal.
NORD1(NNOD)	E	Entera	Lista de nodos en orden de eliminación.
NU(NTUG)	E	Entera	Lista de nodos de voltaje controlado.
M(NNOD)	E	Entera	Vector de apuntadores de primera posición por nodo en la matriz de admitancias nodal.
N(NL2)	E	Entera	Vector de columnas con elemento no cero en la matriz de admitancias. N(M(I)) informa el número de conexiones del nodo I.
NB1(NACT)	E	Entera	Vector de apuntadores de primera posición por nodo en el modelo de potencia reactiva.

Variable	Entrada o Salida	Tipo	Descripción
NB1C(IFIL)	E	Entera	Lista de columnas con elemento no cero en el modelo de potencia reactiva. NB1C(NB1(I)) informa el número de elementos en el triángulo superior del modelo.
B2(IFIL)	E/S	Real	Elemento del modelo de potencia reactiva triangularizado.
YI(NL2)	E/S	Real	Parte imaginaria de la matriz de admitancia nodal.

La figura 3 presenta el diagrama de flujo de la subrutina FORM.

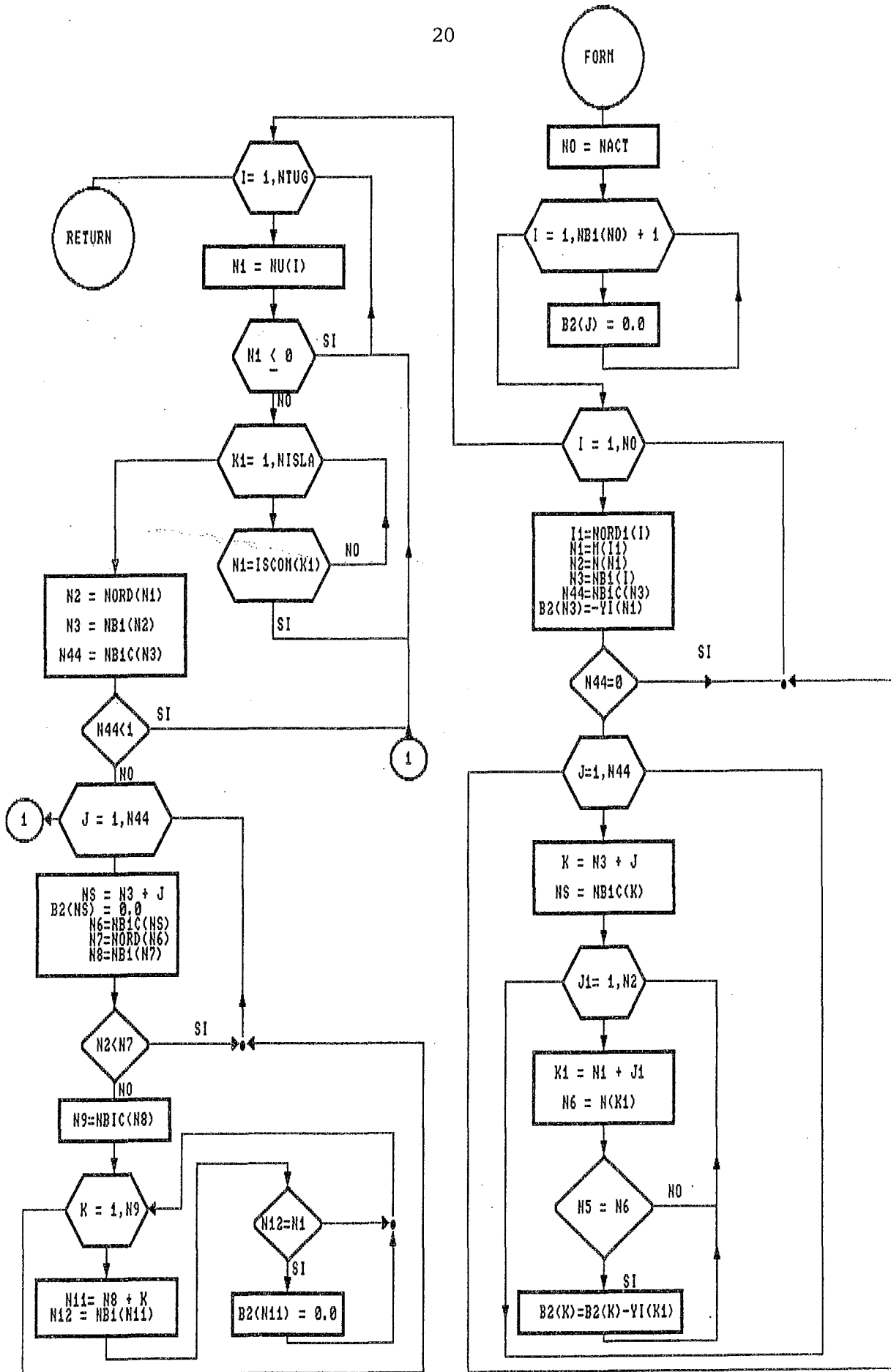


FIGURA.3 SUBROUTINA FORM

SUBROUTINA FACTI

La subrutina FACTI es utilizada por la subrutina AJUSQ para refactorizar el modelo de potencia reactiva. La subrutina trabaja exclusivamente sobre el triángulo superior del modelo.

La subrutina se ejecuta mediante el comando:

CALL FACTI (NNOD, NACT, IFIL, NISLA, NISA, NB1, NBIC, NORD, B2)

La lista de argumentos de la subrutina es la siguiente:

Variable	Entrada o Salida	Tipo	Descripción
NNOD	E	Entera	Número de nodos en la red.
NACT	E	Entera	Número de nodos activos en la red.
IFIL	E	Entera	Longitud del modelo de potencia reactiva.
NISLA	E	Entera	Número de islas eléctricas en la red.
NISA	E	Entera	Número de islas eléctricas.

Variable	Entrada o Salida	Tipo	Descripción
NB1(NACT)	E	Entera	Véctor de primeras posiciones por nodo en el modelo de potencia reactiva.
NB1C(IFIL)	E	Entera	Véctor de columnas con elemento no cero en el modelo de potencia reactiva triangularizado.
NORD(NNOD)	E	Entera	Orden de eliminación de nodos.
B2(NNOD)	E	Real	Elementos del modelo de potencia reactiva.

La figura 4 representa un diagrama de flujo de la subrutina FACTI.

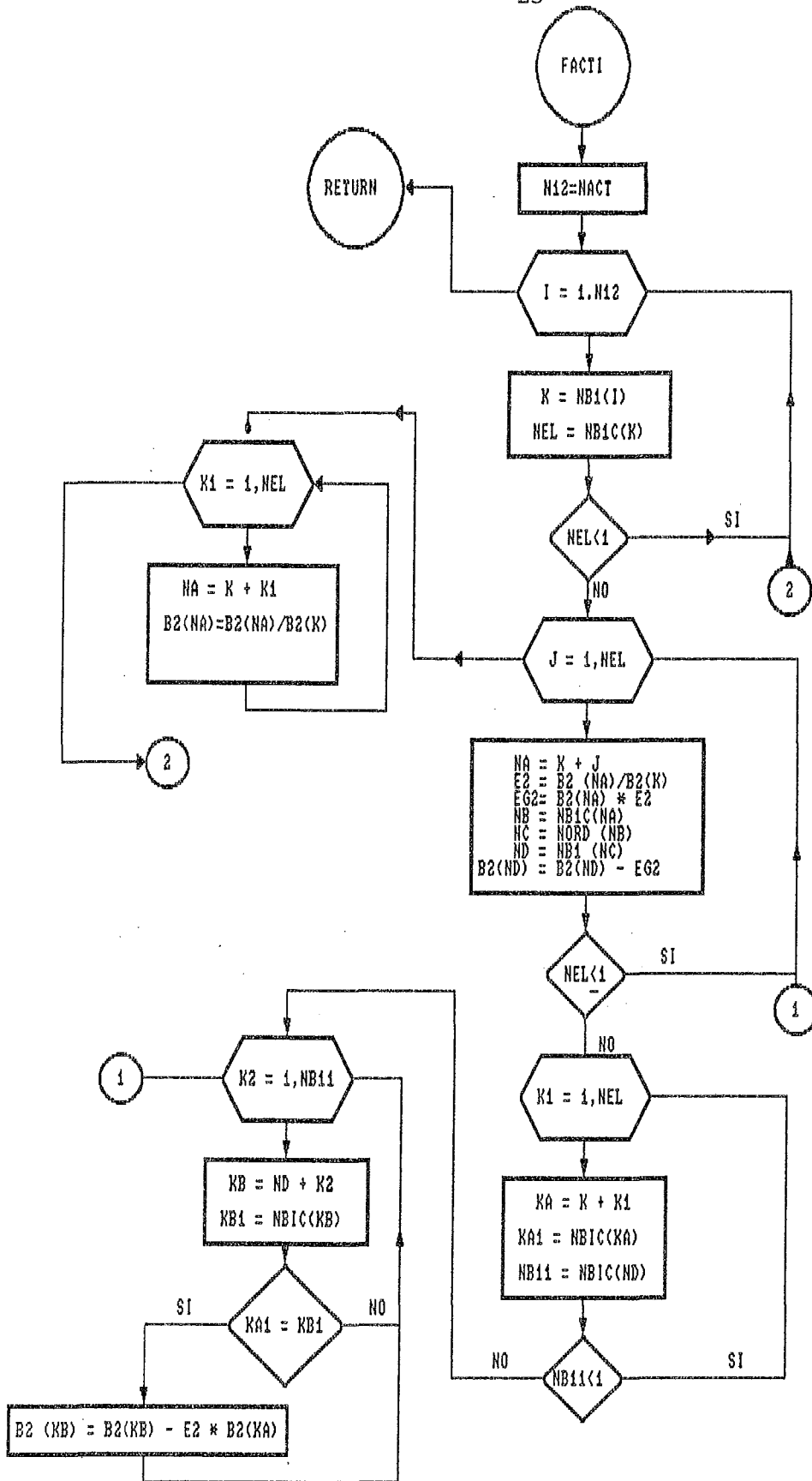


FIGURA 4. SUBRUTINA FACT1

REFERENCIAS

- [1] Mota Palomino Ricardo, "Programa para solución de estudios de flujos por el método desacoplado rápido", Manual de Usuario, 1989.

- [2] Mota Palomino Ricardo, "Bibliotecas para la solución de estudios de flujos convencionales por el método desacoplado rápido", Reporte Interno SGIE-RI-01287, ESIME-IPN, Sección de Graduados, 1987.

Anexo

BIBLIOTECA PARA LA SOLUCION DE ESTUDIOS DE FLUJOS
CONVENCIONALES POR EL METODO DESACOPLADO RAPIDO

RESUMEN

Se presentan en este reporte los detalles técnicos de las rutinas de biblioteca desarrolladas para realizar estudios de flujos en un sistema eléctrico por el método desacoplado rápido. A fin de efectuar estos estudios, se requiere procesar previamente la información básica de la red y la condición de operación del sistema por lo que las rutinas correspondientes se incluyen también en este reporte. Se presentan descripciones breves de cada entrada y salida involucradas así como el diagrama de flujo correspondiente. *

* Las rutinas fuente pueden ser obtenidas previa justificación con el Dr. Ricardo Mota Palomino, Secc. de Graduados e Investigación ESIME, Tel. 5-86-90-01

1. Rutinas para Preprocesamiento de Información de Red.

Todo programa para resolver estudios de flujos requiere de la construcción de modelos adecuados de la red ó sistema, a fin de resolver las ecuaciones no lineales correspondientes. La biblioteca descrita en este reporte utiliza el método desacoplado rápido [1] y por lo tanto requiere de la información de modelos lineales de potencia activa y reactiva así como de sus factores triangulares para realizar iteraciones en forma eficiente.

Las rutinas involucradas en este proceso y el orden en que deben ser ejecutadas se muestra en la Figura 1.

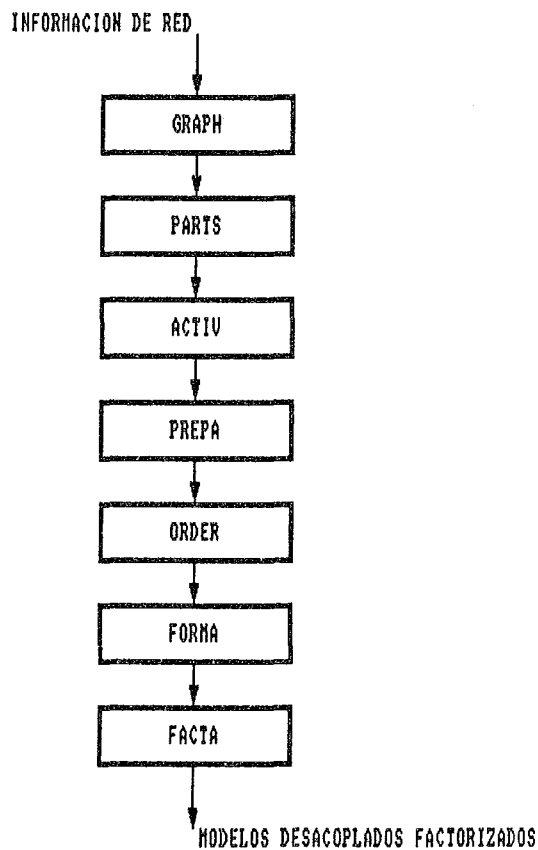


FIG.1.- SECUENCIA DE EJECUCION DE RUTINAS.

Las características y funciones de las rutinas mostradas en la Figura 1 se describen a continuación.

1.1 Subrutina GRAPH.

Partiendo de la información básica de adyacencia de la red eléctrica la subrutina GRAPH forma un grafo no orientado de la red de transmisión representado mediante una lista de adyacencia encadenada.

La subrutina es llamada mediante la instrucción:

```
CALL GRAPH(NNOD, NLIN, NL2, NE, NR, M, N, IPP, ISP)
```

Los argumentos de la subrutina son:

Variable	Entrada ó Salida	Tipo	Descripción
NNOD	E	Entera	Número de nodos del sistema analizado.
NLIN	E	Entera	Número de líneas ó bancos del sistema analizado.
NL2	E	Entera	$2 * NLIN + NNOD$ se calcula antes de llamar a la subrutina
NE(NNOD)	E	Entera	Lista de nodos de envío de las líneas ó bancos del sistema.
NR(NNOD)	E	Entera	Lista de nodos de recepción de líneas ó bancos del sistema.
M(NNOD)	S	Entera	Vector de primera posición de la lista de adyacencia del grafo.

Variable	Entrada ó Salida	Tipo	Descripción
N(NL2)	S	Entera	Vector de adyacencia del grafo. Las posiciones N(M(I)) representan el número de conexiones que tiene el nodo I.
IPP(NL2)	S	Entera	Vector de apuntadores de la lista de adyacencia a la lista de líneas ó bancos de la red. La variable (IPP(J) representa la posición en NE(.) y NR(.) correspondiente al elemento N(J) de la lista.
ISP(NL2)	S	Entera	Vector de encadenamiento de la lista de adyacencia del grafo.

La Figura 2 muestra un diagrama de flujo de la subrutina GRAPH.

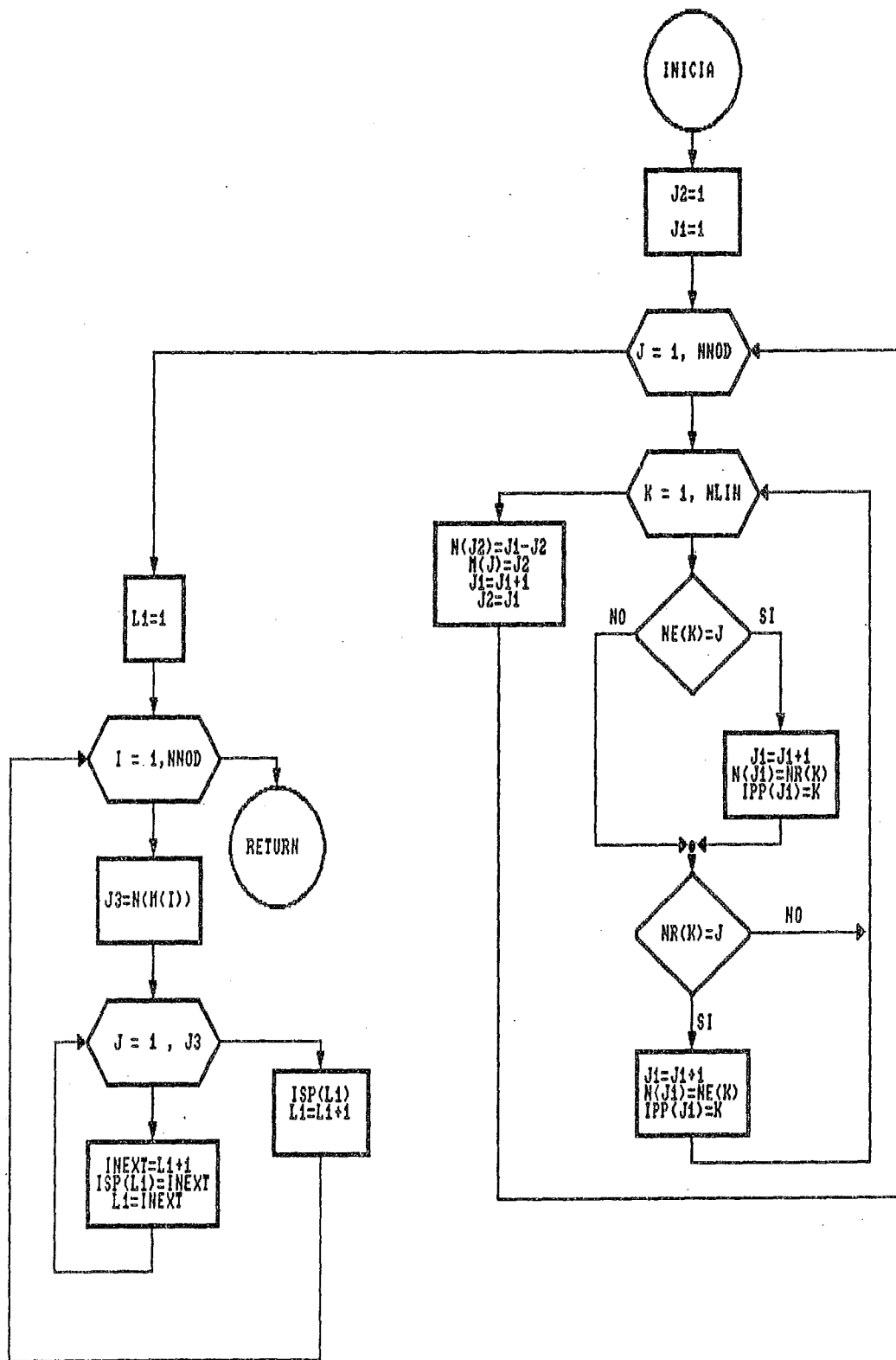


FIG.2.- SUBRUTINA GRAPH.

1.2 Subrutina PARTS.

La subrutina PARTS analiza la conectividad de un grafo no dirigido a fin de identificar las partes conectadas. Esta subrutina es útil para determinar las islas eléctricas que existen en una red.

La subrutina es ejecutada mediante la instrucción:

```
CALL PARTS(NNOD, NL2, M, N, ISP, ISLA, NLIS, NISLA)
```

Los argumentos de la subrutina PARTS son:

Variable	Entrada ó Salida	Tipo	Descripción
NNOD	E	Entera	Número de nodos del sistema analizado.
NL2	E	Entera	$2 * NLIN + NNOD$. Se calcula externamente
M(NNOD)	E	Entera	Vector de primera posición de la lista
N(NL2)	E	Entera	Vector de adyacencia del grafo.
ISP(NL2)	E	Entera	Vector de encadenamiento de la lista de adyacencia del grafo.
ISLA(NNOD)	S	Entera	Indicador de isla a la que pertenece cada nodo.
NLIS(NNOD)	E/S	Entera	Vector auxiliar para mantener una pila de nodos que pertenecen a una isla.

La Figura 3 muestra un diagrama de flujo de la subrutina PARTS.

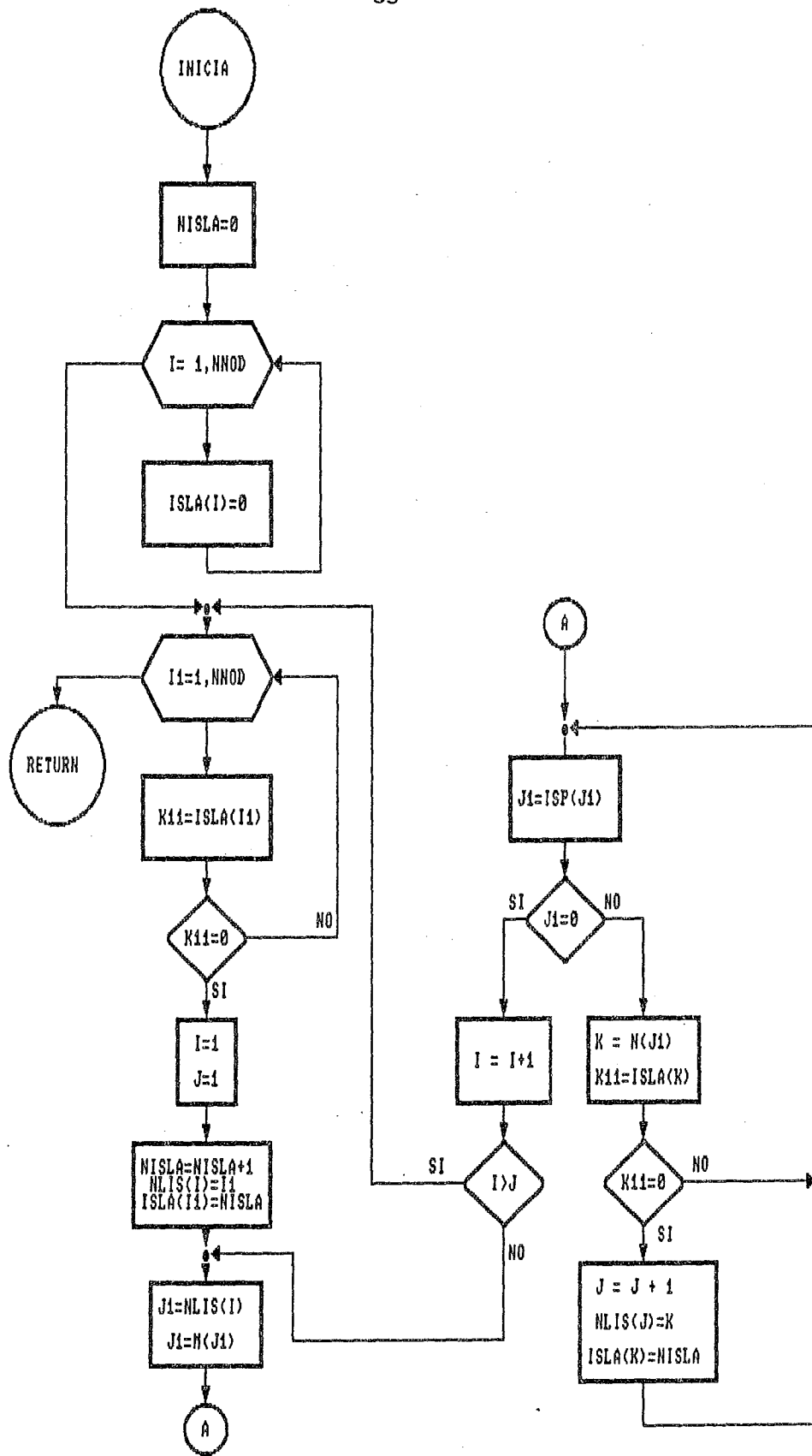


FIG. 3.- SUBRUTINA PARTS.

1.3 Subrutina ACTIV.

La subrutina ACTIV verifica la existencia de generación y/ó carga en cada isla de la red. A continuación se pone una bandera en nodos pertenecientes a islas que no tienen generación y/ó carga y finalmente selecciona un nodo compensador para cada isla de la red que tenga que tenga generación y/ó carga (isla activa). Los nodos compensadores seleccionados serán aquellos que tengan asignada la mayor potencia en el archivo de entrada en cada isla.

La subrutina es ejecutada mediante la instrucción :

CALL ACTIV (NNOD, NISLA, NISA, NTUG, ISLA, NU, PG, PC, ISCOM, PGIS, PCIS, PSLA)

Los argumentos de la subrutina ACTIV son:

Variable	Entrada ó Salida	Tipo	Descripción
NNOD	E	Entera	Número de nodos en el sistema.
NISLA	E	Entera	Número de islas eléctricas en el sistema.
NISA	S	Entera	Número de islas eléctricas activas en el sistema.
NTUG	E	Entera	Número total de nodos de generación en el sistema.
ISLA(NNOD)	E	Entera	Indicador de la isla a que pertenece cada nodo.

Variable	Entrada ó Salida	Tipo	Descripción
NU(NTUG)	E	Entera	Lista de los nodos generadores.
PG(NTUG)	E	Real * 8	Potencia real asignada a cada nodo generador.
PC(NNOD)	E	Real * 8	Potencia real de carga en cada nodo de la red.
ISCOM(NISLA)	S	Entera	Lista de nodos compensadores.
PGIS(NISLA)	S	Real * 8	Potencia real de generación asignada por isla.
PCIS(NISLA)	S	Real * 8	Potencia real de carga por isla.
PSLA(NISLA)	S	Real * 8	Potencia real del compensador asignado por isla.

La Figura 4 muestra un diagrama de flujo de la subrutina ACTIV.

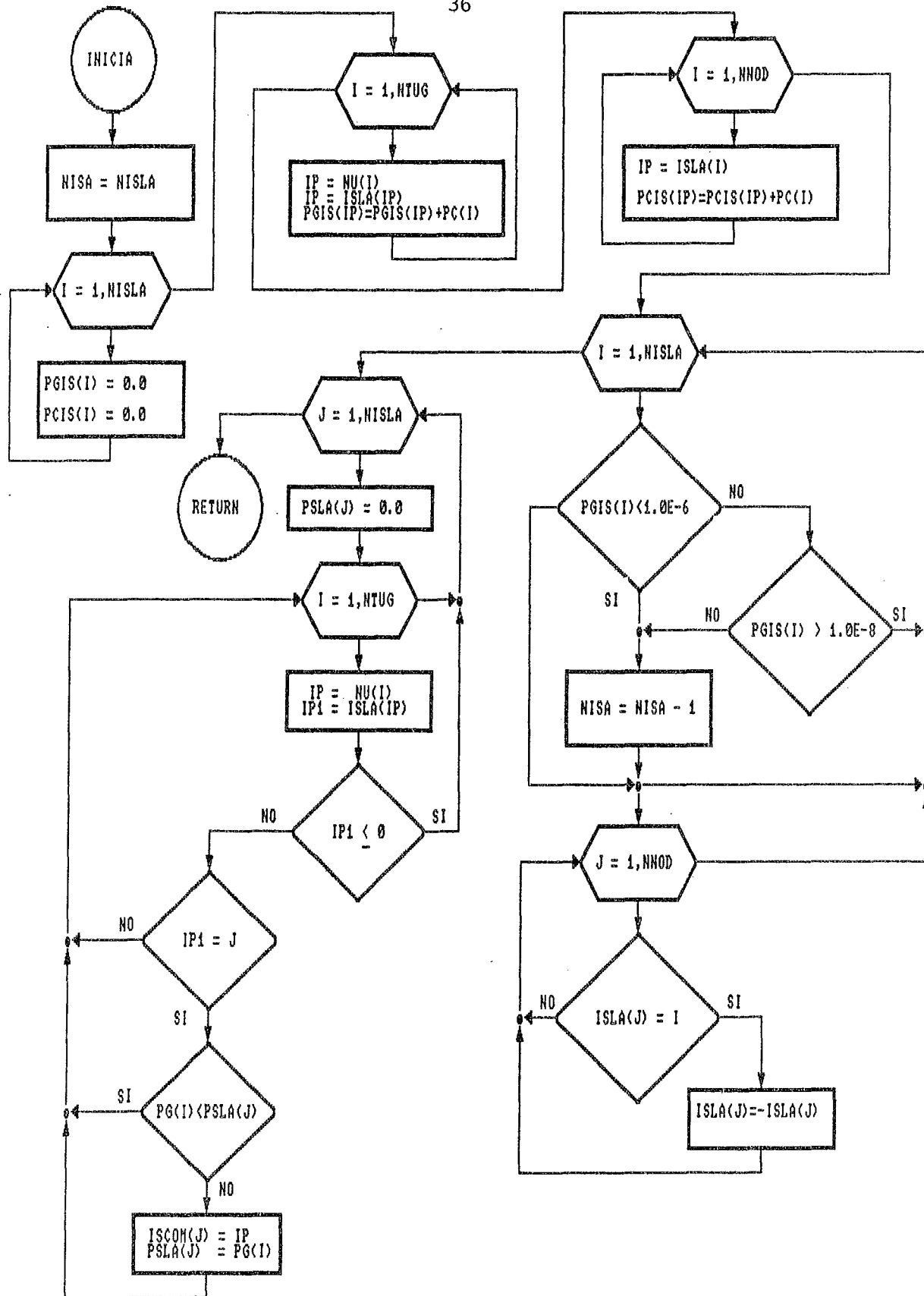


FIG.4.- SUBROUTINA ACTIV

1.4 Subrutina PREPA

La subrutina PREPA transforma el grafo de la red en un grafo simple. Primeramente inicializa una lista de localidades libres en la lista encadenada que representa al grafo a fin de implementar un "recolector de basura" en el proceso de triangulización de modelos; posteriormente elimina ejes paralelos en el grafo, desconecta nodos correspondientes a nodos compensadores en el grafo y finalmente asigna los nodos de orden mayor de eliminación a nodos correspondientes a islas no activas.

La subrutina PREPA es ejecutada mediante la instrucción:

```
CALL PREPA (NNOD, NLIN, NL2, IEXT, NISLA, M, N, ISP, IPP,
           ISLA, ISCOM, LP, NORD, NACT, IDISP)
```

los argumentos de la subrutina PREPA son los siguientes:

Variable	Entrada ó Salida	Tipo	Descripción
NNOD	E	Entera	Número de nodos en el sistema.
NLIN	E	Entera	Número de líneas o bancos en el sistema.
NL2	E	Entera	$2 * NLIN + NNOD$
IEXT	E	Entera	Máxima dimensión de los arreglos dinámicos usados en la simulación de triangularización (arreglos N, ISP).
NISLA	E	Entera	Número de islas eléctricas en el sistema.

Variable	Entrada ó Salida	Tipo	Descripción
M(NNOD)	E	Entera	Vector de primeras posiciones en la lista de adyacencia del grafo.
N(IEXT)	E/S	Entera	Lista de adyacencia del grafo.
ISP(IEXT)	E/S	Entera	Vector de encadenamiento de lista de adyacencia. También contiene lista encadenada de posiciones libres (recolector de basura).
IPP(NL2)	E	Entera	Vector de apuntadores de lista de adyacencia a lista de elementos de transmisión de la red.
ISLA(NNOD)	E	Entera	Indicador de la isla a que pertenece cada nodo de la red.
ISCOM(NI,SLA)	E	Entera	Lista de nodos compensadores por isla.
LP(NLIN)	E	Entera	Vector paralelo a los arreglos NE(.), NR(.), R(.), X(.), B(.) que identifica si la línea ó banco es paralela a otra.

Variable	Entrada ó Salida	Tipo	Descripción
NORD(NNOD)	S	Entera	Vector de orden de eliminación de nodos. A nodos compensadores e inactivos se les asigna un orden alto (mayor a NNOD).
NACT	S	Entera	Número de nodos a eliminar en el proceso de eliminación Gaussiana de modelos.
IDISP	S	Entera	Apuntador a la primera posición de localidades libres en la lista de adyacencia del grafo (recolector de basura).

La Figura 5 ilustra el diagrama de flujo de la subrutina PREPA

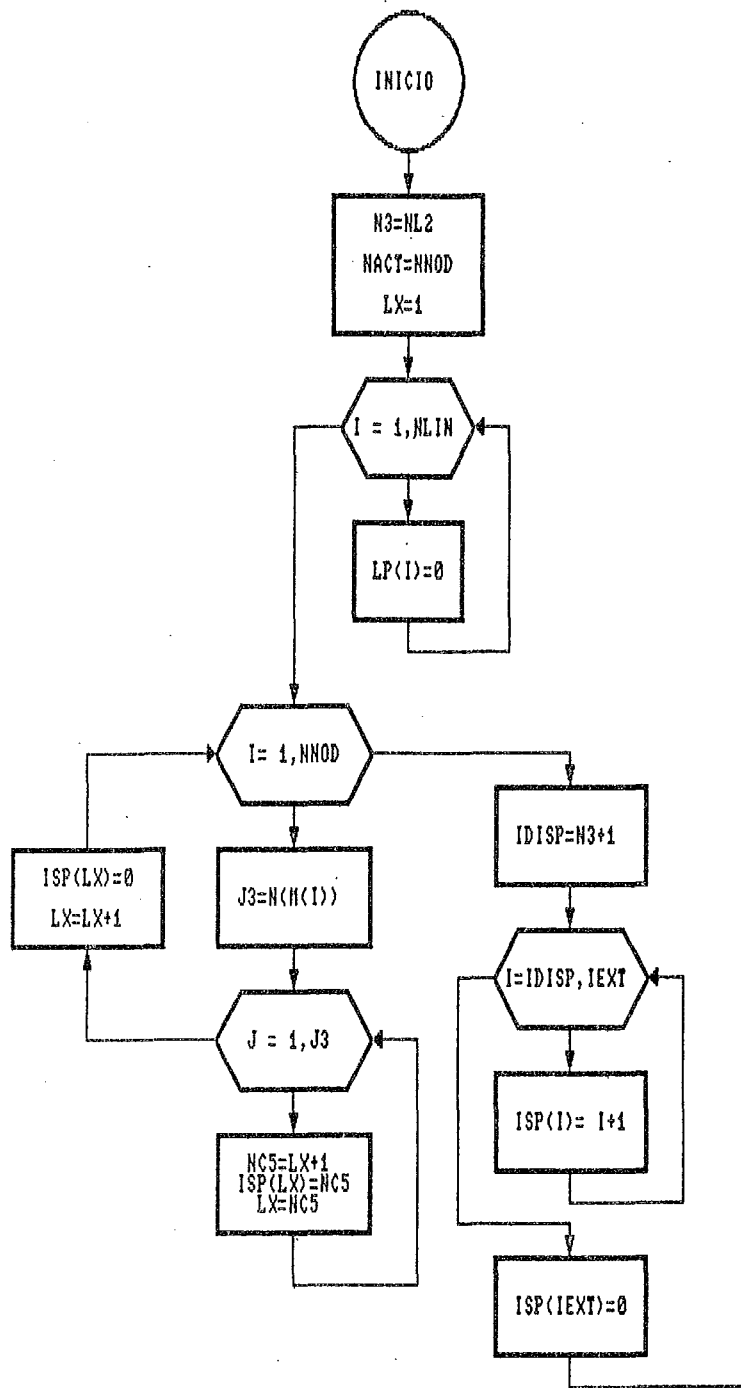
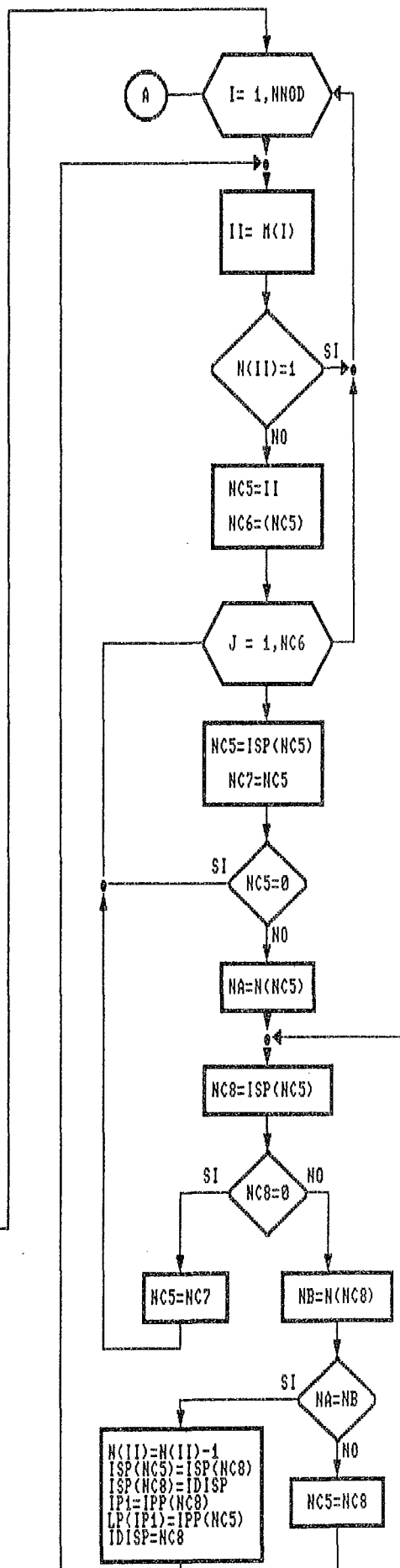
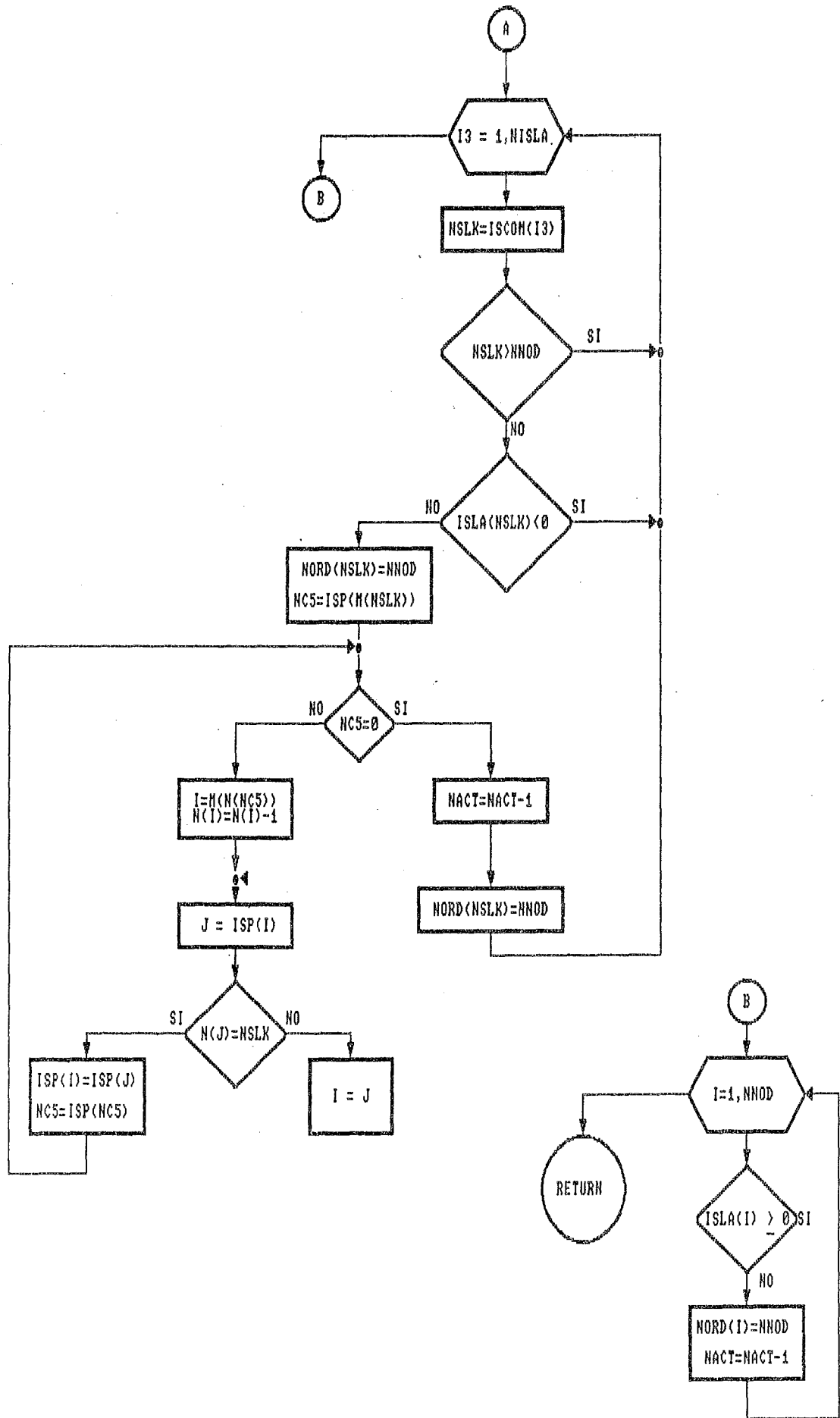


FIG.5.-SUBROUTINA PREPA





1.4 Subrutina ORDER

La subrutina ORDER efectúa un ordenamiento de acuerdo al nodo de "mínimo grado", es decir en cada paso del proceso de eliminación Gaussiana selecciona como siguiente nodo a eliminar a aquel que tiene el menor número de conexiones.

La subrutina ORDER es ejecutada mediante el comando

```
CALL ORDER (NISLA, NNOD, NLIN, NL2, IEXT, NACT, M, N, ISP, IPP,
            NORD, IDISP)
```

Los argumentos de la subrutina ORDER son los siguientes:

Variable	Entrada ó Salida	Tipo	Descripción
NISLA	E	Entera	Número de islas eléctricas en la red.
NNOD	E	Entera	Número de nodos en el sistema.
NLIN	E	Entera	Número de líneas ó bancos en la red analizada.
NL2	E	Entera	$2 * NLIN + NNOD$
IEXT	E	Entera	Máxima dimensión de los arreglos dinámicos usados en la simulación de la eliminación Gaussiana: N(.), ISP(.)

Variable	Entrada ó Salida	Tipo	Descripción
NACT	E	Entera	Número máximo de nodos a eliminar en el proceso de simulación de triangularización.
M(NNOD)	E	Entera	Vector de primeras posiciones en la lista de adyacencia y del grafo.
N(IEXT)	E/S	Entera	Lista de adyacencia por nodo incluyendo elementos creados al triangularizar los arreglos matriciales (Ybus).
ISP(IEXT)	E/S	Entera	Vector de encadenamiento de listas de adyacencia el grafo y lista de posiciones libres.
IPP(INL2)	E/S	Entera	Apuntadores de lista de adyacencia a lista de elementos de transmisión.
NORD(NNOD)	E/S	Entera	Vector de orden de eliminación de nodos.
IDISP	E/S	Entera	Apuntador a primera posición de localidades libres en lista de adyacencia.

La Figura 6 muestra un diagrama de flujo de la subrutina ORDER

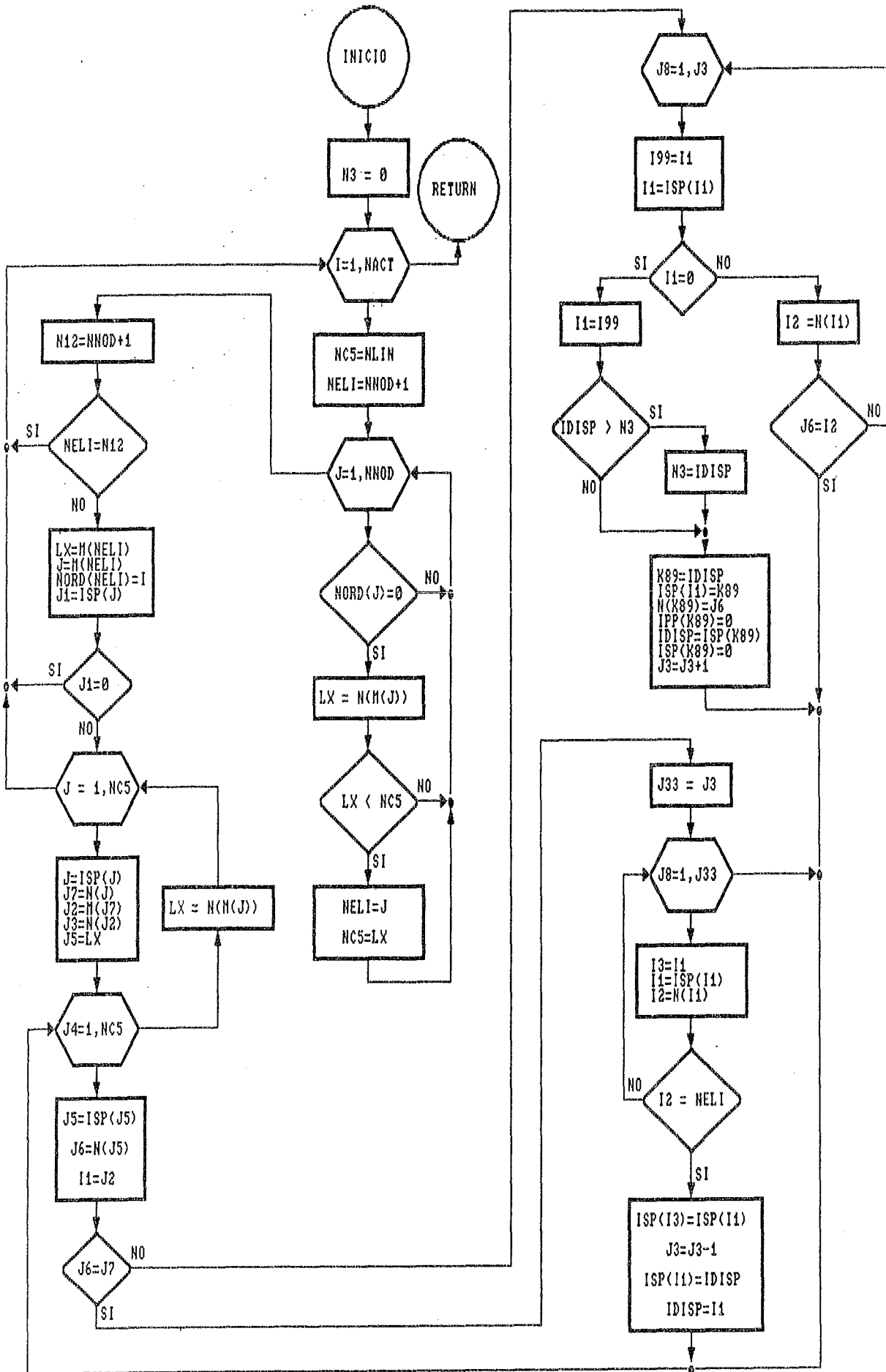


FIG. 6.-SUBROUTINA ORDER

1.5 Subrutina FORMA

La subrutina FORMA toma la información del "grafo de eliminación perfecta" resultado de la rutina de ordenamiento y forma los modelos de potencia real y de potencia reactiva para el método desacoplado rápido.

La subrutina FORMA se ejecuta mediante la instrucción:

CALL FORMA (NNOD, NACT, NLIN, NL2, NISLA, NISA, IEXT, IFIL, NTUG, NSHU, NTR, LP, M, N, NB1, NB1C, B1, B2, IPP, R, X, B, ISP, NORD, NCT, NST, T1, NSH, VLS, NU, ISCOM, NE, NR, NLIS).

Los argumentos de la subrutina FORMA son los siguientes:

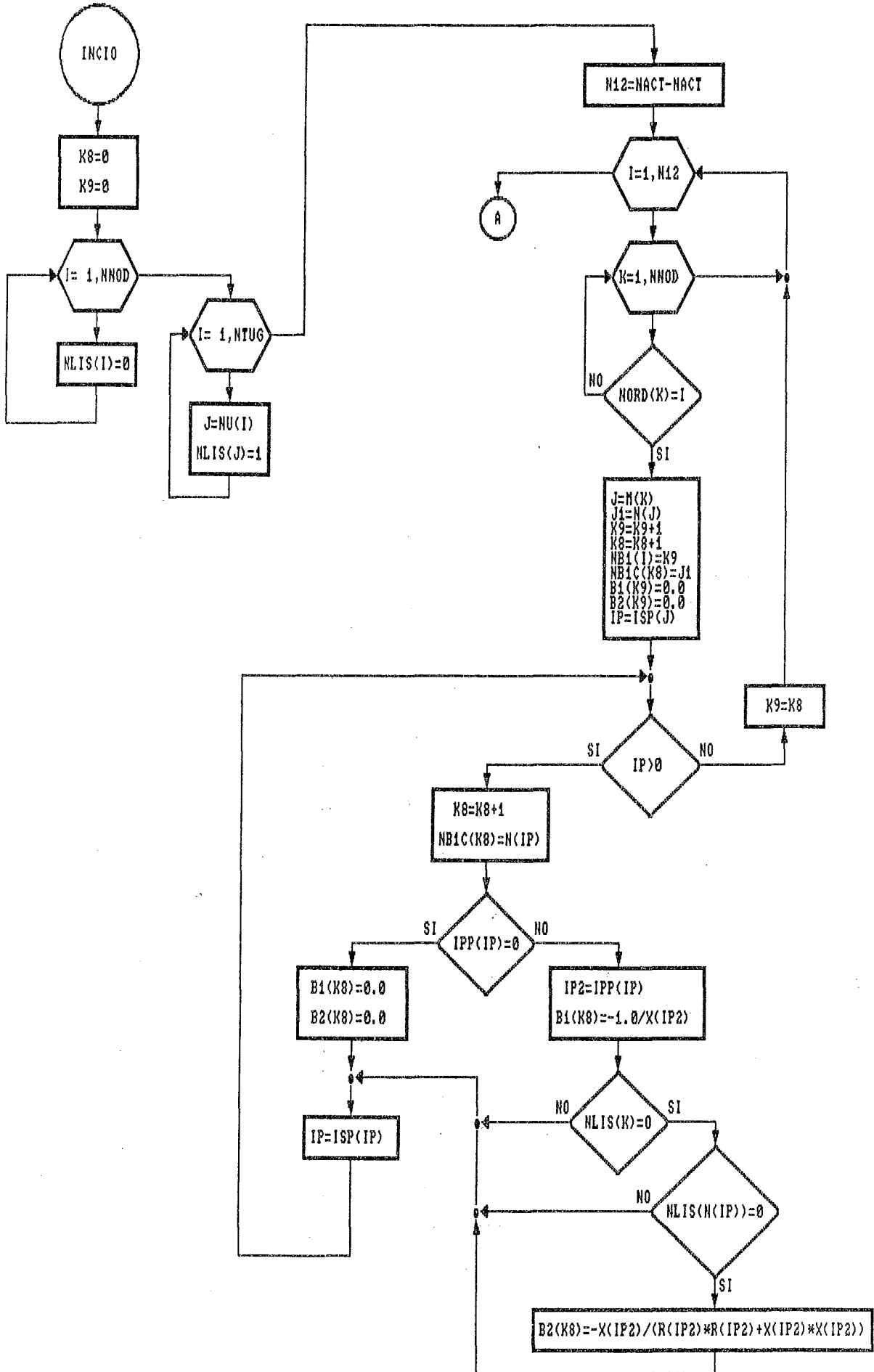
Variable	Entrada ó Salida	Tipo	Descripción
NNOD	E	Entera	Número de nodos en la red.
NACT	E	Entera	Número de nodos a eliminar el proceso de triangulización.
NLIN	E	Entera	Número de elementos de transmisión en la red.
NL2	E	Entera	$2 * NLIN + NNOD$
NISLA	E	Entera	Número de islas eléctricas en el sistema.

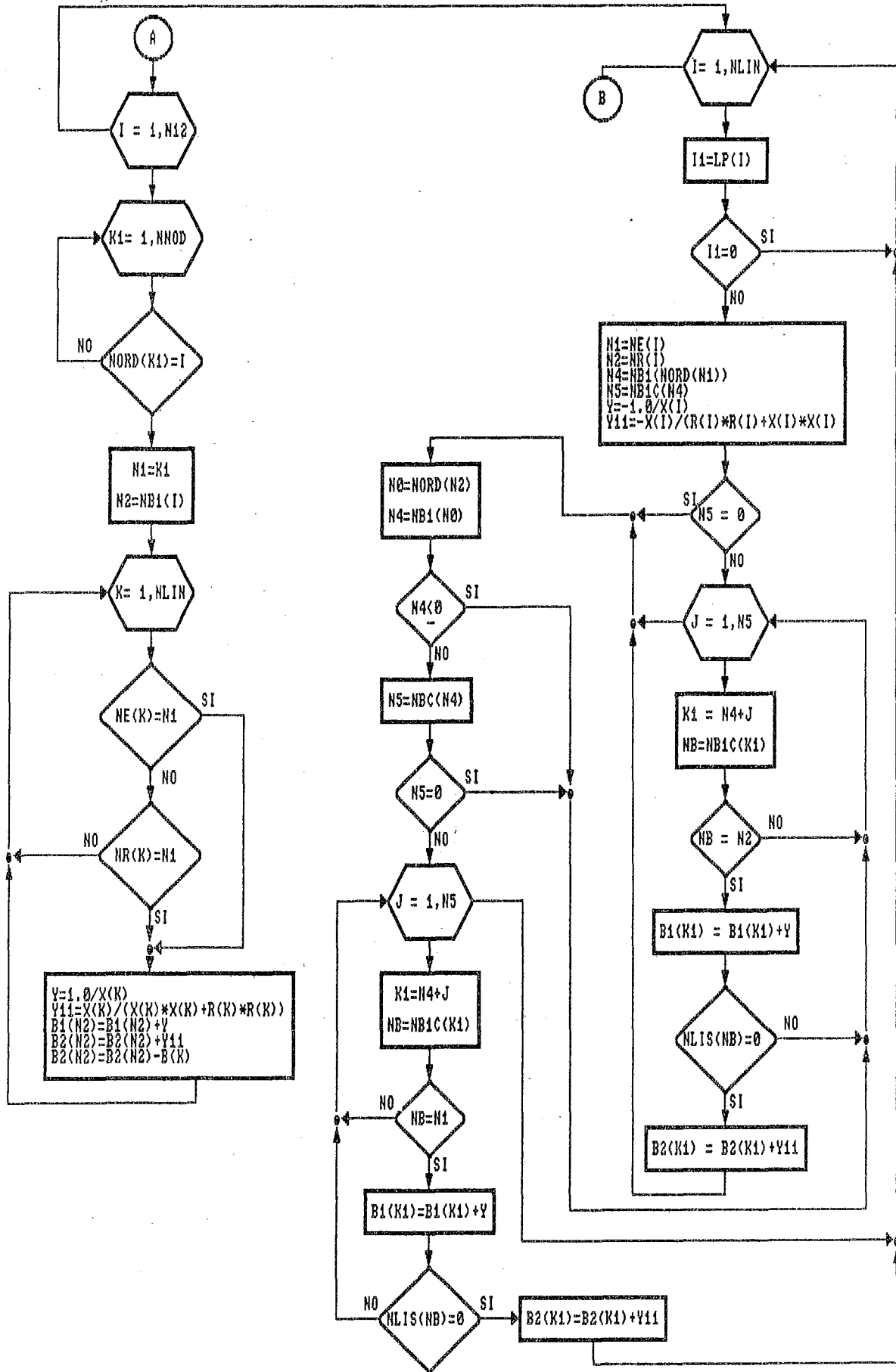
Variable	Entrada ó Salida	Tipo	Descripción
NISA	S	Entera	Número de islas eléctricas activas en el sistema.
IEXT	E	Entera	Máxima dimensión de arreglos en lista de adyacencia del grafo.
IFIL	S	Entera	Máxima dimensión de arreglo triangular del modelo de potencia real y reactiva.
NTUG	E	Entera	Número de nodos de generación en el sistema.
NSHU	E	Entera	Número de compensadores en paralelo en el sistema.
NTR	E	Entera	Número de bancos de transformación en el sistema.
LP(NLIN)	E	Entera	Vector paralelo a lista de líneas que indica con que línea es paralelo el elemento.
M(NNOD)	E	Entera	Vector de primera posición en lista de adyacencia del grafo.

Variable	Entrada ó Salida		Descripción
N(IEXT)	E	Entera	Vector de adyacencias del grafo de eliminación perfecta.
NB1(NNOD)	S	Entera	Vector de primera posición de modelos triangularizados.
NB1C(IFIL)	S	Entera	Vector de adyacencia de modelos triangularizados.
B1(IFIL)	S	Real * 8	Modelo de potencia real.
B2(IFIL)	S	Real * 8	Modelo de potencia reactiva.
IPP(IEXT)	E	Entera	Apuntadores de lista de adyacencia de grafo de eliminación perfecta a la lista de elementos de transmisión.
R(NLIN)	E	Real * 8	Resistencia de elementos de transmisión.
X(NLIN)	E	Real * 8	Reactancias de elementos de transmisión.
B(NLIN)	E	Real * 8	Suceptancia de una rama del pi equivalente de elementos de transmisión.

Variable	Entrada ó Salida	Tipo	Descripción
ISP(IEXT)	E	Entera	Vector de encadenamiento del grafo de eliminación perfecta.
NORD(NNOD)	E	Entera	Orden de eliminación de nodos.
NCT(NTR)	E	Entera	Lista de nodos con tap de transformadores.
NST(NTR)	E	Entera	Lista de nodos sin tap de transformadores.
T1(NTR)	E	Real * 8	Lista de taps en transformadores.
NSH(NSHU)	E	Entera	Lista de nodos con compensación en derivación.
VLS(NSHU)	E	Real * 8	Lista de compensadores en la red.
NU(NTUG)	E	Entera	Lista de nodos generadores.
ISCOM(NISLA)	E	Entera	Lista de nodos compensadores.
NE(NLIN)	E	Entera	Nodos de envío de elementos de transmisión.
NR(NLIN)	E	Entera	Nodos de recepción de elementos de transmisión.
NLIS(NNOD)	E/S	Entera	Vector auxiliar para abanderar nodos generadores en el proceso de formación de modelos.

La Figura 7 ilustra un diagrama de flujo de la Subrutina FORMA





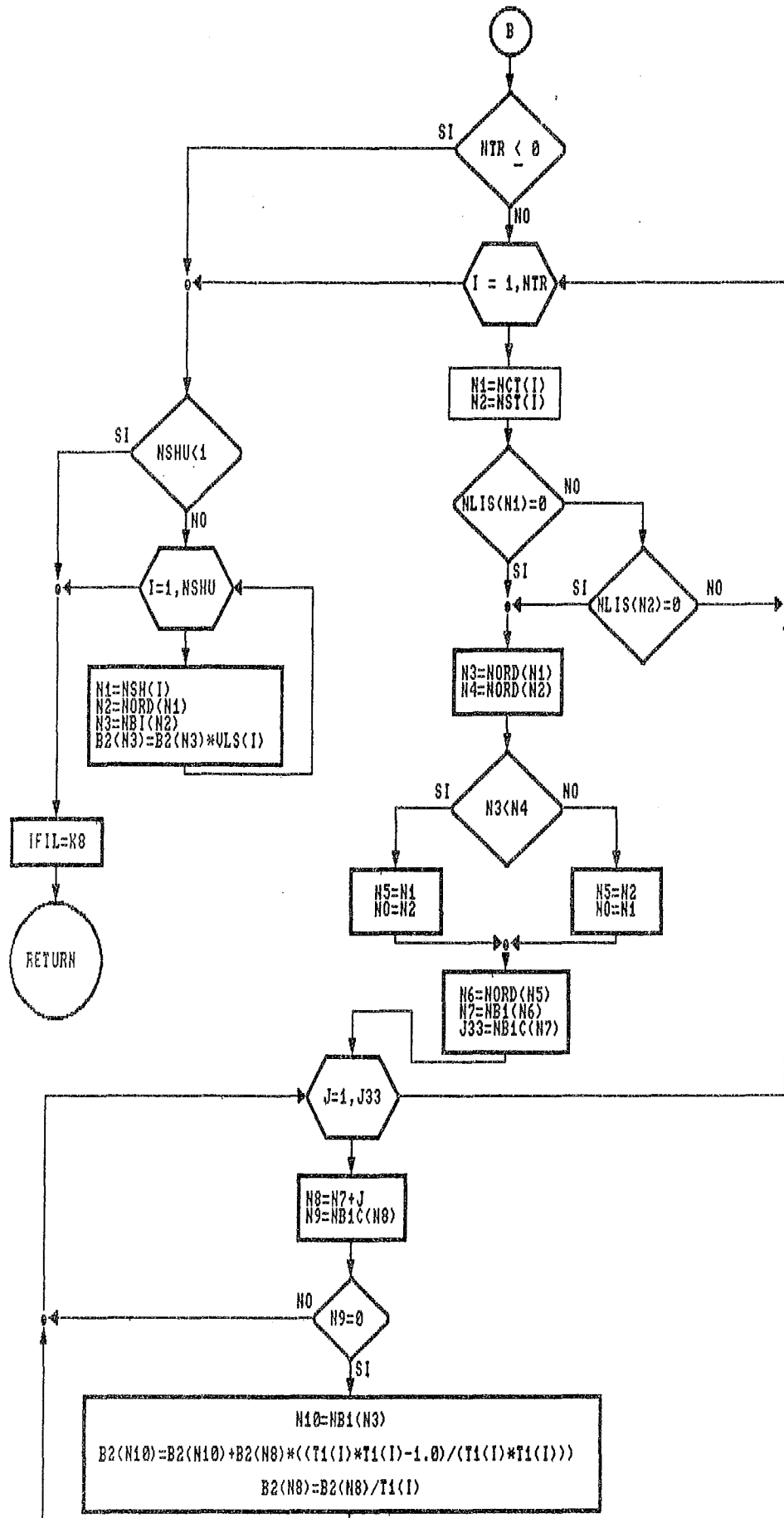


FIG.7.- SUBROUTINA FORMA

1.6 Subrutina FACTA

La Subrutina FACTA genera los factores triangulares de los modelos de potencia activa y reactiva del método desacoplado rápido explotando su simetría.

La subrutina FACTA se ejecuta mediante la instrucción:

CALL FACTA (NNOD, NACT, IFIL, NISLA, NISA, NB1, NB1C, NORD, B1, B2)

Los argumentos de la subrutina FACTA son los siguientes:

Variable	Entrada ó Salida	Tipo	Descripción
NNOD	E	Entera	Número de nodos del sistema.
NACT	E	Entera	Número de nodos a eliminar en el proceso de triangulización.
IFIL	E	Entera	Dimensión máxima de los arreglos del "grafo de eliminación perfecta".
NISLA	E	Entera	Número de islas eléctricas en el sistema.
NISA	E	Entera	Número de islas eléctricas activas en el sistema.

Variable	Entrada ó Salida	Tipo	Descripción
NB1(NNOD)	E	Entera	Vector de primeras posiciones en los modelos triangularizados
NB1C(IFIL)	E	Entera	Vector de listas de adyacencia de los modelos triangularizados
NORD(NNOD)	E	Entera	Orden de eliminación de nodos.
B1(IFIL)	E/S	Real * 8	Modelo triangularizado de potencia real.
B2(IFIL)	E/S	Real * 8	Modelo triangularizado de potencia reactiva.

La Figura 8 muestra un diagrama de flujo de la subrutina FACTA.

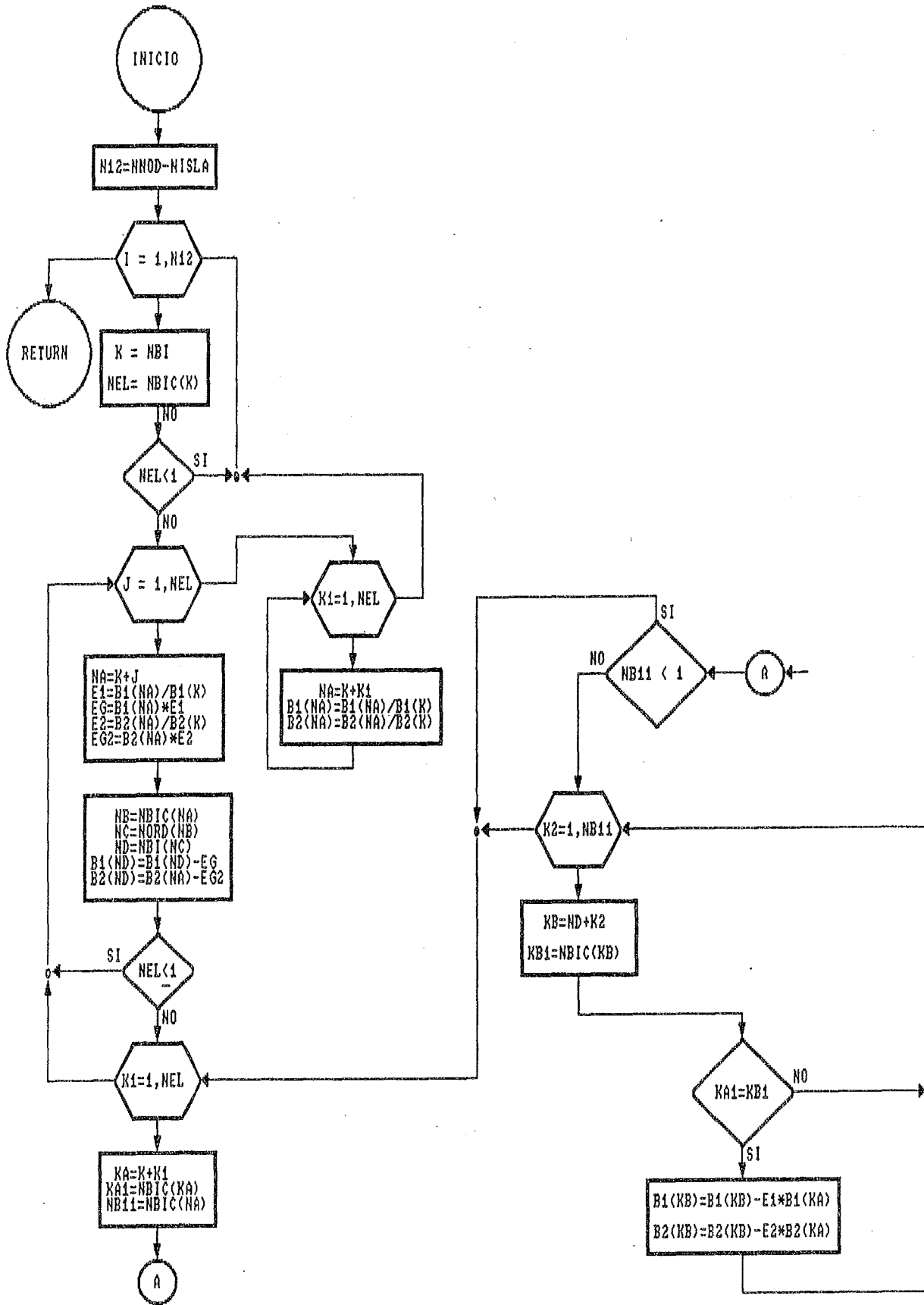


FIG. 8.- SUBROUTINA FACTA

2. Rutinas para resolver estudios de flujos (Método Desacoplado Rápido)

A continuación se describen las rutinas empleadas para resolver estudios convencionales mediante el método desacoplado rápido.

Dado que la solución de un estudio de este tipo puede ser considerada una actividad más en otras aplicaciones, el algoritmo del método desacoplado rápido ha sido desarrollado como una subrutina, la cual puede ser ejecutada mediante la instrucción:

```
CALL LFSOL (NNOD, NACT, NTUG, NISLA, NISA, NL2, IFIL, ISCOM, BASE,
           NORD, NORD1, PG, VG, NU, E3, F3, P2, Q2, PC, QC, YR,
           YI, NB1, NB1C, B1, B2, E, F, NITER, TOLP, TOLQ, M, N,
           DNOMB, TINIA, QMA, QMI).
```

La mayoría de los argumentos de la subrutina LFSOL deben ser formados antes de ejecutarla. Inclusive antes debe ser formada la matriz de admitancias nodal (ejecutando la rutina YBUSC(.)).

La lista total de argumentos de LFSOL se describen a continuación:

Variable	Entrada ó Salida	Tipo	Descripción
NNOD	E	Entera	Número de nodos en la red.
NACT	E	Entera	Número de nodos a eliminar en el proceso de triangularización.
NTUG	E	Entera	Número de nodos generadores.

Variable	Entrada ó Salida	Tipo	Descripción
NISLA	E	Entera	Número de islas eléctricas en el sistema.
NISA	E	Entera	Número de islas activas en el sistema.
NL2	E	Entera	$2 * NLIN + NNOD$
IFIL	E	Entera	Extensión de arreglos conteniendo los modelos triangularizados.
ISCOM(NISLA)	E	Entera	Lista de nodos compensadores.
BASE	E	Real * 8	Base de potencia (MVA) usada.
NORD(NNOD)	E	Entera	Orden de eliminación de nodos.
NORD1(NNOD)	E	Entera	Lista de nodos en orden de eliminación.
PG(NTUG)	E	Real * 8	Potencias activas asignadas por nodo generador.
VG(NTUG)	E	Real * 8	Voltajes asignados (pu) en nodos generadores.
NU(NTUG)	E	Entera	Lista de nodos generadores.
E3(NNOD)	E/S	Real * 8	Variable auxiliar.
F3(NNOD)	E/S	Real * 8	Variable auxiliar.

Variable	Entrada ó Salida	Tipo	Descripción
P2(NNOD)	E/S	Real * 8	Variable auxiliar.
Q2(NNOD)	E/S	Real * 8	Variable auxiliar.
PC(NNOD)	E	Real * 8	Potencial real de carga por nodo.
QC(NNOD)	E	Real * 8	Potencial reactiva de carga por nodo.
YR(NL2)	E	Real * 8	Parte real de la matriz de admitancias nodal.
YI(NL2)	E	Real * 8	Parte imaginaria de la matriz de admitancias nodal.
NB1(NNOD)	E	Entera	Vector de primeras posiciones de los arreglos factorizados.
NB1C(IFIL)	E	Entera	Vector de columnas de arreglos factorizados.
B1(IFIL)	E	Real * 8	Vector de modelo de potencia activa factorizado .
B2(IFIL)	E	Real * 8	Vector de modelo de potencia reactiva factorizado.
E (NNOD)	S	Real * 8	Parte real de los voltajes nodales.
F (NNOD)	S	Real * 8	Parte imaginaria de los voltajes nodales.
NITER	E	Entera	Número máximo de iteraciones permitidas

Variable	Entrada ó Salida	Tipo	Descripciones
TOLP	E	Real * 8	Tolerancia en el modelo de potencia activa.
TOLQ	E	Real * 8	Tolerancia en el modelo de potencia reactiva.
M(NNOD)	E	Entera	Vector de primeras posiciones para matriz de admitancias nodal.
N (NL2)	E	Entera	Vector de columnas para la matriz de admitancias nodal. (N(M(I))) es el número de conexiones al nodo I. YR(M(I)) Y YI(M(I)) son los valores diagonales Y _{ii} en parte real e imaginaria, respectivamente.
DNOME(NNOD)	E	Real * 8	Lista de nombres de los nodos de la red.
TINIA	E	Entera	Tolerancia a partir de que se empieza a ajustar límites de potencia reactiva en nodos generadores.
QMA(NTUG)	E	Real * 8	Vector de límites máximos de potencia reactiva generada en nodos de voltaje controlado.
QMI(NTUG)	E	Real * 8	Vector de límites máximos de potencia reactiva generada en nodos de voltaje controlado.

La Figura 9 muestra un diagrama de flujo de la subrutina LFSOL.

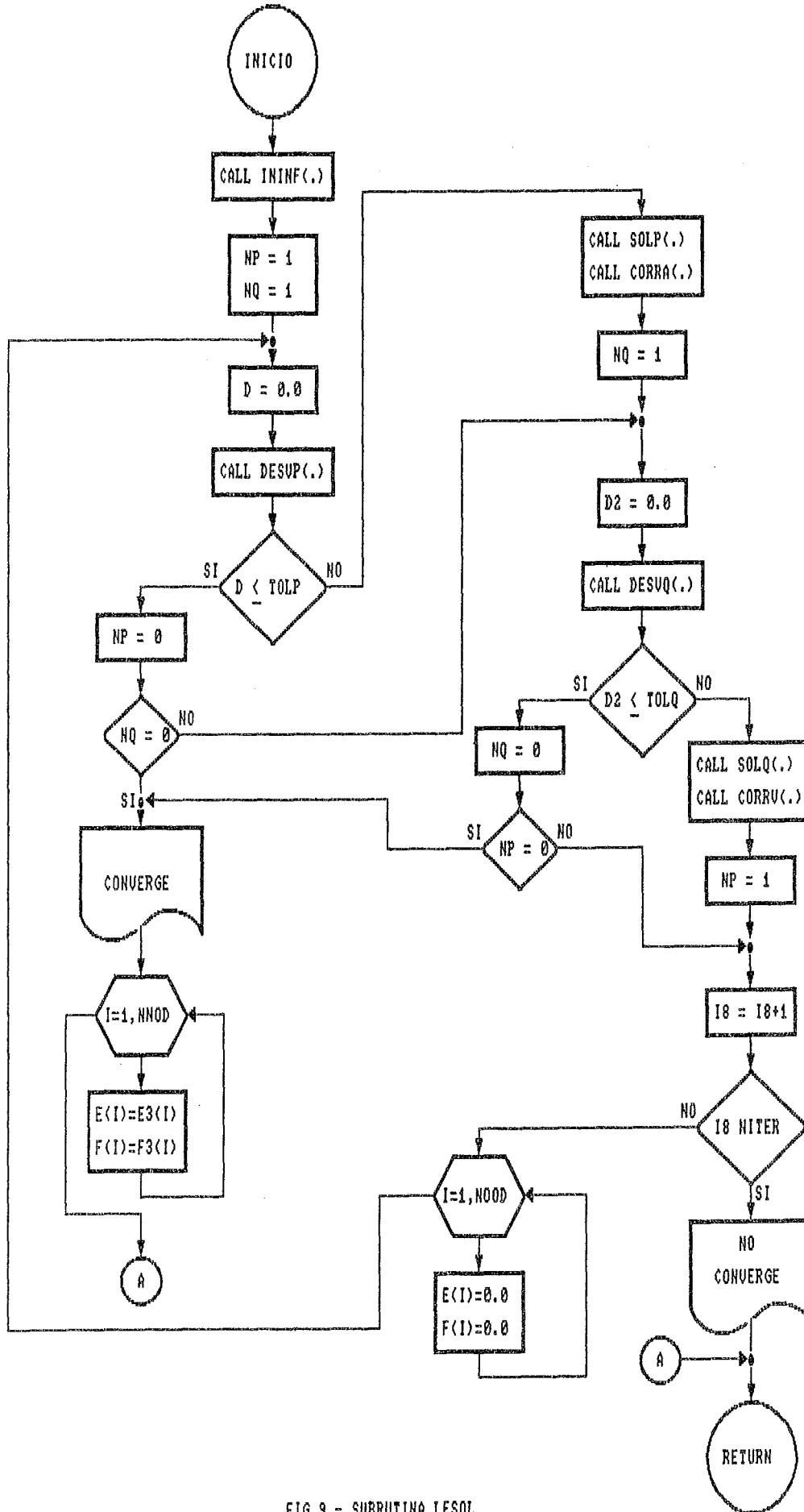


FIG. 9.- SUBROUTINA LFSOL

2.1 Subrutina YBUSC

Como se mencionó en el punto anterior, antes de ejecutar la rutina LFSOL es necesario formar la matriz de admitancias nodal ejecutando la instrucción:

```
CALL YBUSC (NNOD, NLIN, NTR, NSHU, NL2, M, N, YR, YI, BI, NE, NR,
           R, X, B, NCT, NST, T1, NSH, VLS)
```

La subrutina YBUSC forma la matriz de admitancias nodal de la red analizada en listas lineales de adyacencia.

Los argumentos de la subrutina son:

Variable	Entrada ó Salida	Tipo	Descripción
NNOD	E	Entera	Número de nodos en la red.
NLIN	E	Entera	Número de líneas en la red.
NTR	E	Entera	Número de bancos de transformación.
NSHU	E	Entera	Número de elementos compensadores en derivación.
NL2	E	Entera	$2 * NLIN + NNOD$
M(NNOD)	S	Entera	Vector de primeras posiciones en listas de la matriz de admitancias.

Variable	Entrada ó Salida	Tipo	Descripción
N(NL2)	S	Entera	Lista de columnas de la matriz de admitancias nodal. N(M(I)) contiene el número de conexiones adyacentes al nodo I.
YR(NL2)	S	Real * 8	Parte real de la matriz de admitancias nodal. YR(M(I)) contiene la parte real de Yii.
YI(NL2)	S	Real * 8	Parte imaginaria de la matriz de admitancias nodal YI(M(I)) la parte imaginaria de Yii.
BI(NL2)	S	Real * 8	Contiene la rama II de cada rama que genera un elemento no diagonal en la matriz de admitancias nodal.
NE(NLIN)	E	Entera	Lista de nodos de envío.
NR(NLIN)	E	Entera	Lista de nodos de recepción.
R(NLIN)	E	Real * 8	Lista de resistencias de cada línea o banco.

Variable	Entrada ó Salida	Tipo	Descripción
X(NLIN)	E	Real * 8	Lista de reactancias de cada línea o banco.
B(NLIN)	E	Real * 8	Lista de suceptancias en derivación de una rama del circuito II de cada línea de transmisión.
NCT(NTR)	E	Entera	Lista de nodos con cambiador de derivaciones de bancos.
NST(NTR)	E	Entera	Lista de nodos sin cambiador de derivaciones de bancos.
T1(NTR)	E	Real * 8	Lista de relaciones de transformación de cada banco.
NSH(NSHU)	E	Entera	Lista de nodos con compensadores en derivación.
VLS(NSHU)	E	Real * 8	Lista de suceptancias en derivación en nodos compensados (pu).

La Figura 10 muestra un diagrama de flujo de la subrutina YBUSC

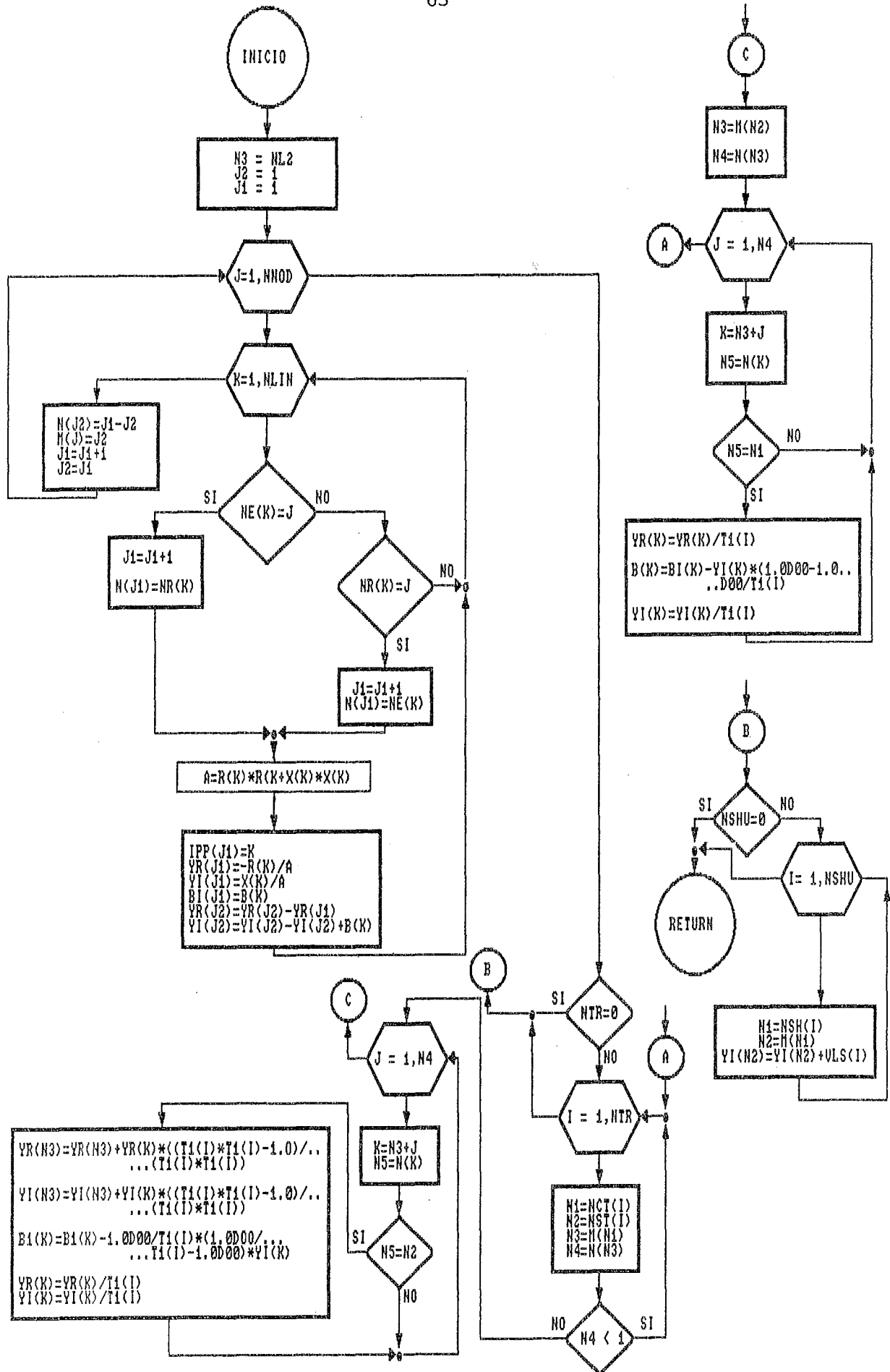


FIG.10.- SUBROUTINA YBUSC

2.2 Subrutina ININF

La Subrutina ININF realiza algunas actividades de inicialización de variables; primeramente calcula las potencias netas inyectadas en cada nodo , inicializa los voltajes de nodos generadores y forma un vector (NORD1(.)) en que se coloca una lista de nodos en el orden en que son eliminados.

Esta rutina es ejecutada usando la instrucción :

```
CALL ININF (NNOD, NTUG, BASE, NORD, NORD1, PG, VG, NU, E3, P2,
           Q2, PC, QC)
```

Los argumentos de la subrutina

Variable	Entrada ó Salida	Tipo	Descripción
NNOD	E	Entera	Número de nodos en la red.
NTUG	E	Entera	Número de nodos generadores.
BASE	E	Real * 8	Base de potencia usada en cantidades por unidad.
NORD(NNOD)	E	Entera	Vector con el orden de eliminación de nodos.
NORD1(NNOD)	S	Entera	Vector con nodos en orden progresivo de eliminación.

Variable	Entrada ó Salida	Tipo	Descripción
PG(NTUG)	E	Real * 8	Potencias asignadas a nodos de generación (MW).
VG(NTUG)	E	Real * 8	Voltajes asignados en nodos de generación (pu)
NU(NTUG)	E	Entera	Lista de nodos con generación.
E3(NNOD)	E/S	Real * 8	Vector auxiliar con magnitudes de voltaje iniciales del estudio.
P2(NNOD)	E/S	Real * 8	Vector auxiliar con potencias netas reales por nodo (pu).
Q2(NNOD)	E/S	Real * 8	Vector auxiliar con potencias reactivas netas reales por nodo.
PC(NNOD)	E	Real * 8	Lista de potencias activas de carga por nodo (MW).
QC(NNOD)	E	Real * 8	Lista de potencias reactivas de carga por nodo (MVAR).

La Figura 11 muestra un diagrama de flujo de la subrutina ININF(.) .

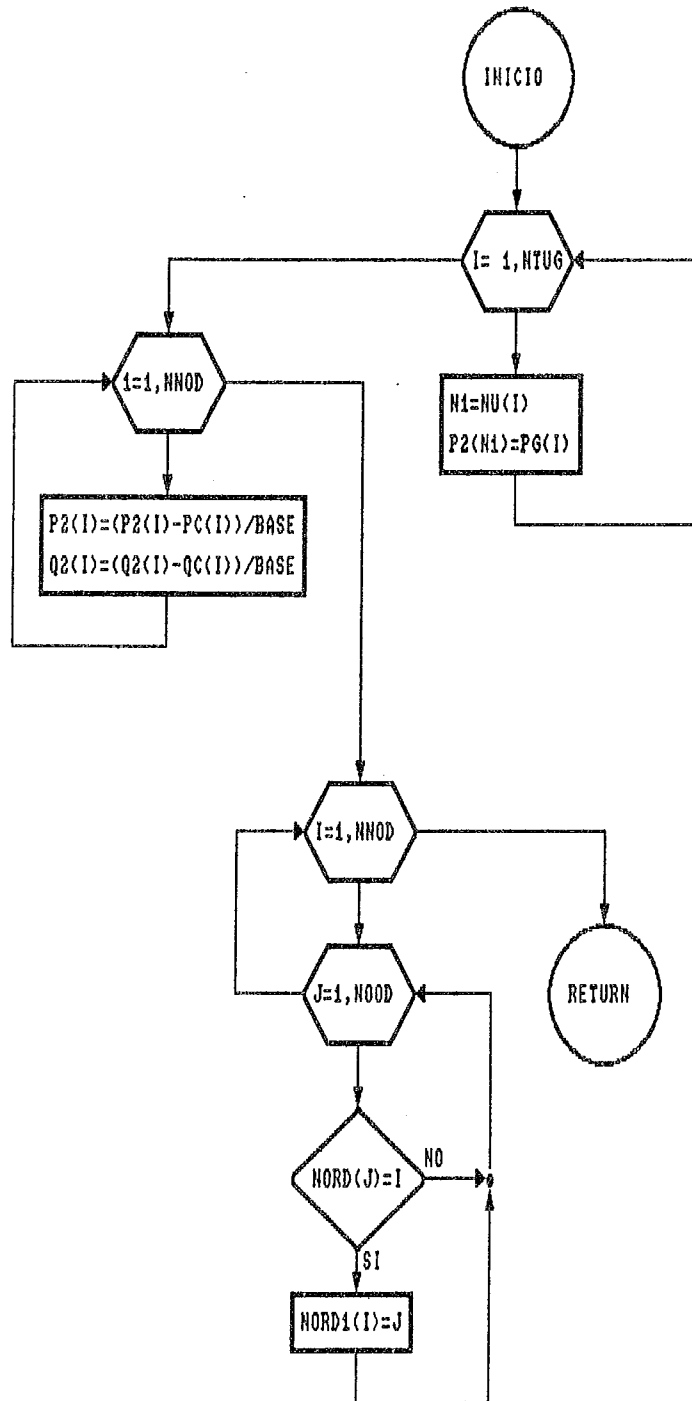


FIG.11.- SUBRUTINA ININF.

2.3 Subrutina DESVP

La Subrutina DESVP calcula las potencias reales en los nodos y las diferencias con los valores especificados para estas potencias. La Subrutina es ejecutada con la instrucción :

```
CALL DESVP(NNOD, NISLA, NL2, ISCOM, M, N, P2, F, E3, F3, YR,
           YI, D, L8)
```

Los argumentos de la subrutina DESVP son los siguientes :

Variable	Entrada ó Salida	Tipo	Descripción
NNOD	E	Entera	Número de nodos en la red.
NISLA	E	Entera	Número de islas eléctricas en la red.
NL2	E	Entera	$2 * NLIN + NNOD$.
ISCOM(NISLA)	E	Entera	Lista de compensadores para cada isla.
M(NNOD)	E	Entera	Vector de primeras posiciones en listas de adyacencias de la matriz de admitancias nodales.
N(NL2)	E	Entera	Lista de columnas en la matriz de admitancias nodal.

Variable	Entrada ó Salida	Tipo	Descripción
P2(NNOD)	E	Entera	Potencias reales netas inyectadas por nodo.
F(NNOD)	S	Real * 8	Vector de diferencias de potencias linealizadas ($\Delta P/V$).
E3(NNOD)	E	Real * 8	Vector de voltajes (defasamientos) por nodo en la iteración.
F3(NNOD)	E	Real * 8	Vector de voltajes (defasamientos) por nodo en la iteración actual.
YR(NL2)	E	Real * 8	Parte real de la matriz de admitancias nodal.
YI(NL2)	E	Real * 8	Parte imaginaria de la matriz de admitancias nodal.
D	S	Real * 8	Máxima desviación de potencial real (pu) en un nodo.
L8	S	Entera	Nodo con máxima desviación de potencia.

Esta subrutina llama a la subrutina IC (.).

La Figura 12 muestra un diagrama de flujo de la subrutina DESVP.

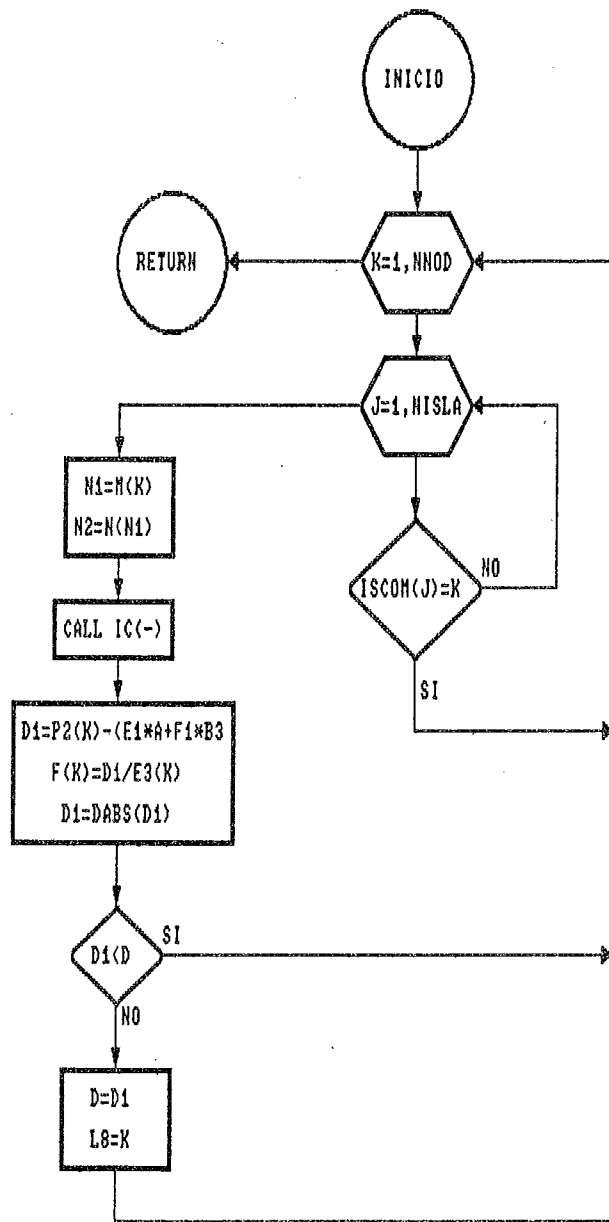


FIG.12.- SUBROUTINA DESUP.

2.4 Subrutina SOLP

La subrutina SOLP resuelve el modelo de potencia real del método desacoplado, conociendo los factores triangulares de dicho modelo y mediante un proceso de sustitución progresiva y regresiva.

La subrutina es ejecutada utilizando la instrucción:

```
CALL SOLP(NNOD, NACT, NISLA, NISA, IFIL, NB1, NB1C, NORD1, F, B1)
```

Los argumentos de la subrutina son los siguientes:

Variable	Entrada ó Salida	Tipo	Descripción
NNOD	E	Entera	Número de nodos en la red.
NACT	E	Entera	Orden del modelo de potencia real.
NISLA	E	Entera	Número de islas eléctricas .
NISA	E	Entera	Número de islas eléctricas activas.
IFIL	E	Entera	Dimensión máxima de los arreglos factorizados del modelo de potencia real.

Variable	Entrada ó Salida	Tipo	Descripción
NB1(NNOD)	E	Entera	Vector de primeras posiciones en los arreglos factorizados.
NB1C(IFIL)	E	Entera	Vector de columnas en los arreglos triangularizados.
NORD1(NNOD)	E	Entera	Lista de nodos en orden de terminación.
F(NNOD)	E/S	Real * 8	En este nodo se introducen los valores ($\Delta P/V$) y se obtienen los defasamientos de los voltajes nodales ($\Delta \delta$).
B1(IFIL)	E	Real * 8	Modelo de potencia real triangulizado.

La subrutina 13 muestra un diagrama de flujo de la subrutina SOLP.

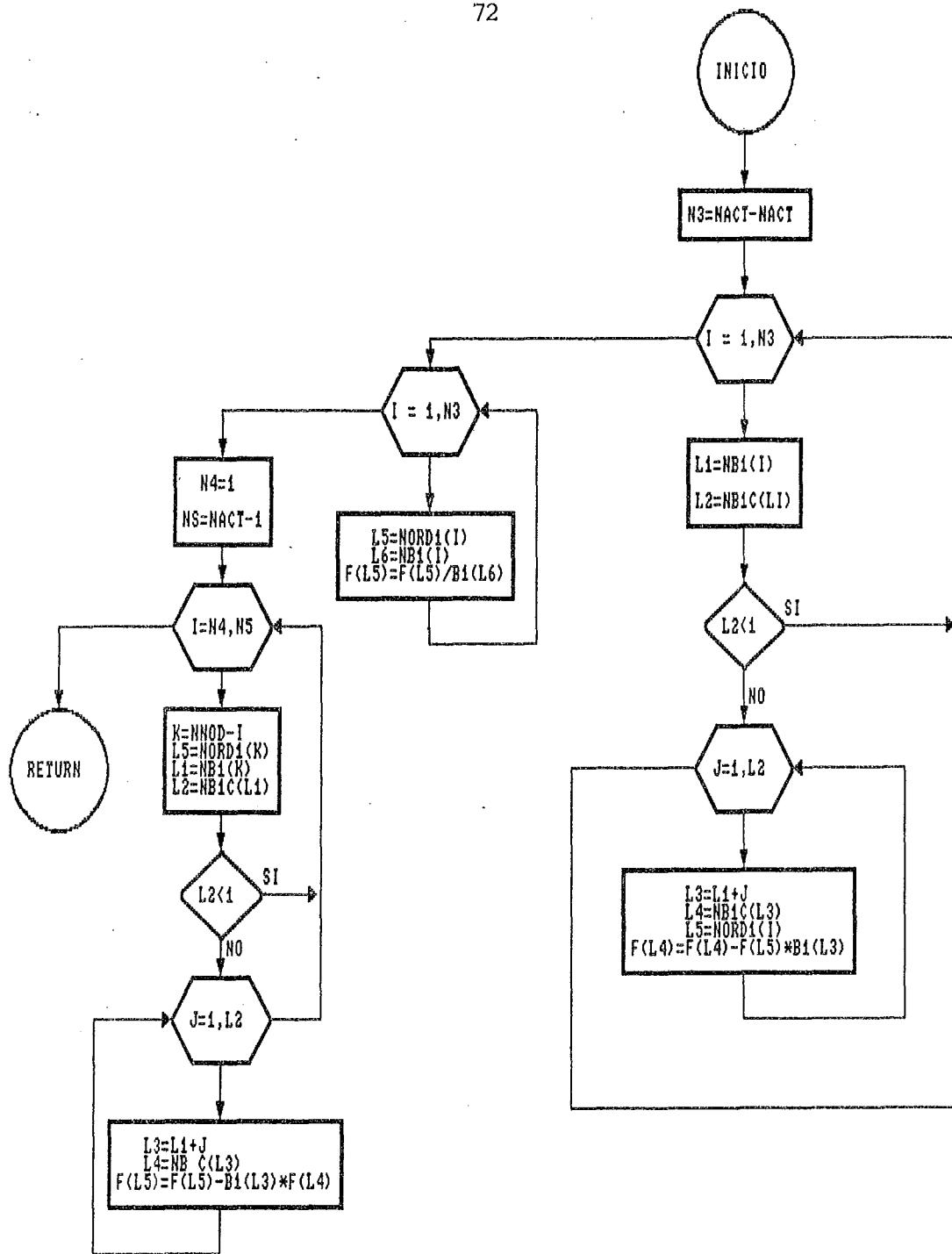


FIG.13.- SUBROUTINA SOLP.

2.5 Subrutina DESVQ

La subrutina DESVQ calcula las potencias reactivas inyectadas en cada nodo y las diferencias con los valores especializados para estas potencias. Además forma el vector ($\Delta Q/V$) y proporciona el nodo en que se tiene la máxima desviación en potencia reactiva y el valor de esta desviación.

La subrutina DESVQ se ejecuta mediante la instrucción :

```
CALL DESVQ (NNOD, NTUG, NL2, NU, NORD, M, N, Q2,
           E, E3, F3, YI, D2, L9)
```

Los argumentos de esta subrutina se describen a continuación :

Variable	Entrada ó Salida	Tipo	Descripción
NNOD	E	Entera	Número de nodos en la red.
NTUG	E	Entera	Número total de generadores.
NL2	E	Entera	$2 * NLIN + NNOD$
NU(NTUG)	E	Entera	Lista de nodos generadores.

Variable	Entrada ó Salida	Tipo	Descripción
NORD(NNOD)	E	Entera	Lista de nuevo orden de eliminación nodal.
M(NNOD)	E	Entera	Vector de primeras posiciones en la matriz de admitancias nodal.
N(NL2)	E	Entera	Lista de columnas para la matriz de admitancias nodal.
Q2(NNOD)	E	Real * 8	Vector auxiliar con las potencias reactivas netas inyectadas por nodo .
E(NNOD)	S	Real * 8	Vector auxiliar con los valores $\Delta Q/V$ por nodo.
E3(NNOD)	E	Real * 8	Vector de magnitudes de voltaje por nodo (pu) en la iteración.
F3(NNOD)	E	Real * 8	Vector de defasamientos de voltaje por nodo (en radianes).
YR(NL2)	E	Real * 8	Parte real de la matriz de admitancias nodal.
YI(NL2)	S	Real * 8	Parte imaginaria de la matriz de admitancias nodal.
D2	S	Real * 8	Máxima desviación en potencia reactiva (pu)
L9	S	Real * 8	Nodo con máxima desviación de potencia reactiva.

La Subrutina DESVQ utiliza la subrutina IC(.).

La Figura 14 muestra un diagrama de flujo de la subrutina DESVQ.

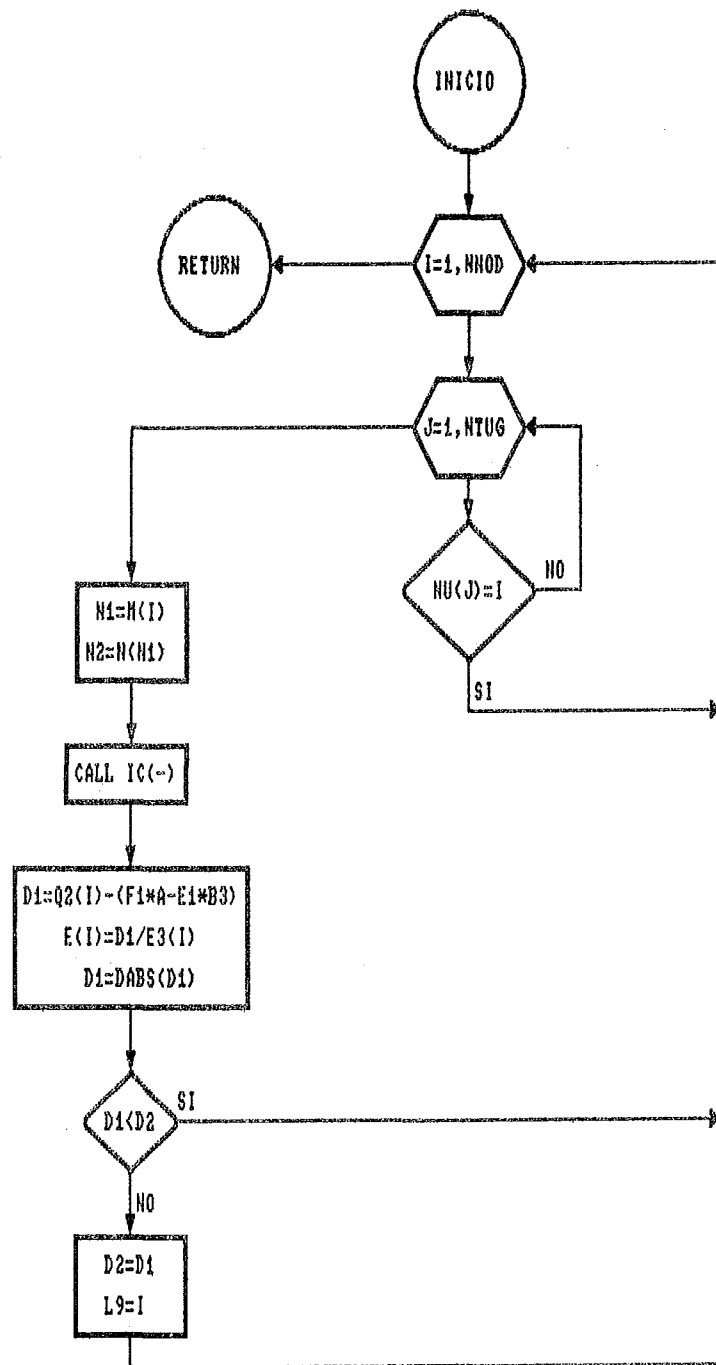


FIG.14.- SUBROUTINA DESUQ.

2.6 Subrutina SOLQ

La Subrutina SOLQ resuelve el modelo de potencia reactiva del método desacoplado mediante un proceso de sustitución progresiva y regresiva conociendo los factores triangulares de dicho modelo.

La subrutina es ejecutada mediante la instrucción:

CALL SOLQ (NNOD, NACT, NISLA, NISA, IFIL, NBI, NB1C, NORD1, E, B2)

Los argumentos de la subrutina son los siguientes :

Variable	Entrada ó Salida	Tipo	Descripción
NNOD	E	Entera	Número de nodos en la red.
NACT	E	Entera	Orden del modelo de potencia real
NISLA	E	Entera	Número de islas eléctricas.
NISA	E	Entera	Número de islas eléctricas activas

Variable	Entrada ó Salida	Tipo	Descripción
IFIL	E	Entera	Dimensión máxima de los arreglos factorizados del modelo de potencia real.
NB1(NNOD)	E	Entera	Vector de primeras posiciones en los arreglos factorizados.
NB1C(NNOD)	E	Entera	Vector de columna en los arreglos triangularizados.
NORD1(NNOD)	E	Entera	Lista de nodos en orden de eliminación.
E(NNOD)	E/S	Real * 8	Variable auxiliar, se entra con los valores $\Delta Q/V$ por nodo y sale con la solución de magnitudes de voltaje (ΔV).
B2(IFIL)	E	Real * 8	Modelo triangulizado de potencia reactiva.

La Figura 15 muestra un diagrama de bloques de la Subrutina SOLQ.

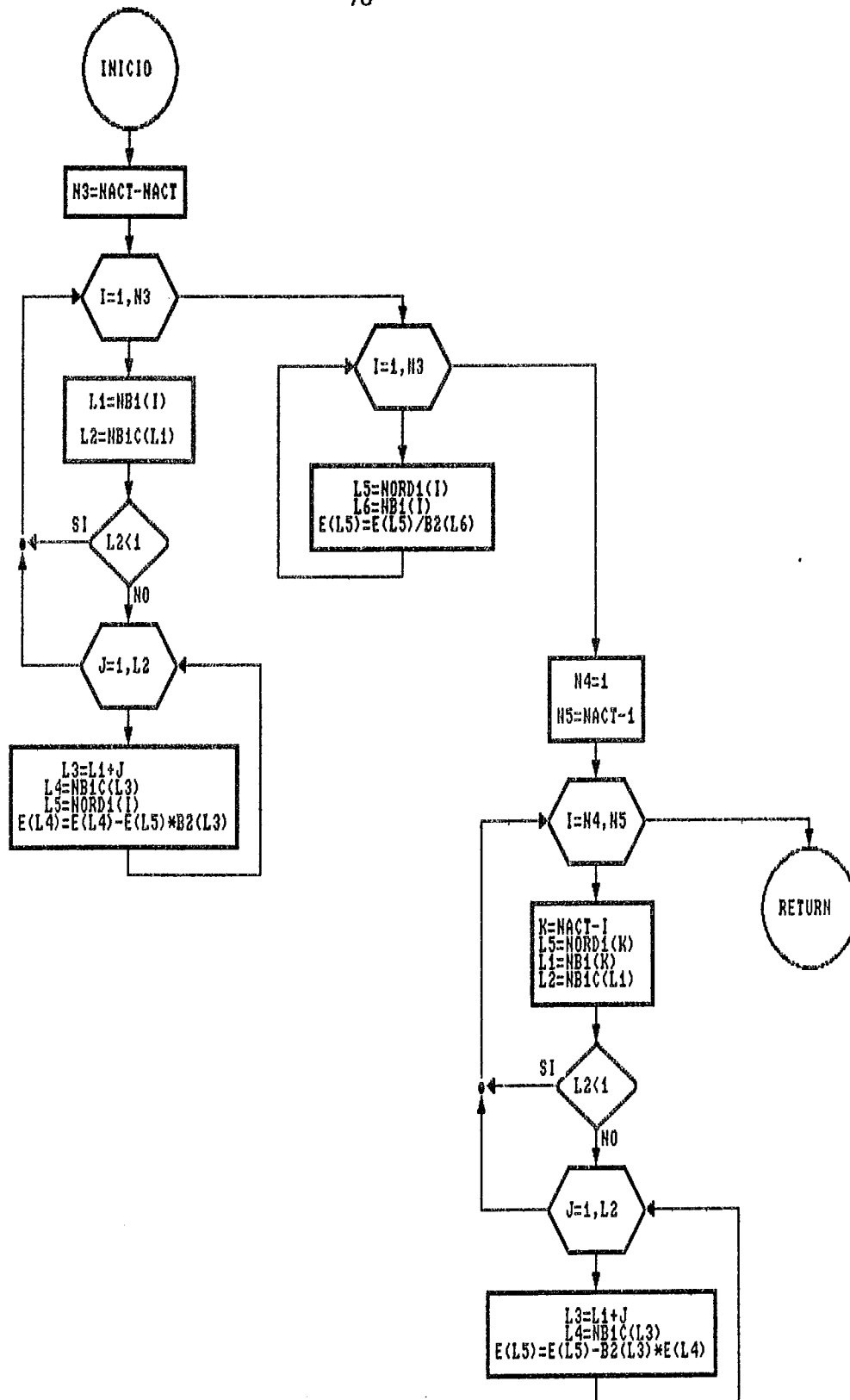


FIG. 15.- SUBROUTINA SOLQ.

2.7 Subrutina IC

La Subrutina IC calcula la corriente inyectada en un nodo de coordenadas cartesianas y es una rutina importante para calcular potencias nodales. La Subrutina se ejecuta mediante la instrucción:

```
CALL IC (K, N1, N2, A, B3, NL2, E3, F3, YR, Y1, NNOD, N)
```

Los argumentos de la subrutina IC se describen a continuación:

Variable	Entrada ó Salida	Tipo	Descripción
K	E	Entera	Nodo en el que se calcula la corriente inyectada.
N1	E	Entera	Apuntador a la primera posición de la lista de adyacencia del nodo K.
N2	E	Entera	Número de elementos conectados al nodo K.
A	S	Real * 8	Parte real de la corriente inyectada al nodo K.
B3	S	Real * 8	Parte reactiva de la corriente inyectada a un nodo K.

Variable	Entrada ó Salida	Tipo	Descripción
E1	S	Real * 8	Variable auxiliar con la parte real del voltaje en el nodo K.
F1	S	Real * 8	Variable auxiliar con la parte imaginaria del voltaje en el nodo K.
NL2	E	Entera	$2 * NLIN + NNOD$
E3(NNOD)	E	Real * 8	Magnitudes de voltaje nodales.
F3(NNOD)	E	Real * 8	defasamientos de voltajes nodales.
YR(NL2)	E	Real * 8	Parte real de la matriz de admitancias nodal.
YI(NL2)	E	Real * 8	Parte imaginaria de la matriz de admitancias nodal.
NNOD	E	Entera	Número de nodos en la red.
N(NL2)	E	Entera	Listas de columnas de la matriz de admitancias nodal.

La Figura 16 muestra un diagrama de flujo de Subrutina IC.

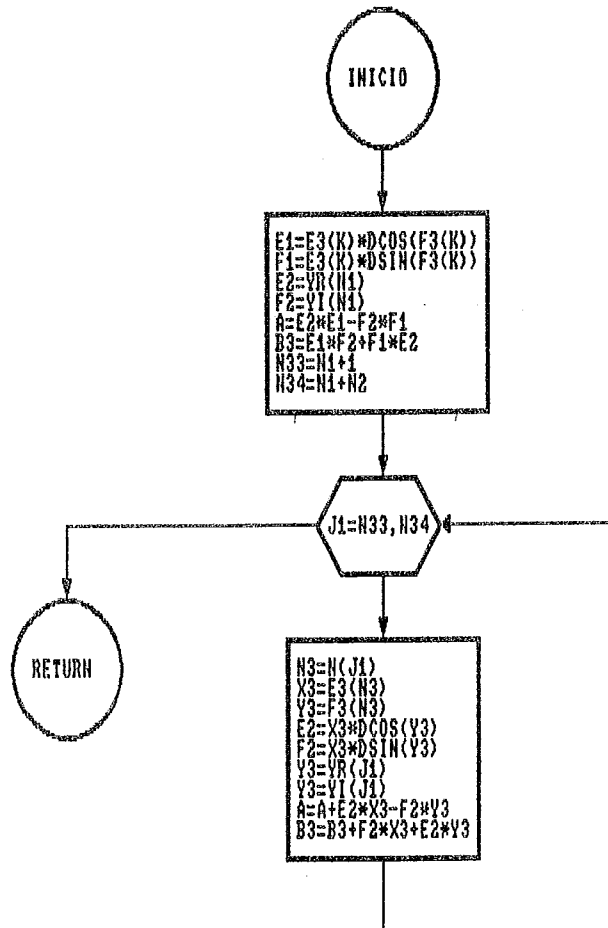


FIG.16.- SUBRUTINA IC.

2.8 Subrutina CORRA

La Subrutina CORRA actualiza ó corrige los defasamientos de los voltajes nodales.

La subrutina se ejecuta utilizando la instrucción;

CALL CORRA(NNOD, NISLA, ISCOM, F, F3)

Variable	Entrada ó Salida	Tipo	Descripción
NNOD	E	Entera	Número de nodos de la red.
NISLA	E	Entera	Número de islas eléctricas en la red.
ISCOM(NISLA)	E	Entera	Lista de nodos compensadores.
NORD(NNOD)	E	Entera	Lista de orden de eliminación nodal
F(NNOD)	E	Real * 8	Lista de correcciones en defasamientos de voltajes nodales.
F3(NNOD)	E/S	Real * 8	Defasamientos de voltajes nodales corregidos.

La Figura 17 muestra un diagrama de flujo de la Subrutina CORRA.

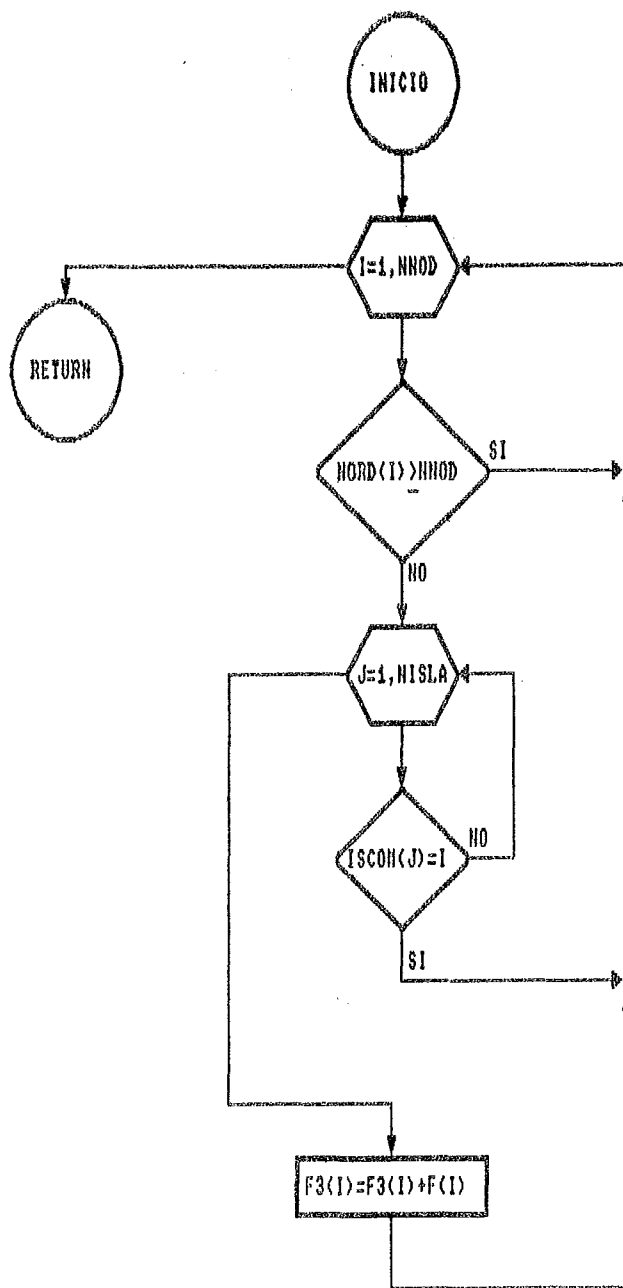


FIG.17.- SUBROUTINA CORRA.

2.9 Subrutina CORRV

La Subrutina CORRV actualiza ó corrige las magnitudes de voltajes nodales después de resolver el modelo de potencia reactiva.

La subrutina se ejecuta utilizando la instrucción :

```
CALL CORRV(NNOD, NTUG, NU, NORD, E3, E)
```

Los argumentos de la subrutina CORRV son los siguientes:

Variable	Entrada ó Salida	Tipo	Descripción
NNOD	E	Entera	Número de nodos en la red.
NTUG	E	Entera	Número de generadores en la red.
NU(NTUG)	E	Entera	Lista de nodos generadores.
NORD(NNOD)	E	Entera	Lista de orden de eliminación nodal
E3(NNOD)	E/S	Real * 8	Magnitudes de voltajes nodales actualizadas.
E(NNOD)	E	Real * 8	Correcciones a magnitudes de voltajes nodales.

La Figura 18 muestra un diagrama de flujo de la Subrutina CORRV.

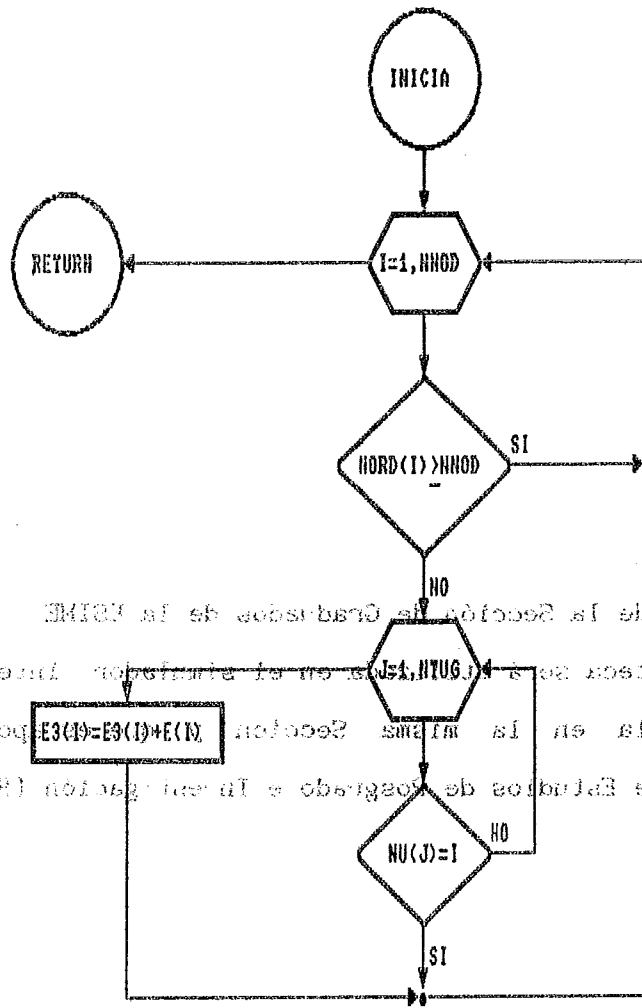


FIG.18.- SUBROUTINA CORR.U.

3. CONCLUSIONES

Se ha presentado un conjunto de subrutinas básicas para resolver estudios de flujos utilizando el método desacoplado rápido. Los estudios de flujos tienen que ser resueltos en una gran variedad de problemas de sistemas eléctricos, tales como en estudios de fallas (cálculo de condiciones de prefalla), estudios de estabilidad (condiciones de predisturbio), análisis de seguridad, despacho económico, etc. Las rutinas aquí descritas se han integrado a la biblioteca FLUJOS disponible en el sistema VAX-11-730 de la Sección de Graduados de la ESIME.

Esta biblioteca será utilizada en el simulador interactivo que se desarrolla en la misma Sección y que es apoyado por la Dirección de Estudios de Posgrado e Investigación (Reg. DEPI - 871443).

4. BIBLIOGRAFIA

- (1) R. Mota-Palomino, "Electric Energy Systems:
Computer Methods for the Solution of Decomposed Power Flows
and Optimal Linear Power Flows ", Ph. D. Thesis, University
of Waterloo, Waterloo, Ontario, 1984.
- (2) R. Mota-Palomino, "Técnicas Computacionales Eficientes para el
Análisis de Sistemas Eléctricos de Potencia ", Tesis de
Maestría, Sección de Graduados ESIME, IPN, 1979.
- (3) B. Stott, O.A. Alsac, "Fast Decoupled Load Flow", IEEE,
Trans. on PAS, Vol. PAS. 93, pp. 859-867, Junio, 1974.

II. CODIGO FUENTE

```

C*****
C****
C**** CEPFL1 ES UN PROGRAMA PARA REALIZAR ESTUDIOS DE FLUJOS
C**** DE CARGA POR EL METODO DESACOPLADO RAPIDO EN MODO BATCH
C**** R. MOTA-PALOMINO 890421
C*****
C
C      INTEGER NE(600),NR(600),NCT(250),NST(250),
2      NU(150),NAU(150),NSH(75),ISLA(500),NORD(500),
3      ISCOM(10),M(500),N(1900),ISP(1900),IPP(1900),NB1(500),NB1C(1200)
C      INTEGER LP(600),NLIS(500),NORD1(500),NM(50)
C
C
C      REAL R(600),X(600),B(600),PC(500),QC(500),T1(250),PG(150),
2      VG(150),QMA(150),QMI(150),VLS(75),QG(150)
C      REAL YR(1700),YI(1700),E(500),F(500),E3(500),F3(500),P2(500),
2      Q2(500),BI(1700),B1(1700),B2(1200),PGIS(10),PCIS(10),PSLA(10)
C      REAL FF,PL(50),QL(50),T2(50),BASE,D,D2
C
C
C      CHARACTER*20 ARCH
C      CHARACTER*70 LETR1
C      CHARACTER*70 LETR2
C      CHARACTER*8 NOMB1
C      CHARACTER*8 NOMB2
C      CHARACTER*8 IFIN
C      CHARACTER*8 DNOMB(500)
C      LOGICAL IEX,IERR
C
C
C*****
C
C
C      LECTU = 5
C      LTERM = 6
C      IFIN = '0'
C
C
C
C      LFFLU = 9
C      NNOD = 0
C      NPRO = 1
C      NLIN = 0
C      NTUG = 0
C      NTR = 0
C      NSHU = 0
C      ITIP = 1
C      IERR = .TRUE.
C      CALL SCREEN
C
C
C **** NNOD ES EL NUMERO TOTAL DE NODOS
C **** NLIN ES EL NUMERO TOTAL DE LINEAS

```

```

C **** NTUG ES EL NUMERO TOTAL DE NODOS DE GENERACION
C **** NTR ES NUMERO TOTAL DE TAPS FUERA DEL NOMINAL EN TRANSFORMADORES
C **** NSHU NUMERO TOTAL DE REACTORES EN PARALELO
C
C
C ***** INICIA LECTURA DE DATOS
      CALL LECTUR(ARCH, DNOMB, LETR1, LETR2, NOMB1, NOMB2, IFIN, NE, NR,
      2 NCT, NST, NU, NAU, NSH, R, X, B, PC, QC, T1, PG, VG, QMA, QMI, VLS, QG,
      3 NNOD, NLIN, NTUG, NTR, NSHU)
C*****
C*****
C***** SE INICIA FORMACION DE MODELOS
C*****
C*****
C
      CALL ENCA
      NL2 = 2*NLIN + NNOD
      IEXT = 1900
C
      CALL CLEAR
      CALL GRAPH(NNOD, NLIN, NL2, NE, NR, M, N, IPP, ISP)
      CALL PARTS(NNOD, NL2, M, N, ISP, ISLA, NLIS, NISLA)
      CALL ACTIV(NNOD, NISLA, NISA, NTUG, ISLA, NU, PG, PC, ISCOM, PGIS, PCIS,
      2 PSLA)
      CALL PREPA(NNOD, NLIN, NL2, IEXT, NISLA, M, N, ISP, IPP, ISLA, ISCOM, LP,
      2 NORD, NACT, IDISP)
C
C
C      WRITE(6, 90)NISLA
C 90 FORMAT(3X, ' ISLAS : ', I5)
C      DO 91 K = 1, NNOD
C          IF(ISLA(K).LT.2)GO TO 91
C          WRITE(6, 95)K, DNOMB(K), ISLA(K)
C 95      FORMAT(/3X, I5, 2X, A8, 2X, I5)
C 91 CONTINUE
      CALL ORDER(NISLA, NNOD, NLIN, NL2, IEXT, NACT, M, N, ISP, IPP, NORD, IDISP)
      IFIL = NL2
      CALL FORMA(NNOD, NACT, NLIN, NL2, NISLA, NISA, IEXT, IFIL, NTUG, NSHU, NTR,
      2 LP, M, N, NB1, NB1C, B1, B2, IPP, R, X, B, ISP, NORD, NCT, NST, T1, NSH, VLS, NU,
      3 ISCOM, NE, NR, NLIS)
      CALL FACTA(NNOD, NACT, IFIL, NISLA, NISA, NB1, NB1C, NORD, B1, B2)
      WRITE(6, 10)
      10 FORMAT(/3X, 'SE PROCESO CORRECTAMENTE TODA LA INFORMACION ')
C*****
C*****
C***** SE DESARROLLA EL ALGORITMO DESACOPADO RAPIDO
C*****
C*****
C
      CALL ENCA
      CALL PARCOR(LECTU, LTERM, TOLP, TOLQ, TINIA, NITER, BASE)
      CALL CLEAR

```

```

DO 250 J = 1, NNOD
  E3(J) = 1.000
  F3(J) = 0.000
  E(J) = 0.000
  F(J) = 0.000
  P2(J) = 0.000
  Q2(J) = 0.000
250 CONTINUE
  CALL YBUSC(NNOD, NLIN, NTR, NSHU, NL2, M, N, YR, YI, BI, NE, NR,
2 R, X, B, NCT, NST, T1, NSH, VLS, IPP)
  CALL LFSOL3(NNOD, NACT, NTUG, NISLA, NISA, NL2, IFIL, ISCOM, BASE, NORD,
2 NORD1, PG, VG, NU, E3, F3, P2, Q2, PC, QC, YR, YI, NB1, NB1C, B1, B2, E, F,
3 NITER, TOLP, TOLQ, M, N, DNOMB, TINIA, QMA, QMI)
  CALL ENCA

C
C SE IMPRIMEN RESULTADOS
C
DO 200 I = 1, NNOD
  FF = F(I)
  E3(I) = E(I)*COS(FF)
  F3(I) = E(I)*SIN(FF)
200 CONTINUE
358 CONTINUE
  CALL CLEAR
  WRITE(LTERM, 300)
300 FORMAT(/10X, ' REPORTE DE FLUJOS ', 24X, ' 1 ')
  WRITE(LTERM, 310)
310 FORMAT(/10X, ' REPORTE DE PARAMETROS ', 20X, ' 2 ')
  WRITE(LTERM, 320)
320 FORMAT(/10X, ' REPORTES RESUMIDOS ', 23X, ' 3 ')
  WRITE(LTERM, 330)
330 FORMAT(/10X, ' NO MAS REPORTES ', 26X, ' 4 ')
  WRITE(LTERM, 350)
350 FORMAT('// ' QUE TIPO DE REPORTE DESEA : '$)
  READ(LTERM, 355, ERR=356) IREP
355 FORMAT(I2)
  IF(IREP.LT.1.OR.IREP.GT.4)GO TO 356
  IF(IREP.EQ.1)GO TO 360
  IF(IREP.EQ.2)GO TO 370
  IF(IREP.EQ.3)GO TO 380
  IF(IREP.EQ.4)GO TO 3000
356 CONTINUE
  WRITE(LTERM, 357)
357 FORMAT(/10X, ' TIPOS DISPONIBLES DE 1 A 4, FORMATO I2')
  GO TO 358
360 CONTINUE
  CALL REPFLU(LETR1, LETR2, NNOD, NL2, NTUG, NTR, NSHU, M, N, YR, YI, BI,
2 NCT, NST, T1, NSH, VLS, BASE, E, F, E3, F3, DNOMB, PG, QG, PC, QC, NU, QMA, QMI,
3 NM, PL, QL, T2, ISLA, LTERM)
  GO TO 358
370 CONTINUE
  CALL REPDAT(LETR1, LETR2, NNOD, NL2, NTUG, NTR, NSHU, M, N, YR, YI, BI,
2 NCT, NST, T1, NSH, VLS, BASE, E, F, E3, F3, DNOMB, PG, QG, PC, QC,
3 NU, QMA, QMI, NM, PL, QL, T2, ISLA, LTERM, IPP, R, X, B, NLIN)

```

GO TO 358
380 CONTINUE
CALL REPRES(LTERM, NNOD, NACT, NLIN, NTUG, NISLA, NISA, ISCOM, R, X, B,
2DNOMB, NU, M, N, PG, QG, QMA, QMI, BASE, PC, QC, ISLA, E, F, E3, F3,
3NE, NR, IPP, YR, YI, NL2, NSHU, NSH, VLS)
GO TO 358

C*****

3000 CONTINUE

C

CALL CLEAR
STOP
END

C*****

C

C\$INCLUDE SCREEN.FOR
C\$INCLUDE LECTUR.FOR
C\$INCLUDE PARCOR.FOR
C\$INCLUDE FLUJOS.FOR
C\$INCLUDE REPFLU.FOR

```
SUBROUTINE SCREEN
CHARACTER*1 ESC
CHARACTER*3 HOME, OFF, INV, BOL
PARAMETER (HOME='[2J', OFF='[Om', BOL='[1m')
ESC=CHAR(27)
WRITE(6,100)ESC,HOME
100 FORMAT(20A)
WRITE(6,105)
105 FORMAT(/////10X,' ESTUDIOS DE FLUJOS ',
2/////,12X,' METODO DESACOPLADO RAPIDO **')
WRITE(6,110)
110 FORMAT(/////////,16X,'** @ R. Mota-Palomino
21989 **')
RETURN
END
```

```

C*****
C****
C****   LECTUR ES UN PROGRAMA DE LECTURA DE DATOS PARA
C****   ESTUDIOS DE FLUJOS DE POTENCIA
C****   R. MOTA-PALOMINO   890420
C*****
C
C      SUBROUTINE LECTUR(ARCH, DNOMB, LETR1, LETR2, NOMB1, NOMB2, IFIN,
2      NE, NR, NCT, NST, NU, NAU, NSH, R, X, B, PC, QC, T1, PG, VG, QMA, QMI, VLS, QG,
3      NNOD, NLIN, NTUG, NTR, NSHU)
      INTEGER NE(1), NR(1), IDES(70), NCT(1), NST(1),
2      NU(1), NAU(1), NSH(1)

C
C
C      REAL R(1), X(1), B(1), PC(1), QC(1), T1(1), PG(1),
2      VG(1), QMA(1), QMI(1), VLS(1), QG(1), BASE

C
C      CHARACTER*20 ARCH
      CHARACTER*70 LETR1
      CHARACTER*70 LETR2
      CHARACTER*8  NOMB1
      CHARACTER*8  NOMB2
      CHARACTER*8  IFIN
      CHARACTER*8  DNOMB(1)
      CHARACTER*8  IRES
      CHARACTER*8  IRE1
      LOGICAL IEX, IERR

C
C*****
C
C      LECTU = 5
      LTERM = 6
      IFIN = '0'
      BASE = 100.
      IWR = LTERM
      KT = 0

C
C      LFFLU = 9
      NNOD = 0
      NPRO = 1
      NLIN = 0
      NTUG = 0
      NTR = 0
      NSHU = 0
      ITIP = 1
      IERR = .TRUE.
C

```



```

C **** NNOD ES EL NUMERO TOTAL DE NODOS
C **** NLIN ES EL NUMERO TOTAL DE LINEAS
C **** NTUG ES EL NUMERO TOTAL DE NODOS DE GENERACION
C **** NTR ES NUMERO TOTAL DE TAPS FUERA DEL NOMINAL EN TRANSFORMADORES
C **** NSHU NUMERO TOTAL DE REACTORES EN PARALELO
C
C
C ***** INICIA LECTURA DE DATOS
C SE PREGUNTA POR ARCHIVO DE DATOS
C
201 CONTINUE
    WRITE(LTERM,500)
500 FORMAT(//,5X,' * NOMBRE DEL ARCHIVO DE DATOS: '$)
    READ(LECTU,501)ARCH
501 FORMAT(A20)
    CALL CLEAR
    WRITE(LTERM,502)ARCH
502 FORMAT(//3X,' ** EL ARCHIVO DE DATOS ES : ',A20)
    OPEN(UNIT=9, FILE=ARCH, STATUS='OLD', ERR=2000,
1 ACCESS='SEQUENTIAL', FORM='FORMATTED')
    GO TO 2002
2000 CONTINUE
    WRITE(LTERM,203)
203 FORMAT(//3X,' *** ERROR EN ARCHIVO DE ENTRADA ')
    GO TO 201
2002 CONTINUE
    WRITE(LTERM,800)
800 FORMAT(//3X,' ** DESEA LISTAR DATOS DE ENTRADA ? '$)
    READ(LTERM,801)NOMB1
801 FORMAT(A8)
    IF(NOMB1.EQ.'SI'.OR.NOMB1.EQ.'si')KT = 3
    IF(KT.EQ.0)GO TO 810
    WRITE(LTERM,811)
811 FORMAT(//3X,' ** DESEA LISTADO PARA IMPRESORA (IM) ? '$)
    READ(LTERM,801)NOMB2
    IF(NOMB2.EQ.'IM'.OR.NOMB2.EQ.'im')IWR = 8
    IF(IWR.EQ.8)OPEN(UNIT=8,FILE='DATOS.SAL')
810 CONTINUE
    IF(IWR.NE.8)CALL CLEAR
    LFFLU = 9
    READ(LFFLU,23,ERR=22)LETR1
    WRITE(6,23)LETR1
    IF(IWR.EQ.8)WRITE(IWR,23)LETR1
23 FORMAT(1X,A70)
    READ(LFFLU,23,ERR=22)LETR2
    WRITE(6,23)LETR2
    IF(IWR.EQ.8)WRITE(IWR,23)LETR2
    GO TO 24
22 WRITE(6,25)
25 FORMAT(//5X,'***** ERROR EN LA DESCRIPCION DEL CASO ')
    IERR=.FALSE.
    GO TO 3000

```

C
C
C
C
C

***** LECTURA DE INFORMACION NODAL

```

24 I=1
28 READ(LFFLU,27,ERR=26)NOMB1,PC(I),QC(I)
27 FORMAT(2X,A8,2X,F8.2,2X,F8.2)
  IF(KT.EQ.3)WRITE(IWR,27)NOMB1,PC(I),QC(I)
  IF(NOMB1.EQ.IFIN)GO TO 30
    DNOMB(I) = NOMB1
    NNOD = NNOD + 1
    I = I + 1
    GO TO 28
26 WRITE(6,29)
29 FORMAT(/5X,'*** ERROR EN LA INFORMACION NODAL')
  IERR = .FALSE.
  GO TO 3000
30 WRITE(6,31)NNOD
31 FORMAT(/5X,'SE LEE LA INFORMACION DE ',I6,' NODOS',/)

```

C
C
C

***** INICIO DE LECTURA DE GENERADORES

```

  IF(IWR.EQ.LTERM.AND.KT.EQ.3)CALL ENCA
  IF(IWR.EQ.LTERM.AND.KT.EQ.3)CALL CLEAR
  I = 1
32 READ(LFFLU,34,ERR=33)NOMB1,NOMB2,PG(I),QMA(I),QMI(I),
  2 VG(I)
34 FORMAT(2X,A8,2X,A8,2X,3(F8.2,2X),F7.5)
  IF(KT.EQ.3)WRITE(IWR,34)NOMB1,NOMB2,PG(I),QMA(I),QMI(I),VG(I)

```

C
C

```

  IF(NOMB1.EQ.IFIN)GO TO 36
    NTUG = NTUG + 1
    I = I + 1
    CALL VERIF(NNOD,I'TIP,JX,JY,I, DNOMB,NOMB1,NOMB2,IEX)
    NU(NTUG) = JX
    NAU(NTUG) = JY
    QMA(NTUG) = QMA(NTUG)/BASE
    QMI(NTUG) = QMI(NTUG)/BASE
    IF(.NOT.IEX)NTUG = NTUG - 1
    GO TO 32
33 WRITE(6,35)NTUG
35 FORMAT(/5X,'***** ERROR EN DATOS DE GENERACION',/5X,
  2 'EN LA LINEA NUMERO ',I6)
  IERR=.FALSE.
  GO TO 3000
36 WRITE(6,37)NTUG
37 FORMAT(/5X,'SE LEYO INFORMACION DE ',I6,
  2 ' NODOS DE GENERACION')

```

C
C
C

***** INICIO DE LA LECTURA DE DATOS DE LINEAS Y TRANSFORMADORES

C
C

```

IF(IWR.EQ.LTERM.AND.KT.EQ.3)CALL ENCA
IF(IWR.EQ.LTERM.AND.KT.EQ.3)CALL CLEAR
I=1
40 READ(LFFLU,42,ERR=43)NOMB1,NOMB2,R(I),X(I),B(I),T,ITR
42 FORMAT(2X,A8,2X,A8,2X,4(F7.5,2X),I1)
IF(KT.EQ.3)WRITE(IWR,42)NOMB1,NOMB2,R(I),X(I),B(I),T
IF(NOMB1.EQ.IFIN)GO TO 45
NLIN=NLIN+1
I=I+1
CALL VERIF(NNOD,ITIP,JX,JY,I,DNOMB,NOMB1,NOMB2,IEX)
IF(.NOT.IEX)NLIN=NLIN-1
IF(.NOT.IEX)GO TO 40
NE(NLIN)=JX
NR(NLIN)=JY
IF(T.LE.1.0E-8)GO TO 40
IF(NTR.EQ.0)GO TO 13
DO 11 K=1,NTR
IF(NCT(K).NE.JX)GO TO 11
IF(NST(K).NE.JY)GO TO 11
WRITE(6,12)DNOMB(JX),DNOMB(JY),T1(K)
12 FORMAT(5X,' ENTRE ',A8,' Y ',A8,' EXISTE TRANSF. ',/5X,
2 ' EN PARALELO CON TAP ',F12.6)
11 CONTINUE
13 NTR=NTR+1
NCT(NTR)=JX
NST(NTR)=JY
T1(NTR) = T
GO TO 40
43 WRITE(6,48)NLIN
48 FORMAT(/5X,' ERROR EN LOS DATOS DE LINEA ',/5X,
2 ' EN LA LINEA NUMERO ',I6)
IERR=.FALSE.
GOTO 3000
45 NL=NLIN-NTR
WRITE(6,49)NL,NTR
49 FORMAT(/5X,' LEIMOS LA INFORMACION DE ',I6,' LINEAS Y ',/27X,
2 I6,' TRANSFORMADORES ')

```

C
C
C
C
C

```

***** INICIO DE LA LECTURA DE ELEMENTOS 'SHUNT'

```

```

IF(IWR.EQ.LTERM.AND.KT.EQ.3)CALL ENCA
IF(IWR.EQ.LTERM.AND.KT.EQ.3)CALL CLEAR
I=1
ITIP=2
50 READ(LFFLU,52,ERR=53)NOMB1,VLS(I)
52 FORMAT(2X,A8,2X,F8.5)
IF(KT.EQ.3)WRITE(IWR,52)NOMB1,VLS(I)
IF(NOMB1.EQ.IFIN)GO TO 56

```

C
C
C

```

NSHU = NSHU + 1
I=I+1
CALL VERIF(NNOD, ITIP, JX, JY, I, DNOMB, NOMB1, NOMB2, IEX)
IF(IEX)NSH(NSHU)=JX
IF(.NOT. IEX)NSHU=NSHU-1
GO TO 50
53 WRITE(6,58)NSHU
58 FORMAT(/5X, '**** ERROR EN LOS DATOS DE SHUNTS ', /5X,
2' EN LA LINEA NUMERO ', I6)
IERR=.FALSE.
GO TO 3000
56 WRITE(6,59)NSHU
59 FORMAT(/5X, 'SE LEYO INFORMACION DE ', I6, ' SHUNTS '//)
C*****
3000 CONTINUE
IF(IWR. EQ. 8)CLOSE(UNIT=8)
C
RETURN
END
C*****
C****
C**** LA SUBROUTINA VERIF REvisa SI LOS NODOS LISTADOS EN CADA PARTE --
C**** DE DATOS ESTAN EN LA LISTA DE NODOS.
C****
C*****
C
SUBROUTINE VERIF(NNOD, ITIP, JX, JY, I, DNOMB, NOMB1, NOMB2, IEX)
C
C
CHARACTER*8 DNOMB(1), NOMB1, NOMB2, NOMB3, NOMB4
LOGICAL IEX
C
C
IEX=.TRUE.
NOMB3=NOMB1
NOMB4=NOMB2
C
C
DO 110 J=1, NNOD
IF(DNOMB(J).NE. NOMB1)GO TO 110
JX=J
GO TO 112
110 CONTINUE
WRITE(6, 113)NOMB1
113 FORMAT(/5X, 'EL NODO ', A8, 'NO EXISTE EN EL SISTEMA ')
I=I-1
IEX=.FALSE.
RETURN
112 IF(ITIP.EQ.2)GO TO 118

```

```
DO 116 J=1, NNOD
IF(DNOMB(J).NE.NOMB2)GO TO 116
JY=J
GO TO 118
116 CONTINUE
WRITE(6, 113)NOMB2
I=I-1
IEX=.FALSE.
118 RETURN
END
```

```

SUBROUTINE PARCOR(LECTU, LTERM, TOLP, TOLQ, TINIA, NITER, BASE)
  CHARACTER*1 ESC
  CHARACTER*3 HOME, BOL, OFF
  CHARACTER*8 NOMB1
  PARAMETER (HOME=' [2J', BOL=' [1m', OFF=' [0m')
  ESC=CHAR(27)
  WRITE(LTERM, 10)ESC, HOME
10  FORMAT(20A)
C
C*****      VALORES DE DEFAULT DE PARAMETROS DE CORRIDA DE FLUJOS
C
      TOLP = 0.001
      TOLQ = 0.001
      TINIA = 0.00001
      NITER = 60
      BASE = 100.0
C
C*****      SE INTRODUCEN VALORES DESEADOS
C
      WRITE(LTERM, 20)
20  FORMAT(///3X, ' ** TOLERANCIA EN P : '$)
      READ(LECTU, 21)TOLP
21  FORMAT(F8.5)
      IF(TOLP.GT.10.0 .OR. TOLP.LT..00001)TOLP=0.001
      WRITE(LTERM, 25)TOLP
25  FORMAT(3X, ' ** LA TOLERANCIA EN P ES : ',
2   F8.5)
      WRITE(LTERM, 30)
30  FORMAT(/3X, ' ** TOLERANCIA EN Q : '$)
      READ(LECTU, 21)TOLQ
      IF(TOLQ.GT.10.0 .OR. TOLQ.LT.0.00001)TOLQ=0.001
      WRITE(LTERM, 32)TOLQ
32  FORMAT(3X, ' ** LA TOLERANCIA EN Q ES : ',
2   F8.5)
      WRITE(LTERM, 600)
600  FORMAT(/3X, ' ** DESEA CORRIDA AJUSTADA ? '$)
      READ(LTERM, 608)NOMB1
608  FORMAT(A8)
      IF(NOMB1.NE.'SI'.AND.NOMB1.NE.'si')GO TO 610
      WRITE(LTERM, 40)
40  FORMAT(/3X, ' ** INICIO DE AJUSTES EN Qg : '$)
      READ(LECTU, 21)TINIA
      IF(TINIA.GT.1000.0 .OR. TINIA .LT. 0.00001)TINIA=0.1
      WRITE(LTERM, 45)TINIA
45  FORMAT(3X, ' ** LOS AJUSTES EN Qg SE INICIAN EN : ',
2   F8.5)
610  CONTINUE
      WRITE(LTERM, 50)
50  FORMAT(/3X, ' ** NUMERO MAXIMO DE ITERACIONES : '$)
      READ(LECTU, 51)NITER
51  FORMAT(I5)
      IF(NITER.GT.200.OR.NITER.LT.1)NITER=15

```

```
WRITE(LTERM,55)NITER
55  FORMAT(3X,' ** EL NUMERO MAXIMO DE ITERACIONES ES :
2  ',I5)
WRITE(LTERM,60)
60  FORMAT(//3X,' ** BASE DE POTENCIA UTILIZADA : '$)
READ(LTERM,62)BASE
62  FORMAT(F10.5)
IF(BASE.GT.501.0.OR.BASE.LT.2.0)BASE=100.0
WRITE(LTERM,65)BASE
65  FORMAT(3X,' ** LA BASE DE POTENCIA EMPLEADA : ',
2  F10.5)
CALL ENCA
RETURN
END
```

```

C*****
C*****
C***** LA SUBROUTINA GRAPH CONSTRUYE EL GRAFO DE LA MATRIZ DE ADMI-
C***** TANCIAS NODAL PARA EL SISTEMA ANALIZADO
C*****
C*****
C
C      SUBROUTINE GRAPH(NNOD, NLIN, NL2, NE, NR, M, N, IPP, ISP)
C
C      INTEGER NE(1), NR(1), M(1), N(1), IPP(1), ISP(1)
C
C      WRITE(6, 400)
400  FORMAT(/3X, ' SE FORMA GRAFO DE Ybus ...')
      J2 = 1
      J1 = 1
      DO 10 J=1, NNOD
          DO 12 K=1, NLIN
              IF(NE(K).NE.J)GO TO 20
              J1=J1+1
              N(J1)=NR(K)
              IPP(J1)=K
          20 IF(NR(K).NE.J)GO TO 12
              J1=J1+1
              N(J1)=NE(K)
              IPP(J1)=K
          12 CONTINUE
              N(J2)=J1-J2
              M(J)=J2
              J1=J1+1
          10 J2=J1
C
C      SE INICIALIZA VECTOR DE ENCADENAMIENTO
C
C      L1=1
      DO 13 I=1, NNOD
          J3=N(M(I))
          DO 14 J=1, J3
              INEXT=L1+1
              ISP(L1)=INEXT
          14 L1=INEXT
              ISP(L1)=0
          13 L1=L1+1
C
C
C      J2=J2-1
      WRITE(6, 16)J2
16  FORMAT(/3X, '** EL GRAFO DE YBUS TIENE : ', I6, ' RAMAS')
C
C
C      RETURN
      END
C

```



```

C   SUBROUTINA ACTIV VERIFICA SI HAY GENERACION Y CARGA EN CADA
C   ISLA ELECTRICA.  DEFINE COMPENSADOR POR ISLA
C
C   SUBROUTINE ACTIV(NNOD, NISLA, NISA, NTUG, ISLA, NU, PG, PC, ISCOM, PGIS,
174  2PCIS, PSLA)
C
C   INTEGER ISLA(1), ISCOM(1), NU(1)
C
C   REAL PGIS(1), PCIS(1), PSLA(1), PG(1), PC(1)
C
C   NISA = NISLA
C   DO 10 I=1, NISLA
C   PGIS(I)=0.0
175  10 PCIS(I)=0.0
C
C   SE SUMA GENERACION Y CARGA EN CADA ISLA
C
C   DO 11 I=1, NTUG
C   IP=NU(I)
C   IP=ISLA(IP)
176  11 PGIS(IP)=PGIS(IP)+PG(I)
C   DO 12 I=1, NNOD
C   IP=ISLA(I)
177  12 PCIS(IP)=PCIS(IP)+PC(I)
C
C   SE PONE BANDERA EN ISLAS SIN CARGA O GENERACION
C   ISLA SIN GENERACION SE CONSIDERA INACTIVA  8900701
C
C   DO 13 I=1, NISLA
C   IF(PGIS(I).LT.1.0E-6)GO TO 130
C   IF(PCIS(I).GT.1.0E-8)GO TO 13
178  130 CONTINUE
C   NISA = NISA - 1
C   WRITE(6,150)NISA
179  150 FORMAT(/2X,' SE AJUSTA EL NUMERO DE ISLAS ACTIVAS A',I5)
C   DO 14 J=1, NNOD
C   IF(ISLA(J).EQ.I)ISLA(J)=-ISLA(J)
180  14 CONTINUE
181  13 CONTINUE
C
C   SE SELECCIONA NODO COMPENSADOR PARA CADA ISLA
C   EL NODO GENERADOR CON MAYOR GENERACION ES ESCOGIDO
C
C   DO 18 J=1, NISLA
C   PSLA(J)=0.0
C   DO 17 I=1, NTUG
C   IP = NU(I)
C   IP1 = ISLA(IP)

```

```

      IF(IP1.LE.0)GO TO 18
      IF(IP1.NE.J)GO TO 17
      IF(PG(I).LT.PSLA(J))GO TO 17
      ISCOM(J)=IP
      PSLA(J)=PG(I)
17    CONTINUE
      890701
      IF(PGIS(J).LT.1.0E-06)ISCOM(J) = NNOD + 1
C
C
18    CONTINUE
C
      RETURN
      END
C
C    LA SUBROUTINA PREPA TRANSFORMA EL GRAFO DE Ybus EN UN GRAFO
C    SIMPLE Y DESCONECTA COMPENSADORES
C
C    SUBROUTINE PREPA(NNOD,NLIN,NL2,IEXT,NISLA,M,N,ISP,IPP,ISLA,ISCOM,
C    2LP,NORD,NACT,IDISP)
C
C    INTEGER LP(1),M(1),N(1),ISP(1),IPP(1),ISLA(1),
C    2NORD(1),ISCOM(1)
C
C    SE INICIALIZA EL VECTOR DE ENCADENAMIENTO PARA TENER UN RECOLEC-
C    TOR DE BASURA
C
      N3 = NL2
      NACT = NNOD
      LX=1
      DO 70 I=1,NLIN
70    LP(I)=0
      DO 100 I=1,NNOD
          NORD(I) = 0
          J3=N(M(I))
          DO 110 J=1,J3
              NC5=LX+1
              ISP(LX)=NC5
110    LX=NC5
          ISP(LX)=0
100    LX=LX+1
          IDISP=N3+1
          DO 30 I=IDISP,IEXT
30    ISP(I)=I+1
          ISP(IEXT)=0
C
C
C
C    SE ELIMINAN RAMAS EN PARALELO
C
C
C
C

```

```

DO 80 I=1, NNOD
81 II=M(I)
  IF(N(II)-1)82, 80, 82
82 NC5=II
  NC6=N(NC5)
  DO 83 J=1, NC6
  NC5=ISP(NC5)
  NC7=NC5
  IF(NC5)84, 83, 84
84 NA=N(NC5)
85 NC8=ISP(NC5)
  IF(NC8)86, 87, 86
87 NC5=NC7
  GO TO 83
86 NB=N(NC8)
  IF(NA-NB)88, 89, 88
88 NC5=NC8
  GO TO 85
89 N(II)=N(II)-1
  ISP(NC5)=ISP(NC8)
  ISP(NC8)=IDISP
  IP1=IPP(NC8)
  LP(IP1)=IPP(NC5)
C
C *** ERROR CORREGIDO EL 881220
C
  IDISP=NC8
C ***
  GO TO 81
83 CONTINUE
80 CONTINUE
C
C
C
C SE DESCONECTAN NODOS COMPENSADORES
C
DO 701 I3=1, NISLA
  NSLK=ISCOM(I3)
  890701
  IF(NSLK.GT.NNOD)GO TO 701
C
  IF(ISLA(NSLK).LT.0)GO TO 701
  NORD(NSLK)=NNOD
  NC5=ISP(M(NSLK))
140 IF(NC5)120, 130, 120
120 I=M(N(NC5))
  N(I)=N(I)-1
125 J=ISP(I)
  IF(N(J)-NSLK)145, 135, 145
135 ISP(I)=ISP(J)
  NC5=ISP(NC5)
  GO TO 140

```

```

145 I=J
    GO TO 125
130 NACT = NACT - 1
C   890701
701 NORD(NSLK)=NNOD
C
C
C   SE ASIGNA ORDEN ALTO A NODOS INACTIVOS
C
    DO 702 I=1, NNOD
    IF(ISLA(I).GE.0)GO TO 702
    NORD(I)=NNOD
    NACT = NACT - 1
702 CONTINUE
C
C
    RETURN
    END
C
C
C   LA SUBROUTINA ORDER EJECUTA UN ORDENAMIENTO DE MINIMO GRADO
C
    SUBROUTINE ORDER(NISLA, NNOD, NLIN, NL2, IEXT, NACT, M, N, ISP, IPP,
2NORD, IDISP)
C
C
    INTEGER NORD(1), M(1), N(1), ISP(1), IPP(1)
C
C
    N3=0
    DO 200 I=1, NACT
    NC5=NLIN
    NELI=NNOD+1
    DO 220 J=1, NNOD
    IF(NORD(J))220, 240, 220
240 LX=N(M(J))
    IF(LX-NC5)245, 220, 220
245 NELI=J
    NC5=LX
220 CONTINUE
    N12=NNOD+1
    IF(NELI-N12)556, 200, 556
556 LX=M(NELI)
    J=M(NELI)
    NORD(NELI)=I
    J1=ISP(J)
    IF(J1)6, 200, 6
    6 DO 250 J1=1, NC5
    J=ISP(J)
    J7=N(J)
    J2=M(J7)
    J3=N(J2)
    J5=LX

```

```

DO 270 J4=1, NCS
J5=ISP(J5)
J6=N(J5)
I1=J2
IF(J6-J7)280, 290, 280
290 J33=J3
DO 300 J8=1, J33
I3=I1
I1=ISP(I1)
I2=N(I1)
IF(I2-NELI)300, 315, 300
315 ISP(I3)=ISP(I1)
J3=J3-1
ISP(I1)=IDISP
IDISP=I1
GO TO 270
300 CONTINUE
GO TO 270
280 DO 330 J8=1, J3
I99=I1
I1=ISP(I1)
IF(I1)150, 151, 150
150 I2=N(I1)
IF(J6-I2)330, 270, 330
330 CONTINUE
GO TO 152
151 I1=I99
152 IF(IDISP-N3)33, 33, 34
34 N3=IDISP
33 K89=IDISP
ISP(I1)=K89
N(K89)=J6
IPP(K89)=0
IDISP=ISP(K89)
ISP(K89)=0
J3=J3+1
270 CONTINUE
N(J2)=J3
250 CONTINUE
200 CONTINUE
RETURN
END

```

C
C
C
C
C

LA SUBROUTINA FORMA LLENA LOS MODELOS DE FLUJOS

SUBROUTINE FORMA(NNOD, NACT, NLIN, NL2, NISLA, NISA, IEXT, IFIL, NTUG,
2NSHU, NTR, LP, M, N, NB1, NB1C, B1, B2, IPP, R, X, B, ISP, NORD, NCT, NST, T1, NSH,
3VLS, NU, ISCOM, NE, NR, NLIS)

C
C
C

```

      INTEGER LP(1),M(1),N(1),NB1(1),NB1C(1),IPP(1),
      2ISP(1),NORD(1),NCT(1),NST(1),NSH(1),NU(1),
      3ISCOM(1),NE(1),NR(1)
      INTEGER NLIS(1)

```

C
C

```

      REAL B1(1),B2(1),R(1),X(1),B(1),T1(1),VLS(1)

```

C
C

C*****

C
C

```

      LOS ELEMENTOS DE LOS MODELOS SE FORMAN Y COLOCAN EN B1() Y
      B11() SOLAMENTE PARA LA TRIANGULAR SUPERIOR

```

C
C

```

      WRITE(6,400)
400 FORMAT(/3X,' SE FORMAN MODELOS ...')

```

```

      K8 = 0

```

```

      K9 = 0

```

```

      DO 80 I = 1, NNOD

```

```

         NLIS(I) = 0

```

```

80 CONTINUE

```

```

      DO 81 I = 1, NTUG

```

```

         J = NU(I)

```

```

         NLIS(J) = 1

```

```

81 CONTINUE

```

C
C

```

      N12 = NACT

```

```

      DO 12 I = 1, N12

```

```

         DO 13 K = 1, NNOD

```

C
C

```

            IF(NORD(K).EQ.I)GO TO 14

```

```

13 CONTINUE

```

```

      GO TO 12

```

```

14 J = M(K)

```

```

      J1 = N(J)

```

```

      K9 = K9 + 1

```

```

      K8 = K8 + 1

```

```

      NB1(I) = K9

```

```

      NB1C(K8) = J1

```

```

      B1(K9)=0.0

```

```

      B2(K9)=0.0

```

```

      IP=ISP(J)

```

```

18 IF(IP.GT.0)GO TO 17

```

```

      K9=K8

```

```

      GO TO 12

```

```

17 K8=K8+1

```

```

      NB1C(K8)=N(IP)

```

```

      IF(IPP(IP).NE.0)GO TO 19

```

```

      B1(K8)=0.0

```

```

      B2(K8)=0.0

```

```

GO TO 20
19 IP2=IPP(IP)
   B1(K8)=-1.0/X(IP2)
     B2(K8) = 0.0
   IF(NLIS(K) .NE. 0) GO TO 20
   IF(NLIS(N(IP)) .NE. 0) GO TO 20
   B2(K8)=-X(IP2)/(R(IP2)*R(IP2)+X(IP2)*X(IP2))
20 IP=ISP(IP)
   GO TO 18
12 CONTINUE

```

C
C
C
C
C

SE FORMAN ELEMENTOS DIAGONALES DE B1()

```

DO 60 I=1,N12
DO 65 K1=1,NNOD
IF(NORD(K1).EQ.I)GO TO 66
65 CONTINUE
GO TO 60
66 N1=K1
   N2=NB1(I)
   DO 70 K=1,NLIN
   IF(NE(K).EQ.N1)GO TO 78
   IF(NR(K).EQ.N1)GO TO 78
   GO TO 70
78 Y=1.0/X(K)
   Y11=X(K)/(X(K)*X(K)+R(K)*R(K))
   B1(N2)=B1(N2)+Y
   B2(N2)=B2(N2)+Y11
   B2(N2)=B2(N2)-B(K)
70 CONTINUE
60 CONTINUE

```

C
C
C
C
C
C

SE MODIFICAN MODELOS PARA INCLUIR LINEAS EN PARALELO

```

DO 75 I=1,NLIN
I1=LP(I)
IF(I1.EQ.0)GO TO 75
N1=NE(I)
N2=NR(I)
N4=NB1(NORD(N1))
N5=NB1C(N4)
Y=-1.0/X(I)
Y11=-X(I)/(R(I)*R(I)+X(I)*X(I))
IF(N5.EQ.0)GO TO 71
DO 72 J=1,N5
K1=N4+J
NB=NB1C(K1)
IF(NB.NE.N2)GO TO 72

```



```

B1(K1)=B1(K1)+Y
IF(NLIS(NB) .NE. 0) GO TO 71
B2(K1)=B2(K1)+Y11
GO TO 71
72 CONTINUE
71 NO=NORD(N2)
N4=NB1(NO)
IF(N4.LE.0)GO TO 75
N5=NB1C(N4)
IF(N5.EQ.0)GO TO 75
DO 77 J=1,N5
K1=N4+J
NB=NB1C(K1)
IF(NB.NE.N1)GO TO 77
B1(K1)=B1(K1)+Y
IF(NLIS(NB) .NE. 0) GO TO 77
B2(K1)=B2(K1)+Y11
GO TO 75
77 CONTINUE
75 CONTINUE

```

C
C
C
C
C

SE INCLUYEN TAPS EN TRANSFORMADORES EN B2()

```

IF(NTR.LE.0)GO TO 200
DO 30 I=1,NTR
N1=NCT(I)
N2=NST(I)
IF(NLIS(N1) .NE. 0 .OR. NLIS(N2) .NE. 0) GO TO 30
N3=NORD(N1)
N4=NORD(N2)
IF(N3.LT.N4)GO TO 31
N5=N2
N0=N1
GO TO 32
31 N5=N1
N0=N2
32 N6=NORD(N5)
N7=NB1(N6)
J33=NB1C(N7)
DO 33 J=1,J33
N8=N7+J
N9=NB1C(N8)
IF(N9.NE.N0)GO TO 33
N10=NB1(N3)
B2(N10)=B2(N10)+B2(N8)*((T1(I)*T1(I)-1.0)/(T1(I)*T1(I)))
B2(N8)=B2(N8)/T1(I)
33 CONTINUE
30 CONTINUE

```

C
C
C

```

C   SE INCLUYEN COMPENSADORES EN B2()
C
C
200 IF(NSHU.LT.1)GO TO 45
    DO 35 I=1, NSHU
        N1=NSH(I)
        N2=NORD(N1)
        N3=NB1(N2)
    35 B2(N3)=B2(N3)-VLS(I)
    45 CONTINUE
C
C
    IFIL=K8
    DO 800 KK = 1, IFIL
        WRITE(6,801)KK,NB1C(KK),B2(KK)
    801  FORMAT(3X,' * K,NB1C, B2 ',2I5,E14.7)
    800 CONTINUE
C
    RETURN
    END
C
C
C   LA SUBROUTINA FACTA GENERA LOS FACTORES TRIANGULARES DE
C   UNA MATRIZ SIMETRICA REAL USANDO UN TRIANGULO DE LA
C   MATRIZ QUE INCLUYA LOS ELEMENTOS CREADOS DURANTE
C   EL ORDENAMIENTO
C
C   SUBROUTINE FACTA(NNOD,NACT,IFIL,NISLA,NISA,NB1,NB1C,NORD,B1,B2)
C
C
C-----
C
C
C   INTEGER NB1(1),NB1C(1),NORD(1)
C
C
C   REAL B1(1),B2(1)
C
C
C   WRITE(6,400)
    400 FORMAT(/3X,' SE FACTORIZAN LOS MODELOS ...')
        N12=NACT
        DO 10 I=1,N12
            K=NB1(I)
            NEL=NB1C(K)
            IF(NEL.LT.1)GO TO 10
            DO 20 J=1,NEL
                NA=K+J
                E1=B1(NA)/B1(K)
                EG=B1(NA)*E1
                E2=B2(NA)/B2(K)
                EG2=B2(NA)*E2

```

```

C
C   SE MODIFICA EL ELEMENTO DIAGONAL DE UN NODO ADYACENTE
C
  NB=NB1C(NA)
  NC=NORD(NB)
  ND=NB1(NC)
  B1(ND)=B1(ND)-EG
  B2(ND)=B2(ND)-EG2
C
C   SE MODIFICAN ELEMENTOS DEBAJO DE LA FILA PIVOTE SI EXISTE
C   CORRESPONDENCIA
C
  IF(NEL.LE.1)GO TO 20
  DO 30 K1=1,NEL
  KA=K+K1
  KA1=NB1C(KA)
  NB11=NB1C(ND)
  IF(NB11.LT.1)GO TO 30
  DO 31 K2=1,NB11
  KB=ND+K2
  KB1=NB1C(KB)
  IF(KA1.NE.KB1)GO TO 31
  B1(KB)=B1(KB)-E1*B1(KA)
  B2(KB)=B2(KB)-E2*B2(KA)
31 CONTINUE
30 CONTINUE
20 CONTINUE
C
C   NORMALIZACION DE ELEMENTOS EN LA FILA PIVOTE
C
  DO 15 K1=1,NEL
  NA=K+K1
  B1(NA)=B1(NA)/B1(K)
15 B2(NA)=B2(NA)/B2(K)
10 CONTINUE
C
  WRITE(6,70)
70 FORMAT(/3X,' SE HAN FACTORIZADO MODELOS ... '/')
C   DO 100 I=1,NNOD
C   100 WRITE(6,101)I,NB1(I)
C
C
C   DO 100 I=1,NNOD
C 100 WRITE(6,101)I,NB1(I)
C 101 FORMAT(3X,' I ',I5,' NB1 ',I5)
C   NN = NNOD - NISLA
C   NN = NB1(NN)
C   DO 103 I = 1, NN
C     WRITE(6,104)I,NB1C(I),B1(I),B2(I)
C 103 CONTINUE
C 104 FORMAT(3X,' I ',I5,' NB1C ',I6,' B1 ',F12.5,' B2 ',F12.5)
  RETURN
  END

```

```

C
C   LA SUBROUTINA ININF INICIALIZA ALGUNAS VARIABLES Y ARREGLOS
C
C   SUBROUTINE ININF(NNOD, NTUG, BASE, NORD, NORD1, PG, VG, NU, E3, P2, Q2, PC,
C   2QC)
C
C
C   INTEGER NORD(1), NORD1(1), NU(1)
C
C   REAL BASE, PG(1), VG(1), E3(1), P2(1), Q2(1)
C   REAL PC(1), QC(1)
C
C   DO 11 I=1, NTUG
C     N1=NU(I)
11  E3(N1)=VG(I)
C
C   CALCULO DE INYECCIONES DE POTENCIA NETA EN CADA NODO
C
C   DO 13 I=1, NTUG
C     N1=NU(I)
13  P2(N1)=PG(I)
C     DO 14 I=1, NNOD
C       P2(I)=(P2(I)-PC(I))/BASE
14  Q2(I)=(Q2(I)-QC(I))/BASE
C
C   SE FORMA UN VECTOR AUXILIAR PARA COLOCAR NODOS EN ORDEN
C   DE ELIMINACION
C
C   DO 20 I=1, NNOD
C     DO 21 J=1, NNOD
C       IF(NORD(J).NE. I)GO TO 21
C       NORD1(I)=J
C   21 CONTINUE
C   20 CONTINUE
C     DO 20 I = 1, NNOD
C       IF(NORD(I).GE. NNOD)GO TO 20
C       NORD1(NORD(I)) = I
20  CONTINUE
C
C   RETURN
C   END

```

```

C
C   LA SUBROUTINA DESVP CALCULA DESVIACIONES DE POTENCIA EN
C   CADA NODO
C
C   SUBROUTINE DESVP(NNOD, NISLA, NL2, ISCOM, NORD, M, N, P2, F, E3, F3,
2   YR, YI, D, L8)
C
C   INTEGER ISCOM(1), M(1), N(1), NORD(1)
C
C   REAL P2(1), F(1), E3(1), F3(1), YR(1), YI(1)
C   REAL A, B3, E1, F1, D, D1
C
C   DO 10 K = 1, NNOD
C       IF(NORD(K).GE.NNOD)GO TO 10
C       DO 11 J = 1, NISLA
C           IF(ISCOM(J).EQ.K)GO TO 10
C   11  CONTINUE
C       N1 = M(K)
C       N2 = N(N1)
C
C       CALL IC(K, N1, N2, A, B3, E1, F1, NL2, E3, F3, YR, YI, NNOD, N)
C
C       AUX = E1*A + F1*B3
C       D1=P2(K)-AUX
C       F(K)=D1/E3(K)
C       D1=ABS(D1)
C       IF(D1.LT.D)GO TO 10
C       D = D1
C       L8 = K
C   10 CONTINUE
C
C   RETURN
C   END
C
C   LA SUBROUTINA SOLP RESUELVE EL MODELO DE POTENCIA REAL
C
C   SUBROUTINE SOLP(NNOD, NACT, NISLA, NISA, IFIL, NB1, NB1C, NORD1, F, B1)
C
C   INTEGER NB1(1), NB1C(1), NORD1(1)
C
C   REAL F(1), B1(1)

```

C
C
C
C

SUSTITUCION PROGRESIVA

```

N3 = NACT
DO 10 I=1, N3
L1=NB1(I)
L2=NB1C(L1)
IF(L2.LT.1)GO TO 10
DO 11 J=1, L2
L3=L1+J
L4=NB1C(L3)
L5=NORD1(I)
11 F(L4)=F(L4)-F(L5)*B1(L3)
10 CONTINUE

```

C
C
C
C
C

SE NORMALIZA CON LOS ELEMENTOS DIAGONALES DE LOS FACTORES

```

DO 12 I=1, N3
L5=NORD1(I)
L6=NB1(I)
12 F(L5)=F(L5)/B1(L6)

```

C
C
C
C
C

SUSTITUCION REGRESIVA

```

N4 = 1
N5 = NACT - 1
DO 13 I = N4, N5
K = NACT - I
L5=NORD1(K)
L1=NB1(K)
L2=NB1C(L1)
IF(L2.LT.1)GO TO 13
DO 14 J=1, L2
L3=L1+J
L4=NB1C(L3)
14 F(L5)=F(L5)-B1(L3)*F(L4)
13 CONTINUE

```

C
CRETURN
ENDC
C
C
C
CLA SUBRUTINA DESVQ CALCULA DESVIACIONES DE POTENCIA REACTIVA
EN CADA NODO

```

C
C
SUBROUTINE DESVQ(NNOD, NTUG, NL2, NU, NORD, M, N, Q2, E, E3, F3, YR, YI, D2, L9)
C
C
INTEGER NU(1), M(1), N(1), NORD(1)
C
C
REAL Q2(1), E(1), E3(1), F3(1), D1, D2, A, B3, E1, F1
REAL YR(1), YI(1)
C
C
DO 10 I=1, NNOD
  IF(NORD(I).GE. NNOD)GO TO 10
  DO 11 J=1, NTUG
    IF(NU(J).EQ. I)GO TO 10
11  CONTINUE
    N1 = M(I)
    N2 = N(N1)
C
C
CALL IC(I, N1, N2, A, B3, E1, F1, NL2, E3, F3, YR, YI, NNOD, N)
C
C
D1 = Q2(I)-(F1*A-E1*B3)
E(I) = D1/E3(I)
D1 = ABS(D1)
IF(D1.LT. D2)GO TO 10
D2 = D1
L9 = I
10 CONTINUE
C
C
RETURN
END

C
C
LA SUBROUTINA SOLQ RESUELVE EL MODELO DE POTENCIA REACTIVA
C
SUBROUTINE SOLQ(NNOD, NACT, NISLA, NISA, IFIL, NB1, NB1C, NORD1, E, B2)
C
C
INTEGER NB1(1), NB1C(1), NORD1(1)
C
C
REAL B2(1), E(1)
C
C
C
C

```

C SUSTITUCION HACIA ADELANTE
C

```

N3 = NACT
DO 10 I=1,N3
L1=NB1(I)
L2=NB1C(L1)
IF(L2.LT.1)GO TO 10
DO 11 J=1,L2
L3=L1+J
L4=NB1C(L3)
L5=NORD1(I)
11 E(L4)=E(L4)-E(L5)*B2(L3)
10 CONTINUE

```

C
C
C
C

SE NORMALIZA CON ELEMENTOS DIAGONALES DE LOS FACTORES

```

DO 12 I=1,N3
L5=NORD1(I)
L6=NB1(I)
12 E(L5)=E(L5)/B2(L6)

```

C
C
C
C
C

SUSTITUCION HACIA ATRAS

```

N4 = 1
N5 = NACT - 1
DO 13 I=N4,N5
K=NACT - I
L5=NORD1(K)
L1=NB1(K)
L2=NB1C(L1)
IF(L2.LT.1)GO TO 13
DO 14 J=1,L2
L3=L1+J
L4=NB1C(L3)
14 E(L5)=E(L5)-B2(L3)*E(L4)
13 CONTINUE

```

C
C

RETURN
END

C
C
C
C
C

LA SUBROUTINA IC CALCULA LA CORRIENTE NETA INYECTADA EN UN NODO
EN COORDENADAS CARTESIANAS

SUBROUTINE IC(K, N1, N2, A, B3, E1, F1, NL2, E3, F3, YR, YI, NNOD, N)

C
C

INTEGER N(NL2)
REAL E3(NNOD), F3(NNOD), YR(NL2), YI(NL2), E1, F1, E2, F2, A, X3, Y3, B3


```

C
E1=E3(K)*COS(F3(K))
F1=E3(K)*SIN(F3(K))
E2=YR(N1)
F2=YI(N1)
A=E2*E1-F2*F1
B3=E1*F2+F1*E2
N33=N1+1
N34=N1+N2
DO 10 J1=N33,N34
N3=N(J1)
X3=E3(N3)
Y3=F3(N3)
E2=X3*COS(Y3)
F2=X3*SIN(Y3)
X3=YR(J1)
Y3=YI(J1)
A=A+E2*X3-F2*Y3
10 B3=B3+F2*X3+E2*Y3

```

```

C
RETURN
END

```

C *****
C LA SUBRUTINA CORRA CORRIGE ANGULOS DE FASE EN CADA ITERACION
C *****

```

C
SUBROUTINE CORRA(NNOD, NISLA, ISCOM, NORD, F, F3)
C
C
INTEGER ISCOM(1), NORD(1)
C
REAL F(1), F3(1)
C
DO 10 I=1, NNOD
IF(NORD(I).GE.NNOD)GO TO 10
DO 11 J=1, NISLA
IF(ISCOM(J).EQ.1)GO TO 10
11 CONTINUE
F3(I)=F3(I)+F(I)
10 CONTINUE
C
RETURN
END

```

C *****
C LA SUBRUTINA CORRV VORRIGE MAGNITUDES DE VOLTAJE EN CADA
C ITERACION
C *****

```

C
SUBROUTINE CORRV(NNOD, NTUG, NU, NORD, E3, E)
C
INTEGER NU(NTUG), NORD(1)
C
C
REAL E3(1), E(1)
C
DO 10 I=1, NNOD
  IF(NORD(I).GE. NNOD)GO TO 10
  DO 11 J=1, NTUG
    IF(NU(J).EQ. I)GO TO 10
11  CONTINUE
    E3(I)=E3(I)+E(I)
10 CONTINUE
C
RETURN
END

SUBROUTINE YBUSC(NNOD, NLIN, NTR, NSHU, NL2, M, N, YR, YI, BI, NE, NR,
2R, X, B, NCT, NST, T1, NSH, VLS, IPP)
C*****
C*****
C***** SUBROUTINE YBUSC PARA FORMAR LA MATRIZ DE ADMITANCIAS NODAL
C*****
C*****
C
C
INTEGER M(1), N(1), NE(1), NR(1), NCT(1), NST(1), IPP(1),
2NSH(1)
C
REAL YR(1), YI(1), R(1), X(1), B(1), T1(1), VLS(1)
REAL A, BI(1)
C
DO 1000 I = 1, NL2
  YR(I) = 0.0
  YI(I) = 0.0
1000 CONTINUE
  N3 = NL2
  J2 = 1
  J1 = 1
  DO 100 J = 1, NNOD
    DO 102 K = 1, NLIN
      IF(NE(K) - J)20, 31, 20
31      J1=J1+1
        N(J1) = NR(K)
      GO TO 33
20 IF(NR(K)-J)102, 32, 102
32 J1 = J1 + 1
        N(J1) = NE(K)
33 A = R(K)*R(K) + X(K)*X(K)

```

```

C ***** CAMBIO DEL 890623
      IPP(J1) = K
C *****
      YR(J1) = - R(K)/A
      YI(J1) = X(K)/A
      BI(J1) = B(K)
      YR(J2) = YR(J2) - YR(J1)
      YI(J2) = YI(J2) - YI(J1) + B(K)
102 CONTINUE
      N(J2)=J1-J2
      M(J)=J2
      J1=J1+1
100 J2=J1

C
C
C
C
C
C
      CORRECCION DE Ybus PARA INCLUIR TAPS EN
      TRANSFORMADORES

      IF(NTR.EQ.0)GO TO 110
      DO 10 I=1,NTR
      N1=NCT(I)
      N2=NST(I)
      N3=M(N1)
      N4=N(N3)
      IF(N4.LT.1)GO TO 10
      DO 11 J=1,N4
      K=N3+J
      N5=N(K)
      IF(N5.NE.N2)GO TO 11
      YR(N3)=YR(N3)+YR(K)*((T1(I)*T1(I)-1.0)/(T1(I)*T1(I)))
      YI(N3)=YI(N3)+YI(K)*((T1(I)*T1(I)-1.0)/(T1(I)*T1(I)))
      BI(K) = BI(K) - 1.0D00/T1(I)* (1.0D00/T1(I) - 1.0D00) * YI(K)
      YR(K)=YR(K)/T1(I)
      YI(K)=YI(K)/T1(I)
11 CONTINUE
      N3=M(N2)
      N4=N(N3)
      DO 12 J=1,N4
      K=N3+J
      N5=N(K)
      IF(N5.NE.N1)GO TO 12
      YR(K)=YR(K)/T1(I)
      BI(K) = BI(K) - YI(K) * ( 1.0D00 - 1.0D00/T1(I))
      YI(K) = YI(K)/T1(I)
12 CONTINUE
10 CONTINUE

C
C
C
C
C
      CORRECCION IN Ybus PARA INCLUIR SHUNTS

```

```

110 IF(NSHU.EQ.0)GO TO 14
    DO 13 I=1,NSHU
        N1=NSH(I)
        N2=M(N1)
13  YI(N2)=YI(N2)+VLS(I)
14  CONTINUE
C
    RETURN
C*****
C*****  END OF SUBROUTINE YBUSC
C*****
    END

    SUBROUTINE AJUSQ(NNOD, NACT, NTUG, NISLA, NISA, TINIA, L9, D2, ISCOM, NU,
2   M, N, IFIL, NB1, NB1C, NORD, NORD1, B1, B2, QMA, QMI, Q2, E, E3, F3,
3   YR, YI, IE2, IE3, IE4, DNOMB, QC, NL2, BASE)
C
    INTEGER NU(1), M(1), N(1), NB1(1), NB1C(1), NORD(1), NORD1(1),
2   ISCOM(1), NNOD, NTUG, NISLA, L9
    REAL Q2(1), E(1), E3(1), D1, D2, A, B3, E1, F1, YI(1), BASE,
2   B1(1), B2(1), QMA(1), QMI(1), QC(1), YR(1), F3(1)
    CHARACTER*8 DNOMB(1)
C
    C
    C
    C
    IF(IE2 .EQ. 0) GO TO 20
    IF(IE3 .NE. 0) GO TO 15
    IF(D2 .GE. TINIA) GO TO 20
    IE3 = 1
15  CONTINUE
    DO 16 IG = 1, NTUG
        I1 = NU(IG)
        IF(I1 .LE. 0)GO TO 16
        N1 = M(I1)
        N2 = N(N1)
        D1 = 0.0
        CALL IC(I1, N1, N2, A, B3, E1, F1, NL2, E3, F3, YR, YI, NNOD, N)
        D1 = F1*A - E1*B3
C
        WRITE(6, 400)I1, D1, QC(I1), QMA(IG), QMI(IG)
C 400  FORMAT(2X, ' **G, QCAL, QC, QMA, QMI *', 15, 2X, 4(E12.5, 2X))
        D1 = D1 + QC(I1) / BASE
        IF(D1 .GT. QMA(IG)) GO TO 17
        IF(D1 .LT. QMI(IG)) GO TO 18
        E(I1) = 0.0
        GO TO 16
17  CONTINUE
        D1 = QMA(IG) - D1
        Q2(I1) = Q2(I1) + QMA(IG)
        GO TO 19
18  CONTINUE
        D1 = QMI(IG) - D1

```

```

      Q2(I1) = Q2(I1) + QMI(IG)
19  CONTINUE
      E(I1) = D1 / E3(I1)
      D1 = ABS(D1)
      IF(D1 .LE. D2) GO TO 21
      L9 = I1
      D2 = D1
21  CONTINUE
      IF(NU(IG) .LE. 0)GO TO 16
      NU(IG) = -NU(IG)
      IE4 = 0
16  CONTINUE
      IF(IE4 .NE. 0)GO TO 20
      DO 30 IK = 1, NTUG
        IF(NU(IK) .GT. 0)GO TO 30
        IK1 = ABS(NU(IK))
        WRITE(6,31)IK1, DNOMB(IK1), Q2(IK1)*BASE
31  FORMAT(/10X, 'NODO ', I6, 1X, A8, ' SE CONVIERTE EN NODO PQ ',
2    ' CON ', F8.2, ' MVAR' )
30  CONTINUE
      CALL FORM(NNOD, NACT, NISLA, NISA, NTUG, ISCOM, NORD, NORD1, NU, M, N,
2    NB1, NB1C, B2, YI)
      CALL FACTI(NNOD, NACT, IFIL, NISLA, NISA, NB1, NB1C, NORD, B2)
      IE4 = 1
20  CONTINUE
C    WRITE(6,50)
C    50 FORMAT(/' **** PASA POR AJUSTE ****')
C
      RETURN
      END

```

```

      SUBROUTINE FORM(NNOD, NACT, NISLA, NISA, NTUG, ISCOM, NORD, NORD1, NU,
2  M, N, NB1, NB1C, B2, YI)
C
      INTEGER NNOD, NISLA, NTUG, ISCOM(1), NORD(1), NORD1(1), NU(1),
2  M(1), N(1), NB1(1), NB1C(1)
      REAL B2(1), YI(1)
C
      SE LIMPIA EL ARREGLO B2(.)
C
      NO = NACT
      DO 30 I = 1, NB1(NO) + 1
        B2(I) = 0.0
30  CONTINUE
C
      SE LLENA B2(.) CON ELEMENTOS DE YI(.)
C
      DO 10 I = 1, NO
        I1 = NORD1(I)
        N1 = M(I1)
        N2 = N(N1)
        N3 = NB1(I)

```

```

N44 = NB1C(N3)
B2(N3) = -YI(N1)
IF(N44 .EQ. 0)GO TO 10
  DO 11 J = 1, N44
    K = N3 + J
    N5 = NB1C(K)
    DO 12 J1 = 1, N2
      K1 = N1 + J1
      N6 = N(K1)
      IF(N5 .NE. N6)GO TO 12
      B2(K) = B2(K) - YI(K1)
12    CONTINUE
11    CONTINUE
10 CONTINUE

C
C   SE ELIMINAN ELEMENTOS DE CONEXIONES CON NODOS P-V AUN
C   REGULANDO VOLTAJE EN B2(.)
C

DO 36 I = 1, NTUG
  N1 = NU(I)
  IF(N1 .LE. 0)GO TO 36
  DO 39 K1 = 1, NISLA
    IF(N1 .EQ. ISCOM(K1))GO TO 36
39  CONTINUE
  N2 = NORD(N1)
  N3 = NB1(N2)
  N44 = NB1C(N3)
  IF(N44 .LT. 1)GO TO 36
  DO 37 J = 1, N44
    N5 = N3 + J
    B2(N5) = 0.0D00
    N6 = NB1C(N5)
    N7 = NORD(N6)
    N8 = NB1(N7)
    IF(N2 .LT. N7)GO TO 37
    N9 = NB1C(N8)
    DO 38 K = 1, N9
      N11 = N8 + K
      N12 = NB1C(N11)
      IF(N12 .NE. N1)GO TO 38
      B2(N11) = 0.0D00
38    CONTINUE
37    CONTINUE
36 CONTINUE
  RETURN
  END

C
C   SUBROUTINE FACTA GENERATES TRIANGULAR FACTORS FOR A
C   SYMMETRICAL MATRIX USING ONLY A TRIANGULAR OF THE O-
C   RIGINAL MATRIX AND FILL IN POSITIONS DERIVED FROM
C   AN ORDERING PROGRAM.
C

```

```

C
SUBROUTINE FACTI(NNOD, NACT, IFIL, NISLA, NISA, NB1, NB1C, NORD, B2)
C
C
INTEGER NB1(1), NB1C(1), NORD(1)
C
C
REAL B2(1)
C
C
N12 = NACT
DO 10 I=1, N12
K=NB1(I)
NEL=NB1C(K)
IF(NEL.LT.1)GO TO 10
DO 20 J=1, NEL
NA=K+J
C
E1=B1(NA)/B1(K)
C
EG=B1(NA)*E1
E2=B2(NA)/B2(K)
EG2=B2(NA)*E2
C
C
C
WE MODIFY DIAGONAL ELEMENT OF AN ADJACENT NODE
C
C
NB=NB1C(NA)
NC=NORD(NB)
ND=NB1(NC)
C
B1(ND)=B1(ND)-EG
B2(ND)=B2(ND)-EG2
C
C
C
WE MODIFY ELEMENTS UNDER THE PIVOT ROW IF THERE EXISTS
CORRESPONDENCE.
C
C
C
IF(NEL.LE.1)GO TO 20
DO 30 K1=1, NEL
KA=K+K1
KA1=NB1C(KA)
NB11=NB1C(ND)
IF(NB11.LT.1)GO TO 30
DO 31 K2=1, NB11
KB=ND+K2
KB1=NB1C(KB)
IF(KA1.NE.KB1)GO TO 31
C
B1(KB)=B1(KB)-E1*B1(KA)
B2(KB)=B2(KB)-E2*B2(KA)
31 CONTINUE
30 CONTINUE
20 CONTINUE

```

```

C
C
C   NORMALIZATION OF ELEMENTS IN PIVOT ROW
C
C   DO 15 K1=1,NEL
C   NA=K+K1
C   B1(NA)=B1(NA)/B1(K)
15  B2(NA)=B2(NA)/B2(K)
10  CONTINUE
C
C
C
C
C   WRITE(6,70)
70  FORMAT(///3X,'SE RETRIANGULARIZAN MODELOS... '/')
C   DO 100 I=1,NNOD
C   100 WRITE(6,101)I,NB1(I)
C
C
C   DO 100 I=1,NNOD
C   100 WRITE(6,101)I,NB1(I)
C   101 FORMAT(3X,' I ',I5,' NB1 ',I5)
C   NN = NNOD - NISLA
C   NN = NB1(NN)
C   DO 103 I = 1, NN
C   WRITE(6,104)I,NB1C(I),B2(I)
C   103 CONTINUE
C   104 FORMAT(3X,' I ',I5,' NB1C ',I6,' B2 ',F12.5)
C   RETURN
C   END

SUBROUTINE LFSOL3(NNOD, NACT, NTUG, NISLA, NISA, NL2, IFIL, ISCOM, BASE,
2NORD, NORD1, PG, VG, NU, E3, F3, P2, Q2, PC, QC, YR, YI, NB1, NB1C, B1, B2, E, F,
3NITER, TOLP, TOLQ, M, N, DNOMB, TINIA, QMA, QMI)
C*****
C***** SUBROUTINE LFSOL(.) PARA RESOLVER FLUJOS DE POTENCIA USANDO
C***** EL METODO DESACOPLADO RAPIDO
C*****
C
C   INTEGER NNOD, NTUG, NISLA, NL2, IFIL, ISCOM(1), NORD(1),
2NORD1(1), NB1(1), NB1C(1), M(1), N(1), NU(1), NITER
C   REAL BASE, PG(1), VG(1), E3(1), F3(1), E(1),
2F(1), P2(1), Q2(1), PC(1), QC(1), YR(1), YI(1), QMA(1), QMI(1),
3B1(1), B2(1), TOLP, TOLQ, D, D2
C   CHARACTER*8 DNOMB(1)
C   I8 = 0
C   CALL ININF(NNOD, NTUG, BASE, NORD, NORD1, PG, VG, NU, E3, P2, Q2, PC, QC)
C   WRITE(6,35)
35  FORMAT(///4X,'ITER DES. MAX. P NODO',5X,'DES. MAX. Q',
14X,'NODO '/')

```



```

C
C
      IE2 = 0
      IE3 = 0
      IE4 = 1
      NP = 1
      NQ = 1
10  CONTINUE
      D = 0.0
      CALL DESVP(NNOD, NISLA, NL2, ISCOM, NORD, M, N, P2, F, E3, F3, YR, YI, D, L8)
C
      IF(D.LE.TOLP)GO TO 11
      CALL SOLP(NNOD, NACT, NISLA, NISA, IFIL, NB1, NB1C, NORD1, F, B1)
C
      CALL CORRA(NNOD, NISLA, ISCOM, NORD, F, F3)
C
      NQ = 1
C
C
20  CONTINUE
      D2 = 0.0
      890823.
      L9 = ISCOM(1)
C
      CALL DESVQ(NNOD, NTUG, NL2, NU, NORD, M, N, Q2, E, E3, F3, YR, YI, D2, L9)
      WRITE(6, 50) I8, D, DNOMB(L8), D2, DNOMB(L9)
C
50  FORMAT(3X, I6, 3X, 2(F10.5, 2X, A8, 5X))
C
      SE INTRODUCE EL AJUSTE DE REACTIVOS EN NODOS P-V
C
      CALL AJUSQ(NNOD, NACT, NTUG, NISLA, NISA, TINIA, L9, D2, ISCOM, NU,
2   M, N, IFIL, NB1, NB1C, NORD, NORD1, B1, B2, QMA, QMI, Q2, E, E3, F3,
3   YR, YI, IE2, IE3, IE4, DNOMB, QC, NL2, BASE)
C
C
      IF(D2 .LE. TOLQ)GO TO 15
C
      CALL SOLQ(NNOD, NACT, NISLA, NISA, IFIL, NB1, NB1C, NORD1, E, B2)
      WRITE(6, 215)XXXX
C
      CALL CORRV(NNOD, NTUG, NU, NORD, E3, E)
      NP = 1
18  CONTINUE
      I8 = I8 + 1
      IF(I8.GT.NITER)GO TO 26
      DO 17 I = 1, NNOD
          E(I) = 0.0
17  F(I) = 0.0
      GO TO 10
C
C
C

```

```

11 NP = 0
   IF(NQ .EQ. 0)GO TO 25
   GO TO 20
C
C
15 NQ = 0
   IF(NP.NE.0)GO TO 18
   GO TO 25
C
C
26 WRITE(6,27)I8
27 FORMAT(/5X,'EL ESTUDIO NO CONVERGE EN ',I6,' ITERACIONES
   2...'/)
   GO TO 3000
C
C
25 WRITE(6,31)I8
31 FORMAT(/5X,'EL ESTUDIO CONVERGE EN ',I6,' ITERACIONES '/')
C
C
30 CONTINUE
   DO 37 I = 1, NNOD
     E(I) = E3(I)
     F(I) = F3(I)
37 CONTINUE
3000 CONTINUE
   RETURN
   END
C*****
C*****  END OF SUBROUTINE LFSOL(.)
C*****

```

```

SUBROUTINE FLUCAL(I1, NNOD, NL2, NTUG, NTR, NSHU, M, N, YR, YI, BI,
2NCT, NST, T1, NSH, VLS, BASE, E, F, E3, F3, DNOMB, PG, QG, PC, QC,
3NU, QMA, QMI, NM, PL, QL, T2, ILE, ISIZE, IWR)
C*****
C*****
C*****  SUBROUTINE FLUCAL PARA CALCULAR FLUJOS DE POTENCIA IN
C*****  TABULAR
C*****
C

```

```

INTEGER M(1), N(1), NCT(1), NST(1), NSH(1), NM(1)
2, NU(1)
CHARACTER*8 DNOMB(1)
REAL E(1), F(1), E3(1), F3(1), PL(1), QL(1), T2(1)
REAL VLS(1), T1(1), QMA(1), QMI(1)
REAL PG(1), QG(1), PC(1), QC(1)
REAL SUM, QUM, EH, AB, CD, YR(1), YI(1), BASE, SI, SR, SG
REAL BI(1), SN, SQ, PC1, QC1
NS1 = 0

```

```

DATA IF6/2H /
DATA IFO/2HRR/
DATA IF1/2HHH/
DATA IF2/2HLL/
DATA IF3/2HTR/
IG = 0
SUM = 0.0
QUM = 0.0
N1 = M(I1)
N2 = N(N1)
KZ = 0
EH = E3(I1)*E3(I1) + F3(I1)*F3(I1)

```

C
C

```

DO 10 K = 1, N2
  J = N1 + K
  KZ = KZ + 1
  MQ = N(J)
  AB = E3(I1)*E3(MQ) + F3(I1)*F3(MQ)
  CD = F3(I1)*E3(MQ) - E3(I1)*F3(MQ)
  PL(KZ) = -EH*YR(J) + YR(J)*AB + YI(J)*CD
  QL(KZ) = EH*(YI(J) - BI(J)) - YI(J)*AB + YR(J)*CD
  PL(KZ) = PL(KZ)*BASE
  QL(KZ) = QL(KZ)*BASE
  NM(KZ) = MQ
  T2(KZ) = 0.0
  SUM = PL(KZ) + SUM
10 QUM = QL(KZ) + QUM

```

C
C
C
C

SE CHECAN TAPS EN TRANSFORMADORES CONECTADOS A ESTE NODO

```

IF(NTR.EQ.0)GO TO 11
DO 12 K1 = 1, NTR
  IF(NCT(K1).NE. I1)GO TO 13
  DO 14 K2 = 1, KZ
    IF(NM(K2).NE. NST(K1))GO TO 14
    T2(K2) = T1(K1)
    NM(K2) = - NM(K2)
14 CONTINUE
  GO TO 12
13 IF(NST(K1).NE. I1)GO TO 12
  DO 15 K2 = 1, KZ
    IF(NM(K2).NE. NCT(K1))GO TO 15
    T2(K2) = 1.0E-7
    NM(K2) = - NM(K2)
15 CONTINUE
12 CONTINUE
  DO 16 K2 = 1, KZ
    NM(K2) = IABS(NM(K2))
16 CONTINUE

```

C
C

C
C
C
C

SE CHECAN SHUNTS EN EL NODO

```

11 IF(NSHU.EQ.0)GO TO 23
   DO 21 K1 = 1, NSHU
     IF(NSH(K1).EQ.I1)GO TO 22
21 CONTINUE
   GO TO 23
22 SI = - EH * VLS(K1)
   SI = SI*BASE
   SR = 0.0
   QUM = QUM + SI
   NS1 = 1
23 CONTINUE

```

C
C
C
C
C

SE IMPRIMEN FLUJOS DE POTENCIA

```

AMAG = E(I1)
ANG = F(I1) * 180.0/3.1416
ILE = ILE + 4
IF(ILE .GT. ISIZE)CALL ENCA
WRITE(IWR, 1)I1, DNOMB(I1), AMAG, ANG, I1
1 FORMAT('/ DE', I6, 1X, A8, 4X, 'MW', 5X, 'MVAR', 5X, 'MVA',
23X, ' TAP', F7.4, F8.2, I5)

```

C
C
C
C
C

SE CHECA SI EL NODO ES GENERADOR

```

DO 25 K1 = 1, NTUG
   IF(IABS(NU(K1)).EQ.I1)GO TO 26
25 CONTINUE
   GO TO 30
26 IG = 1
   PG1 = SUM + PC(I1)
   QG1 = QUM + QC(I1)
   IFL = IFO
   IF(QG1.GE.QMA(K1)*BASE)IFL=IF1
   IF(QG1.LE.QMI(K1)*BASE)IFL=IF2
   SG = SQRT(PG1*PG1 + QG1*QG1)
   WRITE(IWR, 2)PG1, QG1, IFL, SG
2 FORMAT(' GENERADOR', 7X, 2F8.1, A1, F7.1)
   QG(K1) = QG1
30 CONTINUE

```

C
C
C
C
C

SE CHECA SI HAY CARGA EN EL NODO

```

PC1 = ABS(PC(I1))
QC1 = ABS(QC(I1))
IF(PC1.GT.1.OE-4)GO TO 70
IF(QC1.LT.1.OE-4)GO TO 71
70 SG = SQRT(PC(I1)*PC(I1) + QC(I1)*QC(I1))
WRITE(IWR,3)PC(I1),QC(I1),SG
3 FORMAT('    A CARGA',8X,3F8.1)

```

C
C
C
C
C

SE IMPRIME SI HAY UN COMPENSADOR

```

71 IF(NS1.NE.1)GO TO 72
SG = 0.0
SN = ABS(SI)
WRITE(IWR,4)SG,SI,SN
4 FORMAT('    A SHUNT',7X,3F8.1)

```

C
C
C
C
C

SE IMPRIMEN FLUJOS EN CADA LINEA

```

72 CONTINUE
IF(KZ.GT.ISIZE)CALL ENCA

```

C

```

DO 73 K1 = 1, KZ
SG = SQRT(PL(K1)*PL(K1) + QL(K1)*QL(K1))
IFL = IF6
IF(T2(K1).GE.1.OE-8)IFL=IF3
73 WRITE(IWR,5)NM(K1),DNOMB(NM(K1)),PL(K1),QL(K1),SG,T2(K1),IFL
5 FORMAT('    A',I6,1X,A8,F7.1,2F8.1,F8.4,1X,A2)
SI = PC(I1) + SUM
SQ = QC(I1) + QUM
IF(IG.EQ.0)GO TO 74
SI = SI - PG1
SQ = SQ - QG1
74 IF(ABS(SQ).GT.0.5)GO TO 75
IF(ABS(SI).LT.0.5)GO TO 76
75 S = SQRT(SI*SI + SQ*SQ)
WRITE(IWR,6)SI,SQ,SG
6 FORMAT(' DESAJUSTE ',9X,F7.1,2F8.1)
76 CONTINUE
RETURN
END

```

```

C*****
C*****
C*****   REPFLU.FOR  SELECTOR INTERACTIVO DE REPORTES
C*****   DE ESTUDIOS DE FLUJOS
C*****   890501      @R. MOTA-PALOMINO
C*****
C
C
C      SUBROUTINE REPFLU(LETR1, LETR2, NNOD, NL2, NTUG, NTR, NSHU, M, N,
2  YR, YI, BI, NCT, NST, T1, NSH, VLS, BASE, E, F, E3, F3, DNOMB, PG, QG,
3  PC, QC, NU, QMA, QMI, NM, PL, QL, T2, ISLA, LTERM)
C
C
C      CHARACTER*8 NOMB, DNOMB(1)
C      CHARACTER*70 LETR1, LETR2
C      INTEGER M(1), N(1), NCT(1), NST(1), NSH(1), NM(1), NU(1)
C      INTEGER IRES, ISLA(1)
C      REAL E(1), F(1), E3(1), F3(1), PL(1), QL(1), T2(1)
C      REAL VLS(1), T1(1), QMA(1), QMI(1)
C      REAL PG(1), QG(1), PC(1), QC(1), BASE
C      REAL YR(1), YI(1), BI(1)
C
C
62 CONTINUE
   IWR = LTERM
   CALL CLEAR
   WRITE(LTERM, 20)
20  FORMAT(///3X, ' ** REPORTE COMPLETO ', 12X, ' 1 ')
   WRITE(LTERM, 25)
25  FORMAT(/3X, ' ** REPORTE POR NODO ', 12X, ' 2 ')
   WRITE(LTERM, 21)
21  FORMAT(/3X, ' ** FIN DE REPORTE ', 14X, ' 3 ')
   WRITE(LTERM, 30)
30  FORMAT(///' QUE OPCION DESEA : '$)
   READ(LTERM, 40, ERR=60) IRES
40  FORMAT(I2)
   IF(IRES.LT.1 .OR. IRES.GT.3)GO TO 60
   IF(IRES.EQ.2)GO TO 65
   IF(IRES.EQ.3)GO TO 3000
   GO TO 70
60 CONTINUE
   WRITE(LTERM, 61)
61  FORMAT(///10X, ' *** TIPOS DISPONIBLES DE 1 A 3, FORMATO I2')
   GO TO 62
65 CONTINUE
   CALL CLEAR
   WRITE(LTERM, 80)
80  FORMAT(///' ( QUE NODO ?'/8HXXXXXXXXXX)
   READ(LTERM, 82)NOMB
82  FORMAT(A8)
   IF(NOMB.EQ.'          ')GO TO 62
   DO 85 J = 1, NNOD
   IF(NOMB.EQ.DNOMB(J))GO TO 105
85  CONTINUE

```

```

WRITE(LTERM,106)NOMB
106 FORMAT(//1X,' ** EL NODO ',A8,' NO EXISTE ')
GO TO 62
105 CONTINUE
I1 = J
ILE = 0
ISIZE = 25
IF(ISLA(I1).GE.1)GO TO 500
WRITE(6,102)I1, DNOMB(I1)
ILE = ILE + 5
CALL ENCA
GO TO 65
500 CONTINUE
CALL CLEAR
CALL FLUCAL(I1, NNOD, NL2, NTUG, NTR, NSHU, M, N, YR, YI, BI, NCT,
2 NST, T1, NSH, VLS, BASE, E, F, E3, F3, DNOMB, PG, QG, PC, QC, NU,
3 QMA, QMI, NM, PL, QL, T2, ILE, ISIZE, IWR)
CALL ENCA
GO TO 65
70 CONTINUE
WRITE(LTERM,350)
350 FORMAT(//10X,' REPORTE EN LA IMPRESORA (IM) : '$)
READ(LTERM,357,ERR=401)NOMB
357 FORMAT(A2)
IF(NOMB.EQ.'IM'.OR.NOMB.EQ.'im')GO TO 400
IF(NOMB.NE.' ')GO TO 401
ISIZE = 21
IWR = LTERM
GO TO 402
401 CONTINUE
WRITE(LTERM,410)
410 FORMAT(//10X,' * RESPUESTA INCORRECTA, FORMATO A2')
GO TO 70
400 CONTINUE
IWR = 8
OPEN(UNIT=8, FILE='FLUJO. SAL')
ISIZE = 50
WRITE(IWR,700)LETR1, LETR2
700 FORMAT(//1X, A70, /, 1X, A70)
402 CONTINUE
DO 260 I1 = 1, NNOD
200 CONTINUE
ILE = 0
IF(ISLA(I1).GE.1)GO TO 201
WRITE(IWR,102)I1, DNOMB(I1)
102 FORMAT(//10X,' ** EL NODO ',A8,' ESTA AISLADO '//)
ILE = ILE + 5
GO TO 260
201 CONTINUE
IF(IWR.NE.8)CALL CLEAR
CALL FLUCAL(I1, NNOD, NL2, NTUG, NTR, NSHU, M, N, YR, YI, BI,
2 NCT, NST, T1, NSH, VLS, BASE, E, F, E3, F3, DNOMB, PG, QG, PC, QC,
3 NU, QMA, QMI, NM, PL, QL, T2, ILE, ISIZE, IWR)

```

```

        IF(IWR.NE.8)CALL ENCA
260  CONTINUE
        IF(IWR.EQ.8)CLOSE(UNIT=8)
        GO TO 62
3000 CONTINUE
        RETURN
        END

```

```

SUBROUTINE ENCA
CHARACTER*2 NOM
WRITE(6,100)
100 FORMAT(//3X,' ** Oprima la tecla de "entrada" para continuar...'$)
READ(6,102)NOM
102 FORMAT(A2)
RETURN
END

```

```

SUBROUTINE CLEAR
CHARACTER*3 HOME
CHARACTER*1 ESC
PARAMETER (HOME='[2J')
ESC = CHAR(27)
WRITE(6,10)ESC,HOME
10 FORMAT(A1,A3)
RETURN
END

```

```

C*****
C*****
C*****  REPDAT.FOR  SELECTOR INTERACTIVO DE REPORTE
C*****  DE PARAMETROS PARA ESTUDIOS DE FLUJOS
C*****  890501      @R. MOTA-PALOMINO
C*****
C

```

```

SUBROUTINE REPDAT(LETR1,LETR2,NNOD,NL2,NTUG,NTR,NSHU,M,N,
2  YR,YI,BI,NCT,NST,T1,NSH,VLS,BASE,E,F,E3,F3,DNOMB,PG,QG,
3  PC,QC,NU,QMA,QMI,NM,PL,QL,T2,ISLA,LTERM,IPP,R,X,B,NLIN)
CHARACTER*8 NOMB, DNOMB(1)
CHARACTER*70 LETR1,LETR2
INTEGER M(1),N(1),NCT(1),NST(1),NSH(1),NM(1),NU(1)
INTEGER IRES,ISLA(1),IPP(1)
REAL E(1),F(1),E3(1),F3(1),PL(1),QL(1),T2(1)
REAL VLS(1),T1(1),QMA(1),QMI(1)
REAL PG(1),QG(1),PC(1),QC(1),BASE
REAL YR(1),YI(1),BI(1),R(1),X(1),B(1)
62 CONTINUE
IWR = LTERM
CALL CLEAR
WRITE(LTERM,20)
20 FORMAT(///3X,' ** REPORTE COMPLETO ',12X,' 1 ')

```



```

WRITE(LTERM,25)
25 FORMAT(/3X,' ** REPORTE POR NODO ',12X,' 2 ')
WRITE(LTERM,21)
21 FORMAT(/3X,' ** FIN DE REPORTE ',14X,' 3 ')
WRITE(LTERM,30)
30 FORMAT(///' QUE OPCION DESEA : '$)
READ(LTERM,40,ERR=60)IRES
40 FORMAT(I2)
IF(IRES.LT.1 .OR. IRES.GT.3)GO TO 60
IF(IRES.EQ.2)GO TO 65
IF(IRES.EQ.3)GO TO 3000
GO TO 70
60 CONTINUE
WRITE(LTERM,61)
61 FORMAT(///10X,' *** TIPOS DISPONIBLES DE 1 A 3, FORMATO I2')
GO TO 62
65 CONTINUE
WRITE(LTERM,80)
80 FORMAT(///' ( QUE NODO ?'/8HXXXXXXXX)
READ(LTERM,82)NOMB
82 FORMAT(A8)
IF(NOMB.EQ.' ')GO TO 62
DO 85 J = 1, NNOD
IF(NOMB.EQ.DNOMB(J))GO TO 105
85 CONTINUE
WRITE(LTERM,106)NOMB
106 FORMAT(//1X,' ** EL NODO ',A8,' NO EXISTE ')
GO TO 62
105 CONTINUE
I1 = J
ILE = 0
ISIZE = 25
IF(ISLA(I1).GE.1)GO TO 500
WRITE(6,102)I1,DNOMB(I1)
ILE = ILE + 5
500 CONTINUE
CALL CLEAR
CALL PARCAL(I1,NNOD,NL2,NTUG,NTR,NSHU,M,N,YR,YI,BI,NCT,
2 NST,T1,NSH,VLS,BASE,E,F,E3,F3,DNOMB,PG,QG,PC,QC,NU,
3 QMA,QMI,NM,PL,QL,T2,ILE,ISIZE,IWR,IPP,R,X,B,NLIN)
CALL ENCA
GO TO 65
70 CONTINUE
WRITE(LTERM,350)
350 FORMAT(//10X,' REPORTE EN LA IMPRESORA (IM) : '$)
READ(LTERM,357,ERR=401)NOMB
357 FORMAT(A2)
IF(NOMB.EQ.'IM'.OR.NOMB.EQ.'im')GO TO 400
IF(NOMB.NE.' ')GO TO 401
ISIZE = 21
IWR = LTERM
GO TO 402
401 CONTINUE

```

```

        WRITE(LTERM,410)
410   FORMAT(//10X,' * RESPUESTA INCORRECTA, FORMATO A2')
        GO TO 70
400   CONTINUE
        IWR = 8
        OPEN(UNIT=8, FILE='PARAM. SAL')
        ISIZE = 50
        WRITE(IWR,700)LETR1,LETR2
700   FORMAT(//1X,A70,/,1X,A70)
402   CONTINUE
        DO 260 I1 = 1, NNOD
200   CONTINUE
        ILE = 0
        IF(ISLA(I1).GE.1)GO TO 201
        WRITE(IWR,102)I1, DNOMB(I1)
102   FORMAT(//10X,' ** EL NODO ',A8,' ESTA AISLADO '//)
        ILE = ILE + 5
        GO TO 260
201   CONTINUE
        IF(IWR.NE.8)CALL CLEAR
        CALL PARCAL(I1, NNOD, NL2, NTUG, NTR, NSHU, M, N, YR, YI, BI,
2       NCT, NST, T1, NSH, VLS, BASE, E, F, E3, F3, DNOMB, PG, QG, PC, QC,
3       NU, QMA, QMI, NM, PL, QL, T2, ILE, ISIZE, IWR, IPP, R, X, B, NLIN)
        IF(IWR.NE.8) CALL ENCA
260   CONTINUE
        IF(IWR.EQ.8) CLOSE(UNIT=8)
        GO TO 62
3000  CONTINUE
        RETURN
        END

```

```

C*****
      SUBROUTINE PARCAL(I1, NNOD, NL2, NTUG, NTR, NSHU, M, N, YR, YI, BI,
      2NCT, NST, T1, NSH, VLS, BASE, E, F, E3, F3, DNOMB, PG, QG, PC, QC,
      3NU, QMA, QMI, NM, PL, QL, T2, ILE, ISIZE, IWR, IPP, R, X, B, NLIN)
C*****
C*****
C*****      SUBROUTINE PARCAL PARA CALCULAR FLUJOS DE POTENCIA IN
C*****      TABULAR
C*****
C
      INTEGER M(1), N(1), NCT(1), NST(1), NSH(1), NM(1)
      2, NU(1), IPP(1)
      CHARACTER*8 DNOMB(1)
      REAL E(1), F(1), E3(1), F3(1), PL(1), QL(1), T2(1)
      REAL VLS(1), T1(1), QMA(1), QMI(1), R(1), X(1), B(1)
      REAL PG(1), QG(1), PC(1), QC(1)
      REAL SUM, QUM, EH, AB, CD, YR(1), YI(1), BASE, SI, SR, SG
      REAL BI(1), SN, SQ, PC1, QC1
      NS1 = 0
      DATA IF6/2H /

```

```

DATA IFO/2HRR/
DATA IF1/2HHH/
DATA IF2/2HLL/
DATA IF3/2HTR/
C   DO 1200 K = 1, NLIN
C   WRITE(6, 1202)K, R(K), X(K), B(K)
C1202  FORMAT(2X, I5, 2X, 3(F10.5, 2X))
C1200  CONTINUE
      IG = 0
      SUM = 0.0
      QUM = 0.0
      N1 = M(I1)
      N2 = N(N1)
      KZ = 0

C
C
      DO 10 K = 1, N2
        J = N1 + K
        KZ = KZ + 1
        MQ = N(J)
        IP2 = IPP(J)
        PL(KZ) = R(IP2)
        QL(KZ) = X(IP2)
        QG(KZ) = B(IP2)
        NM(KZ) = MQ
        T2(KZ) = 0.0
10    CONTINUE

C
C
      SE CHECAN TAPS EN TRANSFORMADORES CONECTADOS A ESTE NODO

      IF(NTR.EQ.0)GO TO 11
      DO 12 K1 = 1, NTR
        IF(NCT(K1).NE.I1)GO TO 13
        DO 14 K2 = 1, KZ
          IF(NM(K2).NE.NST(K1))GO TO 14
          T2(K2) = T1(K1)
          NM(K2) = - NM(K2)
14    CONTINUE
        GO TO 12
13    IF(NST(K1).NE.I1)GO TO 12
        DO 15 K2 = 1, KZ
          IF(NM(K2).NE.NCT(K1))GO TO 15
          T2(K2) = 1.0E-7
          NM(K2) = - NM(K2)
15    CONTINUE
12    CONTINUE
        DO 16 K2 = 1, KZ
          NM(K2) = IABS(NM(K2))
16    CONTINUE

C
C

```

C
C
C
C

SE CHECAN SHUNTS EN EL NODO

```

11 IF(NSHU.EQ.0)GO TO 23
    DO 21 K1 = 1, NSHU
        IF(NSH(K1).EQ.I1)GO TO 22
21 CONTINUE
    GO TO 23
22 SI = VLS(K1)
    SI = - 1.0/SI
    SR = 0.0
    NS1 = 1
23 CONTINUE

```

C
C
C
C
C

SE IMPRIMEN FLUJOS DE POTENCIA

```

AMAG = E(I1)
ANG = F(I1) * 180.0/3.1416
ILE = ILE + 4
IF(ILE .GT. ISIZE)CALL ENCA
WRITE(IWR,1)I1, DNOMB(I1), AMAG, ANG, I1
1 FORMAT(/' DE', I6, 1X, A8, 4X, 'R ', 5X, ' X ', 5X, 'Y/2',
23X, ' TAP', F7.4, F8.2, I5)

```

C
C
C
C
C

SE CHECA SI EL NODO ES GENERADOR

```

DO 25 K1 = 1, NTUG
    IF(IABS(NU(K1)).EQ.I1)GO TO 26
25 CONTINUE
    GO TO 30
26 IG = 1
    PG1 = PG(K1)
    QG1 = 0.0
    IFL = IFO
    N1 = M(I1)
    N2 = N(N1)
    CALL IC(I1, N1, N2, A, B3, E1, F1, NL2, E3, F3, YR, YI, NNOD, N)
    QG1 = QG1 + (F1*A - E1*B3)*BASE
    SG = SQRT(PG1*PG1 + QG1*QG1)
    WRITE(IWR,2)PG1, QG1, IFL, SG
2 FORMAT(' GENERADOR', 7X, 2F8.1, A1, F7.1, 5X, 'MVA')
    QG(K1) = QG1
30 CONTINUE

```

C
C
C
C
C

SE CHECA SI HAY CARGA EN EL NODO

```

C      PC1 = ABS(PC(I1))
C      QC1 = ABS(QC(I1))
C      IF(PC1.GT. 1.0E-4)GO TO 70
C      IF(QC1.LT. 1.0E-4)GO TO 71
C 70 SG = SQRT(PC(I1)*PC(I1) + QC(I1)*QC(I1))
C      WRITE(IWR,3)PC(I1),QC(I1),SG
C      3 FORMAT('      A CARGA',8X,3F8.1,5X,'MVA')
C
C
C      SE IMPRIME SI TIENE SHUNT
C
C
C 71 IF(NS1.NE.1)GO TO 72
C      SG = 0.0
C      SN = ABS(SI)
C      WRITE(IWR,4)SG,SI,SN
C      4 FORMAT('      A SHUNT',7X,3F8.1)
C
C
C      SE IMPRIMEN FLUJOS EN CADA LINEA
C
C
C 72 CONTINUE
C      IF(KZ.GT. ISIZE)CALL ENCA
C
C
C      DO 73 K1 = 1, KZ
C          IFL = IF6
C          IF(T2(K1).GE. 1.0E-8)IFL=IF3
C 73 WRITE(IWR,5)NM(K1),DNOMB(NM(K1)),PL(K1),QL(K1),QG(K1),T2(K1),IFL
C      5 FORMAT('      A',16,1X,A8,F7.4,1X,2(F7.4,1X),F7.4,1X,A2)
C      RETURN
C      END

```

C*****
C*****
C***** SUBROUTINA REPRES(.) PARA FORMAR REPORTES RESUMIDOS
C*****
C*****
SUBROUTINE REPRES(LTERM, NNOD, NACT, NLIN, NTUG, NISLA, NISA,
2 ISCOM, R, X, B, DNOMB, NU, M, N, PG, QG, QMA, QMI, BASE, PC, QC, ISLA, E, F,
3 E3, F3, NE, NR, IPP, YR, YI, NL2, NSHU, NSH, VLS)

```

C
C
C      INTEGER ISCOM(1), NU(1), M(1), N(1), ISLA(1), NE(1), NR(1),
C 2 IPP(1), NSH(1)
C      REAL R(1), X(1), B(1), PG(1), QG(1), QMA(1), QMI(1), PC(1), QC(1)
C      REAL E(1), F(1), E3(1), F3(1), YR(1), YI(1), VLS(1)
C      CHARACTER*8 DNOMB(1)
C      IWR = LTERM
C
C
C      SE CALCULA POTENCIA GENERADA EN COMPENSADORES
C

```

```

DO 510 I = 1, NISLA
  IG = ISCOM(I)
  IF(IG.GE.NNOD)GO TO 510
  DO 520 K = 1, NTUG
    IP = IABS(NU(K))
    IF(IP.EQ.IG)GO TO 525
520  CONTINUE
    GO TO 510
525  CONTINUE
    N1 = M(IG)
    N2 = N(N1)
    CALL IC(IG, N1, N2, A, B3, E1, F1, NL2, E, F, YR, YI, NNOD, N)
    PG(K) = (E1*A + F1*B3)*BASE
510 CONTINUE
19  CONTINUE
    CALL CLEAR
    WRITE(LTERM, 10)
10  FORMAT(/12X, ' RESUMEN DE GENERADORES ', 30X, ' 1 ')
    WRITE(LTERM, 11)
11  FORMAT(/12X, ' RESUMEN DE PARAMETROS ', 30X, ' 2 ')
    WRITE(LTERM, 12)
12  FORMAT(/12X, ' RESUMEN DE NODOS          ', 30X, ' 3 ')
    WRITE(LTERM, 13)
13  FORMAT(/12X, ' RESUMEN DE ISLAS          ', 30X, ' 4 ')
    WRITE(LTERM, 39)
39  FORMAT(/12X, ' RESUMEN IMPRESO          ', 30X, ' 5 ')
    WRITE(LTERM, 14)
14  FORMAT(/12X, ' SELECTOR DE REPORTES      ', 30X, ' 6 ')
    WRITE(LTERM, 15)
15  FORMAT(// ' QUE TIPO DE RESUMEN DESEA : '$)
    READ (LTERM, 16, ERR=17)IRE3
16  FORMAT(I2)
    IF(IRE3.LT.1.OR.IRE3.GT.6)GO TO 17
    IF(IRE3.EQ.1)GO TO 100
    IF(IRE3.EQ.2)GO TO 200
    IF(IRE3.EQ.3)GO TO 300
    IF(IRE3.EQ.4)GO TO 400
    IF(IRE3.EQ.5)GO TO 500
    IF(IRE3.EQ.6)GO TO 20
17  CONTINUE
    WRITE(LTERM, 18)
18  FORMAT(//10X, 'TIPOS DISPONIBLES DE 1 A 5, FORMATO I2 ')
    GO TO 19
500 CONTINUE
    IWR = 8
C
    OPEN(UNIT=8, FILE='RESUME. SAL')
100 CONTINUE
    IF(IWR.EQ.LTERM)CALL CLEAR
    WRITE(IWR, 101)
101  FORMAT(//2X, ' NUM      NODO      VOLTAJE COMPLEJO  POTENCIA ',
2    ' GENERADA  LIMITES INF. Y SUP. ')

```

```

WRITE(IWR, 102)
102 FORMAT(19X, 'MAG(PU) GRADOS      MW', 5X, 'MVAR',
2 5X, 'MVAR      MVAR')
DO 103 I = 1, NTUG
  I1 = IABS(NU(I))
  N1 = M(I1)
  N2 = N(N1)
  CALL IC(I1, N1, N2, A, B3, E1, F1, NL2, E, F, YR, YI, NNOD, N)
  D1 = F1*A - E1*B3
  D1 = D1 * BASE
  QQ = D1 + QC(I)
  ANG = F(I1) * 180.0/3.1416
  WRITE(IWR, 104) I1, DNOMB(I1), E(I1), ANG,
2  PG(I), QQ, QMI(I)*BASE, QMA(I)*BASE
104 FORMAT(2X, I4, 2X, A8, 2X, F7.4, 2X, F7.2, 3X, 4(F7.1, 2X))
103 CONTINUE
  IF(IWR.EQ.LTERM)CALL ENCA
  IF(IWR.EQ.8)GO TO 200
  GO TO 19
200 CONTINUE
  IF(IWR.EQ.LTERM)CALL CLEAR
  WRITE(IWR, 201)
201 FORMAT(/9X, 'DEL NODO AL NODO ', 4X, ' R ', 7X, ' X ', 7X,
2' Y/2 ', /, 34X, 'PU', 8X, 'PU', 9X, 'PU')
  DO 202 I = 1, NLIN
    N1 = NE(I)
    N2 = NR(I)
    WRITE(IWR, 203)DNOMB(N1), DNOMB(N2), R(I), X(I), B(I)
203 FORMAT(10X, A8, 2X, A8, 3X, F7.4, 3X, F7.4, 3X, F7.4)
202 CONTINUE
  IF(IWR.EQ.LTERM)CALL ENCA
  IF(NSHU.LT.1)GO TO 2001
  WRITE(IWR, 2005)
2005 FORMAT(/2X, ' NODO CON SHUNT ', 2X, ' SUC. (PU) ')
  DO 2000 IS = 1, NSHU
    WRITE(IWR, 2002)NSH(IS), DNOMB(NSH(IS)), VLS(IS)
2002 FORMAT(3X, I5, 2X, A8, 2X, F10.5)
2000 CONTINUE
  IF(IWR.EQ.LTERM)CALL ENCA
2001 CONTINUE
  IF(IWR.EQ.8)GO TO 300
  GO TO 19
300 CONTINUE
  IF(IWR.EQ.LTERM)CALL CLEAR
  WRITE(IWR, 301)
301 FORMAT(/2X, ' NUMERO      NODO      VOLTAJE COMPLEJO POTENCIA ',
2' DE CARGA')
  WRITE(IWR, 302)
302 FORMAT(5X, '      MAG(PU) GRADOS      MW      MVAR ')
  DO 303 K1 = 1, NNOD
    WRITE(IWR, 304)K1, DNOMB(K1), E(K1), F(K1)*180.0/3.1416, PC(K1),
2 QC(K1)
304 FORMAT(2X, I6, 2X, A8, 2X, F7.4, 2X, F7.2, 3X, F7.1, 2X, F7.1)
303 CONTINUE

```

```

IF(IWR.EQ.LTERM)CALL ENCA
IF(IWR.EQ.8)GO TO 400
GO TO 19
400 CONTINUE
IF(IWR.EQ.LTERM)CALL CLEAR
DO 401 I = 1, NTUG
  QG(I) = 0.0
401 CONTINUE
DO 402 I = 1, NTUG
  I1 = IABS(NU(I))
  N1 = M(I1)
  N2 = N(N1)
  CALL IC(I1, N1, N2, A, B3, E1, F1, NL2, E, F, YR, YI, NNOD, N)
  QG(I) = QG(I) + (F1*A - E1*B3)*BASE
402 CONTINUE
WRITE(IWR, 403)
403 FORMAT(/2X, ' ISLA COMPEN POTENCIA DE GENERACION ',
2 ' POTENCIA DE CARGA PERDIDAS ', /, 11X, ' SADOR MW ',
3 , 8X, ' MVAR', 8X, ' MW', 6X, ' MVAR ', 7X, ' MW', 6X, ' MVAR')
DO 404 I=1, NISLA
  QER = 0.0
  PER = 0.0
  PGT = 0.0
  QGT = 0.0
  PCT = 0.0
  QCT = 0.0
DO 405 J = 1, NNOD
  IF(IABS(ISLA(J)) .NE. I)GO TO 405
  PCT = PCT + PC(J)
  QCT = QCT + QC(J)
405 CONTINUE
DO 406 J = 1, NTUG
  I1 = IABS(NU(J))
  IF(ISLA(I1) .NE. I)GO TO 406
  QGT = QGT + QG(J)
  PGT = PGT + PG(J)
406 CONTINUE
PER = PGT - PCT
QER = QGT - QCT
WRITE(IWR, 407) I, DNOMB(ISCOM(I)), PGT, QGT, PCT, QCT, PER, QER
407 FORMAT(/6X, I2, 2X, A8, 1X, F8.2, 5X, 5(F8.2, 2X))
IF(ISCOM(I).LE.NNOD)GO TO 404
WRITE(IWR, 4000) I
4000 FORMAT(/3X, ' ** LA ISLA ', I5, 2X, ' ESTA INACTIVA ')
404 CONTINUE
IF(IWR.EQ.LTERM)CALL ENCA
IF(IWR.EQ.8)CLOSE(UNIT=8)
IWR = LTERM
GO TO 19
20 CONTINUE
RETURN
END

```