

ESCUELA SUPERIOR POLITECNICA DEL LITORAL

DEPARTAMENTO DE INGENIERIA ELECTRICA

"USO DE PASCAL EN LA ELABORACION DE UN PROGRAMA PARA
RESOLVER EL FLUJO DE CARGA USANDO EL METODO DESACOPLADO
RAPIDO".

TESIS DE GRADO

PREVIA A LA OBTENCION DEL TITULO DE:

INGENIERO EN ELECTRICIDAD

Especializacion: POTENCIA

PRESENTADA POR :

ALFREDO VILLACRESES PEÑA

GUAYAQUIL - ECUADOR

1.983

A G R A D E C I M I E N T O

AL ING. SERGIO FLORES MACIAS
DIRECTOR DE TESIS, POR SU AYUDA
Y COLABORACION PARA LA REALI
ZACION DE ESTE TRABAJO.

D E D I C A T O R I A

- A MI FAMILIA



ING. SERGIO FLORES MACIAS
DIRECTOR DE TESIS

DECLARACION EXPRESA

"LA RESPONSABILIDAD POR LOS HECHOS, IDEAS Y DOCTRINAS EXPUESTOS EN ESTA TESIS, ME CORRESPONDEN EXCLUSIVAMENTE; Y, EL PATRIMONIO INTELECTUAL DE LA MISMA, A LA ESCUELA SUPERIOR POLITECNICA DEL LITORAL".

(REGLAMENTO DE EXAMENES Y TITULOS PROFESIONALES DE LA ESPOL).



ALFREDO VILLACRESES PENA

R E S U M E N

A través de esta tesis fundamentalmente se elabora un programa para ser usado en la ESPOL, en el estudio de flujos de carga.

El programa corre en un microcomputador, los cuales son de bajo costo, como el APPLE II, esto es una gran ventaja ya que permite que cualquier compañía consultora, empresa eléctrica o ingeniero independiente pueda analizar sistemas de potencia.

En el CAPITULO I, se hace un breve análisis del problema de flujo de carga.

El CAPITULO II, da a conocer los principales métodos usados para obtener la solución del problema.

Uno de los grandes inconvenientes que se han presentado al resolver el flujo de carga es la velocidad para

su resolución, aquí se da como solución a este inconveniente, el uso del método desacoplado rápido. La descripción del mismo se la da en el CAPITULO III.

Con la finalidad de que el programa elaborado pueda posteriormente optimizarse, para la escritura del mismo se ha usado un lenguaje que tiene como una de sus características principales el dar facilidad para que cualquier programa sea fácilmente entendido y posteriormente modificado. El lenguaje usado es PASCAL y se hace un análisis del mismo en el CAPITULO IV.

La elaboración del programa se la tratará en el CAPITULO V y en él se explica la técnica que ha sido usada para ahorrar memoria de computadora, ya que este es otro de los inconvenientes que existen al procesar problemas de sistemas de potencia, peor aún si se usa una microcomputadora como en nuestro caso.

Para probar la validez del programa se han corrido dos sistemas, uno de ellos el Guaya-

quil, el otro es un sistema tipo que aparece en - las publicaciones de IEEE, los resultados se muestran en el CAPITULO VI..

Con el objeto de dar facilidad para poder mejorar el programa, se hacen recomendaciones que a juicio del autor son claves para ello.

Finalmente en el APENDICE A, se dan las reglas para usar en un microcomputador APPLE I, el programa elaborado.

INDICE GENERAL

PAGS

RESUMEN -----	VI
INDICE GENERAL -----	IX
INDICE DE FIGURAS -----	XII
INDICE DE TABLAS -----	XIII
INTRODUCCION -----	14
CAPITULO I	
ANALISIS DEL PROBLEMA DE FLUJO DE CARGA -----	17
1.1. GENERALIDADES -----	17
1.2. DEFINICION DEL PROBLEMA DE FLUJO DE CARGA-----	18
1.3. ECUACIONES QUE DETERMINA EL FLUJO DE CARGA	19
1.4. CARACTERISTICAS DE LAS ECUACIONES -----	23
1.5. SOLUCION DEL PROBLEMA -----	23
CAPITULO II	
BREVE DESCRIPCION DE LOS PRINCIPALES METODOS - USADOS EN LA RESOLUCION DEL FLUJO DE CARGA-----	25
2.1. METODOS ITERATIVOS -----	25
2.1.1. Método de Gauss -----	26
2.1.2. Método de Gauss - Seidel -----	28

Inv. No. POT- 038

	PAGS
2.1.3. Método de Newton - Raphson -----	29
2.2. METODOS DIRECTOS -----	34
2.2.1. Eliminación Gaussiana -----	34
2.2.2. Inversión de Matrices -----	37
2.3. METODOS DESACOPLADOS -----	39
2.4. OTROS METODOS (7) -----	39
CAPITULO III	
ESTUDIO DEL METODO DESACOPLADO RAPIDO -----	42
3.1. GENERALIDADES -----	42
3.2. NOTACION USADA EN ESTE METODO -----	47
3.3. ALGORITMO BASICO (6) -----	48
3.4. ALMACENAMIENTO Y COMPACTACION -----	53
3.5. ESQUEMA BASICO PARA LA SOLUCION -----	54
CAPITULO IV	
PROGRAMACION EN PASCAL PARA RESOLVER PROBLEMAS DE SISTEMAS DE POTENCIA -----	56
4.1. CARACTERISTICAS PRINCIPALES DE ESTE LENGUAJE -----	56
4.2. COMPARACION CON OTROS LENGUAJES CIENTIFICOS-----	58
4.3. USO EN PASCAL EN SISTEMAS DE POTENCIA -----	59
CAPITULO V	
ELABORACION DE UN PROGRAMA EN PASCAL PARA SOLUCIONAR EL PROBLEMA DE FLUJO DE CARGA POR EL METODO - DESACOPLADO RAPIDO -----	62

5.1. ESTRUCTURA DEL PROGRAMA -----	62
5.1.1. Entradas del Programa -----	62
5.1.2. Procesamiento de Datos -----	63
5.1.3. Salidas del programa -----	64
5.2. DIAGRAMA DE FLUJO DEL PROGRAMA -----	65
5.3. METODO Y TECNICAS NUMERICAS USADAS EN EL PROGRAMA -----	86
5.3.1. Descomposición triangular -----	86
5.3.2. Bi-factorización -----	88
5.3.3. Ordenamiento óptimo de matrices ---	99
5.3.4. Empaquetamiento -----	102
5.3.5. Flujo de carga por el método desaco- plado rápido -----	107
CAPITULO VI	
APLICACIONES DEL PROGRAMA ELABORADO -----	108
6.1. SISTEMA TIPO IEEE (Figura 6.1)-----	108
6.2. SISTEMA GUAYAQUIL -----	112
6.3. COMPARACION CON EL PROGRAMA EXISTENTE EN LA ESPOL -----	112
CONCLUSIONES Y RECOMENDACIONES -----	123
APENDICE A -----	128
BIBLIOGRAFIA -----	131

INTRODUCCION

Debido a la importancia que tiene el flujo de carga en el análisis de sistemas de potencia se ha desarrollado esta tesis, con el fin de proporcionar un programa que pueda ser usado para dicho estudio en una microcomputadora.

A mediados de los años cincuenta, desde que se comenzaron a disponer de computadoras digitales empezó la preparación de programas para ser empleados en sistemas de potencia, desde aquella época hasta hoy se han venido desarrollando una serie de técnicas y métodos para ser usados en los programas con el objeto de optimizarlos.

Los mejores programas son aquellos que dan solución al flujo de carga en forma rápida y usando la menor cantidad de memoria de computadora.

En este trabajo se describen los métodos más conocidos y se desarrolla el programa en base a uno que es completamente nuevo, el mismo que se conoce como desaco-

plado rápido. Este trata el flujo de carga como dos problemas independientes, tomando como uno de ellos la relación Potencia activa-ángulo del voltaje y el otro la relación Potencia reactiva-módulo del voltaje.

Así también se ha tomado en cuenta la cantidad de memoria para almacenamiento a utilizar, por ello se han descrito técnicas que permiten disponer el menor espacio de almacenamiento y también reducir el número de operaciones a efectuar; las técnicas aquí usadas son relativamente nuevas en nuestro medio, son conocidas como compactación (para el almacenamiento) y bi-factorización (para reducir el número de operaciones). La compactación consiste en almacenar solamente los elementos que son diferentes a cero y la bi-factorización obtiene la inversa de una matriz utilizando los elementos almacenados en la compactación.

Lo dicho anteriormente puede ser utilizado en todo lenguaje empleado para programación de computadoras, sin embargo, con el fin de que el trabajo presentado pueda ser mejorado y por ende optimizado se ha usado PASCAL, que es un lenguaje estructurado, que tiene como una de sus principales características el permi-

tir facilmente entender y luego modificar programas existentes.

CAPITULO I

ANALISIS DEL PROBLEMA DE FLUJO DE CARGA

1.1. GENERALIDADES

Flujo de carga (o flujo de potencias) es la solución bajo condiciones estáticas de operación de un sistema eléctrico de potencias y es la rutina usada con más frecuencia en computadoras digitales dentro del campo de redes eléctricas.

Generalmente los estudios de flujo de carga se realizan para investigar lo siguiente:

- Flujo de potencias activa y reactiva en cada línea de una red.
- Voltaje en cada barra del sistema.
- Efecto que tiene en el sistema el hecho de - rearreglar e incorporar nuevos circuitos.
- Efectos en el sistema al ocurrir pérdidas - temporales de generación y/o carga.
- Optimizar las pérdidas del sistema.

En resumen se puede afirmar que los estudios de flujo de carga se usan para planificar y controlar la operación en estado estable de los sistemas de potencia.

Los estudios normalmente se realizan para condiciones mínimas de carga para analizar la posibilidad de que haya inestabilidad en el sistema - debido a la existencia de niveles altos de voltaje y para condiciones máximas de carga para investigar la posibilidad de que haya inestabilidad sincrónica. El diseño y operación de una red de potencia para obtener una economía óptima es de suma importancia y la promoción de esta idea ha dado lugar al uso de controles automáticos centrales en las estaciones de generación.

1.2. DEFINICION DEL PROBLEMA DE FLUJO DE CARGA

El estudio de flujo de carga consiste en el cálculo de voltajes y flujos de potencia en una red para determinadas condiciones de barras. Con cada barra se encuentran asociadas cuatro variables:

Las potencias activa y reactiva, la magnitud del voltaje y el desplazamiento fasorial o angular - del voltaje.

De acuerdo a lo especificado para cada una de las barras se las puede clasificar como sigue:

- Barra oscilante: Es necesario seleccionar una barra que deberá proporcionar las potencias adicionales requeridas para satisfacer las pérdidas en las líneas de transmisión. Para esta barra la magnitud y ángulo del voltaje son constantes (se toma generalmente este ángulo 0° como referencia para los demás voltajes del sistema).
- Barra de carga: Son aquellas en las que se conoce la potencia compleja $S = P \pm jQ$.
- Barras de generación (o de voltaje controlado) en estas se conocen la magnitud del voltaje y la potencia activa.

1.3. ECUACIONES QUE DETERMINA EL FLUJO DE CARGA

A continuación se detalla la simbología usada en

las fórmulas que se van a exponer:

[I] : representa la matriz de las corrientes que fluyen de o a cada uno de los nodos de la red.

[E] : representa la matriz de los voltajes de los nodos del sistema,

[Y] : representa la matriz de admitancias de la red.

P_p : potencia activa de la barra "p".

E_p : voltaje de la barra "p" (fasor).

E_p^* : conjugada del voltaje de la barra "p"

I_p : corriente de la barra "p" (fasor).

Y_{pq} : admittance de la línea que une las barras "p" y "q".

Y'_{pq} : admittance de carga de la línea.

P_{pq} : potencia activa que fluye de la barra "p" a la barra "q".

Q_{pq} : potencia reactiva que fluye de la barra "p" a la "q".

V_p : magnitud del voltaje en el nodo "p".

θ_p : ángulo del voltaje en el nodo "p".

P_p^{sp} : potencia activa especificada para el nodo "p".

Q_p^{sp} : potencia reactiva especificada para el nodo "p".

θ_{pq} : $\theta_p - \theta_q$.

Una red eléctrica tal como un sistema de potencia puede ser representada por un sistema de ecuaciones nodales en la siguiente forma:

$$[I] = [Y] [E] \quad (1.1)$$

Las potencias real y reactiva en una barra "p" esta dada por:

$$P_p + jQ_p = E_p^* I_p \quad (1.2)$$

$$\therefore I_p = \frac{P_p + jQ_p}{E_p^*} \quad (1.3)$$

La corriente que sale de una barra cualquiera "p" hacia una barra "q" esta dada por:

$$I_{pq} = (E_p - E_q) Y_{pq} + E_p (Y'_{pq}/2) \quad (1.4)$$

Las potencias real y reactiva fluyendo en la linea de la barra "p" a la barra "q" se calculan en base a la siguiente formula:

$$P_{pq} - jQ_{pq} = E_p^* (E_p - E_q) Y_{pq} + E_p^* E_p (Y'_{pq}/2) \quad (1.5)$$

1.4. CARACTERISTICAS DE LAS ECUACIONES

Las características principales son las siguientes:

- Las ecuaciones son algebraicas debido a que ellas representan el modelo de un sistema estático, o un sistema operando en estado estable.
- Las ecuaciones son no lineales.
- Es importante hacer notar que en la ecuación matricial (1.1) la matriz $[Y]$ es una matriz cuadrada, dispersa (con bastantes elementos iguales a cero) y simétrica.

1.5. SOLUCION DEL PROBLEMA

La solución de las ecuaciones algebraicas que describen un sistema de potencia, está basada fundamentalmente en una técnica iterativa debida a su no-linealidad. La solución debe satisfacer las leyes de Kirchoff, así por ejemplo la suma algebraica de las potencias entrando y saliendo de cada nodo debe ser igual a cero.

tiendo de una barra debe ser igual a cero.

La solución se obtiene resolviendo un sistema de ecuaciones formado por dos ecuaciones por cada nodo de carga (ya que se conoce la potencia reactiva y activa de cada nodo) y una ecuación por cada nodo de generación (se conoce la potencia activa). A partir del sistema antes mencionado se calculan los voltajes para cada barra y con ellos se calculan para el nodo oscilante, las potencias activas y reactivas generadas así como los flujos de carga por cada una de las líneas de transmisión del sistema que se está estudiando.

CAPITULO II

BREVE DESCRIPCION DE LOS PRINCIPALES METODOS USADOS EN LA RESOLUCION DEL FLUJO DE CARGA

Una gran parte del tiempo empleado en la resolución del flujo de carga es usado en la solución de las ecuaciones (1.1) y (1.2). Por lo tanto es importante analizar los métodos más conocidos, los mismos que de acuerdo a la clasificación que se encuentra en álgebra lineal se dividen en directos e iterativos.

2.1. METODOS ITERATIVOS

Estos métodos resuelven simultáneamente las ecuaciones (1.1) y (1.2), los más conocidos en orden de simplicidad son:

- Método de Gauss
- Método de Gauss - Seidel
- Método de Newton - Raphson

2.1.1. Método de Gauss

El sistema de ecuaciones a resolver es de la forma:

$$f_1(x_1, x_2, \dots, x_n) = 0$$

$$f_2(x_1, x_2, \dots, x_n) = 0$$

.....

.....

$$f_n(x_1, x_2, \dots, x_n) = 0 \quad (2.1)$$

El método de Gauss consiste en obtener de cada una de las ecuaciones anteriores las variables buscadas, resultando un sistema de ecuaciones como el siguiente:

$$x_1 = g_1(x_1, x_2, \dots, x_n)$$

$$x_2 = g_2(x_1, x_2, \dots, x_n)$$

.....

.....

(2.2)

$$x_n = g_n(x_1, x_2, \dots, x_n)$$

Luego se asumen valores iniciales arbitrarios para cada una de las variables $(x_1^0, x_2^0, \dots, x_n^0)$ y usando las ecuaciones (2.2) se calculan nuevos valores para las variables buscadas:

$$\begin{aligned} x_1^1 &= g_1(x_1^0, x_2^0, \dots, x_n^0) \\ x_2^1 &= g_2(x_1^0, x_2^0, \dots, x_n^0) \\ &\dots \\ &\dots \\ x_n^1 &= g_n(x_1^0, x_2^0, \dots, x_n^0) \end{aligned} \quad (2.3)$$

En general para una iteración "m" se tiene que:

$$\begin{aligned} x_1^m &= g_1(x_1^{m-1}, x_2^{m-1}, \dots, x_n^{m-1}) \\ x_2^m &= g_2(x_1^{m-1}, x_2^{m-1}, \dots, x_n^{m-1}) \\ &\dots \\ &\dots \\ x_n^m &= g_n(x_1^{m-1}, x_2^{m-1}, \dots, x_n^{m-1}) \end{aligned} \quad (2.4)$$

El procedimiento anterior se continua hasta que la diferencia entre las variables calculadas en dos iteraciones consecutivas es menor que un cierto valor de tolerancia

$$|x_i^m - x_i^{m-1}| < \text{tolerancia} \quad (2.5)$$

donde $i = 1, 2, \dots, n$. Se puede ver claramente que éste método es muy sencillo, pero su principal desventaja radica en que converge lentamente a la solución.

Generalmente se requieren unas 100 iteraciones antes de llegar al resultado con suficiente exactitud.(2)

2.1.2. Método de Gauss - Seidel

Este método surgió al hacer una modificación al método de Gauss para aumentar la velocidad de convergencia. Al igual que el método de Gauss se parte de un sistema de ecuaciones como el (2.2) y con valores iniciales arbitrarios para cada -

una de las variables. Usando la primera ecuación del sistema (2.2) se obtiene - x_1^1 :

$$x_1^1 = g_1(x_1^0, x_2^0, \dots, x_n^0) \quad (2.6)$$

El siguiente paso consiste en obtener - x_2^1 , pero en lugar de usar x_1^0 como en el método de Gauss, se calcula x_2^1 usando - el valor recién obtenido para x_1 :

$$x_2^1 = g_2(x_1^1, x_2^0, \dots, x_n^0) \quad (2.7)$$

Para una variable cualquiera "n" en la iteración "m" se tendría:

$$x_n^m = g_n(x_1^m, x_2^m, \dots, x_{n-1}^m) \quad (2.8)$$

De la misma forma que en el método de - Gauss el procedimiento se sigue hasta - que se satisface la ecuación (2.3).

2.1.3. Método de Newton - Raphson

Los métodos tratados anteriormente, como

se ha podido observar son fáciles de comprender y por lo mismo la elaboración de programas, para resolver el flujo de carga usando computadoras, no es complicada. Sin embargo ambos métodos convergen a menor velocidad que el método de Newton.

La mayor ventaja del método de Newton es que garantiza convergencia en la mayoría de los casos.

Para una mejor explicación del método consideremos el sistema de ecuaciones (2.1).

Si asumimos que los valores iniciales para las variables buscadas son $x_1^0, x_2^0, \dots, x_n^0$ tendremos que a cada una de ellas le hará falta un incremento ($\Delta x_1, \Delta x_2, \dots, \Delta x_n$) para obtener la solución exacta.

Por lo dicho anteriormente el sistema de ecuaciones (2.1) se puede expresar en la siguiente forma:

$$f_1(x_1^0 + \Delta x_1, x_2^0 + \Delta x_2, \dots, x_n^0 + \Delta x_n) = 0$$

$$f_2(x_1^0 + \Delta x_1^0, x_2^0 + \Delta x_2^0, \dots, x_n^0 + \Delta x_n^0) = 0$$

.....

$$f_n(x_1^0 + \Delta x_1^0, x_2^0 + \Delta x_2^0, \dots, x_n^0 + \Delta x_n^0) = 0$$

Expandiendo estas ecuaciones en series de Taylor tenemos:

$$f_1(x_1^0, x_2^0, \dots, x_n^0) + x_1 \frac{\delta f_1}{\delta x_1} \Big|_{x_1^0} + \dots + \Delta x_n \frac{\delta f_1}{\delta x_n} \Big|_{x_n^0}$$

$$f_2(x_1^0, x_2^0, \dots, x_n^0) + \Delta x_1^0 \frac{\delta f_2}{\delta x_1} \Big|_{x_1^0} + \dots$$

$$\dots + \Delta x_n \frac{\delta f_2}{\delta x_n} \Big|_{x_n^0}$$

.....

$$f_n(x_1^0, x_2^0, \dots, x_n^0) + \Delta x_1 \frac{\delta f_n}{\delta x_1} \Big|_{x_1^0} + \dots$$

$$\dots + \Delta x_n \frac{\delta f_n}{\delta x_n} \Big|_{x_n^0} \quad (2.9)$$

Donde:

$$\frac{\delta f_n}{\delta x_1} \Big|_{x_1^0}$$

significa derivada parcial de f_n con respecto a la variable x_1 evaluada para x_1 igual a x_1^0 .

Escribiendo las ecuaciones (2.9) en forma matricial tenemos que:

$$[f^{(0)}] + [J^{(0)}] [\Delta x^{(0)}] = 0 \quad (2.10)$$

Donde:

$$[f^{(0)}] = \begin{bmatrix} f_1(x_1^0, x_2^0, \dots, x_n^0) \\ f_2(x_1^0, x_2^0, \dots, x_n^0) \\ \dots \\ \dots \\ f_n(x_1^0, x_2^0, \dots, x_n^0) \end{bmatrix}$$

$$[J(0)] = \begin{bmatrix} \frac{\delta f_1}{\delta x_1} |_{x_1^0} & \dots & \frac{\delta f_1}{\delta x_n} |_{x_n^0} \\ \dots & \ddots & \dots \\ \frac{\delta f_n}{\delta x_1} |_{x_1^0} & \dots & \frac{\delta f_n}{\delta x_n} |_{x_n^0} \end{bmatrix}$$

$$[\Delta x^{(0)}] = \begin{bmatrix} \Delta x_1 \\ \Delta x_2 \\ \vdots \\ \vdots \\ \Delta x_n \end{bmatrix} \quad (3.11)$$

De la ecuación (3.10) tenemos que:

$$[\Delta x^{(0)}] = -[J^{(0)}]^{-1} [f^{(0)}] \quad (3.12)$$

Resolviendo la ecuación (3.12) se obtienen los incrementos deseados, y la solución buscada será:

$$x_i^1 = x_i^0 + \Delta x_i^0 \quad i = 1, 2, \dots, n \quad (3.13)$$

El proceso se continua al igual que en los métodos anteriores hasta que se satisfaga la ecuación (2.3)

2.2. METODOS DIRECTOS

Los métodos directos únicamente resuelven sistemas lineales tal como el sistema representado por la ecuación (1.1). El hecho de que en el estudio de flujo de carga se especifican las potencias convierte el problema en no lineal ya que para obtener los voltajes usando la ecuación (1.1), es necesario previamente calcular las corrientes usando la ecuación (1.3). Lo dicho anteriormente implica que para obtener la solución de las ecuaciones (1.1) y (1.3), se pueden emplear los métodos directos, pero al igual que en los métodos iterativos - se tiene que partir de valores arbitrarios de voltajes para obtener las corrientes con la ecuación (1.3).

2.2.1. Eliminación Gaussiana

Para describir este método consideremos el sistema de ecuaciones que mostramos en la página siguiente:

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + a_{14}x_4 &= b_1 \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + a_{24}x_4 &= b_2 \end{aligned} \quad (2.13)$$

$$a_{31}x_1 + a_{32}x_2 + a_{33}x_3 + a_{34}x_4 = b_3$$

$$a_{41}x_1 + a_{42}x_2 + a_{43}x_3 + a_{44}x_4 = b_4$$

El primer paso consiste en dividir la primera ecuación para a_{11} obteniendo:

$$a_{ij}^{(1)} = (1/a_{11})a_{1j} \quad j = 2, \dots, n$$

$$b_1^{(1)} = (1/a_{11})b_1 \quad (2.14)$$

Los superíndices indican la etapa del proceso. La segunda etapa es eliminar a_{21} de la segunda ecuación y hacer uno el coeficiente de x_2 , esto se consigue multiplicando la primera ecuación modificada por a_{21} , luego restándola de la segunda y finalmente dividiéndola por el nuevo coeficiente de x_2 obteniendo:

$$x_1 + a_{12}^{(1)}x_2 + a_{13}^{(1)}x_3 + a_{14}^{(1)}x_4 = b_1^{(1)}$$

$$x_2 + a_{23}^{(2)}x_3 + a_{24}^{(2)}x_4 = b_2^{(2)} \quad (2.15)$$

$$a_{31}x_1 + a_{32}x_2 + a_{33}x_3 + a_{34}x_4 = b_3$$

$$a_{41}x_1 + a_{42}x_2 + a_{43}x_3 + a_{44}x_4 = b_4$$

Donde:

$$a_{2j}^{(1)} = a_{2j} - a_{21}a_{1j}^{(1)} \quad j = 2, \dots, n$$

$$b_2^{(1)} = b_2 - a_{21}b_1^{(1)} \quad (2.16)$$

$$a_{2j}^{(2)} = (1/a_{22}^{(1)}) a_{2j}^{(1)} \quad j = 3, \dots, n$$

$$b_2^{(2)} = (1/a_{22}^{(1)}) b_2^{(1)} \quad (2.17)$$

De igual forma se continua con las ecuaciones 3 y 4 del sistema (2.13) hasta finalmente obtener:

$$x_1 + a_{12}^{(1)}x_2 + a_{13}^{(1)}x_3 + a_{14}^{(1)}x_4 = b_1^{(1)}$$

$$x_2 + a_{23}^{(2)}x_3 + a_{24}^{(2)}x_4 = b_2^{(2)}$$

$$x_3 + a_{34}^{(3)}x_4 = b_3^{(3)} \quad (2.18)$$

$$x_4 = b_4^{(4)}$$

Es fácil notar que a partir de la ecuación cuatro del sistema (2.18) se obtiene el valor de x_4 . Para obtener los valores de las variables restantes se lo hace por sustitución inversa, lo cual expresado en forma general se tiene la siguiente ecuación:

$$x_i = b_i^{(i)} - \sum_{j=i+1}^n a_{ij}^{(i)}x_j \quad (2.19)$$

2.2.2. Inversión de Matrices

Esta es la forma más directa por la cual se resuelven las ecuaciones (1.1) y (1.2).

Despejando los voltajes de la ecuación (1.1) tenemos;

$$[\epsilon] = [Y^{-1}] \cdot [i] \quad (2.20)$$

donde $[Y^{-1}]$ es la inversa de la matriz Y .

La forma más común por la que se obtiene la inversa de una matriz es por medio de la evaluación del determinante de la matriz y los cofactores de cada uno de los elementos de ella.

Así por ejemplo se tiene que:

$$[Y] = \begin{bmatrix} Y_{11} & Y_{12} & Y_{13} & \dots & Y_{1n} \\ Y_{21} & Y_{22} & Y_{23} & \dots & Y_{2n} \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ Y_{n1} & Y_{n2} & Y_{n3} & \dots & Y_{nn} \end{bmatrix}$$

La inversa de ésta matriz será:

$$[Y^{-1}] = \begin{bmatrix} \frac{Y_{11}}{|Y|} & \frac{Y_{12}}{|Y|} & \dots & \frac{Y_{1n}}{|Y|} \\ \frac{Y_{21}}{|Y|} & \frac{Y_{22}}{|Y|} & \dots & \frac{Y_{2n}}{|Y|} \\ \dots & \dots & \dots & \dots \\ \frac{Y_{n1}}{|Y|} & \frac{Y_{n2}}{|Y|} & \dots & \frac{Y_{nn}}{|Y|} \end{bmatrix}$$

Donde Y_{11} , Y_{12} , etc., son los cofactores de los elementos y_{11} , y_{12} , etc., y $|Y|$ es el determinante de la matriz Y . Actualmente éste método es el menos usado debido a la gran cantidad de operaciones y memoria que se requieren para su utilización.

2.3. METODOS DESACOPLADOS

Una característica inherente de cualquier sistema de potencia operando en estado estable es la fuerte interdependencia entre las potencias activas y los ángulos de los voltajes, y entre las potencias reactivas y las magnitudes de los voltajes.

Así mismo es notable que el acoplamiento entre las componentes " $P = \theta$ " y " $Q = V$ " es débil.

Los métodos que se valen de ésta característica - son conocidos como métodos desacoplados, ya que se tratan separadamente los problemas " $P = \theta$ " y " $V-Q$ ".

2.4. OTROS METODOS (7)

En este punto se da a conocer la existencia de métodos que no entran en las categorías mencionadas

anteriormente.

Existen dos métodos que son de interés, uno de ellos es conocido como método de Minimización y el otro se conoce como transiente - respuesta análoga.

A. El método de Minimización convierte el problema de flujo de carga en la minimización de una función arbitraria f_0 , usualmente igual a la suma de los cuadrados de los desajustes de potencias o corrientes. El punto mínimo global en que $f_0 = 0$ coincide con la solución del flujo de carga.

B. El otro método que ha sido considerado de interés en éste trabajo es el que hace una analogía del problema de flujo de carga con una respuesta transiente. Este método sostiene que si el cálculo digital etapa por etapa de una respuesta transiente es continuado hasta que el sistema alcanza su estado estable, ésta será la solución al flujo de carga.

Esta idea se adapta expresando cada barra P-V

como una ecuación diferencial:

$$\frac{d^2\theta_i}{dt^2} = \Delta P_i / H_i$$

Donde H_i es una inercia artificial. Cada barra $P - Q$ es representada por la siguiente ecuación dinámica:

$$\frac{d^2V_i}{dt^2} = \Delta Q_i / G_i$$

El sistema se resuelve integrando etapa por etapa, donde los valores iniciales erróneos de voltajes para cada barra proveen la perturbación al sistema.

Existen otros métodos que combinan varios métodos aquí tratados y por ende no se los menciona como algo nuevo.

CAPITULO III

ESTUDIO DEL METODO DESACOPLADO RAPIDO

3.1. GENERALIDADES

Este método es simple, práctico y además proporciona con extremada rapidez la solución del flujo de carga, dando como resultado un amplio campo de aplicaciones.

Es usado en soluciones exactas cuando se trata por ejemplo de un estudio de pérdidas de un sistema o en soluciones aproximadas (se requiere con mayor rapidez la solución) en tiempo real de operación cuando el sistema es sometido a algún tipo de contingencia. El método puede ser aplicado para redes de cualquier tamaño y debido al ahorro de operaciones y memoria de almacenamiento que permite, puede ser implementado eficientemente para computadoras de capacidad limitada (micro-computadoras). Anteriormente ya se mencionó que el método se basa en el principio de que existe un desacoplamiento entre "P - θ" y "Q - V" en un sistema de potencia.

"Este método da soluciones en aproximadamente 4 a 7 iteraciones para soluciones con una exactitud típica de 0.01 y cada iteración demora aproximadamente 1 1/2 veces más que una iteración usando el método de Gauss - Seidel y 1/5 de lo que demora una iteración usando el método de Newton" (6).

Al hablar de desacoplamiento entre "P-θ" y "Q-V", se refiere a que, por ejemplo al variar la carga que pasa por una línea de transmisión, son afectados: la magnitud del voltaje por la variación de la potencia reactiva y el ángulo entre los voltajes de las barras, que une la línea de transmisión, por la variación de la potencia activa.

Para demostrar lo dicho anteriormente consideremos la figura 3.1 que presenta un sistema simple de potencia, en el cual un generador está enviando potencia a la carga L a través de la línea que tiene una impedancia $Z = R + jX$.

Observando el diagrama fasorial mostrado en la figura 3.1.b. tenemos que:

$$\epsilon^2 = (V + \Delta V)^2 + \delta V^2$$

$$\epsilon^2 = (V + R_I \cos \theta + X_I \sin \theta)^2 + (X_I \cos \theta - R_I \sin \theta)^2$$

$$\epsilon^2 = (V + \frac{RP}{V} + \frac{XQ}{V})^2 + (\frac{XP}{V} - \frac{RQ}{V})^2$$

De lo anterior tenemos que:

$$\Delta V = \frac{RP + XQ}{V}$$

$$Y \quad \delta V = \frac{XP - RQ}{V}$$

$$SI \quad \delta V \ll (V + \Delta V)$$

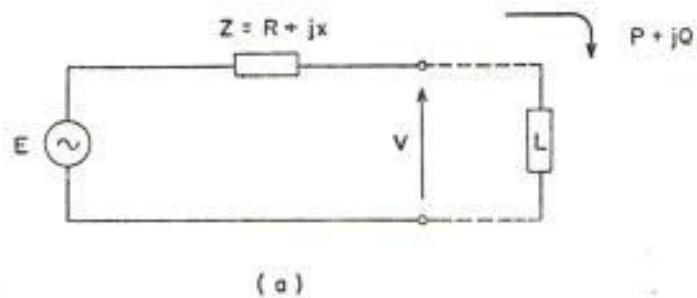
$$\epsilon^2 = (V + \frac{RP + XQ}{V})^2$$

$$E - V = \frac{RP + XQ}{V} = \Delta V$$

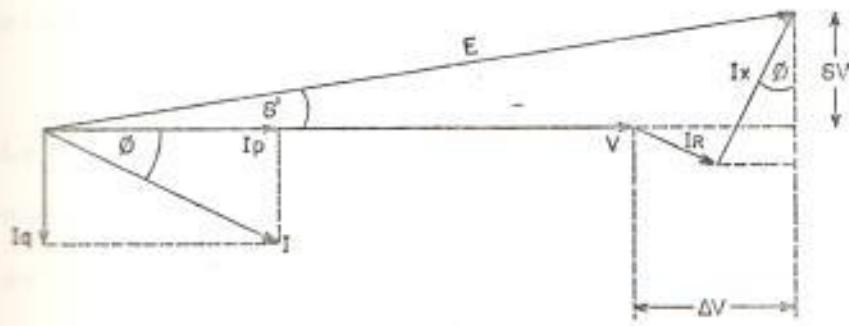
Por lo tanto la variación de voltaje (magnitud) estará dada por la ecuación anterior.

$$SI \quad R = 0$$

$$E - V = \frac{XQ}{V}$$



(a)



(b)

Fig. 3.1.- Diagrama fasorial de un sistema transmitiendo potencia a traves de una impedancia en serie a una carga L .

La variación del voltaje dependerá de la variación de la potencia reactiva Q.

El ángulo se obtiene de la siguiente relación:

$$\operatorname{sen}^{-1} \left(\frac{\delta V}{E} \right) = \frac{(X_P - RQ)/V}{E}$$

La variación de δ' dependerá de P ya que $R=0$.

Como conclusión se puede decir que el desacoplamiento dependerá de la relación $X = R$.

Lo asumido anteriormente es a menudo cierto ya que para la mayoría de las líneas de un sistema de potencias la resistencia es despreciable comparada con la reactancia y se introduce tan sólo un pequeño error al despreciarla.

Por ejemplo si $R = 0.1 X$ el error en el cálculo de los voltajes es de 0.49 por ciento y si $R = 0.4 X$ el error es de 7.7 por ciento (16).

3.2. NOTACION USADA EN ESTE METODO

Otra forma de expresar la ecuación (1.2) es la siguiente:

$$P_i + jQ_i = V_i e^{j\theta_i} \sum_{k=1}^n Y_{ik} (V_i e^{j\theta_i} - V_k e^{j\theta_k})^* \quad i, k = 1, \dots, n \quad (3.1)$$

$$\text{Haciendo } Y_{ik} = G_{ik} + jB_{ik} \quad (3.2)$$

Tenemos que los desajustes de potencia estarán dados por:

$$P_i = P_i^{sp} - V_i \sum_{k=1}^n V_k (G_{ik} \cos \theta_{ik} + B_{ik} \sin \theta_{ik})$$

$$Q_i = Q_i^{sp} - V_i \sum_{k=1}^n V_k (G_{ik} \sin \theta_{ik} - B_{ik} \cos \theta_{ik}) \quad (3.3)$$

Donde los superíndices "sp" indican que son las potencias especificadas para la barra "i".

Como ya se ha dicho anteriormente el sistema de ecuaciones a resolver estará formado de dos ecuaciones por cada nodo de carga (las dos ecuaciones (3.3)) y de una ecuación por cada nodo generador

(la primera de las ecuaciones (3.3)).

3.3. ALGORITMO BASICO (6)

De las ecuaciones 3.3 tenemos que:

$$\Delta P = P_i - P_i^{sp} = -V_i \sum_{k=1}^n V_k (G_{ik} \cos \theta_{ik} + B_{ik} \sin \theta_{ik})$$

$$\Delta Q = Q_i - Q_i^{sp} = -V_i \sum_{k=1}^n V_k (G_{ik} \sin \theta_{ik} - B_{ik} \cos \theta_{ik})$$

(3.4)

En las ecuaciones 3.4. se puede observar que toda variación en P y Q dependerá de cambios en el voltaje ($V = V \theta$) de cada nodo, así para una barra cualquiera "J" se tendrá de acuerdo al método de Newton Raphson:

$$\Delta P_J = \frac{\partial P_j}{\partial \theta_1} \Delta \theta_1 + \frac{\partial P_j}{\partial \theta_2} \Delta \theta_2 + \dots + \frac{\partial P_j}{\partial \theta_n} \Delta \theta_n +$$

$$+ \frac{\partial P_j}{\partial V_1} \Delta V_1 + \frac{\partial P_j}{\partial V_2} \Delta V_2 + \dots + \frac{\partial P_j}{\partial V_n} \Delta V_n \quad (3.5)$$

lo anterior se puede escribir en forma condensada:

$$\Delta P_J = \sum_{i=1}^n \frac{\partial P_j}{\partial \theta_i} \Delta \theta_i + \sum_{i=1}^n \frac{\partial P_j}{\partial V_i} \Delta V_i \quad (3.6)$$

de igual forma se puede demostrar que:

$$\Delta Q_j = \sum_{i=1}^n \frac{\partial Q_j}{\partial \theta_i} \Delta \theta_i + \sum_{i=1}^n \frac{\partial Q_j}{\partial V_i} \Delta V_i \quad (3.7)$$

Expresando el sistema de ecuaciones para todos los nodos en forma matricial se tiene:

$$\begin{bmatrix} \Delta P_1 \\ \vdots \\ \Delta P_n \\ \hline \Delta Q_1 \\ \vdots \\ \Delta Q_n \end{bmatrix} = \begin{bmatrix} \frac{\partial P_1}{\partial \theta_1} & \dots & \frac{\partial P_1}{\partial \theta_n} & \frac{\partial P_1}{\partial V_1} & \dots & \frac{\partial P_1}{\partial V_n} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \frac{\partial P_n}{\partial \theta_1} & \dots & \frac{\partial P_n}{\partial \theta_n} & \frac{\partial P_n}{\partial V_1} & \dots & \frac{\partial P_n}{\partial V_n} \\ \hline \frac{\partial Q_1}{\partial \theta_1} & \dots & \frac{\partial Q_1}{\partial \theta_n} & \frac{\partial Q_1}{\partial V_1} & \dots & \frac{\partial Q_1}{\partial V_n} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \frac{\partial Q_n}{\partial \theta_1} & \dots & \frac{\partial Q_n}{\partial \theta_n} & \frac{\partial Q_n}{\partial V_1} & \dots & \frac{\partial Q_n}{\partial V_n} \end{bmatrix} \begin{bmatrix} \Delta \theta_1 \\ \vdots \\ \Delta \theta_n \\ \hline \Delta V_1 \\ \vdots \\ \Delta V_n \end{bmatrix} \quad (3.8)$$

Por conveniencia para el análisis la ecuación 3.8 se puede escribir en la siguiente forma:

$$\begin{bmatrix} \Delta P \\ \Delta Q \end{bmatrix} = \begin{bmatrix} H & N \\ J & L \end{bmatrix} \begin{bmatrix} \Delta \theta \\ \Delta V/V \end{bmatrix} \quad (3.9)$$

En la ecuación 3.9 los elementos de la matriz jacobiana estarán dados por la evaluación de las siguientes ecuaciones:

$$H = \frac{\partial P}{\partial \theta}$$

$$N = \frac{\partial P}{\partial V} + V$$

$$J = \frac{\partial Q}{\partial \theta}$$

$$L = \frac{\partial Q}{\partial V} + V \quad (3.10)$$

Aplicando el principio de desacoplamiento, se desprecian las submatrices N y J quedando dos ecuaciones matriciales separadas:

$$[\Delta P] = [H] [\Delta \theta] \quad (3.11)$$

$$[\Delta Q] = [L] [\Delta V / V] \quad (3.12)$$

Usando las ecuaciones 3.10, se obtiene que los elementos de las submatrices H y L son:

$$H_{km} = L_{km} = V_k V_m (G_{km} \sin \theta_{km} - B_{km} \cos \theta_{km}) \quad (3.13)$$

$$H_{kk} = -B_{kk} V_k^2 + Q_k \quad (3.14)$$

$$L_{kk} = -B_{kk} V_k^2 + Q_k \quad (3.15)$$

Generalmente en sistemas de potencia se tiene que:

$$\cos \theta_{km} = 1;$$

$$G_{km} \sin \theta_{km} \ll B_{km};$$

$$Q_k \ll B_{kk} V_k^2$$

Por lo tanto las ecuaciones anteriores, que permiten calcular los elementos de las matrices H y L quedan reducidas a las siguientes expresiones:

$$H_{km} = L_{km} = -V_k V_m B_{km} \quad (3.16)$$

$$H_{kk} = L_{kk} = -B_{kk} V_k^2 \quad (3.17)$$

Una buena aproximación de las ecuaciones 3.11 y 3.12., en base a las ecuaciones 3.16 y 3.17., es:

$$[\Delta P] = [V \cdot B' \cdot V] [\Delta \theta] \quad (3.18)$$

$$[\Delta Q] = [V \cdot B'' \cdot V] [\Delta V/V] \quad (3.19)$$

Donde los elementos de las matrices B' y B'' son elementos de la matriz de susceptancias $-B$.

En el cálculo de B' y B'' , habiendo aceptado el principio de desacoplamiento se tienen que hacer las siguientes consideraciones en sus cálculos.

- a. Omitir de B' la representación de aquellos elementos de la red que afectan el flujo de MVAR, tales como reactancias en paralelo y transformadores con tapas diferentes al nominal. De

acuerdo al principio arriba mencionado el flujo de MVAR afecta estrictamente a la magnitud de voltaje y por ende no deben ser considerados, estos elementos en el cálculo de $\Delta\theta$ y por consiguiente en B' .

- b. Se debe omitir al calcular los elementos de B'' la variación del ángulo producido por los cambiadores de fase, ya que esta no afecta el cálculo de la magnitud del voltaje en la ecuación 3.16, de acuerdo al principio de desacoplamiento en la variación de ΔV solo influirán los cambios en ΔQ .

Pasando el término V a la izquierda de B' y B'' en 3.18 y 3.19 al lado izquierdo de las ecuaciones, eliminando en 3.18 la influencia del flujo de MVAR en el cálculo de $\Delta\theta$ ajustando a 1 p.u., los términos V a la derecha de B' , y simplificando en 3.19 los términos V a la derecha de B'' se obtiene:

$$[\Delta P/V] = [B'] [\Delta\theta] \quad (3.20)$$

$$[\Delta Q/V] = [B''] [\Delta V] \quad (3.21)$$

Es importante hacer notar que los términos V en el lado izquierdo de las ecuaciones 3.20 y 3.21 afectan el comportamiento de funciones definidas y no el acoplamiento.

Las matrices B' y B'' son reales, dispersas y tienen la misma estructura que las matrices H y L . La ventaja fundamental que se obtiene como consecuencia de lo anterior, es que estas matrices contienen solo elementos de la matriz de admitancias, por lo tanto son constantes y debido a esto se obtiene solamente una vez su inversa para todo el proceso iterativo.

3.4. ALMACENAMIENTO Y COMPACTACION

Como ya se dijo anteriormente las matrices B' y B'' son matrices dispersas y simétricas.

Estas características nos permiten guardar en memoria la mitad de los elementos diferentes de cero fuera de la diagonal de la matriz, obteniéndose así un gran ahorro de memoria de almacenamiento. La compactación o empaquetamiento aprovecha lo dicho anteriormente, la forma como se ha hecho en este trabajo se describirá en el capítulo V.

3.5. ESQUEMA BASICO PARA LA SOLUCION

El esquema seguido para obtener la solución es el mostrado en la figura 3.2.

Claramente se puede observar que la estrategia que se sigue usando el esquema mostrado es resolver alternadamente las ecuaciones 3.20 y 3.21, ya que conocemos ΔP y ΔQ en ellas, por lo tanto obtenemos $\Delta\theta$ y ΔV .

Lo anterior se repite hasta que ΔP y ΔQ sean menores que el valor escogido como tolerancia de error para la solución buscada.

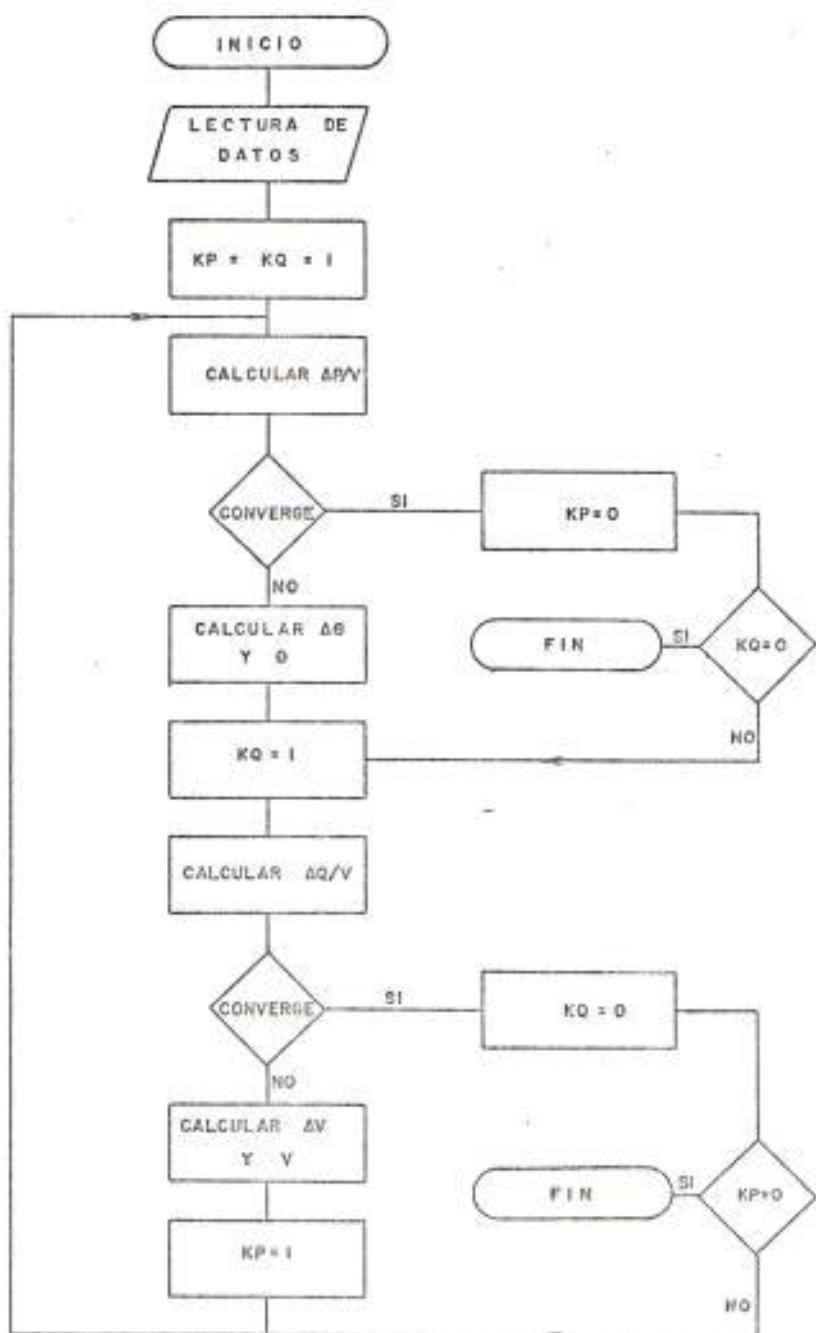


Fig. 3.2.— Esquema seguido para la resolución del flujo de carga por el método desacoplado rápido

CAPITULO IV

PROGRAMACION EN PASCAL PARA RESOLVER PROBLEMAS DE SISTEMAS DE POTENCIA

4.1. CARACTERISTICAS PRINCIPALES DE ESTE LENGUAJE

La primera versión de Pascal apareció en el año 1.968, la misma que fue elaborada por el profesor Niklaus Wirth de la Universidad de Zurich , Suiza.

Este lenguaje fue anunciado oficialmente por Wirth en el año de 1.971 y su manual fue publicado en 1.973.

Sus características más importantes son:

- a. Todas las variables son definidas en forma explícita al comienzo de cada programa y esto trae como consecuencia la existencia de dos piezas fundamentales en la documentación; una lista de las variables usadas y el rango en

- el cual entran sus valores.
- b. No existe limitación en el número de caracteres usados en un identificador, permitiendo así, que las variables puedan tener nombres significativos, fáciles de entender y recordar. Los ocho primeros caracteres diferencian los identificadores.
- c. Además de los tipos de variables conocidos en otros lenguajes tales como real, integer, char, boolean, Pascal ofrece tipos estructurados tales como arrays, sets, records, files y un tipo muy especial que se conoce como pointer.
- Pascal también da facilidad al programador para que pueda crear sus propios tipos.
- d. Pascal proporciona una extensa lista de instrucciones estructuradas para : asignaciones, procedimientos, selecciones (if, case) e iteraciones (for, while, repeat).
- e. Pascal permite el uso de variables con el ca-

rácter de locales que son usadas y declaradas en un procedimiento, estas variables solo ocupan espacio de memoria cuando se ejecuta el procedimiento en el que están incluidas, luego este espacio de memoria queda libre. Las variables que tienen el carácter de globales se definen en el programa principal y pueden ser usadas en cualquier parte del mismo.

f. Es necesario indicar que la programación estructurada tiene algunas ventajas tales como: relativa facilidad al iniciar la programación gran facilidad para hacer modificaciones, modularidad de los programas, documentación y facilidad para que otras personas puedan entender el programa sin que hayan comentarios. Una vez conocidas las ventajas de la programación estructurada encontramos una de las más importantes características de Pascal ya que este lenguaje es el que más facilidades presenta, intrínsecamente, para la programación estructurada.

4.2. COMPARACION CON OTROS LENGUAJES CIENTIFICOS

Cada lenguaje científico tiene sus ventajas y

desventajas, lo cual nos permite concluir que no hay un lenguaje ideal para todas las aplicaciones.

En la tabla 4.1, se presenta una comparación entre los principales lenguajes, en la misma que, para las diferentes características se los califica para una nota máxima de 10. Esta comparación fue elaborada por Texas Instruments (16).

4.3. USO DE PASCAL EN SISTEMAS DE POTENCIA

Del cuadro anterior se puede observar que Pascal tiene muchas ventajas sobre algunos de los lenguajes científicos con los que se los ha comparado y por lo tanto su uso en sistemas de potencia va a ser importante.

La característica "e" mencionada en el punto 4.1 hace de este lenguaje una valiosa herramienta para elaborar programas en la resolución de problemas de sistemas de potencia ya que por lo general éstos fundamentalmente requieren de bastante memoria para su ejecución y almacenamiento de datos.

Otra característica que hace a PASCAL un lenguaje

je muy útil es la existencia del tipo de variable conocido como "puntero" que permite por ejemplo el almacenamiento en forma compacta de matrices de tal suerte que solo se almacenan los elementos diferentes de cero en forma indexada con gran facilidad para su posterior acceso. Este tipo de variable ayuda al programador a crear y manipular datos estructurados en forma sumamente flexible, esta flexibilidad permite el fácil manejo de datos en los problemas de sistemas de potencia.

TABLA 4.1.

CARACTERÍSTICA	COMPARACIÓN ENTRE PASCAL, FORTRAN, BASIC, PL/M Y COBOL				
	LENGUAJE	CONSIDERADO	PASCAL	FORTRAN	BASIC
Aplicación para programas largos			10	6	3
Aplicabilidad para programas pequeños			8	7	10
Facilidad para aprenderlo como primer lenguaje			6	8	10
Facilidad para aprenderlo como segundo lenguaje si el primero fue estructurado			8	6	7
Facilidad para aprenderlo como segundo lenguaje si el primero no fue estructurado			5	9	10
Apoyo a la técnica "de arriba hacia abajo"			10	5	3
Apoyo para crear programas estructurados			10	4	4
Ecritura de programas - largos			10	5	3
Capacidad de generar códigos eficientes			8	7	6
Flexibilidad en tipo de datos			10	5	1
Capacidad de compilar para chequear consistencias y errores			10	5	4
Acceso controlado a datos			10	6	1
					3
					4

CAPITULO V

ELABORACION DE UN PROGRAMA EN PASCAL PARA SOLUCIONAR
EL PROBLEMA DE FLUJO DE CARGA POR EL METODO DESCOPLA-
DO RAPIDO

5.1. ESTRUCTURA DEL PROGRAMA

El programa se lo puede dividir en tres partes , las mismas que son:

- Entradas del programa
- Procesamiento de datos
- Salidas del programa

5.1.1. Entradas del Programa

Los datos de entrada del programa deben ser dados en p.u., y se introducen al ejecutar el procedimiento DATOS, en forma de respuestas a preguntas que hace el computador al usuario, son los siguientes:

- Para líneas: resistencia, reactancia y susceptancia paralelo.

- Para nodos: Potencias activa y reactiva de carga para todos los nodos, potencia activa generada para los nodos generadores, voltaje deseado para los nodos de generación incluida la barra oscilante.

Además de los datos anteriores también se requieren otros de carácter general como nombre del usuario, nombre del sistema, fecha, etc.

5.1.2. Procesamiento de Datos

Con los datos leídos en el procedimiento anterior se procede a obtener la matriz de admitancias lo cual se consigue al ejecutar el procedimiento MATAD. Para poder obtener posteriormente la inversa de la matriz encontrada en MATAD (matriz de admisiones) se procede a bifactorizarla usando el procedimiento BIFACTORIZA, la bifactorización se la hace en forma ordenada comenzando con la fila que contiene menor número de elementos diferentes a cero, para conseguir esto, en el programa a

cada paso de la bifactorización se ejecuta el procedimiento PIVOTEAR.

Finalmente para resolver el flujo de carga usando el método desacoplado rápido se ejecuta el procedimiento FLUJO, el mismo que para resolver el problema planteado recibe la ayuda del procedimiento CALCULOS y para obtener todas las salidas deseadas ejecuta el procedimiento CALFLUJO.

5.1.3. Salidas del programa

Las salidas del programa son:

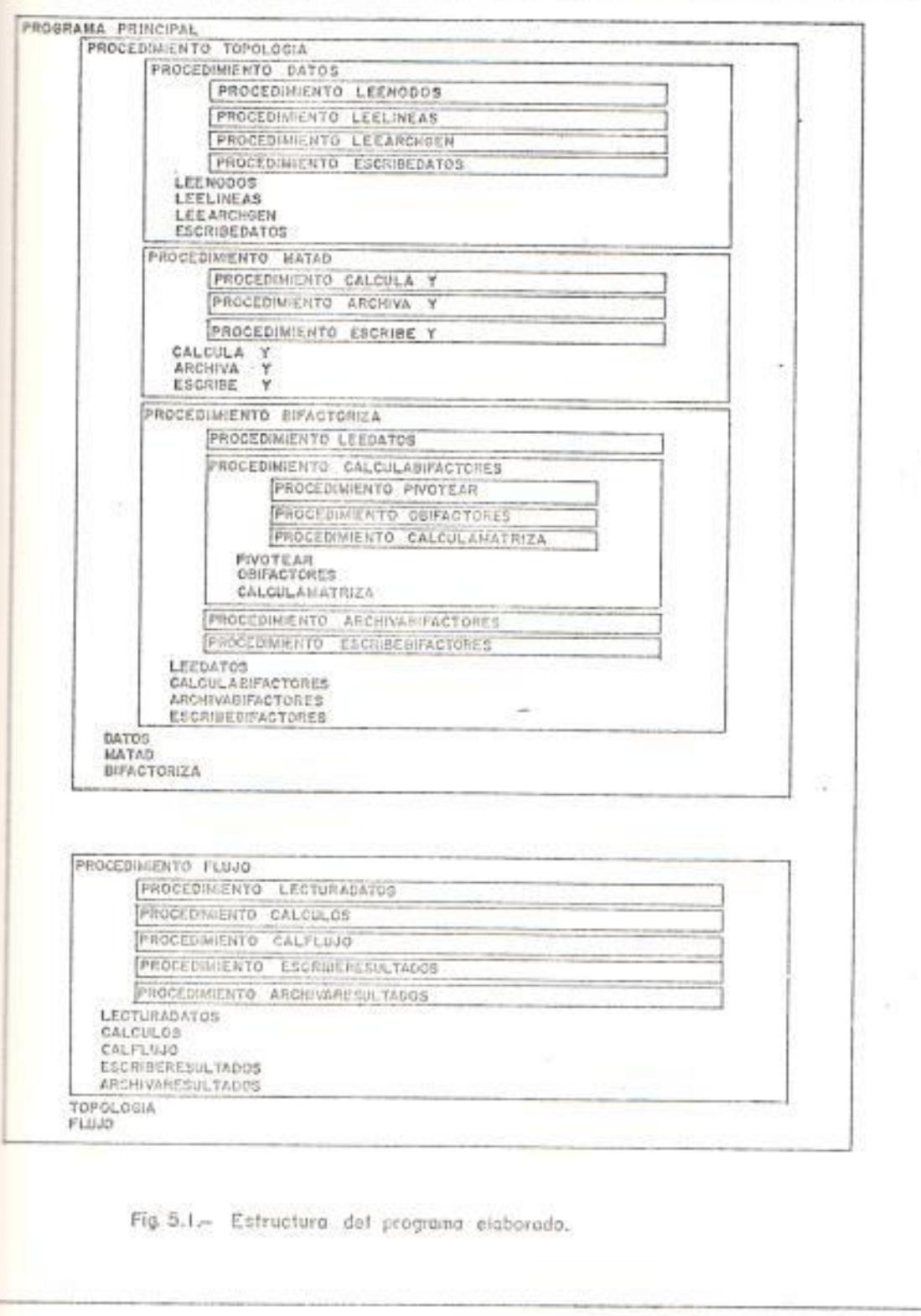
- Para líneas las potencias activa y reactiva que conducen, las pérdidas en cada una de ellas (activas y reactivas),
- Para nodos: cuando son nodos generadores se obtiene la potencia reactiva generada y el ángulo del voltaje deseado, cuando son nodos de carga se obtiene el voltaje módulo y ángulo);, para la barra oscilante se obtienen las potencias activas y reactivas generadas,

Una salida adicional de mucha importancia es el número de iteraciones en que se ha obtenido la solución.

Los resultados pueden ser sacados por pantalla o impresos al ejecutar el procedimiento ESCRIBERRESULTADOS, también pueden ser almacenados en discos al ejecutar el procedimiento ARCHIVARESULTADOS.

5.2. DIAGRAMA DE FLUJO DEL PROGRAMA

En programación estructurada realmente no se elabora un diagrama de flujo. A continuación se presenta el flujo que se sigue en el programa, de acuerdo a las instrucciones usadas en PASCAL, en las figuras 5.1 y 5.2, se presenta gráficamente la estructura del mismo y la secuencia que se sigue en el procesamiento de datos.



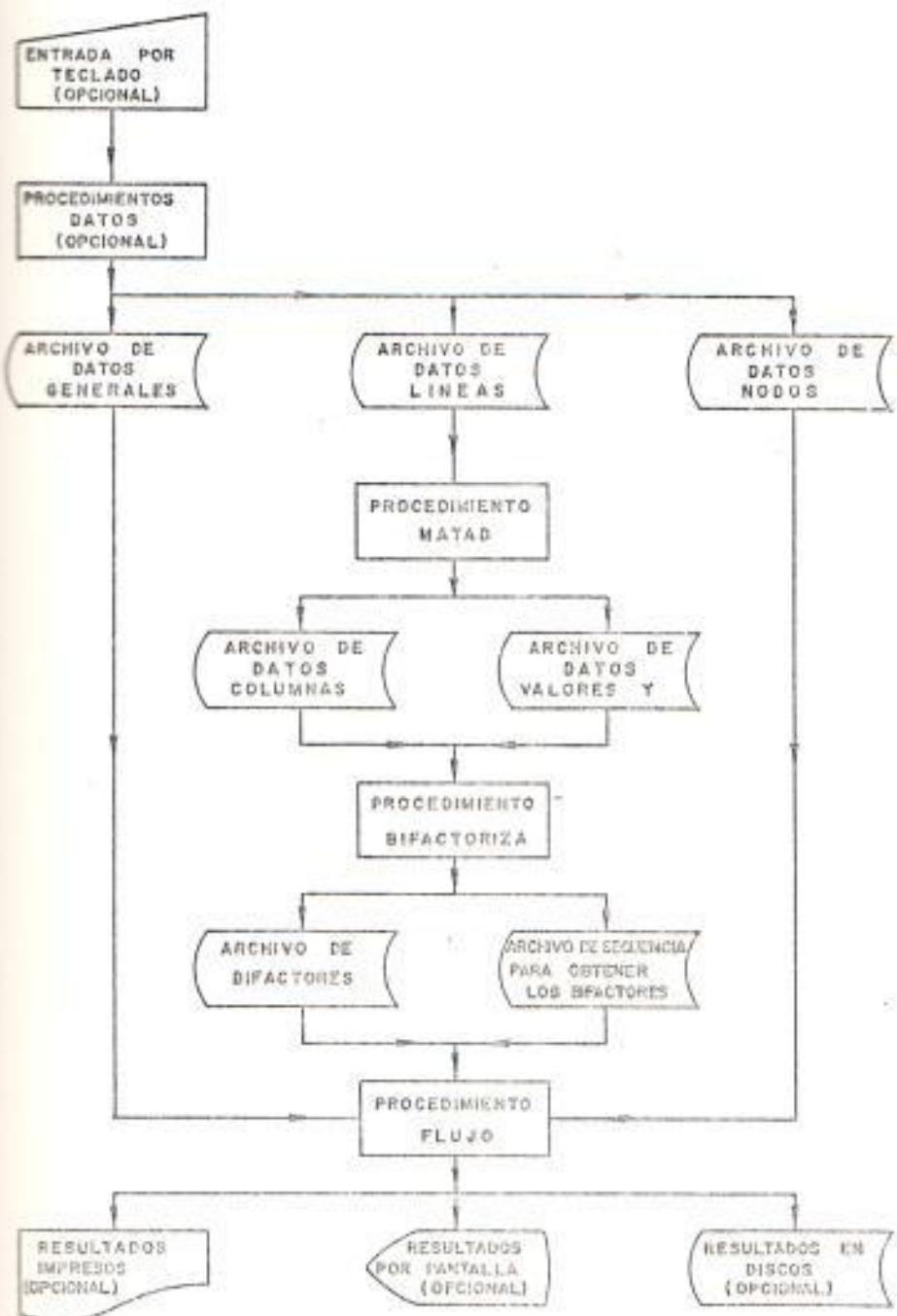


Fig. 5.2.- Secuencia del flujo y procesamiento de datos.

PROGRAM PRINCIPAL

(* ** DESCRIPCION:

Este programa resuelve el problema de flujo de carga en sistemas eléctricos de potencia, en él se aplica el método desacoplado rápido, además se usa compactación de matrices y la inversión de las mismas se la hace usando bi-factorización.

** ARCHIVOS DE ENTRADA:

ARCHGENERAL: Archivo de datos generales

LINEAS: Archivo de datos de líneas

NODOS: Archivo de datos de nodos

Estos archivos pueden haber sido creados anteriormente y estar grabados en discos o pueden ser introducidos por el teclado.

** ARCHIVOS DE SALIDA:

ARCHANODOS: Resultados de nodos grabados en discos.

ARCHALINEAS: Resultados de líneas grabados en discos.

Los resultados también pueden ser sacados por impresora o consola de acuerdo al de seo del usuario.

** DICCIONARIO DE CONSTANTES GLOBALES:

NTOTNODOS: Máximo número de nodos del pro ma.

NTOTLINEAS: Máximo número de líneas del pro grama

NOCEROS: Máximo número de elementos diferen tes de cero en el programa

** DICCIONARIO DE VARIABLES GLOBALES:

NOMBRE1 : Nombre del archivo de datos

NOMBRE2: Nombre del archivo de líneas

NOMUSUARIO: Nombre del usuario

NOMSISTEMA: Nombre del sistema

NBGENERACION: Número de barras generadoras

NTOTNOD: Número de nodos del sistema a estudiar

NTOTLIN: Número de líneas del sistema a estudiar

NTOTNOCE: Número de elementos diferentes de cero del sistema a estudiar

*)

PROCEDURE TOPOLOGIA

(*) ** DESCRIPCION:

Este procedimiento lee los datos de en trada del sistema a ser estudiado, calcu la y compacta la matriz de admitancias - para luego bifactorizarla.

** ARCHIVOS DE ENTRADA:

ARCHGENERAL: Datos generales del sistema

NODOS: Datos de cada nodo

LINEAS: Datos de cada linea

** ARCHIVOS DE SALIDA:

ARCHISECUENCIA: Secuencia del proceso de la matriz "H"

ARCH2SECUENCIA: Secuencia del proceso de la matriz "L"

ARCHIVO1FACTORES: Bifactores de la matriz "H"

ARCHIVO2BIFACTORES: Bifactores de la matriz "L"

** TIPOS:

RGENERAL: Registro del archivo de datos ge
nerales, contiene:

NUMNOD : número de nodos del sis
tema

NUMLIN: Número de líneas del sistema

NBGEN: Número de barras generadoras

NUMNOCE: Número de elementos diferentes de cero

USUARIO: Nombre del usuario

SISTEMA: Nombre del sistema

AGENERAL: Archivo formado por los registros del tipo ARGENERAL

** DICCIONARIO DE VARIABLES:

SEÑAL: Índice si se debe de leer o no el archivo de datos ya creado

CONTESTACION: Indica si se va a crear el archivo de datos

NOMBRE3: Nombre del archivo de datos generales

REGENERAL: Variable del tipo RGENERAL

ARCHGENERAL: Variable del tipo AGENERAL

*)

PROCEDURE DATOS;

(* ** DESCRIPCION

Lee los datos de entrada del programa

**** ARCHIVOS DE ENTRADA:**

ARCHGENERAL: Datos generales

NODOS: Datos de cada nodo

LINEAS: Datos de cada línea

Pueden haber sido grabados en discos anteriormente o pueden ser creados al momento por teclado

**** ARCHIVOS DE SALIDA:**

Son los mismos archivos de entrada ya que son creados en este procedimiento

**** TIPOS:**

LINEA: Registro del archivo de LINEAS, contiene:

NODORECEPCION: Nodo de recepción

RESISTENCIA: Resistencia de cada línea

REACTANCIA: Reactancia de cada línea

SUSCEPPAR: Susceptancia paralelo de cada línea

NODO : Registro del archivo de NO
DOS, contiene:

PGEN: Potencia activa gene
 rada de cada nodo

QGEN: Potencia reactiva ge
 nerada de cada nodo

PCARGA: Potencia activa de
 carga de cada nodo

QCARGA: Potencia reactiva de
 carga de cada nodo

VMODULO: Módulo del voltaje

VANGULO: Ángulo del voltaje

RNOMBRE: Nombre de cada no
 do

** DICCIONARIO DE VARIABLES:

LINRECORD: Variable del tipo LINEA

NODRECORD: Variable del tipo NODO

CONTA1: Contador de nodos

CONTA2: Contador de líneas

*)

PROCEDURE LEEARCHGEN;

(*) DESCRIPTION:

Lee los datos generales del
 sistema a ser estudiado

*)

PROCEDURE LEENODOS;

(* ** DESCRIPCION:

Lee y archiva los datos de c
da nodo

*)

PROCEDURE LEELINEAS;

(* ** DESCRIPCION:

Lee y archiva los datos de c
da linea

*)

PROCEDURE ESCRIBEDATOS;

(* ** DESCRIPCION:

Escribe los datos leídos en -
los procedimientos anteriores

*)

PROCEDURE NUEVONOMBRE;

(* ** DESCRIPCION:

Forma el nombre para el ar
chivo de datos en discos

*)

BEGIN (* PROCEDIMIENTO DATOS*)

NUEVONOMBRE;

```
LEEARCHGEN;

IF "DESEA CREAR UN NUEVO AR
CHIVO DE NODOS"
LEENODOS
ELSE
IF "DESEA CREAR UN NUEVO ARCHI
VO DE LINEAS"
LEELINEAS;
ESCRIBE DATOS
END; (*PROCEDIMIENTO DATOS*)

PROCEDURE MATAD;
(* ** DESCRIPCION:
Forma y compacta la matriz de admitan
cias
** ARCHIVOS DE ENTRADA:
LINEAS: Datos de líneas
** ARCHIVOS DE SALIDA:
COLUMNAS: Almacena información de co
lumnas
VALORES: Almacena los valores de ca
da elemento
** DICCIONARIO DE VARIABLES:
YC, YB: vectores que almacenan los va
lores de conductancia y suscep
tancia
```

LÍNEA: registro de líneas

NOCERO: vector que almacena el número
de elementos diferentes de cero
de cada columna

INICIOINF: vector que almacena la posición donde comienza la información de cada columna

SIGUIENTE: Vector que almacena la posición del siguiente elemento diferente a cero

FILA: Vector que almacena la fila -
de cada elemento

J, K, J1, J2: contadores

YCPROPIA, YBPROPIA, YCMUTUA, YBMUTUA:
Variables que sirven para calcular los elementos de la matriz -
de admitancias

*)

PROCEDURE CALCULAY;

(* ** DESCRIPCION:

```
    Calcula las admitancias propias
    y mutuas de cada nodo
*)

PROCEDURE ARCHIVAY;
(* ** DESCRIPCION:
    Archiva la matriz de admitancias
*)

PROCEDURE ESCRIBEY;
(* ** DESCRIPCION:
    Escribe la matriz de admitancias
*)

BEGIN (* PROCEDIMIENTO MATAD*)
    CALCULAY;
    ARCHIVAY;
    ESCRIBEY
END; (* PROCEDIMIENTO MATAD*)

PROCEDURE BIFACTORIZADA;
(* ** DESCRIPCION:
    Bifactoriza las matrices de conduc
    tancias y susceptancias
** ARCHIVOS DE ENTRADA :
```

COLUMNAS: Información de cada columna
a ser procesada

VALORES: Valores de cada elemento de
la matriz

** ARCHIVOS DE SALIDA:

ARCHISECUENCIA, ARCH2SECUENCIA:

Archiva la secuencia con la que se
ha procesado cada matriz

ARCHIVO1BIFACTORES, ARCHIVO2BIFACTORES:
Archiva los valores de la matriz de
bifactores

** DICCIONARIO DE VARIABLES:

INICIOINF: Vector que almacena la po-
sición donde comienza la
información de cada co-
lumna

NOCERO: Vector que almacena la cant_í-
dad de elementos diferen-
tes de cero de cada colum-
na

SIGUIENTE : Vector que almacena la
posición donde se en-
cuentra el siguiente -
elemento diferente de
cero

FILA: Almacena la fila de cada elemento

MARCA: Indica si una fila ya se ha procesado

VALOR: Vector que contiene el valor de cada elemento

EBIFACTOR: Elemento de la matriz - de Bifactores

NUEVOTOTALNODOS: Dimensión de la matriz B

B, IB1, IB2, IB3: Contadores

*)

PROCEDURE LEEDATOS;

(* ** DESCRIPCION:

Lee la matriz de admitancias

*)

PROCEDURE CALCULABIFACTORES;

(* ** DESCRIPCION:

Calcula los bifactores de la matriz que se esta procesando

*)

PROCEDURE PIVOTEAR;

(* ** DESCRIPCION :

Escoge la fila con menor
 número de elementos diferentes a cero
*)
PROCEDURE OBIFACTORES;
(* ** DESCRIPCION:
 Calcula los bifactores correspondientes a la fila escogida en el procedimiento anterior
*)
PROCEDURE CALCULAMATRIZA;
(* ** DESCRIPCION:
 Calcula la nueva matriz a usarse en el cálculo de los bifactores correspondientes a la siguiente fila a procesar
*)
BEGIN (* CALCULABIFACTORES *)
 FOR CONTADOR = 1 TO (TOTAL DE NODOS - 1) DO

```
BEGIN
    PIVOTEAR;
    OBIFACTORES;
    CALCULAMATRIZA
END
END; (*CALCULABIFACTORES*)

PROCEDURE ESCRIBEBIFACTORES;
(* ** DESCRIPCION:
    Escribe los bifactores obtenidos anteriormente
*)

PROCEDURE ARCHIVABIFACTORES;
(* ** DESCRIPCION:
    Archiva los bifactores escritos en el procedimiento anterior
*)

BEGIN (* BIFATORIZA *)
    LEEDATOS;
    CALCULABIFACTORES;
    ARCHIVABIFACTORES;
    ESCRIBEBIFACTORES
END; (* BIFATORIZA *)
```

```
BEGIN (* PROCEDIMIENTO TOPOLOGIA *)
    DATOS;
    MATAD;
    BIFACTORIZA
END; (* PROCEDIMIENTO TOPOLOGIA *)

PROCEDURE FLUJO;
(* ** DESCRIPCION:
   Resuelve y calcula todo lo requerido por el flujo de carga
   ** ARCHIVOS DE ENTRADA:
   ARCHNODOS: Archivo de datos - nodos
   ARCHLINEAS: Archivo de datos - líneas
   COLUMNAS: Archivo con información de las columnas de la matriz de admitancias
   VALORES: Archivo con los valores de cada uno de los elementos de la matriz de admisiones
   ** ARCHIVOS DE SALIDA:
   ARCHANODOS: Archiva los resultados de nodos en discos
   ARCHALINEAS: Archiva los resultados de líneas en discos
   Además los resultados se obtienen -
```

por pantalla y/o impresos según el deseo del usuario

** DICCIONARIO DE VARIABLES:

VNOMBRE: Nombre de cada nodo

RESPUESTA: Indica si imprime o no los resultados

RESPUESTA: Indica si salen o no los resultados por consola

RESPUESTA: Indica si se archivan o no los resultados

FECHA: Fecha de creación de datos- resultados

PERIFERICO: Nombre del periférico por el que van a salir los resultados

PTOTAL, QTOTAL: Potencias totales de cada nodo

VH, VA: Valor modular y angular de voltaje

DPV, DQV: Desajustes de potencias

PGENERADA, QGENERADA: Potencias generadas

PCAR, QCAR: Potencias de carga de cada nodo

*)

```
PROCEDURE LECTURADATOS;  
(* ** DESCRIPCION:  
    Lee los datos requeridos en el procedimiento FLUJO  
*)  
  
PROCEDURE CALCULOS;  
(* ** DESCRIPCION:  
    Resuelve el flujo de carga  
*)  
  
PROCEDURE CALFLUJO;  
(* ** DESCRIPCION:  
    Calcula con los datos obtenidos en el procedimiento anterior, los flujos de potencia por cada linea y la potencia generada en cada nodo  
*)  
  
PROCEDURE ESCRIBERRESULTADOS;  
(* ** DESCRIPCION:  
    Imprime y/o saca por pantalla los resultados obtenidos en los procedimientos anteriores  
*)
```

```
PROCEDURE ARCHIVARESULTADOS;
(* ** DESCRIPCION:
   Archiva los resultados, si es que
   así lo desea el usuario
*)
BEGIN (* PROCEDIMIENTO FLUJO *)
  LECTURADATOS;
  CALCULOS;
  CALFLUJO;
  ESCRIBERESULTADOS;
  ARCHIVARESULTADOS
END; (* PROCEDIMIENTO FLUJO *)
BEGIN (* PROGRAMA PRINCIPAL *)
  TOPOLOGIA;
  FLUJO
END. (* PROGRAMA PRINCIPAL*)
```

5.3. METODO Y TECNICAS NUMERICAS USADAS EN EL PROGRAMA

5.3.1. Descomposición triangular

Debido a la importancia que tiene en la resolución del flujo de carga, el método que se usa para obtener la solución de un sistema de ecuaciones, - trataremos aquí dos de los más usados para este proceso. En este punto se analizará brevemente la descomposición triangular y en el siguiente la bifactorización.

Supongamos que se tiene que resolver la ecuación $AX = b$ (5.1)

Donde A es una matriz de orden "n", no singular, X son las n incognitas y b es un vector de constantes tal que por lo menos contiene un término diferente de cero; En forma extendida la ecuación (5.1) se puede escribir:

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ \vdots \\ b_n \end{bmatrix} \quad (5.2)$$

Si efectuamos combinaciones con estas ecuaciones hasta que queden en la siguiente forma:

$$\begin{bmatrix} 1 & c_{12} & \dots & c_{1n} \\ 0 & 1 & & c_{2n} \\ \dots & \dots & & \dots \\ 0 & 0 & & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_n \end{bmatrix}$$

(5.3)

Se puede ver que es fácil obtener x_n de la última ecuación, $x_n = d_n$; luego x_{n-1} de la penúltima ecuación

donde aparecen solamente x_{n-1} y x_n , pero esta última ya es conocida, así sucesivamente se va de abajo hacia arriba hasta encontrar x_1 . Una comparación de las ecuaciones (5.2) y (5.3) muestra que los coeficientes de la matriz A y del vector b han sido cambiados hasta obtener una matriz triangular, con ceros en la parte inferior y un nuevo vector de coeficientes d , respectivamente.

El proceso que se sigue para obtener la ecuación (5.3) es el mismo que se describe en el numeral 2.2.1., bajo el subtítulo de Eliminación gaussiana.

5.3.2. Bi-factorización

El método de bi-factorización fue desarrollado por Zollenkopf y es una modificación importante a la eliminación por el método de Gauss.

Es aplicable especialmente a los aná-

lisis de sistemas grandes, dispersos, que tienen los elementos de la diagonal diferentes a cero, los cuales predominan sobre los elementos que se encuentran fuera de la diagonal.

El método combina la triangularización y el producto de matrices para expresar la inversa de la matriz A, de la ecuación (5.1), como el producto de 2xn matrices.

El proceso consiste en ir calculando 2 matrices de tal forma que una de ellas al pre-multiplicarla por la matriz A y la segunda post-multiplicarla se obtenga una nueva matriz A^{-1} de tal suerte que el elemento a_{11} se haga 1 y los elementos de la fila y columna 1 se hagan ceros. Lo anterior se repite hasta obtener la matriz unidad y se lo puede representar por la siguiente ecuación:

$$L_n \dots L_2 L_1 A R_1 R_2 \dots R_n = U \quad (5.4)$$

Donde:

R = matrices factores derechos

L = matrices factores izquierdos

U = matriz unidad

n = orden de la matriz

Para encontrar la inversa de la matriz A se lo hace despejándola de la ecuación (5.4), así:

$$AR_1R_2 \dots R_n = L_1^{-1}L_2^{-1}\dots L_n^{-1}$$

$$AR_1R_2 \dots R_n L_n \dots L_2 L_1 = U$$

$$R_1R_2 \dots R_n L_n \dots L_2 L_1 = A^{-1}$$

Donde:

A^{-1} es la inversa de la matriz A . Para determinar las matrices factor se introduce la siguiente secuencia de matrices intermedias:

$$A = A_0$$

$$A_1 = L_1 A_0 R_1$$

$$A_2 = L_2 A_1 R_2$$

$$\vdots$$

$$\vdots$$

$$A_n = L_n A_{n-1} R_n$$

Si se hace la bifactorización comenzando por la fila y columna 1, se tiene que para una etapa K cualquiera las matrices A_k , L_k y R_k tienen la siguiente forma:

$$A_k = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \cdot & \cdot & a_{k,k} & a_{k,n} \\ \cdot & \cdot & a_{k+1,k} & a_{k+1,n} \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ 0 & \cdot & a_{n,k} & a_{n,n} \end{bmatrix}$$

$$L_k = \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & L_{k,k} & \\ & & L_{k+1,k} & 1 \\ & & L_{n,k} & \\ & & & 1 \end{bmatrix}$$

$$R_k = \begin{bmatrix} 1 & & & & \\ & 1 & R_{k,k+1}, \dots, R_{k,n} & & \\ & & 1 & \dots & \\ & & & & \ddots \\ & & & & 1 \end{bmatrix}$$

Donde todos los elementos que no se muestran son ceros.

Para una matriz simétrica

$$a_{i,k} = a_{k,i}$$

$$R_{i,k} = L_{k,i}$$

Para calcular Λ_k , L_k , R_k se usan -

las siguientes fórmulas:

$$L_{k,k}^k = 1/a_{k,k}^{k-1}$$

$$L_{i,k}^k = -a_{i,k}^{k-1}/a_{k,k}^{k-1}$$

$$R_{k,j}^k = -a_{k,j}^{k-1}/a_{k,k}^{k-1}$$

$$a_{i,j}^{k-1} = a_{i,j}^{k-1} - a_{i,k}^{k-1} \cdot a_{k,j}^{k-1}/a_{k,k}^{k-1}$$

Con el propósito de reducir tiempo y ahorrar memoria en el proceso de bifactorización, se ha efectuado en el programa presentado por esta tesis una eliminación ordenada, realizando la bifactorización comenzando por las filas y/o columnas que tengan menos elementos diferentes de cero, para lo cual previo a cada paso de la bifactorización se efectúa un voto para escoger la fila o la co

lumna que cumple con lo dicho anteriormente.

La ecuación para el cálculo de los elementos de la matriz A en una etapa cualquiera k, cuando la columna "p" se ha procesado queda en la siguiente forma:

$$a_{i,j}^k = a_{i,j}^{k-1} - a_{i,p}^{k-1} \cdot a_{p,j}^{k-1} / a_{p,p}^{k-1}$$

A continuación se muestra un ejemplo donde se encuentra la solución a un sistema de ecuaciones, usando bifactorización.

Consideremos el siguiente sistema:

$$AX = B$$

Donde:

$$A = \begin{bmatrix} 3 & -1 & -1 & 0 \\ -1 & 2 & 0 & 0 \\ -1 & 0 & 2 & -1 \\ 0 & 0 & -1 & 1 \end{bmatrix}$$

$$B = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \quad X = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$$

Usando las fórmulas dadas para calcular los elementos de la nueva matriz A y de las matrices L y R tenemos:

Primera etapa:

$$L_1 = \begin{bmatrix} 1/3 & 0 & 0 & 0 \\ 1/3 & 1 & 0 & 0 \\ 1/3 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad L_1 A R_1 = A_1$$

$$R_1 = \begin{bmatrix} 1 & 1/3 & 1/3 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 5/3 & -1/3 & 0 \\ 0 & -1/3 & 5/3 & -1 \\ 0 & 0 & -1 & 1 \end{bmatrix}$$

Segunda etapa:

$$L_2 A_1 R_2 = A_2$$

$$L_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 3/5 & 0 & 0 \\ 0 & 1/5 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1/5 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 8/5 - 1 & \\ 0 & 0 & -1 & 1 \end{bmatrix}$$

Tercera etapa:

$$L_3 A_2 R_3 = A_3$$

$$L_3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 5/8 & 0 \\ 0 & 0 & 5/8 & 1 \end{bmatrix}$$

$$R_3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 5/8 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 3/8 \end{bmatrix}$$

Cuarta etapa:

$$L_4 A_3 R_4 = A_4$$

$$L_4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 8/3 \end{bmatrix}$$

$$R_4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

El resultado final será:

$$x = A^{-1} b$$

$$x = R_1 R_2 R_3 R_4 L_4 L_3 L_2 L_1 B$$

$$x = \begin{bmatrix} 7/3 \\ 5/3 \\ 13/3 \\ 16/3 \end{bmatrix}$$

5.3.3. Ordenamiento óptimo de matrices

Una matriz dispersa es aquella que contiene un buen porcentaje de elementos fuera de la diagonal iguales a cero. Durante el proceso de eliminación, la tendencia es a llenar la matriz dispersa, osea, destruir los ceros. Como ya se vió, durante el proceso de eliminación, los renglones superiores se van restando - de los inferiores y llenándolos, esto trae como consecuencia incremento de memoria y tiempo de computación. Para demostrar lo último consideremos la siguiente matriz:

$$A = \begin{bmatrix} X & X & X & X & X \\ X & X & & & \\ X & & X & & \\ X & & & X & \\ X & & & & X \end{bmatrix}$$

Donde X representa los elementos diferentes de cero. Después de la -

primera eliminación usando la bifactorización, la matriz resultante será del tipo:

$$A^1 = \begin{bmatrix} 1 & & & \\ X & X & X & X \end{bmatrix}$$

La matriz resultante es una matriz llena de elementos diferentes a cero y la ventaja de tener inicialmente una matriz dispersa se ha perdido.

En cambio si, previo a la bifactorización, recordemos la matriz de la siguiente forma:

$$A = \begin{bmatrix} X & & & X \\ & X & & X \\ & & X & X \\ & & & X & X \\ X & X & X & X & X \end{bmatrix}$$

Las matrices resultantes serán:

$$A^1 = \begin{bmatrix} 1 & & & \\ & X & & X \\ & & X & X \\ & & & X \\ X & X & X & X \end{bmatrix}$$

$$A^2 = \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & X & X \\ & & & X \\ X & X & X & X \end{bmatrix}$$

$$A^3 = \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & X \\ X & X & X & X \end{bmatrix}$$

$$A^4 = \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 1 \\ & & & x \end{bmatrix}$$

Las matrices R y L correspondientes a cada etapa tendrán la misma estructura dispersa.

La estrategia seguida en este trabajo, ha sido, como ya se mencionó anteriormente, escoger en cada etapa de la reducción la columna que tiene el menor número de elementos diferentes a cero,

5.3.4. Empaquetamiento

El empaquetamiento de una matriz consiste en almacenar solamente los elementos que son diferentes a cero, esto se hace con las matrices que son dispersas para así explotar ésta característica.

Para efectuar el empaquetamiento se forman grandes columnas, las mismas que contienen información de los elementos diferentes de cero. Hablando del caso aquí presentado, se han creado los siguientes vectores:

- Un vector que indica el número de columnas.
- Un vector que indica el número de elementos diferentes a cero de cada columna.
- Un vector que contiene la indicación sobre la posición donde comienza la información de cada columna.

Los vectores mencionados tendrán como máximo número de elementos, el número de filas o columnas de la matriz que se va a compactar.

Además de estos vectores existen otros tres que contienen:

- La fila del elemento
- El valor del elemento
- Otro vector que contendrá o indicará en qué posición se encuentra el siguiente elemento diferente a cero de cada columna, éste último valor es cero cuando se trata del último elemento de una columna.

Para ilustrar lo último en la Figura - (5.1) se muestra la forma como se almacena la siguiente matriz:

$$\begin{bmatrix} 2 & -1 & -1 & 0 \\ -1 & 1 & 0 & 0 \\ -1 & 0 & 2 & -1 \\ 0 & 0 & -1 & 2 \end{bmatrix}$$

Aparentemente al observar el ejemplo anterior se podría llegar a la conclusión de que no se justifica usar empaquetamiento, ya que para almacenar

todos los elementos de la matriz dada se necesitarían 16 elementos (4×4), mientras que si se almacena en forma empaquetada se necesitarían por lo menos 52 elementos de memoria ($4 \times 3 + 4 \times 10$). Ver figura 5.1.

Sin embargo, al tratar un sistema de potencia nos encontramos con matrices llenas de ceros y es así como en el sistema tratado en ésta tesis se tiene una matriz de 37×37 en la cual el número de elementos diferentes a cero es 113.

Para ésta matriz, si se almacenan todos sus términos, se necesitarían 1369 elementos de memoria, mientras que si se almacena en forma empaquetada solo se necesitan 563 elementos ($37 \times 3 + 4 \times 113$).

Esta característica es más apreciable cuando se trata de sistemas de mayor tamaño.

	Columna	Dirección (Posición)	No ceros
1	1	1	3
2	4	4	2
3	6	6	3
4	9	9	2

Dirección	Fila	Valor	Siguiente
1	1	2	2
2	2	-1	3
3	3	-1	0
4	1	-1	5
5	2	1	0
6	1	-1	7
7	3	2	8
8	4	-1	0
9	3	-1	10
10	4	2	0

TABLEA 5.1. EJEMPLO DE COMPACTACION DE UNA MATRIZ CUALQUIERA.

5.3.5. Flujo de carga por el método desacoplado rápido

El método desacoplado rápido ha sido analizado ampliamente en el capítulo III. Las fórmulas y el esquema seguido por el programa son los que se han presentado en dicho capítulo, por ende se ha considerado innecesario insistir en los mismos criterios que ya han sido expuestos, sin embargo es importante hacer notar que en el programa se ha considerado como desajuste máximo para la exactitud de la solución un valor de 0,01 (6).

CAPITULO VI

APLICACIONES DEL PROGRAMA ELABORADO

Para comprobar el funcionamiento del programa se lo hizo trabajar con dos sistemas: Un sistema "tipo" - que normalmente aparece en las publicaciones de la IEEE y el Sistema Guayaquil.

6.1. SISTEMA TIPO IEEE (Figura 6.1)

Este sistema está compuesto por 5 barras y 7 líneas, de las 5 barras dos tienen generación y una de ellas se la escogió como barra oscilante, las barras restantes son de carga. Los datos - de entrada y salida del programa son los mostrados en las tablas 6.1. Se puede observar que se ha obtenido la solución en 5 iteraciones con una tolerancia de 0.001 mientras que para el mismo sistema, usando el método de Gauss-Seidel con una tolerancia igual la solución se obtiene en 7 iteraciones (5).

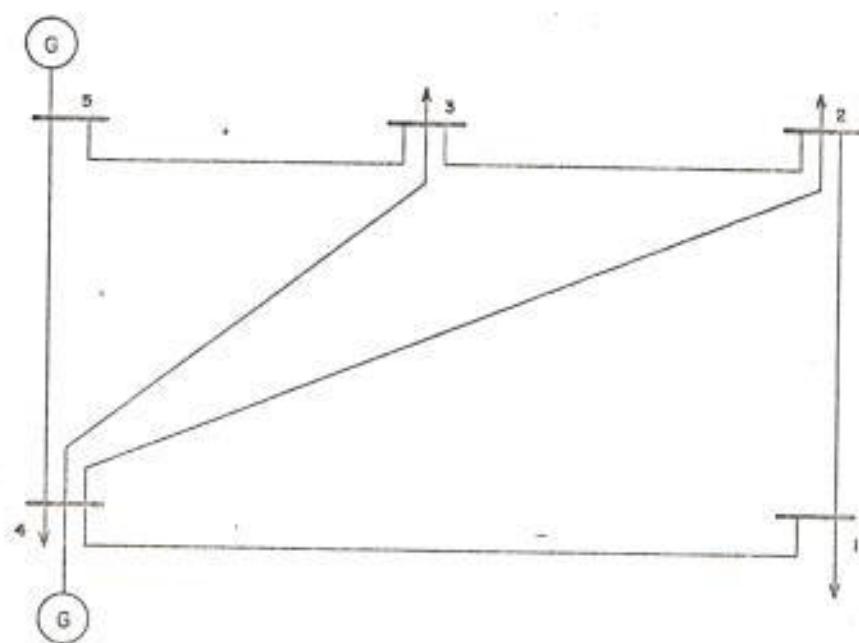


Fig. 6.1.- Sistema tipo IEEE

NOMBRE DEL USUARIO: ALFREDO VILLACRESES P

NOMBRE DEL SISTEMA: TIPO IEEE - 5 BARRAS

FECHA: 6 DE SEPTIEMBRE DE 1982

DATOS DE ENTRADA - NODOS

NODO	NOMBRE	VOLT	PGEN	QGEN	PCAR	QCAR
#		P.U.	P.U.	P.U.	P.U.	P.U.
1	UNO	1.000	0.000	0.000	0.600	0.100
2	DOS	1.000	0.000	0.000	0.400	0.050
3	TRES	1.000	0.000	0.000	0.450	0.150
4	CUATRO	1.000	0.400	0.300	0.200	0.100
5	OSC.	1.000	0.000	0.000	0.000	0.000

DATOS DE ENTRADA - LINEAS

LINEA	NOD	NOD	RESIS	REACT	SUSCEP.P
#	ENV	REC	P.U.	P.U.	- P.U.
1	5	4	0.020	0.060	0.030
2	5	3	0.080	0.240	0.050
3	4	3	0.060	0.180	0.040
4	4	2	0.060	0.180	0.040
5	4	1	0.040	0.120	0.030
6	3	2	0.010	0.030	0.020
7	2	1	0.080	0.240	0.050

TABLA 6.1.a. DATOS DE ENTRADA SISTEMA TIPO IEEE

TOTAL ITERACIONES = 5

RESULTADOS NODOS

NODO	PGEN P.U.	QGEN P.U.	PCARGA P.U.	QCARGA P.U.	VMODULO P.U.	VANGULO GRADOS
1	0.000	0.000	0.600	0.100	0.967	-7.069
2	0.000	0.000	0.400	0.050	0.971	-6.140
3	0.000	0.000	0.450	0.150	0.971	-5.764
4	0.400	0.614	0.200	0.100	1.000	-3.400
5	1.306-0.305	0.000	0.000	0.000	1.000	0.000

RESULTADOS LINEAS

LINEA	NE	NR	PENU P.U.	QENU P.U.	FREC P.U.	QREC P.U.	PPER P.U.	QPER P.U.
1	5	4	0.898	-0.285	0.981	-0.308	0.018	0.023
2	5	3	0.408	-0.020	0.395	-0.011	0.013	-0.009
3	4	3	0.250	0.062	0.246	0.089	0.004	-0.026
4	4	2	0.282	0.053	0.277	0.076	0.005	-0.024
5	4	1	0.551	0.091	0.538	0.082	0.013	0.009
6	3	2	0.185	-0.072	0.185	-0.055	0.000	-0.018
7	2	1	0.062	-0.028	0.062	0.018	0.000	-0.046

TABLA 6.1.b. RESULTADOS SISTEMA TIPO IEEE

6.2. SISTEMA GUAYAQUIL

Este sistema de potencia se muestra en la figura 6.2. y está compuesto por 37 barras y 38 líneas. El sistema Guayaquil tiene básicamente dos barras de generación ya que la misma está concentrada en el Salitral y en la planta Guayaquil, pero, para realizar un mejor estudio dentro de dichas plantas se han subdividido en 6 barras de generación y se consideran las unidades a gas del Salitral como barra oscilante. Los datos de entrada y salida son mostrados en las tablas 6.2.

6.3. COMPARACION CON EL PROGRAMA EXISTENTE EN LA ESPOL

Debido a que el programa elaborado en esta tesis ha sido desarrollado usando una microcomputadora(APPLE II) no se puede efectuar una comparación en cuanto a la velocidad para obtener el resultado puesto que éstas son inherentemente más lentas que una computadora, sin embargo si se puede establecer una gran diferencia en lo referente a la velocidad de convergencia de los métodos usados en ambos programas. Así -

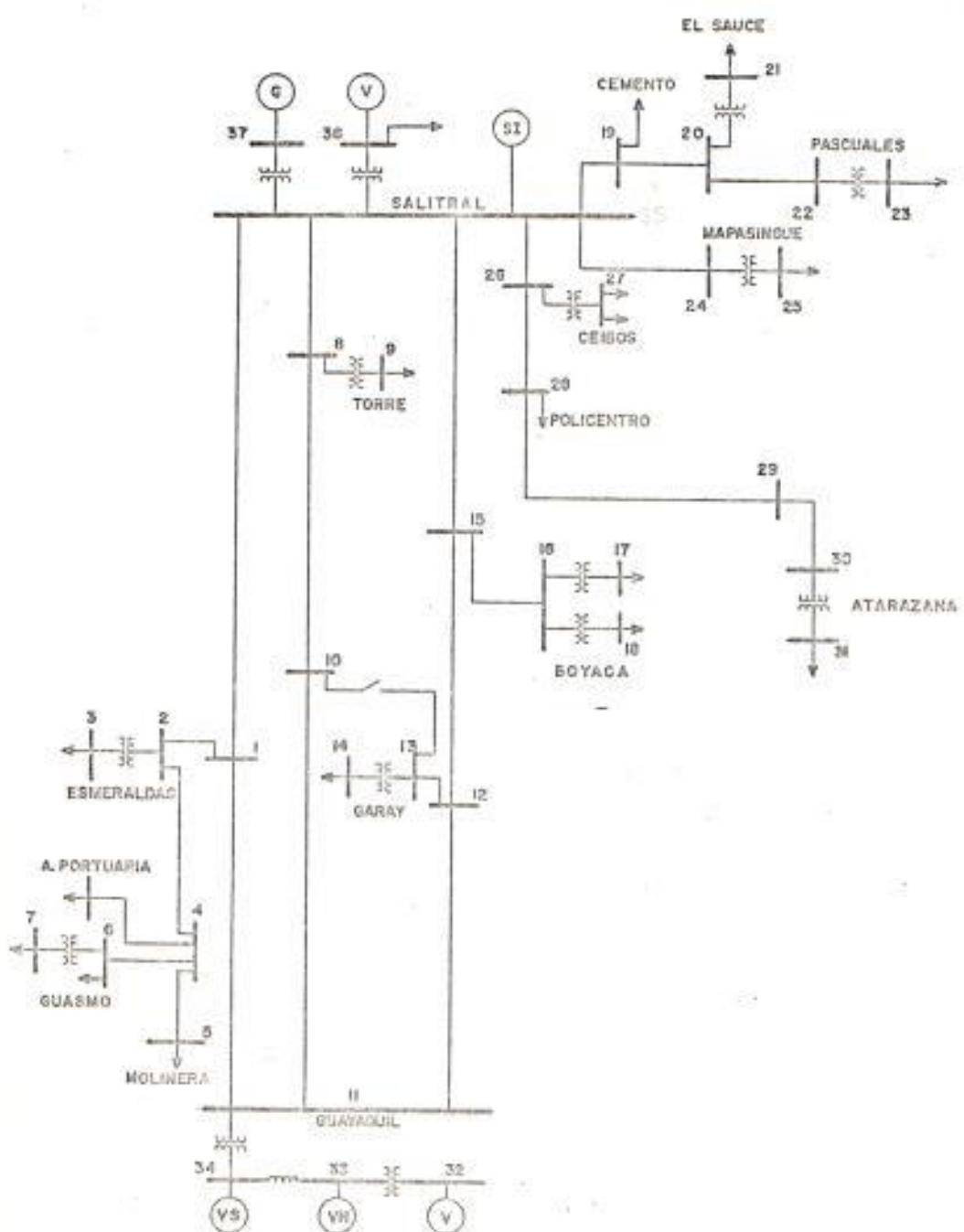


Fig. 6.2.- Sistema Guayaquil

NOMBRE DEL USUARIO: ALFREDO VILLACRESES P.

NOMBRE DEL SISTEMA: GUAYAQUIL - 81

FECHA: 6 DE SEPTIEMBRE DE 1982

DATOS DE ENTRADA - NODOS

NODO #	NOMBRE	VOLT	PGEN	QGEN	PCAR	QCAR
		P.U.	P.U.	P.U.	P.U.	P.U.
1	TAP ESME	1.000	0.000	0.000	0.000	0.000
2	ESMER 69	1.000	0.000	0.000	0.000	0.000
3	ESMER 13	1.000	0.000	0.000	0.275	0.090
4	T. PORTU	1.000	0.000	0.000	0.030	0.010
5	MOLINERA	1.000	0.000	0.000	0.060	0.020
6	GUASM 69	1.000	0.000	0.000	0.030	0.010
7	GUASM 13	1.000	0.000	0.000	0.110	0.040
8	TORRE 69	1.000	0.000	0.000	0.000	0.000
9	TORRE 13	1.000	0.000	0.000	0.200	0.030
10	T GARA S	1.000	0.000	0.000	0.000	0.000
11	GUAYA 69	1.000	0.000	0.000	0.000	0.000
12	T GARA N	1.000	0.000	0.000	0.000	0.000
13	GARAY 69	1.000	0.000	0.000	0.000	0.000
14	GARAY 13	1.000	0.000	0.000	0.275	0.090
15	T BOYACA	1.000	0.000	0.000	0.000	0.000
16	BOYAC 69	1.000	0.000	0.000	0.000	0.000
17	BOYACA 4	1.000	0.000	0.000	0.045	0.013
18	BOYAC 13	1.000	0.000	0.000	0.265	0.087
19	INECEL	1.000	0.000	0.000	0.180	0.057
20	SAUCE 69	1.000	0.000	0.000	0.045	0.015
21	SAUCE 13	1.000	0.000	0.000	0.110	0.060
22	PASCU 69	1.000	0.000	0.000	0.040	0.013
23	PASCU 13	1.000	0.000	0.000	0.100	0.050
24	MAPAS 69	1.000	0.000	0.000	0.030	0.010
25	MAPAS 13	1.000	0.000	0.000	0.120	0.070
26	CEIBO 69	1.000	0.000	0.000	0.000	0.000
27	CEIBO 13	1.000	0.000	0.000	0.295	0.090
28	T POLICE	1.000	0.000	0.000	0.020	0.006
29	T ATARAZ	1.000	0.000	0.000	0.000	0.000
30	ATAR 69	1.000	0.000	0.000	0.000	0.000
31	ATAR 13	1.000	0.000	0.000	0.120	0.035
32	VAPOR 4	1.035	0.050	0.000	0.035	0.010
33	VAPOR N	1.056	0.150	0.000	0.295	0.050
34	VAPOR S	1.060	0.100	0.000	0.000	0.000
35	SALIT 69	1.050	1.400	0.000	0.140	0.050
36	SALIT VA	1.050	0.340	0.000	0.090	0.028
37	S GAS DS	1.020	0.000	0.000	0.000	0.000

TABLA 6.2.a. DATOS DE ENTRADA - NODOS SISTEMA GUAYAQUIL

DATOS DE ENTRADA - LINEAS

LINEA #	NOD ENV	NOD REC	RESIS P.U.	REACT P.U.	SUSCEP.P. P.U.
1	36	35	0.020	0.255	0.000
2	37	35	0.004	0.062	0.000
3	34	11	0.040	0.534	0.000
4	32	33	0.210	1.268	0.000
5	24	25	0.055	0.603	0.000
6	20	21	0.036	0.440	0.000
7	22	23	0.048	0.458	0.000
8	26	27	0.024	0.289	0.000
9	30	31	0.051	0.608	0.000
10	16	17	0.184	1.400	0.000
11	16	18	0.027	0.289	0.000
12	13	14	0.018	0.220	0.000
13	8	9	0.027	0.249	0.000
14	2	3	0.018	0.220	0.000
15	6	7	0.041	0.590	0.000
16	33	34	0.015	0.517	0.000
17	35	24	0.023	0.072	0.000
18	35	19	0.032	0.100	0.000
19	19	20	0.009	0.027	0.000
20	20	22	0.014	0.044	0.000
21	35	26	0.015	0.047	0.000
22	26	28	0.013	0.031	0.000
23	28	29	0.008	0.018	0.000
24	29	30	0.008	0.017	0.000
25	35	15	0.031	0.098	0.000
26	15	16	0.008	0.011	0.000
27	15	12	0.004	0.010	0.000
28	12	13	0.002	0.008	0.000
29	12	11	0.012	0.028	0.000
30	35	8	0.015	0.034	0.000
31	8	10	0.014	0.033	0.000
32	10	11	0.011	0.025	0.000
33	35	1	0.027	0.086	0.000
34	1	2	0.003	0.011	0.000
35	2	4	0.009	0.027	0.000
36	4	5	0.019	0.026	0.000
37	4	6	0.020	0.027	0.000
38	1	11	0.005	0.015	0.000

TABLA 6.2.b. DATOS DE ENTRADA - LINEAS SISTEMA GUAYAQUIL -

TOTAL ITERACIONES = 17

RESULTADOS NODOS

NODO	PGEN P.U.	QGEN P.U.	PCARGA P.U.	QCARGA P.U.	VMODULO P.U.	VANGULO GRADOS
1	0.000	0.000	0.000	0.000	1.026	-4.889
2	0.000	0.000	0.000	0.000	1.022	-5.162
3	0.000	0.000	0.275	0.090	0.996	-8.480
4	0.000	0.000	0.030	0.010	1.018	-5.462
5	0.000	0.000	0.060	0.020	1.016	-5.527
6	0.000	0.000	0.030	0.010	1.013	-5.608
7	0.000	0.000	0.110	0.040	0.983	-9.251
8	0.000	0.000	0.000	0.000	1.039	-3.983
9	0.000	0.000	0.200	0.030	1.025	-6.621
10	0.000	0.000	0.000	0.000	1.032	-4.465
11	0.000	0.000	0.000	0.000	1.027	-4.834
12	0.000	0.000	0.000	0.000	1.022	-5.065
13	0.000	0.000	0.000	0.000	1.021	-5.174
14	0.000	0.000	0.275	0.090	0.994	-8.500
15	0.000	0.000	0.000	0.000	1.023	-5.021
16	0.000	0.000	0.000	0.000	1.019	-5.155
17	0.000	0.000	0.045	0.013	0.990	-8.598
18	0.000	0.000	0.265	0.087	0.983	-9.404
19	0.000	0.000	0.180	0.057	1.013	-5.375
20	0.000	0.000	0.045	0.015	1.006	-5.748
21	0.000	0.000	0.110	0.060	0.974	-8.452
22	0.000	0.000	0.040	0.013	1.001	-6.045
23	0.000	0.000	0.100	0.050	0.972	-8.601
24	0.000	0.000	0.030	0.010	1.040	-3.628
25	0.000	0.000	0.120	0.070	0.989	-7.449
26	0.000	0.000	0.000	0.000	1.036	-4.121
27	0.000	0.000	0.295	0.090	0.999	-8.723
28	0.000	0.000	0.020	0.006	1.032	-4.321
29	0.000	0.000	0.000	0.000	1.030	-4.417
30	0.000	0.000	0.000	0.000	1.029	-4.515
31	0.000	0.000	0.120	0.035	1.000	-8.476
32	0.050-0.009	0.035	0.010	1.035	-8.065	
33	0.150	0.070	0.295	0.050	1.056	-9.276
34	0.100	0.077	0.000	0.000	1.060	-5.831
35	1.400	1.649	0.140	0.050	1.050	-3.168
36	0.340	0.016	0.090	0.020	1.050	0.160
37	0.921-0.527	0.000	0.000	1.020	0.000	

TABLA 6.2.c. RESULTADOS - NODOS SISTEMA GUAYAQUIL

RESULTADOS LINEAS

LINEA #	NE	NR	PEN _V P.U.	QEN _V P.U.	PREC P.U.	QREC P.U.	PPER P.U.	QPER P.U.
1	36	35	0.250	-0.012	0.249	-0.027	0.001	0.014
2	37	35	0.921	-0.527	0.916	-0.594	0.004	0.067
3	34	11	-0.030	0.068	-0.031	0.066	0.000	0.003
4	32	33	0.015	-0.019	0.015	-0.020	0.000	0.001
5	24	25	0.121	0.082	0.120	0.070	0.001	0.012
6	20	21	0.111	0.067	0.110	0.060	0.001	0.007
7	22	23	0.101	0.056	0.100	0.050	0.001	0.006
8	26	27	0.297	0.117	0.295	0.090	0.002	0.027
9	30	31	0.120	0.042	0.120	0.033	0.001	0.009
10	16	17	0.045	0.016	0.045	0.013	0.000	0.003
11	16	18	0.267	0.110	0.265	0.087	0.002	0.023
12	13	14	0.277	0.109	0.275	0.090	0.002	0.019
13	8	9	0.201	0.040	0.200	0.030	0.001	0.010
14	2	3	0.277	0.109	0.275	0.090	0.002	0.019
15	6	7	0.111	0.048	0.110	0.040	0.001	0.008
16	33	34	-0.130	-0.000	-0.130	-0.008	0.000	0.008
17	35	24	0.152	0.094	0.151	0.092	0.001	0.002
18	35	19	0.486	0.239	0.478	0.212	0.009	0.027
19	19	20	0.298	0.155	0.297	0.152	0.001	0.003
20	20	22	0.141	0.070	0.141	0.069	0.000	0.001
21	35	26	0.443	0.183	0.440	0.173	0.003	0.010
22	26	28	0.144	0.056	0.144	0.055	0.000	0.001
23	28	29	0.121	0.048	0.120	0.046	0.000	0.000
24	29	30	0.130	0.050	0.130	0.050	0.000	0.000
25	35	15	0.407	0.167	0.402	0.150	0.005	0.017
26	15	16	0.314	0.128	0.313	0.126	0.001	0.001
27	15	12	0.080	0.023	0.083	0.022	0.000	0.000
28	12	13	0.277	0.109	0.277	0.109	0.000	0.001
29	12	11	-0.188	-0.087	-0.189	-0.088	0.000	0.001
30	35	8	0.512	0.128	0.508	0.119	0.004	0.009
31	8	10	0.307	0.080	0.306	0.077	0.001	0.003
32	10	11	0.304	0.076	0.305	0.074	0.001	0.002
33	35	1	0.429	0.167	0.423	0.151	0.005	0.017
34	1	2	0.509	0.202	0.508	0.199	0.001	0.003
35	2	4	0.232	0.091	0.231	0.089	0.001	0.002
36	4	5	0.060	0.020	0.060	0.020	0.000	0.000
37	4	6	0.141	0.059	0.141	0.058	0.000	0.001
38	1	11	-0.086	-0.052	-0.086	-0.052	0.000	0.000

TABLA 6.2.d. RESULTADOS - LINEAS SISTEMA GUAYAQUIL

por ejemplo, para el sistema Guayaquil, usando el programa LFL en la IBM 1130 se obtuvo la solución con una tolerancia de 0.00005 en 65 iteraciones, mientras que usando el programa aquí elaborado, con una tolerancia de 0.001 se obtuvo la solución en 25 iteraciones. Realmente no se puede hacer una comparación en cuanto a la exactitud de las respuestas en base a las tolerancias puesto que 0.00005 es para voltajes - mientras que 0.001 es para desajustes de potencia, de acuerdo con los métodos usados en cada programa. Sin embargo en las tablas 6.3 y 6.4, se presentan una comparación entre los resultados obtenidos en este trabajo y los obtenidos - usando el LFL, de las mismas se han sacado los siguientes promedios en la variación de los resultados:

módulo de voltaje: 0.408 % de diferencia

ángulo de voltaje: 0.767 % de diferencia

Potencias activas: 0.410 % de diferencia

potencias reactivas: 2.130 % de diferencia

Es necesario hacer notar que no se han tomado en

cuenta los valores obtenidos para los nodos generadores, ya que aunque en algunos de ellos se han sacado resultados idénticos, en otros son sumamente diferentes. La explicación de lo anterior radica en el hecho de que el programa LFL considera límites de generación de potencia reactiva para los nodos generadores mientras que el aquí elaborado no lo hace, es por eso que algunos de los voltajes deseados para los nodos generadores al correr el programa LFL no han sido mantenidos debido a la imposibilidad de sobrepasar los límites de potencia reactiva allí considerados, así por ejemplo para las barras 34 y 35 se descaban voltajes de 1.060 p.u. y 1.050 p.u., respectivamente y luego de ejecutar el programa LFL, para cumplir con las condiciones de generación reactiva, éstos voltajes se han tenido que ajustar a 1.053 p.u. y 1.046 p.u., respectivamente. Aunque la diferencia entre los voltajes deseados y los obtenidos por el programa LFL no es grande, se puede observar que si es notoria la diferencia en el cálculo de las potencias reactivas generadas ya que en el programa presentado por ésta tesis se han rebasado los límites de potencia ya que no han sido considerados.

rados.

En todo caso tomando en cuenta lo dicho anteriormente y observando la variación de los demás resultados, el programa da una solución aceptable en menor número de iteraciones que el programa LFL. Es necesario indicar que una tolerancia - de 0.01 es un buen valor(6) para el método considerado en éste trabajo, una tolerancia menor puede ocasionar que se ejecuten más iteraciones pero sin mejorar el resultado en cuanto a exactitud.

Existe otra diferencia básica entre los programas aquí comparados y es que el programa LFL permite el uso de transformadores con cambio de tap bajo carga.

NODO #	V.MOD. ALF p.u.	V. MOD. TESIS p.u.	%DIF.	V.ANG. LFL °	V.ANG. TESIS °	%DIF.
1	1.021	1.026	0.49	-4.8	-4.889	1.82
2	1.018	1.022	0.39	-5.1	-5.162	1.20
3	0.991	0.996	0.50	-8.5	-8.480	0.24
4	1.013	1.018	0.49	-5.4	5.462	1.14
5	1.012	1.016	0.39	-5.5	5.527	0.49
6	1.009	1.013	0.40	-5.6	-5.608	0.14
7	0.978	0.983	0.51	-9.2	-9.251	0.55
8	1.035	1.039	0.39	3.9	3.983	2.09
9	1.021	1.025	0.39	-6.6	-6.621	0.32
10	1.028	1.032	0.39	-4.4	-4.465	1.46
11	1.023	1.027	0.39	-4.8	-4.834	0.70
12	1.018	1.022	0.39	-5.0	-5.065	1.28
13	1.016	1.021	0.49	-5.1	-5.174	1.43
14	0.990	0.994	0.40	-8.5	-8.500	0
15	1.018	1.023	0.49	-5.0	-5.021	0.42
16	1.015	1.019	0.39	-5.1	-5.155	1.07
17	0.986	0.990	0.40	-8.6	-8.598	0.02
18	0.979	0.983	0.40	-9.4	-9.404	0.04
19	1.009	1.013	0.40	-5.3	-5.375	1.40
20	1.002	1.006	0.40	-5.7	-5.748	0.84
21	0.970	0.974	0.41	-8.4	-8.452	0.62
22	0.997	1.001	0.40	-6.0	-6.045	0.74
23	0.968	0.972	0.41	-8.6	-8.601	0.01
24	1.037	1.040	0.29	-3.6	-3.628	0.77
25	0.985	0.989	0.40	-7.4	-7.449	0.65
26	1.032	1.036	0.39	-4.1	-4.121	0.51
27	0.995	0.999	0.40	-8.7	-8.723	0.26
28	1.029	1.032	0.29	-4.3	-4.321	0.49
29	1.027	1.030	0.29	-4.4	-4.417	0.39
30	1.025	1.029	0.40	-4.5	-4.515	0.33
31	0.995	1.000	0.50	-8.5	-8.476	0.28
32	1.034	1.035	0.10	-8.2	-8.065	1.64
33	1.056	1.056	0.00	-9.2	-9.276	0.81
34	1.053	1.060	0.66	-5.8	-5.831	0.53
35	1.046	1.050	0.38	-3.1	-3.168	2.15
36	1.050	1.050	0.00	0.1	0.160	#
37	1.019	1.020	0.10	0.0	0.000	0

TABLA 6.3.

COMPARACION ENTRE LOS RESULTADOS OBTENIDOS-NODOS

LINEA #	P. ENV. LFL p.u.	P. ENV. TESIS p.u.	% DIF.	Q. ENV. LFL p.u.	Q. ENV. TESIS p.u.	% DIF.
1	0.250	0.250	0	0.106	-0.012	*
2	0.930	0.921	0.97	0.341	-0.527	*
3	-0.030	-0.030	0	0.064	0.068	*
4	0.015	0.015	0	0.002	-0.019	*
5	0.121	0.121	0	0.082	0.082	0
6	0.111	0.111	0	0.067	0.067	0
7	0.101	0.101	0	0.056	0.056	0
8	0.297	0.297	0	0.118	0.117	0.85
9	0.121	0.120	0.83	0.045	0.042	6.67
10	0.045	0.045	0	0.016	0.016	0
11	0.267	0.267	0	0.111	0.110	0.90
12	0.277	0.277	0	0.109	0.109	0
13	0.201	0.201	0	0.040	0.040	0
14	0.277	0.277	0	0.109	0.109	0
15	0.111	0.111	0	0.049	0.048	2.04
16	-0.130	-0.130	0	0.012	0.000	*
17	0.152	0.152	0	0.094	0.094	0
18	0.487	0.486	0.21	0.240	0.239	0.42
19	0.298	0.298	0	0.157	0.155	1.27
20	0.141	0.141	0	0.070	0.070	0
21	0.443	0.443	0	0.182	0.183	0.55
22	0.143	0.144	0.69	0.054	0.056	3.57
23	0.121	0.121	0	0.045	0.048	6.25
24	0.122	0.133	6.15	0.047	0.050	6.00
25	0.410	0.407	0.73	0.173	0.167	3.47
26	0.315	0.314	0.32	0.130	0.128	1.54
27	0.088	0.088	0	0.022	0.023	4.35
28	0.276	0.277	0.36	0.107	0.109	1.83
29	-0.189	-0.188	0.53	-0.088	-0.087	1.14
30	0.514	0.512	0.39	0.134	0.128	4.48
31	0.309	0.307	0.65	0.086	0.080	6.89
32	0.308	0.306	0.65	0.082	0.076	7.32
33	0.431	0.429	0.46	0.171	0.167	2.34
34	0.510	0.509	0.20	0.202	0.202	0
35	0.232	0.232	0	0.091	0.091	0
36	0.061	0.060	1.64	0.021	0.020	4.76
37	0.142	0.141	0.70	0.060	0.059	1.67
38	-0.086	-0.086	0	-0.053	-0.052	1.89

TABLA 6.4.
COMPARACION ENTRE LOS RESULTADOS OBTENIDOS-LINEAS

CONCLUSIONES Y RECOMENDACIONES

Luego de haber comparado el programa elaborado en esta tesis con el programa existente en la ESPOL, el LFL, se puede concluir que el primero, aunque no es completo - arroja resultados bastante exactos y por lo tanto es un programa que podrá ser usado en sistemas que no teñ gan las restricciones que no se han considerado en el mismo. Así por ejemplo puede ser usado en sistemas co mo el Guayaquil que no tiene transformadores con cambio automático de tap bajo carga, además para conseguir resultados que estén completamente de acuerdo con la rea lidad se pueden hacer consideraciones de tal forma que el sistema a estudiar solo presente como barra generadora la oscilante y todas las demás se deben tomar co mo barras de carga, al hacer esto último no tendremos que preocuparnos por las violaciones en la generación de potencia reactiva.

La ejecución del programa tarda bastante. Para solucionar este problema se deberá usar un compilador PAS CAL y el tiempo de ejecución será disminuido entre 10 y 12 veces de lo que demora actualmente (aproximadamente 2 horas). Lo último se debe a que el programa es

critico en PASCAL es compilado y convertido a un programa objeto en lenguaje P para luego mediante un interpretador P ejecutar el programa objeto, en cambio, si existiera un compilador PASCAL este pasaría el programa fuente en PASCAL directamente a programa objeto. La existencia del lenguaje P se debe a que para los fabricantes de lenguajes es más sencillo diseñar un interpretador P que diseñar y elaborar un compilador para cada computador.

La finalidad de haber usado un lenguaje como PASCAL, fue de mantener siempre vigente la intención de someterlo a modificaciones y extensiones como las mencionadas anteriormente.

Las principales recomendaciones para optimizar el programa en cuanto a su valor analítico, son las siguientes:

1. Considerer límites de generación de potencia reactiva para barras generadoras.

Existen dos formas para considerar lo anterior (6).

La primera consiste en, a cada violación de los límites de potencia reactiva, convertir la barra - afectada (PV) en una barra de carga (PQ) y deberá permanecer así durante el resto de los cálculos -

hasta que pueda ser reconvertida nuevamente si es que no comete alguna violación. Lo anterior involucra retriangulizar la matriz B'' cada vez que exista una violación. La segunda forma que sugiere la referencia (6) para ajustar las respuestas obtenidas por el programa de esta tesis es corregir el voltaje de la barra violadora "k" por una cantidad V_k en la siguiente iteración para reducir el error $Q_k = (Q_k \text{ límite} - Q_k)$ a cero. La convergencia de este proceso es rápida cuando se usa un factor aproximado de sensitividad S_k que relaciona V_k y Q_k dado por:

$$V_k = S_k Q_k / V_k$$

Definido S_k de acuerdo a la ecuación 3.21 es el elemento de la diagonal correspondiente a la barra k en la inversa de la matriz B'' aumentada por la fila y la columna k de la barra que se convierte a PQ. En un apendice de la referencia 6 se muestra que S_k se puede calcular fácil y rápidamente sin retriangulizar la matriz B'' que es lo que se sugirió anteriormente. La corrección mencionada se debe dejar de aplicar

si en alguna etapa de la solución el valor de v_k es restablecido.

2. Considerar transformadores que tengan cambio de taps bajo carga.

Esta recomendación no tiene la misma importancia como la anterior cuando se analizan sistemas como los estudiados aquí (por ejemplo Sistema Guayaquil) ya que en ellos no existen este tipo de elementos, sin embargo es necesario dar a conocer como se pueden tomar en cuenta en forma muy sencilla. Así para una barra k que tiene control de voltaje por cambios en las derivaciones se tiene el siguiente algoritmo.

$$t_i \text{ (nuevo)} - t_i \text{ (viejo)} = \pm \alpha (v_k - v_k^{\text{sp}})$$

Donde t_i es el valor del tap en p.u.y α es un factor empírico y normalmente para un sistema bien condicionado α igual a 1 da convergencia rápida. El factor de corrección debe ser aplicado después de calcular los voltajes.

Lo de mayor importancia es que, la rata de convergencia es afectada poco, aunque los cambios del

círcuito del transformador tratado se reflejan en el cálculo de los elementos de la matriz B^{II} en cada cambio de tap, de tal forma que la matriz B^{II} es formada y factorizada al comienzo del estudio asumiendo que todos los taps son nominales y esta permanece constante durante el resto de la solución.

B I B L I O G R A F I A

1. ALVARADO FERNANDO L., "POWER SYSTEM PROGRAMMING IN PASCAL", POWER INDUSTRY COMPUTER APPLICATIONS CONFERENCE (PICA) 15-18 MAYO/79, CLEVELAND OHIO, PAGS. 238 - 244.
2. ELGERD OLLE I., "ELECTRIC ENERGY SYSTEM THEORY", - Mc. GRAW HILL, 1979.
3. LAUGHTON A. Y M.W. HUMPHREY D., "NUMERICAL TECHNIQUES IN SOLUTION OF POWER SYSTEMS LOAD FLOW PROBLEMS" , PROCEEDINGS IEEE, LONDON, SEPT. 1964, VOL.TII PAGS. 1575 - 1588.
4. SASSON ALBERT Y F.J. JAAMES, "DIGITAL METHODS APPLIED TO POWER FLOW STUDIES", IEEE TRANS. POWER APPARATUS AND SYSTEMS, NEW ORLEANS, JULIO 1967, VOL.PAS - 86, PAGS. 860 - 867.
5. STAGG AND EL - ABIAD, "COMPUTER METHODS IN POWER SYSTEM ANALYSIS", NEW YORK: Mc GRAW HILL, 1968.
6. STOTT BRIAN Y O.ALSAC, "FAST DECOUPLED LOAD FLOW", presentado a IEEE PES Summer Meet, VANCOUVER, JULIO 15 , 20, 1973, IEEE TRANS. POWER APP. SYSTEM, VOL PAS - 93 , PAGS. 859 - 869, MAYO/JUNIO, 1974.

7. STOTT BRIAN, "REVIEW OF LOAD FLOW CALCULATION METHODS" PROCEEDINGS DE IEEE, JULIO 1974, VOL 62, PAGS. 916 - 929, MANCHESTER.
8. STOTT BRIAN, "DECOUPLED NEWTON LOAD FLOWS", IEEE TRANS POWER APPARATUS AND SYSTEMS, VOL. PAS - 91, PAGS. 1955- 1959, SEPT/OCT., 1972.
9. TINNEY W.F., POWELL W.L., WALKER J.W., "PROGRAMMING OF SPARSITY - DIRECTED ORDERING SCHEMES", PROC. DE 1975 PSCC, OCT. 1975, PAGS. 1 - 14, PORTLAND, OREGON.
10. TINNEY W.F., WALKER J.W., "DIRECTED SOLUTIONS OF SPARSE NETWORKS EQUATIONS BY OPTIMALLY ORDERED TRIANGULAR FACTORIZATION", PROC. IEEE, VOL 55, PAGS. 1801- 1809, NOV, PORTLAND, OREGON.
11. TINNEY W.F. AND C.E. HART, "POWER FLOW SOLUTION BY NEWTON'S METHOD", IEEE TRANS. POWER APPARATUS AND SYSTEMS, VOL. PAS-86, PAGS. 1449 - 1460, NOV. 1967 , NEW YORK.
12. TINNEY W.F. AND MEYER W. SCOTT, "SOLUTION OF LARGE SPARSE SYSTEMS BY ORDERED TRIANGULAR FACTORIZATION", IEEE TRANSACTIONS ON AUTOMATIC CONTROL, VOL.AC- 18 , PAGS. 333 - 345, PORTLAND, OREGON, AGOSTO 1973.

13. VAN NESS J., "ITERATION METHODS FOR DIGITAL LOAD FLOW STUDIES", AIEE TRANS. POWER APP. SYST., VOL. 78, PAGS 583 - 588, NEW YORK, AGOSTO, 1959.
14. VAN NESS J. Y J.H. GRIFFIN, "ELIMINATION METHODS FOR LOAD FLOW STUDIES", AIEE TRANS. POWER APP.SYST. VOL. 80, PAGS.299 - 304, NEW YORK, JUNIO 1961.
15. WARD J.B., HALE H.W., "DIGITAL COMPUTER SOLUTION - OF POWER FLOW PROBLEMS", AIEE TRANS. POWER APP.SYST. VOL.75, PAGS. 398 - 404, NEW YORK, JUNIO 1.956.
16. WEEDY B.M., "ELECTRIC POWER SYSTEMS", WILEY, 1979.
17. WICKMAN W., "PASCAL IS A "NATURAL" ", IEEE SPECTRUM PAGS. 35 - 41.