



DIVISION DE ESTUDIOS DE POSGRADO
Facultad de Ingeniería

01170
2
zej

Flujos de Potencia para Redes
Eléctricas con Altos
Cocientes R/X

Luis Enrique Díaz Espeleta

TESIS CON
FALLA DE ORIGEN

TESIS

Presentada a la División de Estudios de
Posgrado de la

FACULTAD DE INGENIERIA
de la

UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO
como requisito para obtener
el grado de

MAESTRO EN INGENIERIA ELECTRICA

Asesor: Dr. Hector Sarmiento Uruchurtu

Ciudad Universitaria
Febrero de 1992



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso

DERECHOS RESERVADOS © PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis está protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

INDICE

INTRODUCCION	1
1. MODELOS PARA LA SOLUCION DE FLUJOS DE CARGA	6
1.1 ECUACIONES DE FLUJOS DE CARGA	6
1.1.1 Generalizacion de las ecuaciones de flujos de carga	10
1.2 CRITERIOS DE CONVERGENCIA	11
1.2.1 Desequilibrio de Potencia	12
1.2.2 Cambio en la magnitud de Voltaje	13
1.3 METODO DE NEWTON-RAPHSON (N-R)	14
1.3.1 El caso escalar	14
1.3.2 El caso multidimensional.....	15
1.3.3 Solución de flujos de carga aplicando las ecuaciones de N-R	16
1.4 METODO DESACOPLADO	23
1.5 METODO DESACOPLADO RAPIDO.....	24
1.6 DIFERENCIAS ENTRE LOS METODOS.....	26
2. ALGORITMO PROPUESTO	28
2.1 MODIFICACIONES AL METODO DESACOPLADO RAPIDO	29
2.2 MODIFICACION PROPUESTA PARA SER APLICADA AL METODO DESACOPLADO RAPIDO	32
3. DESCRIPCION DEL ALGORITMO.....	34
3.1 PASOS DEL ALGORITMO	35
3.2 DESCRIPCION DE LAS RUTINAS QUE CONFORMAN EL PROGRAMA.....	38
3.2.1 Lectura de datos	38
3.2.2 Impresión de datos de entrada	40
3.2.3 Cálculo de la matriz Ybus	40
3.2.4 Cálculo de las Matrices B', B'', B'''	44
3.2.5 Función para el Cálculo de Flujos de Carga	44
3.2.6 Función Gauss-Seidell	44

4. APLICACION DEL ALGORITMO PARA ALGUNOS MODELOS DE REDES ELECTRICAS	49
4.1 CASO 1. SISTEMA DE 3 BUSES Y 3 LINEAS CON 1 GENERADOR	49
4.2 CASO 2. SISTEMA DE 6 BUSES Y 7 LINEAS CON 2 GENERADORES	59
CONCLUSIONES	71
BIBLIOGRAFIA.....	75
ANEXO	76

INTRODUCCION

El flujo de carga es el estudio hecho para determinar las condiciones de operación de un sistema eléctrico de potencia, el cual se define como una red eléctrica con muchos nodos o buses, nodos de generación que inyectan energía a la red, nodos de carga que consumen energía de ella y derivaciones o ramas uniendo a los nodos. Por lo que, el estudio de la forma y cantidad en que la energía recorre los diferentes caminos entre los puntos de generación y carga se denomina flujos de carga o flujos de potencia.

El problema de flujos de carga parte del planteamiento de un modelo matemático de una red eléctrica que describe la relación entre los voltajes y las potencias que circulan en toda la complejidad del sistema. Para la aplicación del modelo se consideran condiciones ideales de funcionamiento en la red :

- El sistema se encuentra en perfecto equilibrio de cargas, es decir la energía que se produce es igual a la que se consume.
- La carga no varía con el tiempo
- No existen fenomenos ajenos o inherentes al sistema produciendo fallas que se traduzcan en disturbios de la operación del sistema.

Se sabe, sin embargo, que las condiciones reales son otras precisamente opuestas a las señaladas anteriormente: existen disturbios que traen inestabilidad al

sistema, la carga tiene variaciones considerables a través de las 24 horas del día las cuales pueden llegar a exigir mayor energía de la que se puede suministrar, los conductores eléctricos tienen ciertos límites de ampicidad y de temperatura que limitan la funcionabilidad de la red, entre otras.

En conclusión, la aplicación de los modelos matemáticos que se han estudiado en los últimos veinte años consideran condiciones ideales para el cálculo de voltajes y ángulos en los buses y el cálculo de corrientes y potencias que circulan por las líneas que los interconectan.

El concepto de flujos de potencia está muy relacionado con los estudios de planeación, optimización, operación y control de los Sistemas Eléctricos de Potencia.

Es muy difícil seleccionar el modelo a aplicar para la solución de problemas de flujos de potencia que se presentan frecuentemente en los Sistemas Eléctricos de Potencia. Para ello se requiere hacer un análisis exhaustivo de las ventajas de cada uno de los métodos disponibles. Estos, para poder aplicarse eficazmente en los Sistemas Eléctricos de Potencia, necesitan de algoritmos computacionales cíclicos o iterativos, lo que quiere decir que repiten el mismo proceso de cálculo donde la condiciones de arranque de un nuevo ciclo son las condiciones finales del ciclo anterior hasta llegar a un punto en el que la diferencia entre los resultados de un ciclo y otro están dentro de un margen de tolerancia preestablecido. Cuando ocurre esto se dice que el sistema converge en los resultados obtenidos después de un número de iteraciones. También puede ocurrir que después de un número máximo de iteraciones fijadas, el sistema no alcance la convergencia, se dice

entonces que el sistema diverge.¹ Los algoritmos están muy asociados con el tiempo de ejecución, que depende del número de iteraciones y el tiempo de cada una de éstas depende del modelo matemático, como también de la cantidad de memoria a usar durante el proceso.

Todas estas consideraciones nos permiten concluir que ningún algoritmo es aplicable, con el mismo nivel de eficiencia, a todos los modelos de red que puedan existir, ya que ninguno de ellos posee todas las características deseables de los demás. Por esta razón el problema de los flujos de carga en Sistemas Eléctricos de Potencia, desde hace más de tres décadas sigue siendo motivo de estudio, encaminado éste a la búsqueda de la optimización de algoritmos existentes que permitan su aplicación, en cada vez más modelos de redes eléctricas.

Los nuevos algoritmos tienden a mejorar los tiempos de ejecución, disminuir el uso de memoria y mejorar la exactitud en los resultados. Los métodos numéricos son más eficientes cuando aprovechan las propiedades físicas del sistema en estudio. Es por esto que la solución de las matrices dispersas en redes muy grandes, por medio de las técnicas de eliminación ordenada, trajo mayor eficiencia que la inversión de matrices en los métodos usados. Recientemente se estudió la interacción entre las pérdidas físicas de los flujos de MW y MVAR, mediante el desacoplamiento matemático de MW y el ángulo de voltaje y entre los MVAR y la magnitud de voltaje. Esto derivó en el método Desacoplado rápido, el cual mantiene ciertas ventajas de los métodos existentes y aporta mayor rapidez, es más fácil en su codificación e implementación, posee mayor confiabilidad que el

¹ B. Stott, O. Alsac, "Fast Decoupled Load Flow", IEEE Trans. Power Appl. Syst., vol. Pas-93 May/June, pp. 859.

Newton-Raphson y con requerimientos mínimos de memoria.

El método Desacoplado rápido se convirtió en el método más usado por lo expuesto anteriormente, aunque empezó a manifestar problemas de convergencia para sistemas con altos cocientes de la resistencia entre la reactancia. Para solucionar esto se han empleado varios algoritmos, uno de compensación serie y otro de compensación en paralelo. Sin embargo, estos algoritmos son muy diferentes al Desacoplado rápido por lo que no permiten su implementación en los algoritmos usados comúnmente. Es por esto que la modificación al método Desacoplado rápido, planteada en este trabajo, representa una solución rápida y eficiente para la mayoría de los algoritmos implementados actualmente y que tienen problemas para redes con altos cocientes R/X .²

El modelo propuesto en este trabajo parte de cambios hechos a un modelo muy conocido. Esta modificación permite el uso del algoritmo en forma mejorada para redes eléctricas con altos cocientes R/X , es decir para redes en las que la resistencia de algunas líneas es muy considerable.

El primer capítulo describe brevemente los principales conceptos de flujos de carga, con la derivación de sus ecuaciones y los criterios de convergencia utilizados. Posteriormente se explican los métodos más usados de flujos de carga: Newton-Raphson, Desacoplado y Desacoplado Rápido. Finalmente se mencionan algunas diferencias entre ellos.

² Anjan Bose. "A modification to the fast decoupled power flow for networks with high R/X ratios". IEEE Power Appl. Syst., Vol. 3 No. 2. May 1988, pag. 743.

El segundo capítulo trata sobre el algoritmo propuesto. Se explican las consideraciones y modificaciones que se hacen al método Desacoplado para obtener las ecuaciones correspondientes a este nuevo algoritmo.

El tercer capítulo describe las rutinas que componen el algoritmo propuesto, con sus respectivos diagramas de flujo, así como la forma en que se deben meter los datos al programa.

El cuarto y último capítulo muestra los modelos de redes que se aplican al sistema, los resultados numéricos y gráficos y la conclusiones.

Al final se anexa un listado del programa codificado en lenguaje "C".

1. MODELOS PARA LA SOLUCION DE FLUJOS DE CARGA

En este capítulo se describen en forma breve los algoritmos que se han desarrollado para la solución de flujos de carga en redes eléctricas. Primero se explica como surgen las ecuaciones de flujos de carga a través de un modelo de red sencillo, después se describen los 3 métodos que he considerado más significativos para llegar a entender el propuesto. Finalmente se explican las diferencias entre cada uno.

1.1 ECUACIONES DE FLUJOS DE CARGA

Partamos de una red eléctrica de dos nodos y dos generadores conectados a través de una línea L como se muestra en la Figura 1.1.a. Cada bus recibe alimentación del generador y envía energía a la carga Sc. La Figura 1.1.b muestra el modelo π utilizado para la línea. La figura 1.1.c nos indica como queda la red modelada para poder aplicar los conceptos de redes:

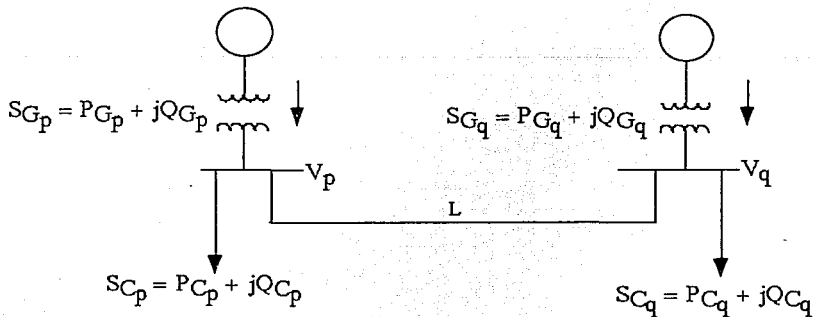


Figura 1.1.a

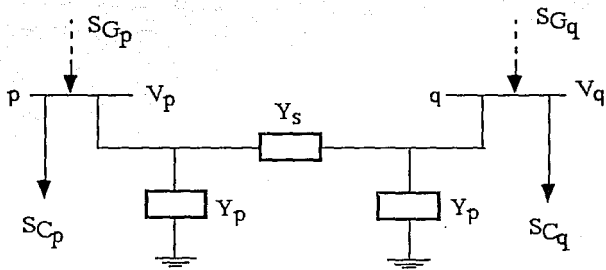


Figura 1.1.b

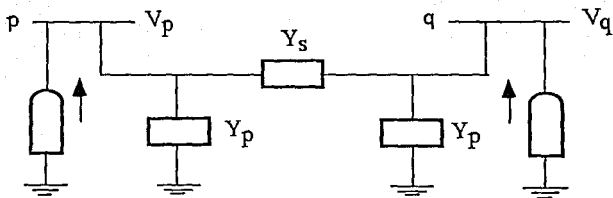


Figura 1.1.c

Aplicando las leyes de Kirchof a la red anterior, tenemos:

$$S_p = V_p I_p^*$$

despejamos el valor de I_p ,

$$I_p = \frac{S_p^*}{V_p^*} = Y_p V_p + (V_p - V_q) Y_s$$

1.1

$$I_q = \frac{S_q^*}{V_q^*} = V_q Y_p + (V_q - V_p) Y_s$$

Introduciendo los elementos de Ybus tenemos:

$$y_{pp} = Y_p + Y_s$$

$$y_{pq} = y_{qp} = -Y_s$$

1.2

$$y_{qq} = Y_q + Y_s$$

Las ecuaciones de corriente quedan :

$$I_p = y_{pp} V_p + y_{pq} V_q$$

1.3

$$I_q = y_{qp} V_p + y_{qq} V_q$$

Las ecuaciones de ybus, corriente y voltaje quedan:

$$I_{bus} = \begin{bmatrix} I_p \\ I_q \end{bmatrix} \quad \text{Vector de corrientes 1.4}$$

$$V_{bus} = \begin{bmatrix} V_p \\ V_q \end{bmatrix} \quad \text{Vector de voltajes 1.5}$$

$$Y_{bus} = \begin{bmatrix} Y_{pp} & Y_{pq} \\ Y_{qp} & Y_{qq} \end{bmatrix} \quad \text{Matriz de admitancias 1.6}$$

En forma compacta las ecuaciones se pueden expresar como :

$$I_{bus} = Y_{bus} V_{bus} \quad 1.7$$

o

$$V_{bus} = Z_{bus} I_{bus} ; Z_{bus} = Y_{bus}^{-1} \quad 1.8$$

Pero como lo que se conoce en un Sistema Eléctrico de Potencia no es la corriente sino la Potencia Real, las ecuaciones pueden ser expresadas en términos de ésta:

$$S_p^* = P_p - jQ_p = y_{pp} V_p V_p^* + y_{pq} V_q V_p^* \quad 1.9$$

$$S_q^* = P_q - jQ_q = y_{qp} V_p V_q^* + y_{qq} V_q V_q^*$$

1.1.2 Generalización de las ecuaciones de flujos de carga. Considere una red eléctrica representada por la Figura 2.2, donde el bus p esta conectado a otros k buses,

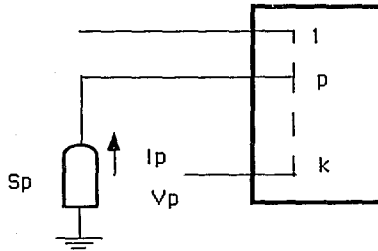


Figura 1.2

de la ecuacion 1.3 tenemos:

$$I_p = \frac{S_p^*}{V_p^*} = y_{p1}V_1 + y_{p2}V_2 + \dots + y_{pp}V_p + \dots + y_{pn}V_n, \quad i = 1, 2, \dots, n$$

donde:

$$y_{pp} = \sum_{\substack{k=1 \\ k \neq p}}^n (y_{pk} + y_{S_{pk}}) \quad 1.9$$

$$y_{pk} = y_{kp} = -y_{S_{pk}}$$

Las matrices quedan:

$$I_{bus} = \begin{bmatrix} I_1 \\ \vdots \\ I_n \end{bmatrix} \quad V_{bus} = \begin{bmatrix} V_1 \\ \vdots \\ V_n \end{bmatrix}$$

$$Y_{bus} = \begin{bmatrix} y_{11} & \cdots & y_{1n} \\ \vdots & \cdots & \vdots \\ y_{n1} & \cdots & y_{nn} \end{bmatrix}$$

Por lo que

$$S_p^* = P_p - jQ_p = V_p^* \sum_{k=1}^n y_{pk} V_k, \quad p=1,2,\dots,n \quad 1.10$$

1.2 CRITERIOS DE CONVERGENCIA.

En un Sistema Eléctrico de potencia se consideran tres tipos de buses:

- a.- Bus de referencia o "Slack". Se identifica como bus V-0. En este bus se conoce el valor de la magnitud del Voltaje y el ángulo, pero se desconoce la potencia real y la potencia reactiva las cuales son calculadas al final del proceso. Debido a que este bus representa el Fasor de referencia, su ángulo es igual a cero como se observa en la siguiente figura.

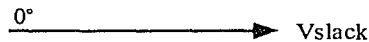


Figura 1.3

b.-Bus de Control de Voltaje (P-V). En el se especifica la potencia real y la magnitud de voltaje se mantiene constante por medio de alimentación de potencia reactiva.

c.-Bus de Carga (P-Q). En este bus se conoce la potencia compleja que se consume y se calcularán los valores de magnitud de voltaje y ángulo.

1.2.1 Desequilibrio de Potencia

El incremento de potencia en un bus esta dado por:

$$\Delta S_p = S_p - V_p I_p^* = P_p + jQ_p - V_p \sum_{q=1}^n Y_{pq}^* V_q^* \quad 1.11$$

considerando que:

$$V_p = e_p + jf_p = |V_p| \theta_p$$

$$Y_{pq} = G_{pq} + jB_{pq} \quad 1.12$$

$$\theta_{pq} = \theta_p - \theta_q$$

Para cada bus P-Q ó P-V tenemos,

En forma rectangular:

$$\Delta P_p = P_p - \operatorname{Re} \left[(e_p + jf_p) \sum_{q=1}^n (G_{pq} - jB_{pq})(e_q - jf_q) \right] \quad 1.13$$

En forma polar

$$\Delta P_p = P_p - |V_p| \left[\sum_{q=1}^n (G_{pq} \cos \theta_{pq} + B_{pq} \sin \theta_{pq}) |V_q| \right] \quad 1.14$$

Para cada bus P-Q tenemos

En forma rectangular:

$$\Delta Q_p = Q_p - \text{Im} \left[(e_p + jf_p) \sum_{q=1}^n (G_{pq} - jB_{pq})(e_q - jf_q) \right] \quad 1.15$$

En forma polar:

$$\Delta Q_p = Q_p - |V_p| \left[\sum_{q=1}^n (G_{pq} \sin \theta_{pq} - B_{pq} \cos \theta_{pq}) |V_q| \right] \quad 1.16$$

El criterio de convergencia que considera el desbalance de potencia en los buses se basa en las siguientes desigualdades:

$$\Delta P_p \leq C_p \quad \forall \text{ buses P-Q y P-V}$$

$$\Delta Q_p \leq C_q \quad \forall \text{ buses P-Q}$$

Los valores C_p y C_q se escogen en el rango de 0.01 a 10 MW ó MVAR.

1.2.2 Cambio en la magnitud de Voltaje.

El cambio de la magnitud de voltaje entre las iteraciones k y $k+1$ esta dado por:

$$\Delta |V_p| = |V_p^{(k+1)}| - |V_p^{(k)}| < C_v \quad \forall \text{ bus P-Q}$$

El valor típico de C_v varía entre 0.01 y 0.0001

1.3 METODO DE NEWTON-RAPHSON (N-R).

1.3.1 El caso escalar. Para una función escalar cuando se encuentra una solución en un punto dado, hay un error asociado de tal forma, que la solución correcta esta dada por la expresion siguiente:

$$f(x(0)) + \Delta x(0) = 0$$

donde $f(x(0))$ es la función evaluada en el punto de solución $x(0)$ y $\Delta x(0)$ es el error asociado con la solución de la función.

Si expandimos la función en series de taylor tenemos:

$$f(x(0)) + \Delta x(0) + f'(x(0))\Delta x(0) + \frac{1}{2}(\Delta x(0))^2 f''(x(0)) + \dots = 0$$

Si se considera que el error es muy pequeño, los terminos de orden mayor se pueden despreciar quedando la expresión anterior de la siguiente manera:

$$f(x(0)) + \Delta x(0)f'(x(0)) = 0$$

Por lo que el error será:

$$\Delta x(0) = - \frac{f(x(0))}{f'(x(0))} \quad 1.17$$

Si le sumamos este error a la solución original obtenemos un valor mejorado de las variables relacionadas con la solución:

$$x(1) = x(0) + \Delta x(0) = x(0) - \frac{f(x(0))}{f'(x(0))}$$

El uso cíclico o repetitivo de este procedimiento nos lleva al algoritmo de Newton-Raphson:

$$x^{(v+1)} = x^{(v)} - \frac{f(x^{(v)})}{f'(x^{(v)})} \quad 1.18$$

1.3.2 El caso multidimensional. Aplicando el caso escalar a la solución de una matriz de ecuaciones de dimensión n tenemos,

$$x^{(0)} = \begin{bmatrix} x_1^{(0)} \\ \vdots \\ x_n^{(0)} \end{bmatrix} \quad 1.19$$

$$\begin{aligned} f_1(x^{(0)}) + \left(\frac{\partial f_1}{\partial x_1}\right)^{(0)} \Delta x_1 + \dots + \left(\frac{\partial f_1}{\partial x_n}\right)^{(0)} \Delta x_n &= 0 \\ \vdots \\ f_n(x^{(0)}) + \left(\frac{\partial f_n}{\partial x_1}\right)^{(0)} \Delta x_1 + \dots + \left(\frac{\partial f_n}{\partial x_n}\right)^{(0)} \Delta x_n &= 0 \end{aligned}$$

En forma matricial el sistema de n funciones se puede expresar como:

$$\begin{bmatrix} f_1(x^{(0)}) \\ \vdots \\ f_n(x^{(0)}) \end{bmatrix} + \begin{bmatrix} \left(\frac{\partial f_1}{\partial x_1}\right)^{(0)} & \dots & \left(\frac{\partial f_1}{\partial x_n}\right)^{(0)} \\ \vdots & & \vdots \\ \left(\frac{\partial f_n}{\partial x_1}\right)^{(0)} & \dots & \left(\frac{\partial f_n}{\partial x_n}\right)^{(0)} \end{bmatrix} \begin{bmatrix} \Delta x_1 \\ \vdots \\ \Delta x_n \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}$$

o en notación compacta:

$$f(x(0)) + J(0) \Delta x(0) = 0$$

o

$$\Delta x(0) = -[J(0)]^{-1} f(x(0)) \quad 1.20$$

de acuerdo a la ecuación 1.18 tenemos:

$$x(v+1) = x(v) - [J(v)]^{-1} f(x(v)) \quad 1.20.a$$

La matriz J es cuadrada de dimensión nxn y se denomina Jacobiano.

1.3.3 Solución de flujos de carga aplicando las ecuaciones de N-R. De la ecuación de una red eléctrica:

$$I_{bus} = Y_{bus} V_{bus}$$

en forma expandida para un nodo tenemos,

$$I_p = \sum_{q=1}^n Y_{pq} V_q, \quad p=1,2,\dots,n$$

en forma conjugada,

$$V_p I_p^* = S_p = V_p \sum_{q=1}^n Y_{pq} V_q = P_q - jQ_p$$

sabemos que ,

$$\begin{aligned} Y_{pq} &= G_{pq} + jB_{pq} = |Y_{pq}| \angle \gamma_{pq} \\ V_p &= e_p + jf_p = |V_p| \angle \theta_p \\ V_q &= e_q + jf_q = |V_q| \angle \theta_q \end{aligned}$$

separando la potencia aparente en sus componentes real e imaginario se obtiene,

$$\begin{aligned} S_p^* &= P_p - jQ_p = \sum_{q=1}^n |Y_{pq}| |V_p| |V_q| e^{j(\theta_q - \theta_p + \gamma_{pq})} \\ P_p &= \sum_{q=1}^n |Y_{pq}| |V_p| |V_q| \cos(\theta_q - \theta_p + \gamma_{pq}) \end{aligned} \quad 1.21$$

$$P_p = |V_p| \sum_{q=1}^n ((G_{pq} \cos \theta_{pq} + B_{pq} \sin \theta_{pq}) |V_q|)$$

$$Q_p = - \sum_{q=1}^n |Y_{pq}| |V_p| |V_q| \sin(\theta_q - \theta_p + \gamma_{pq}) \quad 1.22$$

$$Q_p = |V_p| \sum_{q=1}^n ((G_{pq} \sin \theta_{pq} - B_{pq} \cos \theta_{pq}) |V_q|)$$

Del número total de buses n, existen n1 buses P-Q, n2 buses P-V y un bus de referencia, por lo que el número total de buses es $n = n1 + n2 + 1$.

Las variables que se buscan son las n_1 magnitudes de voltaje en los buses P-Q y $(n_1 + n_2)$ ángulos en los buses P-V y P-Q. De acuerdo a esto nuestras variables planteadas en N-R son:

$$\mathbf{x} = \begin{bmatrix} \theta_2^{(0)} \\ \theta_3^{(0)} \\ \vdots \\ \vdots \\ |V_{n-1}^{(0)}| \\ V_n^{(0)} \end{bmatrix}$$

Expandiendo en series de Taylor tenemos,

$$P_i = f_{ip}^{(0)} + \left(\frac{\partial f_{ip}}{\partial \theta_2} \right)^{(0)} \Delta \theta_2^{(0)} + \dots + \left(\frac{\partial f_{ip}}{\partial |V_n|} \right)^{(0)} \Delta |V_n^{(0)}|, \quad p=1,2,\dots,n$$

1.23

$$Q_i = f_{iq}^{(0)} + \left(\frac{\partial f_{iq}}{\partial \theta_2} \right)^{(0)} \Delta \theta_2^{(0)} + \dots + \left(\frac{\partial f_{iq}}{\partial |V_n|} \right)^{(0)} \Delta |V_n^{(0)}|$$

donde $f_{ip}^{(0)}$ y $f_{iq}^{(0)}$ representan la potencia real y reactiva respectivamente, en el bus i .

El incremento de potencia es,

$$\Delta P_i^{(0)} = P_i - f_{ip}^{(0)}$$

$$\Delta Q_i^{(0)} = Q_i - f_{iq}^{(0)}$$

1.24

...

Las ecuaciones 1.23 pueden ser representadas en forma matricial de la siguiente forma:

$$\begin{bmatrix} \Delta P_2^{(0)} \\ \vdots \\ \Delta Q_n^{(0)} \end{bmatrix} = \begin{bmatrix} \left(\frac{\partial f_{2p}}{\partial \theta_2}\right)^{(0)} & \dots & \left(\frac{\partial f_{2p}}{\partial |V_n|}\right)^{(0)} \\ \vdots & & \vdots \\ \left(\frac{\partial f_{nq}}{\partial \theta_2}\right)^{(0)} & \dots & \left(\frac{\partial f_{nq}}{\partial |V_n|}\right)^{(0)} \end{bmatrix} \begin{bmatrix} \Delta \theta_2^{(0)} \\ \vdots \\ \Delta |V_n|^{(0)} \end{bmatrix} \quad 1.25$$

donde,

$$H = \frac{\partial P_p}{\partial \theta_p}; M = \frac{\partial P_p}{\partial V_p}; N = \frac{\partial Q_p}{\partial \theta_p}; L = \frac{\partial Q_p}{\partial V_p}$$

Por lo que la matriz anterior, expresada en terminos del Jacobiano queda,

$$\begin{bmatrix} \Delta P \\ \Delta Q \end{bmatrix} = \begin{bmatrix} \frac{\partial f_p}{\partial \theta} & \frac{\partial f_p}{\partial |V|} \\ \frac{\partial f_q}{\partial \theta} & \frac{\partial f_q}{\partial |V|} \end{bmatrix} \begin{bmatrix} \Delta \theta \\ \frac{\Delta |V|}{|V|} \end{bmatrix} = \mathbf{J} \begin{bmatrix} \Delta \theta \\ \frac{\Delta |V|}{|V|} \end{bmatrix} \quad 1.26$$

$$\begin{bmatrix} \Delta \theta \\ \frac{\Delta |V|}{|V|} \end{bmatrix} = \mathbf{J}^{-1} \begin{bmatrix} \Delta P \\ \Delta Q \end{bmatrix} \quad 1.27$$

El jacobiano lo podemos dividir en las siguientes cuatro submatrices:

$$\mathbf{J} = \begin{bmatrix} \mathbf{H} & \mathbf{N} \\ \mathbf{M} & \mathbf{L} \end{bmatrix} \quad 1.28$$

donde:

$$P = f(\theta)$$

$$Q = f(V)$$

A continuación se obtienen los valores de las submatrices:

Primero recordemos las siguientes expresiones,

$$S_p^* = P_p - jQ_p = V_p^* \sum_{k=1}^n y_{pk} V_k, p=1,2,\dots,n$$

$$\Delta P_p = P_p - |V_p| \left[\sum_{q=1}^n (G_{pq} \cos \theta_{pq} + B_{pq} \sin \theta_{pq}) |V_q| \right]$$

$$\Delta Q_p = Q_p - |V_p| \left[\sum_{q=1}^n (G_{pq} \sin \theta_{pq} - B_{pq} \cos \theta_{pq}) |V_q| \right]$$

Para $p = q$,

$$\begin{aligned} \frac{\partial \Delta P_p}{\partial \theta_p} = H_{pp} &= \frac{\partial \left\{ P_p - |V_p| \sum_{q=1}^n (G_{pq} \cos (\theta_p - \theta_q) + B_{pq} \sin (\theta_p - \theta_q)) \right\}}{\partial \theta_p} \\ &= -|V_p| \sum_{q=1}^n (G_{pq} \sin \theta_{pq} - B_{pq} \cos \theta_{pq}) |V_q| - B_{pp} |V_p|^2 \\ &= -Q_p - B_{pp} |V_p|^2 \end{aligned}$$

$$\begin{aligned}\frac{\partial \Delta P_p}{\partial V_p} &= N_{pp} = |V_p| \sum_{q=1}^n (G_{pq} \cos \theta_{pq} + B_{pq} \sin \theta_{pq}) |V_q| + G_{pp} |V_p|^2 \\ &= P_p + G_{pp} |V_p|^2\end{aligned}$$

$$\begin{aligned}\frac{\partial \Delta Q_p}{\partial \theta_p} &= M_{pp} = |V_p| \sum_{q=1}^n (G_{pq} \cos \theta_{pq} + B_{pq} \sin \theta_{pq}) |V_q| - G_{pp} |V_p|^2 \\ &= P_p - G_{pp} |V_p|^2\end{aligned}$$

$$\begin{aligned}\frac{\partial \Delta Q_p}{\partial V_p} &= L_{pp} = |V_p| \sum_{q=1}^n (G_{pq} \sin \theta_{pq} - B_{pq} \cos \theta_{pq}) |V_q| - B_{pp} |V_p|^2 \\ &= Q_p - B_{pp} |V_p|^2\end{aligned}$$

Para $p \neq q$,

$$\frac{\partial \Delta P_p}{\partial \theta_q} = H_{pq} = |V_p| |V_q| (G_{pq} \sin \theta_{pq} - B_{pq} \cos \theta_{pq})$$

$$\frac{\partial \Delta P_p}{\partial |V_q|} = N_{pq} = |V_p| |V_q| (G_{pq} \cos \theta_{pq} + B_{pq} \sin \theta_{pq})$$

$$\frac{\partial \Delta Q_p}{\partial \theta_q} = M_{pq} = -|V_p| |V_q| (G_{pq} \cos \theta_{pq} + B_{pq} \sin \theta_{pq}) = -N_{pq}$$

$$\frac{\partial \Delta Q_p}{\partial |V_q|} = L_{pq} = -|V_p| |V_q| (G_{pq} \sin \theta_{pq} - B_{pq} \cos \theta_{pq}) = -H_{pq}$$

De la ecuación 1.28 $\Delta\theta$ es el subvector de incrementos de ángulos en los buses P-Q y P-V. Las variables correspondientes al incremento de la magnitud de voltaje en los buses P-Q son divididos por la magnitud de voltaje para lograr una simetría en los elementos de la matriz de coeficientes.

El problema se puede resumir en dos variables X, Y, donde X es el número de variables desconocidas y Y el número de variables conocidas, pudiéndose plantear lo siguiente:

$$x = \begin{bmatrix} |V| \text{ en cada bus P-Q} \\ \theta \\ \theta \text{ en cada bus P-V} \end{bmatrix} \quad y = \begin{bmatrix} V_s \\ \theta_s \text{ en el bus de referencia} \\ P_p \\ Q_p \text{ en cada bus P-Q} \\ P_p \\ |V_p| \text{ en cada bus P-V} \end{bmatrix}$$

De las ecuaciones 1.21 y 1.22 expresadas en forma polar, seleccionamos un número de ecuaciones igual al número de variables en X,

$$F(x) = \begin{bmatrix} \text{Ecuación 1.21 para cada bus P-Q y P-V} \\ \text{Ecuación 1.22 para cada bus P-Q} \end{bmatrix} = 0$$

Tendremos entonces $2n_1 + n_2$ ecuaciones .

Todas las cantidades en la ecuación lineal 1.28 corresponden a la iteración k. Después de encontrados los vectores de solución para $\Delta\theta$ y $\frac{\Delta|V|}{|V|}$, se aplica la corrección,

$$|V|^{(k+1)} = |V|^{(k)} + \Delta|V|^{(k)}$$

1.29

$$\theta^{(k+1)} = \theta^{(k)} + \Delta\theta^{(k)}$$

El proceso se repite para los valores corregidos, hasta alcanzar el punto de convergencia deseado.

1.4 METODO DESACOPLADO.

Este método se basa en el de Newton-Raphson con pequeñas modificaciones hechas sobre la ecuación lineal 1.28. Para esto se toma en cuenta:

-Que existe una estrecha relación entre la potencia real de un bus y su ángulo de Voltaje.

-Que existe una estrecha relación entre la potencia reactiva de un bus y su magnitud de Voltaje.

Esto lleva a que se pueden despreciar las submatrices N y M, quedando la solución planteada en términos de dos conjuntos de ecuaciones lineales:

$$[\Delta P] = [H] [\Delta \theta] \quad 1.30$$

$$[\Delta Q] = [L] \left[\frac{\Delta |V|}{|V|} \right] \quad 1.31$$

Para $p \neq q$

$$H_{pq} = L_{pq} = |V_p| |V_q| (G_{pq} \sin \theta_{pq} - B_{pq} \cos \theta_{pq}) \quad 1.32$$

Para $p = q$,

$$H_{pp} = -B_{pp} |V_p|^2 - Q_p \quad 1.33$$

$$L_{pp} = -B_{pp} |V_p|^2 + Q_p$$

1.5 METODO DESACOPLADO RAPIDO.

Si consideramos que:

$$\cos \theta_{pq} \cong 1, \quad G_{pq} \sin \theta_{pq} \ll B_{pq}, \quad Q_p \ll B_{pp} |V_p|^2$$

y ajustamos $V_p = V_q = 1$ p.u., las ecuaciones 1.32 y 1.33 quedan:

$$H_{pq} = |V_p| |V_q| (0 - B_{pq} \cos \theta_{pq}) \quad 1.34$$

$$H_{pq} = \frac{\partial \Delta P}{\partial \theta_q} = -B_{pq}$$

$$H_{pp} = \frac{\partial \Delta P}{\partial \theta_p} = -|V_p| \sum_{q=1}^n (0 - B_{pq}) |V_q| - 0$$

$$H_{pp} = - \sum_{q=1}^n -B_{pq} = \sum_{q=1}^n B_{pq}, \quad p \neq q$$

$$L_{pq} = \frac{\partial \Delta Q}{\partial |V_q|} = -|V_p| |V_q| (G_{pq} \sin \theta_{pq} + B_{pq} \cos \theta_{pq})$$

$$L_{pq} = -B_{pq}$$

$$L_{pp} = \frac{\partial \Delta Q}{\partial |V_p|} = |V_p| \sum_{q=1}^n (0 + B_{pq} * 1) |V_q| - 0 = Q_p - B_{pp} |V_p|^2$$

$$L_{pp} = -B_{pp}$$

Las submatrices H y L son cuadradas de dimensión $n_1 + n_2$ y n_1 respectivamente.

Las ecuaciones de flujo quedan de la siguiente forma:

$$\Delta P = B' \Delta \theta$$

$$\Delta Q = B'' \Delta V \quad 1.35$$

Estas ecuaciones también se pueden escribir de la siguiente manera:

$$\frac{\Delta P}{\Delta \theta} = B', \quad \frac{\Delta Q}{\Delta V} = B''$$

Comparandolas con,

$$\frac{\partial P_p}{\partial \theta_q} = -B_{pq} \quad \frac{\partial Q_p}{\partial V_q} = -B_{pq}$$

$$\frac{\partial P_p}{\partial \theta_p} = \sum_{q=1}^n B_{pq} \quad \frac{\partial Q_p}{\partial V_p} = -B_{pp}$$

Se puede concluir que:

$$B'_{pq} = -B_{pq} \quad B''_{pq} = -B_{pq}$$

$$B'_{pp} = \sum_{q=1}^n B_{pq} \quad B''_{pp} = -B_{pp}$$

donde B_{pq} y B_{pp} son elementos de la matriz $[-B]$, o sea la parte imaginaria, negada, de la matriz Y_{bus} .

Finalmente se hacen las siguientes consideraciones:

- a.- En la matriz $[B']$ se omitió la representación de todos los elementos de la red que tienen efecto sobre el flujo de potencia reactiva, tales como reactancias en derivación, transformadores reguladores o derivaciones en los transformadores que producen voltajes no nominales.
- b.- De la matriz $[B'']$ se omitió el efecto de corrimiento de fase o cambio de ángulo de voltaje que provocan los "Phase Shifter".
- c.- En el cálculo de la matriz $[B']$ se desprecian las resistencias serie de la línea.

Las matrices $[B']$ y $[B'']$ son reales y dispersas y tienen la misma estructura de las matrices $[H]$ y $[L]$ respectivamente. Debido a que contienen solamente las admitancias de la red, son constantes y necesitan ser calculadas una sola vez al comienzo del estudio. También se caracterizan por ser simétricas si no se presentan "Phase Shifter".

1.6 DIFERENCIAS ENTRE LOS METODOS.

Como se explicó anteriormente, las tendencias de los métodos de flujos de potencia se traducen en rapidez de proceso, simplicidad y exactitud.

El método de Newton-Raphson tiene una convergencia cuadrática, es decir que el error en una iteración disminuye en proporción al cuadrado del error de la siguiente. Esto se logra bien cuando los valores iniciales o de arranque proporcionados, se aproximan un poco a la solución. El tiempo de solución puede disminuirse cuando la matriz Jacobiana se mantiene constante cada dos iteraciones.

Para redes grandes este método requiere de técnicas especiales de programación tales como el uso de matrices dispersas y técnicas de eliminación ordenada. Con esto se alcanza mucha eficiencia con respecto a otros métodos, si la red tiende a ser muy grande.

El método desacoplado converge de igual manera que el Newton-Raphson. Su ventaja está en que la matriz que usa corresponde a la mitad del orden del Jacobiano, por lo que su necesidad en memoria disminuye también a la mitad. Un sistema en estudio de la IEEE de más de 1080 buses y considerando un criterio de convergencia de 0.01 MW/MVAR, mostró una solución en siete iteraciones y con una velocidad quince veces menor a los resultados obtenidos, para el mismo sistema, con el método de Newton-Raphson.

El método desacoplado rápido converge de igual forma que los dos anteriores , pero tiene la ventaja de que el tiempo de cálculo se disminuye aún más por las simplificaciones hechas.

2. ALGORITMO PROPUESTO.

De todos los algoritmos estudiados en el capítulo anterior, es el método desacoplado rápido el que tiene mas uso debido al poco tiempo requerido en el cálculo y a que es muy exacto para la mayoría de las configuraciones de redes eléctricas. Sin embargo, tiene problemas de convergencia en sistemas con ramales que tienen altos cocientes R/X . Algunas modificaciones al método permiten solucionar estas dificultades.

Las propuestas de Dyliaço y Ramaraο de usar compensación serie y la de Dackman de usar compensación paralela, proporcionaron ciertas mejoras en la convergencia aunque con ciertas inconsistencias, es decir para algunos casos las mejoras funcionaban y para otros no.

El algoritmo que es motivo de este trabajo, proporciona consistencia en la convergencia para Sistemas Elécticos de Potencia con altos cocientes R/X y funciona mejor que los métodos de compensación. Este método también prueba que la rapidez en la convergencia para redes con cocientes R/X dentro de los rangos normales, es solamente un poco menor que el método desacoplado rápido.

Existen otros métodos para manipular redes con altos cocientes R/X , tales como el “Modelo Híbrido para métodos de flujos de carga” y “Flujo de Carga super desacoplado con Bus Slack Distribuido”. Estos métodos no se basan en el Desacoplado rápido por lo que no corresponden a este estudio y sería objeto de otro trabajo poder comparar sus

resultados con el algoritmo propuesto. Es por esto que las comparaciones hecha aquí son relacionadas con el método desacoplado rápido.

2.1 MODIFICACIONES AL METODO DESACOPLADO RAPIDO.

Sabemos que las ecuaciones de potencia en un bus de una red son:

$$P_p = |V_p| \sum_{q=1}^n ((G_{pq} \cos \theta_{pq} + B_{pq} \sin \theta_{pq}) |V_q|) \quad 2.1$$

$$Q_p = |V_p| \sum_{q=1}^n ((G_{pq} \sin \theta_{pq} + B_{pq} \cos \theta_{pq}) |V_q|) \quad 2.2$$

Del método desacoplado rápido sabemos que las ecuaciones de flujo resultantes son:

$$\Delta P = B' \Delta \theta \quad 2.3$$

$$\Delta Q = B'' \Delta V \quad 2.4$$

donde,

$$B'_{pq} = -B_{pq} \quad 2.5$$

$$B'_{pp} = \sum_{q=1}^n B_{pq}$$

$$B''_{pq} = -B_{pq} \quad 2.6$$

$$B''_{pp} = -B_{pq}$$

donde B_{pq} son los elementos de la matriz de susceptancia $[B]$.

Las pruebas hechas al método desacoplado rápido indican que cuando se usan las ecuaciones 2.5, el sistema converge lentamente. Esto se corrige usando la siguiente aproximación:

$$B'_{pq} = -\frac{1}{x_{pq}} = -B_{pq} - \frac{G_{pq}^2}{B_{pq}}$$

$$B''_{pp} = \sum_{q=1}^n B'_{pq}$$

2.7

La figura 2.1 nos muestra la representación de estas aproximaciones

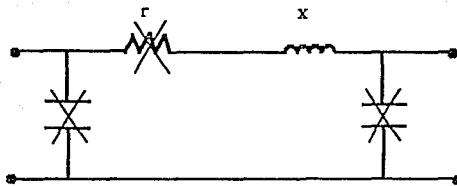


Figura 2.1.a

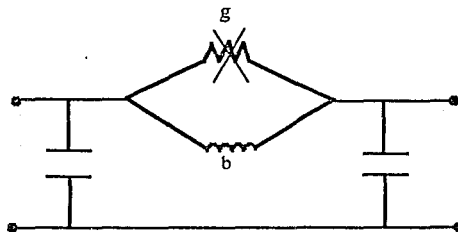


Figura 2.1.b

Estos esquemas representan los elementos que se desprecian en el método desacoplado rápido cuando se usan las aproximaciones: B' , mostrada en la Figura 2.1.a y B'' , mostrada en la Figura 2.1.b.

La siguiente tabla muestra la diferencia en la convergencia al usar las dos aproximaciones de B' (ecuaciones 2.5 y 2.7), utilizando modelos de red de la IEEE.

Número de buses	Cociente R/X mas alto	Número de iteraciones usando 0.01 MW/MVAR de tolerancia	
		B' original (2.4)	B' modificada (2.5)
5	0.333	4.5	4.0
14	1.105	19.5	3.5
30	1.107	17.0	3.5
57	1.089	64.9	4.5
118	0.640	5.5	5.5

Como se puede observar, la diferencia en el número de iteraciones se hace más palpable cuando aumenta el cociente R/X , pero para bajos cocientes la diferencia es mínima.

Según lo anterior es conveniente usar la aproximación 2.7 como estandar para la solución de flujos de carga con el sistema desacoplado rápido.

Pruebas hechas con cocientes R/X mayores a 1 indican que aún la aproximación de B' , no produce una buena convergencia del sistema. Esto se soluciona con la modificación que se presenta a continuación.

2.2 MODIFICACION PROPUESTA PARA SER APLICADA AL METODO DESACOPLADO RAPIDO.

Para la mayoría de las redes de transmisión de Energía Eléctrica el cociente R/X es pequeño, por consiguiente la consideración de $G_{pq} \ll B_{pq}$ es correcta. Sin embargo, en la práctica se ha encontrado que el método desacoplado rápido converge muy bien cuando G_{pq} no es excedido significativamente por B_{pq} . De esta forma las consideraciones hechas en el método desacoplado rápido no afectan la convergencia a menos que sean violadas por un alto margen, en cuyo caso se debe incluir el efecto de la conductancia (G_{pq}) en la iteraciones para mejorar la convergencia.

Una forma de incluir la conductancia se deriva de sumar las ecuaciones de inyección de potencia a un bus:

$$\frac{P_p}{V_p} = V_p G_{pp} + \sum_{q=1}^n V_q [G_{pq} \cos \theta_{pq} + B_{pq} \sin \theta_{pq}] \quad 2.6$$

+

$$\frac{Q_p}{V_p} = -V_p B_{pp} + \sum_{q=1}^n V_q [G_{pq} \sin \theta_{pq} - B_{pq} \cos \theta_{pq}] \quad 2.7$$

$$\frac{(P_p + Q_p)}{|V_p|} = (G_{pp} - B_{pp})|V_p| + \sum_{q=1}^n |V_q| [(G_{pq} - B_{pq}) \cos \theta_{pq} + (G_{pq} + B_{pq}) \sin \theta_{pq}] \quad 2.8$$

La ecuación 2.4 puede ser reemplazada por:

$$\Delta P + \Delta Q = B''' \Delta V \quad 2.9$$

donde B''' puede ser aproximada por las expresiones:

$$B'''_{pq} = G_{pq} - B_{pq} \quad 2.10$$

$$B'''_{pp} = G_{pp} - B_{pp}$$

La ecuación 2.3 no se altera pero el efecto de valores grandes de G_{pq} se reflejan en una nueva aproximación de la matriz B' .

Como se indicó anteriormente la aproximación señalada en la ecuación 2.7 mejora la convergencia del método, sin embargo en los casos donde G_{pq} es un valor muy grande el cuadrado de éste termino puede exagerar su efecto. Una expresión alternativa que modera este efecto se muestra a continuación:

$$B'_{pq} = - B_{pq} - 0.4 G_{pq} - 0.3 \frac{G_{pq}^2}{B_{pq}} \quad 2.11$$
$$B_{pp} = - \sum_{q=1}^n B'_{pq}$$

Los coeficientes 0.4 y 0.3 se encontraron experimentalmente.

3. DESCRIPCION DEL ALGORITMO

El algoritmo que se estudia en este trabajo corresponde exactamente a las etapas por las que pasa el estudio de flujos de carga, es decir lectura de datos de las líneas de la red, lectura de datos de los buses, cálculo de la matriz Ybus, algoritmo de solución de ecuaciones lineales múltiples para obtener los valores de voltaje y de ángulos en los buses y por último la aplicación de los conceptos de Convergencia para obtener los resultados finales.

En este trabajo se hace una comparación entre el método desacoplado rápido y el propuesto para poder tener una referencia de su comportamiento en redes eléctricas con y sin altos cocientes R/X . Se tomó como método de comparación el método desacoplado rápido debido a que es el más usado ya que su convergencia es rápida, su exactitud en los resultados obtenidos es muy confiable y también a que el algoritmo es menos difícil de codificar que los otros.

El objetivo del trabajo es demostrar que el algoritmo en estudio converge en forma más rápida que el Desacoplado Rápido para redes eléctricas con altos cocientes R/X y que se comporta similarmente a éste en redes eléctricas con cocientes R/X pequeños. Debido a que en general las líneas de transmisión en Sistemas eléctricos de Potencia presentan cocientes R/X bajos -para los circuitos de distribución estos cocientes son mas altos-, el algoritmo propuesto utiliza modelos de redes con bajos cocientes R/X y se va repitiendo el proceso para la misma red, multiplicando las resistencias de todas las líneas por un factor de 0.5, 1.0, 1.5, 2.0, 2.5, 3.0, 3.5, 4.0, 4.5 y 5.0. Esto permite alcanzar, para los factores de multiplicación altos, un Sistema Eléctrico de Potencia con cociente R/X próximo a un modelo de red que nos permita hacer el estudio requerido. Consideraremos que un cociente R/X es alto, cuando sea mayor o igual a 0.9.

Al final, el programa de computadora nos entrega un cuadro comparativo del número de iteraciones contra el factor de multiplicación, indicando cual es el cociente R/X más alto en toda la red, para los dos métodos en comparación. De esta forma se podrán sacar las conclusiones correspondientes.

Como se explicó en el capítulo anterior, los dos métodos siguen los mismos pasos, la diferencia radica en que cuando se calculan los valores de la matriz Y_{bus} , se calculan también los valores de las matrices B' y B'' para el método Desacoplado Rápido y B' y B''' para el algoritmo propuesto, las cuales incluyen las correcciones correspondientes al cálculo de sus elementos.

El programa se codificó en Lenguaje C, tomando en cuenta que los sistemas de cómputo tienden cada vez en forma clara, hacia una estandarización que será regida por ciertas normas que implementarán los fabricantes de equipos y los diseñadores de software. En este sentido el lenguaje seleccionado es el que cada vez se usa más con estos criterios de normalización y además contiene ciertas características que permiten, con mayores ventajas, el mismo manejo que se hacía antes con el lenguaje tradicional, Fortran.

3.1 PASOS DEL ALGORITMO.

El algoritmo completo del método propuesto se describe a continuación:

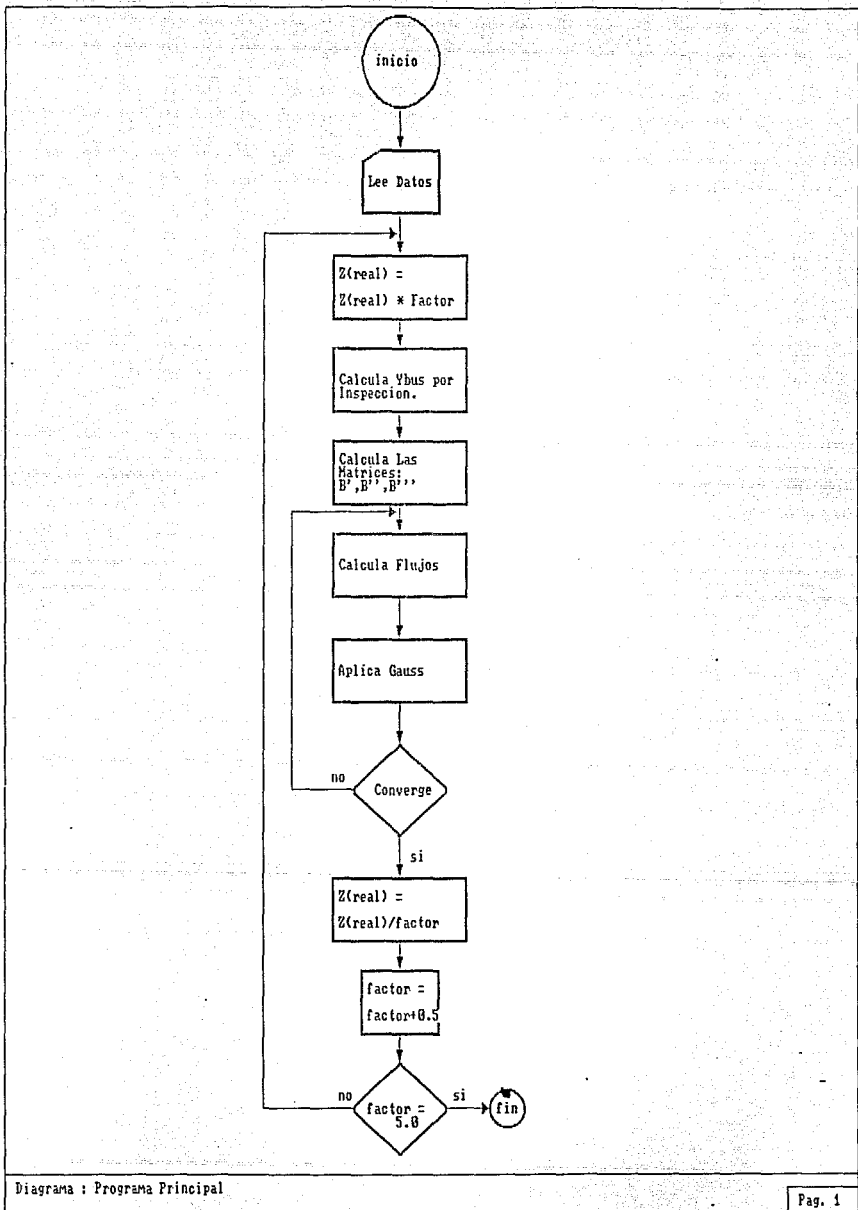
1. Se leen los datos de la RED.
2. Se escriben o imprimen los datos de las líneas y de los buses del sistema.
3. Se multiplican las resistencias de cada línea por el factor " f ".
4. Se imprimen los datos de las líneas y los buses con los valores resultantes de la multiplicación de las resistencias de las líneas por el factor " f ".
5. Se calculan la matrices Y_{bus} , B' , B'' y B''' .
6. Se calculan los incrementos de Potencia Real y Reactiva aplicando la función Flujos.
7. Se construye el sistema de ecuaciones lineales y se le aplica la solución de eliminación de Gauss.

8. Si el sistema converge para el método desacoplado rápido, considerando que los nuevos valores de voltaje y ángulo deben ser menores a un criterio de convergencia ϵ o que el número de iteraciones sea mayor que 100, repetimos todo el proceso desde el punto 3 pero para las matrices B' y B'' , es decir para el método propuesto. Cuando este también converge o diverge, continuamos en el punto siguiente, de lo contrario regresamos al punto 6 con los nuevos valores obtenidos en la eliminación de Gauss.

9. Se reinician todas las variables a los datos originales de la RED y se incrementa $f = f + 0.5$.

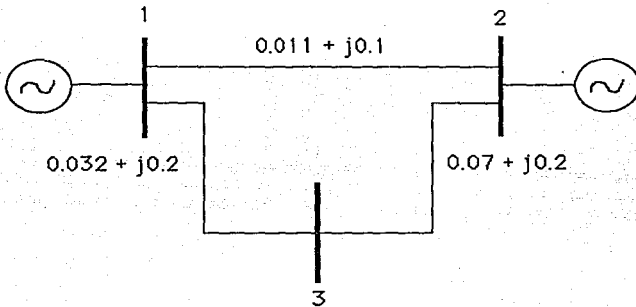
10. Si el factor f es menor que 5, regresamos al punto 3, de lo contrario se imprime la tabla de resultados finales donde se comparan el número de iteraciones contra el factor de multiplicación para los dos métodos utilizados.

El siguiente diagrama de flujo explica todos los pasos o rutinas generales que componen el algoritmo propuesto, posteriormente se explicarán cada una de ellas.



3.2 DESCRIPCION DE LAS RUTINAS QUE CONFORMAN EL PROGRAMA.

3.2.1 Lectura de Datos. Para explicar como se deben introducir los datos para que el programa los interprete y manipule correctamente, me basaré en un ejemplo sencillo que consta del diagrama Unifilar de la red, y las tablas de datos de sus líneas y buses:



Línea No.	Entre buses	Impedancia de la línea
1	1-2	$0.011 + 0.1$
2	2-3	$0.070 + 0.2$
3	1-3	$0.032 + 0.2$

Bus No.	Tipo	Generador		Carga		Magnitud de Voltaje
		P	Q	P	Q	
1	Slack					1.0
2	P-V	9.3217				1.1
3	P-Q			0.036392	0.5339	

Para cargar el programa hay que teclear la RX con el siguiente número de parámetros, en el orden señalado:

1. Factor mínimo, el cual es el número donde comienza el factor de multiplicación "f".
2. Factor máximo, el cual es el número hasta el cual se aplica el factor de multiplicación "f".
3. El criterio de convergencia cv. Se recomienda que sea 0.01.

Al cargar el sistema, se pregunta por el archivo desde donde se leerán los datos. Este archivo además de los datos de las líneas y de los buses, debe contener el número de buses de control de voltaje, el número de buses de carga y el número de líneas de conducción eléctrica. Se asume siempre que el primer bus es el de control de voltaje o Slack.

El Formato para el archivo de datos es el siguiente:

```
n
n1
n2
no_lin;
a[i] b[j] R.x[i] R.y[i]
bus TIPO[bus] Sg.x[bus] Sg.y[bus] Sc.x[bus] Sc.y[bus] VOLTAJE.x[bus]
VOLTAJE.y[bus]
```

donde:

```
n      = Numero de buses del Sistema Eléctrico de Potencia.
n1     = Numero de buses de control de voltaje del Sistema Eléctrico de Potencia.
n2     = Numero de buses de Carga del Sistema Eléctrico de Potencia.
no_lin = Numero de Líneas del Sistema Eléctrico de Potencia.
a[i]   = Nodo de Envío para la línea i.
b[i]   = Nodo de recepción para la línea i.
R      = estructura compleja formada por:
        R.x -> Matriz de las resistencias de las líneas.
        R.y -> Matriz de las reactancias de las líneas.
Sg     = Estructura Compleja para la Potencia aparente de "GENERACION"
        formada por:
        Sg.x -> Potencia real de Generacion
```

Sg.y -> Potencia reactiva de Generacion
Sc = Estructura Compleja para la Potencia aparente de "CARGA" formada por:

Sc.x -> Potencia real de Carga

Sc.y -> Potencia reactiva de Carga

VOLTAJE = Estructura Compleja para leer los datos de voltaje y angulo en los buses

VOLTAJE.x -> Contiene el Voltaje Inicial en cada Bus.

VOLTAJE.y -> Contiene el angulo inicial en cada Bus.

Si se quisiera cargar la red anterior, el archivo de datos deberá ser:

```
3
1
1
3
1 2 0.017 0.1
2 3 0.01 0.2
1 3 0.01 0.2
SLACK 0 0 0 0 1.0 0
P-V 0.3217 0 0 0 1.1 0
P-Q 0 0 0.6392 0.5339 1.0 0
```

3.2.2 Impresión de Datos de Entrada. Consiste de una función que extrae los datos leídos del archivo correspondiente de entrada y los copia al archivo de salida. De esta forma se van construyendo en este archivo, todos los resultados obtenidos a partir de la aplicación del algoritmo al modelo de RED leído.

3.2.3 Cálculo de la matriz Ybus. Este cálculo consiste en aplicar los conceptos siguientes:

1. Cada elemento de la matriz que coincide con un bus, es decir que el renglón y la columna de la matriz sea el mismo, se calcula sumando todas las impedancias de las líneas que llegan a él. Ya obtenido el resultado, se invierte el signo.

2. Cada elemento de la matriz para una línea de transmisión que va de un bus a otro, se calcula invirtiendo el valor de la impedancia de la línea.

El diagrama de flujo que describe esta función se muestra a continuación e incluye la rutina de inversión de un número complejo:

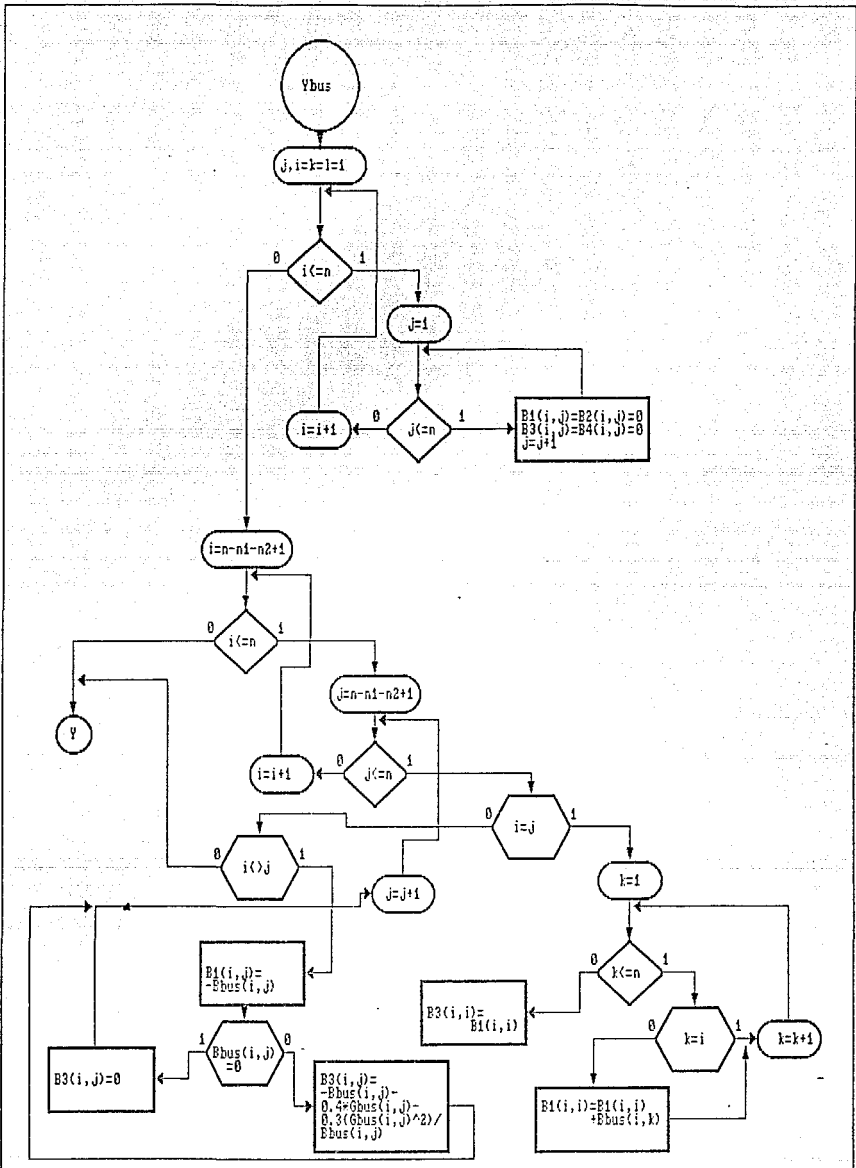


Diagrama : Funcion Ybus.

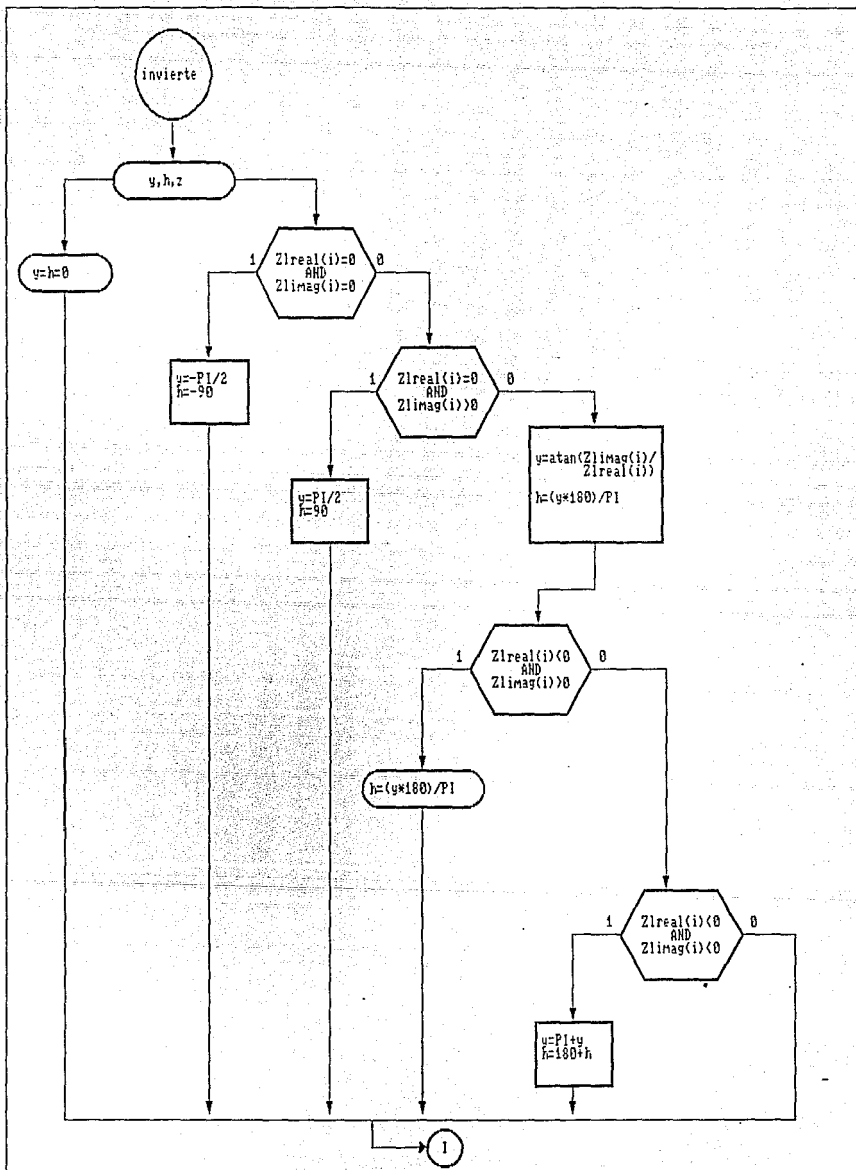
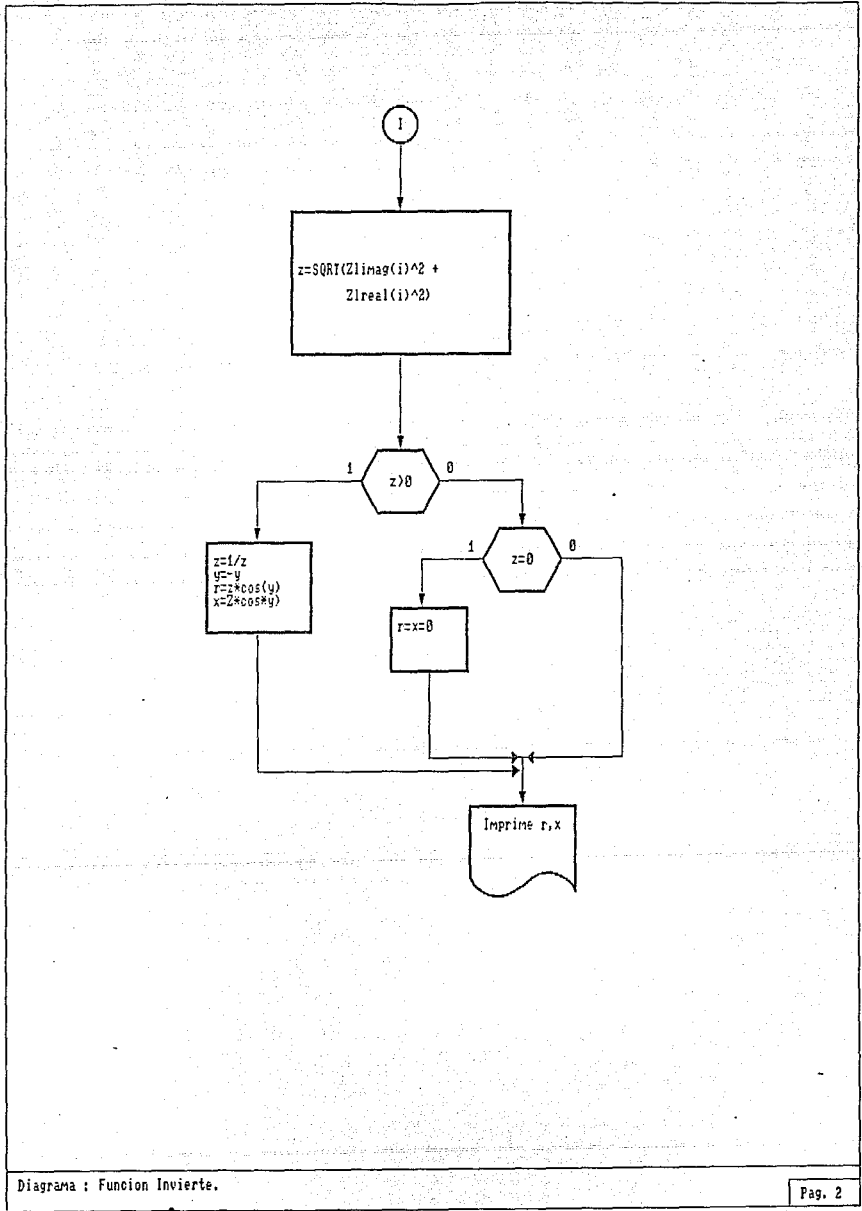


Diagrama : Funcion Invierte.



3.2.4 Cálculo de las Matrices B' , B'' , B''' . Para este cálculo se toma la matriz Y_{bus} con sus componentes real e imaginario, la matriz G_{bus} o matriz de impedancias y la matriz B_{bus} o matriz de susceptancias. El proceso de cálculo se divide en dos pasos, ya que se compararán los resultados para dos algoritmos diferentes de flujos de potencia:

1. Modelo desacoplado Rápido. Las ecuaciones para el cálculo de las matrices B' y B'' se basan en las ecuaciones 2.5 y 2.6 y las correcciones hechas al método para mejorar su convergencia, en las ecuaciones 2.7. Los valores obtenidos se guardan en matrices denominadas $B1$ y $B3$ respectivamente.

2. Modelo propuesto. Las ecuaciones para el cálculo de las matrices B' y B''' se basan en las ecuaciones 2.5 y 2.10 y las correcciones hechas al método desacoplado rápido mejorado (Como se indicó en punto anterior), en las ecuaciones 2.11. Los valores obtenidos se guardan en matrices denominadas $B2$ y $B4$ respectivamente.

Para estos cálculos se considera que el primer bus es siempre el bus de referencia o Slack.

El diagrama de flujo para esta función se puede observar en el diagrama de la función Y_{bus} .

3.2.5. Función para el Cálculo de Flujos de Carga. Aquí se calculan los incrementos de Potencia real y reactiva que serán aplicados en la construcción del sistema de ecuaciones lineales indicados en las ecuaciones 2.3, 2.4 y 2.9.

3.2.6 Función Gauss-Seidell. Aquí se calculan los incrementos de voltaje y de ángulo para cada bus, luego esos incrementos se aplican a los respectivos valores de las magnitudes de voltaje y ángulo obtenidos en la iteración anterior hasta que cada incremento, en cada bus, sea menor al criterio de convergencia cv .

Los correspondientes diagramas de flujo se muestran en la página siguiente. El listado de la codificación del algoritmo aparece en el anexo 1.

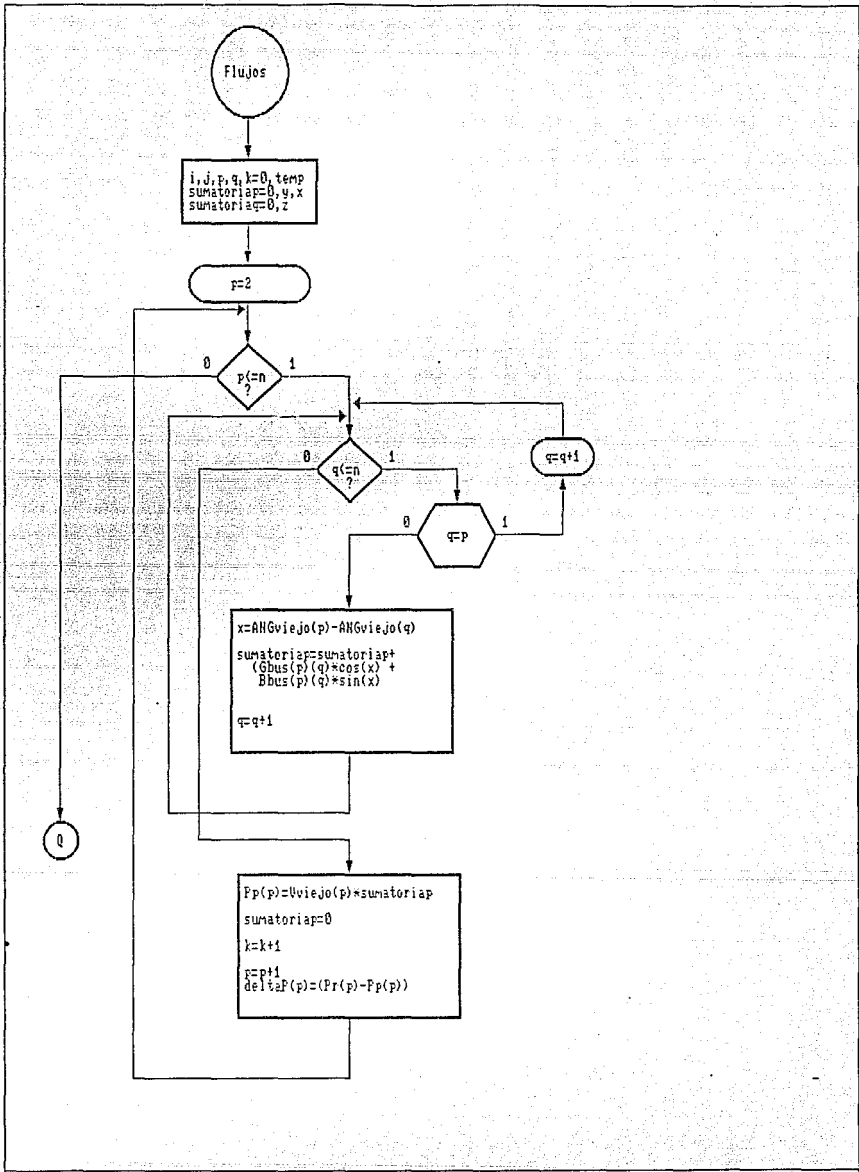
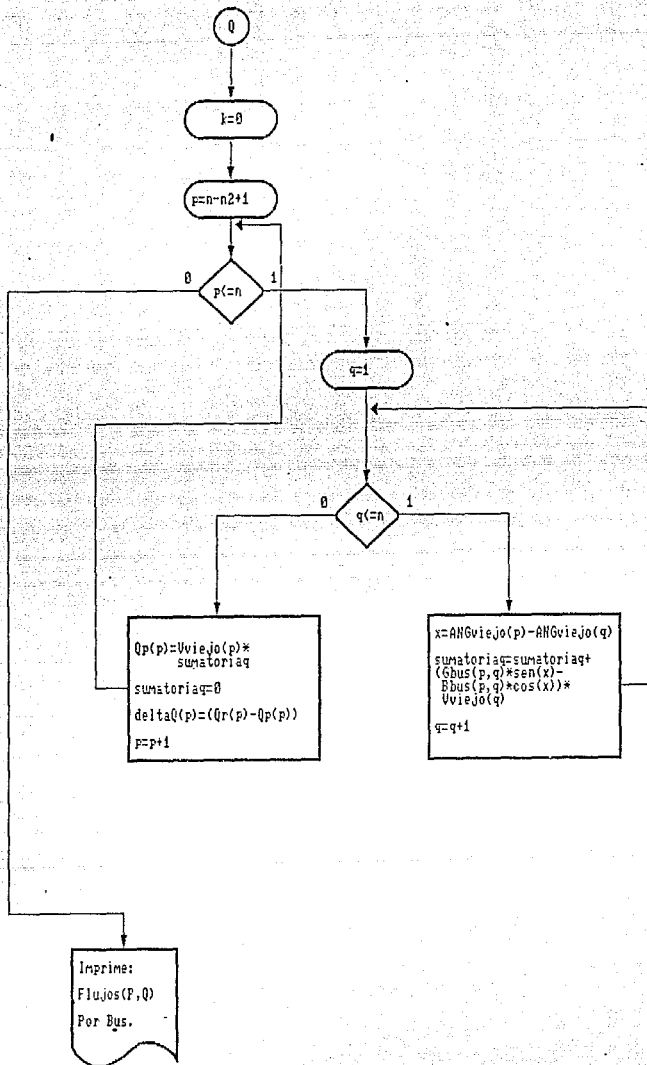


Diagrama : Funcion Flujos.



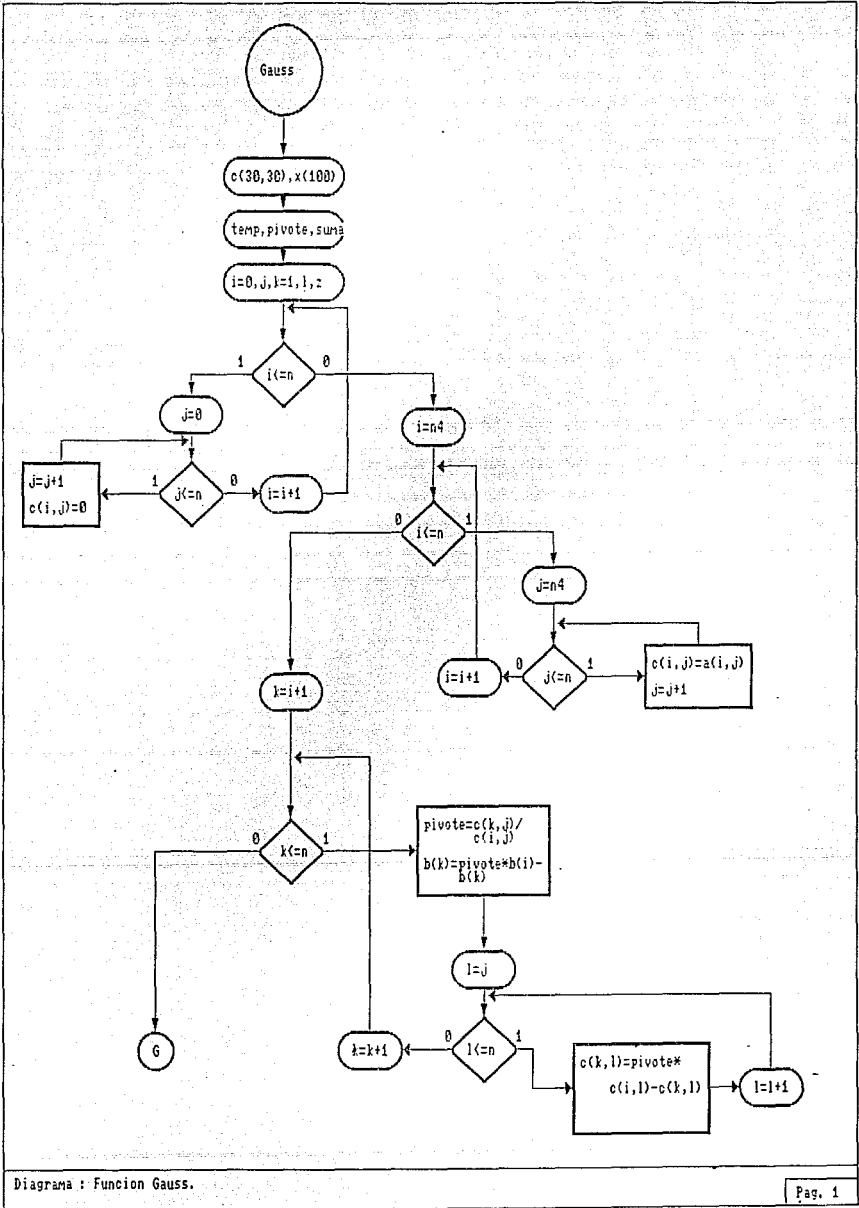
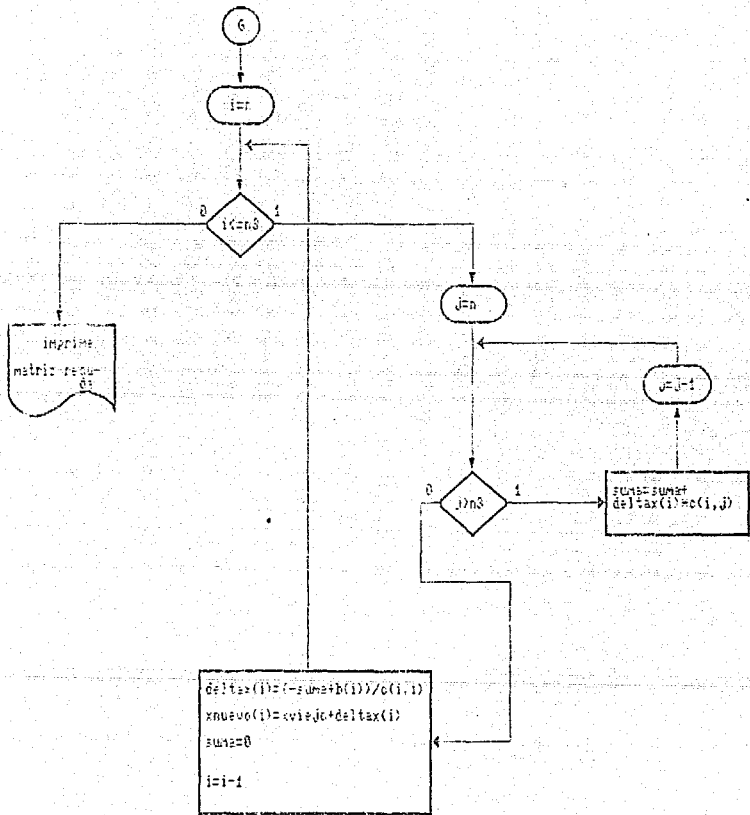


Diagrama : Funcion Gauss.



4. APLICACION DEL ALGORITMO PARA ALGUNOS MODELOS DE REDES ELECTRICAS.

En este capítulo se aplicará el algoritmo estudiado a redes eléctricas reales, para estudiar su comportamiento y poder sacar las conclusiones correspondientes.

Los datos de las líneas están en p.u. y los datos de las potencias en los buses toman como potencia base 100 MVA.

4.1. CASO 1. SISTEMA DE 3 BUSES y 3 LINEAS CON 1 GENERADOR.

En la figura 4.1 se muestra el diagrama unifilar del sistema. Los parámetros de las líneas se dan en la tabla 4.1 y los datos de los buses se dan en la tabla 4.2.

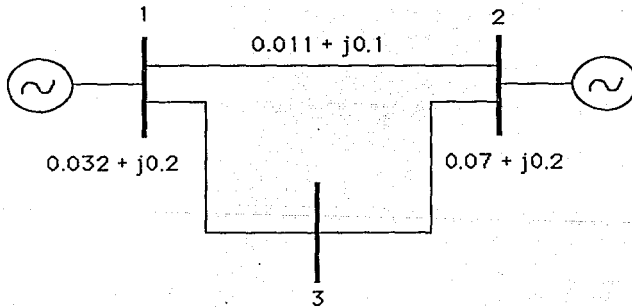


Figura 4.1.

Línea No.	Entre buses	Impedancia de la línea
1	1-2	0.011 + 0.1
2	2-3	0.070 + 0.2
3	1-3	0.032 + 0.2

Tabla 4.1.

Bus No.	Tipo	Generador		Carga		Magnitud de Voltaje
		P	Q	P	Q	
1	Slack					1.0
2	P-V	9.3217				1.1
3	P-Q			0.036392	0.5339	

Tabla 4.2.

A continuación aparecen los resultados de la ejecución del programa para el caso 1:

 LOS DATOS QUE A CONTINUACION APARECEN, SON LOS DATOS DE LA RED CUYAS RESISTENCIAS DE CADA LINEA, SON MULTIPLICADAS POR EL FACTOR ESPECIFICADO

NUMERO DE BUSES: 3 NUMERO DE LINEAS: 3

LINEA No.	ENTRE BUSES	IMPEDANCIA DE LA LINEA
1	1- 2	0.011 + J0.100
2	2- 3	0.070 + J0.200
3	1- 3	0.032 + J0.200

DATOS DE LOS BUSES :

BUS No.	TIPO	GENERACION		CARGA		VOLTAJE	
		P	Q	P	Q	V	ang.
1	SLACK	0.0000	0.0000	0.0000	0.0000	1.0000	0.0000
2	P-V	9.3217	0.0000	0.0000	0.0000	1.1000	0.0000
3	P-Q	0.0000	0.0000	0.0364	0.5339	1.0000	0.0000

DATOS DE LA RED:

FACTOR DE MULTIPLICACION : 0.50

NUMERO DE BUSES: 3 NUMERO DE LINEAS: 3

LINEA No.	ENTRE BUSES	IMPEDANCIA DE LA LINEA
1	1- 2	0.006 J0.100
2	2- 3	0.035 J0.200
3	1- 3	0.016 J0.200

EL SISTEMA DESACOPLADO RAPIDO, CON FACTOR 0.5, CONVERGE EN LA 4a. ITERACION

LOS RESULTADOS DESPUES DE 4 ITERACIONES SON:

BUS	VOLTAJE	ANGULO(rad)
1	1.0000	0.0000
2	1.1000	0.7232
3	0.9057	0.3830

EL SISTEMA CON EL METODO PROPUESTO, CON FACTOR 0.5, CONVERGE EN LA 5a. ITERACION

LOS RESULTADOS DESPUES DE 5 ITERACIONES SON:

BUS	VOLTAJE	ANGULO(rad)
1	1.0000	0.0000
2	1.1000	0.7258
3	0.9082	0.3859

DATOS DE LA RED:

FACTOR DE MULTIPLICACION : 1.00

NUMERO DE BUSES: 3 NUMERO DE LINEAS: 3

LINEA No.	ENTRE BUSES	IMPEDANCIA DE LA LINEA
1	1- 2	0.011 J0.100
2	2- 3	0.070 J0.200
3	1- 3	0.032 J0.200

EL SISTEMA DESACOPLADO RAPIDO, CON FACTOR 1.0, CONVERGE EN LA 4a. ITERACION

LOS RESULTADOS DESPUES DE 4 ITERACIONES SON:

BUS	VOLTAJE	ANGULO(rad)
1	1.0000	0.0000
2	1.1000	0.7078
3	0.8897	0.3800

EL SISTEMA CON EL METODO PROPUESTO, CON FACTOR 1.0, CONVERGE EN LA 6a. ITERACION

LOS RESULTADOS DESPUES DE 6 ITERACIONES SON:

BUS	VOLTAJE	ANGULO(rad)
1	1.0000	0.0000
2	1.1000	0.7075
3	0.8950	0.3779

DATOS DE LA RED:

FACTOR DE MULTIPLICACION : 1.50

LINEA No.	ENTRE BUSES	IMPEDANCIA DE LA LINEA
1	1- 2	0.017 J0.100
2	2- 3	0.105 J0.200
3	1- 3	0.048 J0.200

EL SISTEMA DESACOPLADO RAPIDO, CON FACTOR 1.5, CONVERGE EN LA 5a. ITERACION

LOS RESULTADOS DESPUES DE 5 ITERACIONES SON:

BUS	VOLTAJE	ANGULO(rad)
1	1.0000	0.0000
2	1.1000	0.7018
3	0.8763	0.3776

EL SISTEMA CON EL METODO PROPUESTO, CON FACTOR 1.5, CONVERGE EN LA 6a. ITERACION

LOS RESULTADOS DESPUES DE 6 ITERACIONES SON:

BUS	VOLTAJE	ANGULO(rad)
1	1.0000	0.0000
2	1.1000	0.7016
3	0.8841	0.3741

DATOS DE LA RED:

FACTOR DE MULTIPLICACION : 2.00

LINEA No.	ENTRE BUSES	IMPEDANCIA DE LA LINEA
1	1- 2	0.022 J0.100
2	2- 3	0.140 J0.200
3	1- 3	0.064 J0.200

EL SISTEMA DESACOPLADO RAPIDO, CON FACTOR 2.0, CONVERGE EN LA 6a. ITERACION

LOS RESULTADOS DESPUES DE 6 ITERACIONES SON:

BUS	VOLTAJE	ANGULO(rad)
1	1.0000	0.0000
2	1.1000	0.7033
3	0.8621	0.3741

EL SISTEMA CON EL METODO PROPUESTO, CON FACTOR 2.0, CONVERGE EN LA 7a. ITERACION

LOS RESULTADOS DESPUES DE 7 ITERACIONES SON:

BUS	VOLTAJE	ANGULO(rad)
1	1.0000	0.0000
2	1.1000	0.6870
3	0.7806	0.3834

DATOS DE LA RED:

FACTOR DE MULTIPLICACION : 2.50

LINEA No.	ENTRE BUSES	IMPEDANCIA DE LA LINEA
1	1- 2	0.028 J0.100
2	2- 3	0.175 J0.200
3	1- 3	0.080 J0.200

EL SISTEMA DESACOPLADO RAPIDO, CON FACTOR 2.5, CONVERGE EN LA 7a. ITERACION

LOS RESULTADOS DESPUES DE 7 ITERACIONES SON:

BUS	VOLTAJE	ANGULO(rad)
1	1.0000	0.0000
2	1.1000	0.7114
3	0.8454	0.3764

EL SISTEMA CON EL METODO PROPUESTO, CON FACTOR 2.5, CONVERGE EN LA 7a. ITERACION

LOS RESULTADOS DESPUES DE 7 ITERACIONES SON:

BUS	VOLTAJE	ANGULO(rad)
1	1.0000	0.0000
2	1.1000	0.7114
3	0.8616	0.3669

DATOS DE LA RED:

FACTOR DE MULTIPLICACION : 3.00

LINEA No.	ENTRE BUSES	IMPEDANCIA DE LA LINEA
1	1- 2	0.033 JO.100
2	2- 3	0.210 JO.200
3	1- 3	0.096 JO.200

EL SISTEMA DESACOPLADO RAPIDO, CON FACTOR 3.0, CONVERGE EN LA 8a. ITERACION

LOS RESULTADOS DESPUES DE 8 ITERACIONES SON:

BUS	VOLTAJE	ANGULO(rad)
1	1.0000	0.0000
2	1.1000	0.7248
3	0.8358	0.3877

EL SISTEMA CON EL METODO PROPUESTO, CON FACTOR 3.0, CONVERGE EN LA 10a. ITERACION

LOS RESULTADOS DESPUES DE 10 ITERACIONES SON:

BUS	VOLTAJE	ANGULO(rad)
1	1.0000	0.0000
2	1.1000	0.7243
3	0.8295	0.3826

DATOS DE LA RED:

FACTOR DE MULTIPLICACION : 3.50

LINEA No.	ENTRE BUSES	IMPEDANCIA DE LA LINEA
1	1- 2	0.039 J0.100
2	2- 3	0.245 J0.200
3	1- 3	0.112 J0.200

EL SISTEMA DESACOPLADO RAPIDO, CON FACTOR 3.5, CONVERGE EN LA 11a. ITERACION

LOS RESULTADOS DESPUES DE 11 ITERACIONES SON:

BUS	VOLTAJE	ANGULO(rad)
1	1.0000	0.0000
2	1.1000	0.7417
3	0.8218	0.3821

EL SISTEMA CON EL METODO PROPUESTO, CON FACTOR 3.5, CONVERGE EN LA 11a. ITERACION

LOS RESULTADOS DESPUES DE 11 ITERACIONES SON:

BUS	VOLTAJE	ANGULO(rad)
1	1.0000	0.0000
2	1.1000	0.7418
3	0.8163	0.3944

DATOS DE LA RED:

FACTOR DE MULTIPLICACION : 4.00

LINEA No.	ENTRE BUSES	IMPEDANCIA DE LA LINEA
1	1- 2	0.044 J0.100
2	2- 3	0.280 J0.200
3	1- 3	0.128 J0.200

EL SISTEMA DESACOPLADO RAPIDO, CON FACTOR 4.0, CONVERGE EN LA 17a. ITERACION

LOS RESULTADOS DESPUES DE 17 ITERACIONES SON:

BUS	VOLTAJE	ANGULO(rad)
1	1.0000	0.0000
2	1.1000	0.7628
3	0.8209	0.3986

EL SISTEMA CON EL METODO PROPUESTO, CON FACTOR 4.0, CONVERGE EN LA 12a. ITERACION

LOS RESULTADOS DESPUES DE 12 ITERACIONES SON:

BUS	VOLTAJE	ANGULO(rad)
1	1.0000	0.0000
2	1.1000	0.7628
3	0.8072	0.4093

DATOS DE LA RED:

FACTOR DE MULTIPLICACION : 4.50

NUMERO DE BUSES: 3 NUMERO DE LINEAS: 3

LINEA No.	ENTRE BUSES	IMPEDANCIA DE LA LINEA
1	1- 2	0.050 J0.100
2	2- 3	0.315 J0.200
3	1- 3	0.144 J0.200

EL SISTEMA DESACOPLADO RAPIDO, CON FACTOR 4.5, CONVERGE EN LA 30a. ITERACION

LOS RESULTADOS DESPUES DE 30 ITERACIONES SON:

BUS	VOLTAJE	ANGULO(rad)
1	1.0000	0.0000
2	1.1000	0.7862
3	0.8133	0.4072

EL SISTEMA CON EL METODO PROPUESTO, CON FACTOR 4.5, CONVERGE EN LA 16a. ITERACION

LOS RESULTADOS DESPUES DE 16 ITERACIONES SON:

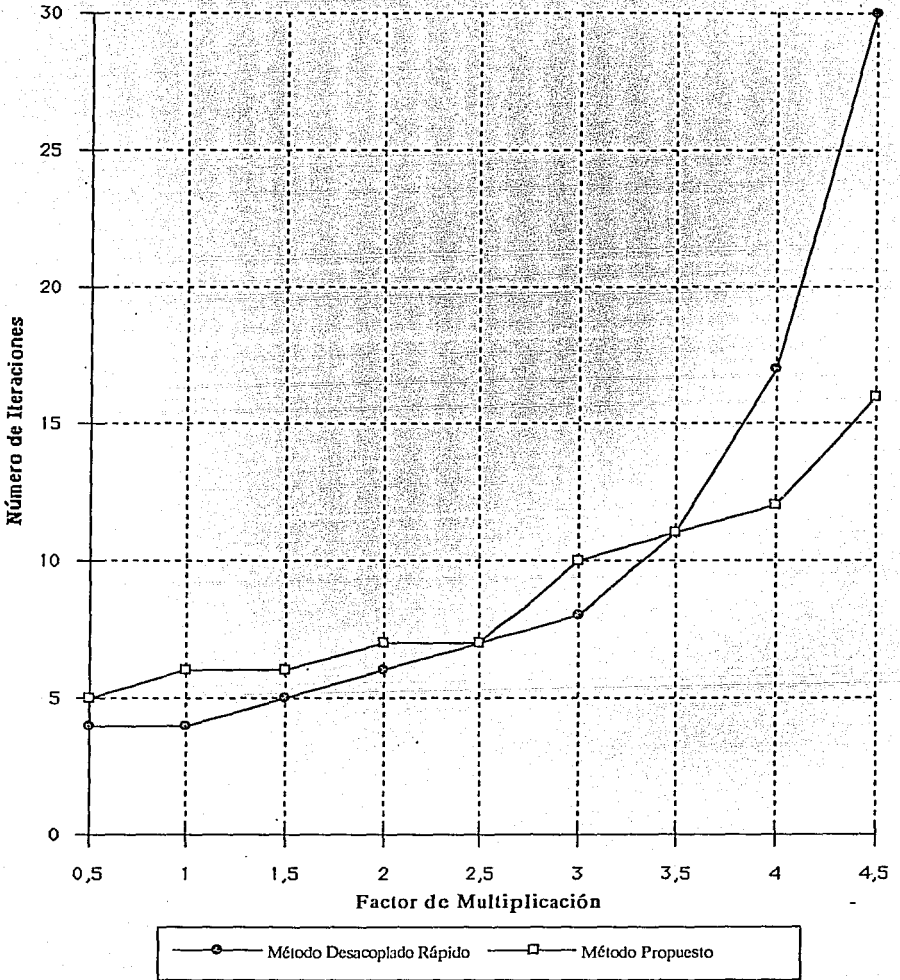
BUS	VOLTAJE	ANGULO(rad)
1	1.0000	0.0000
2	1.1000	0.7859
3	0.8213	0.3865

Número de iteraciones cuando todas las resistencias de las líneas son multiplicadas por el factor :

factor	0.50	1.00	1.50	2.00	2.50	3.00	3.50	4.00	4.5
Des. Rápido	4.0	4.0	5.0	6.0	7.0	8.0	11.0	17.0	30
Propuesto	5.0	6.0	6.0	7.0	7.0	10.0	11.0	12.0	16

Los resultados se muestran graficados en la siguiente página.

RESULTADOS DEL CASO 1



4.2. CASO 2. SISTEMA DE 6 BUSES y 7 LINEAS CON 2 GENERADORES.

En la figura 4.2 se muestra el diagrama unifilar del sistema. Los parámetros de las líneas se dan en la tabla 4.3 y los datos de los buses se dan en la tabla 4.4.

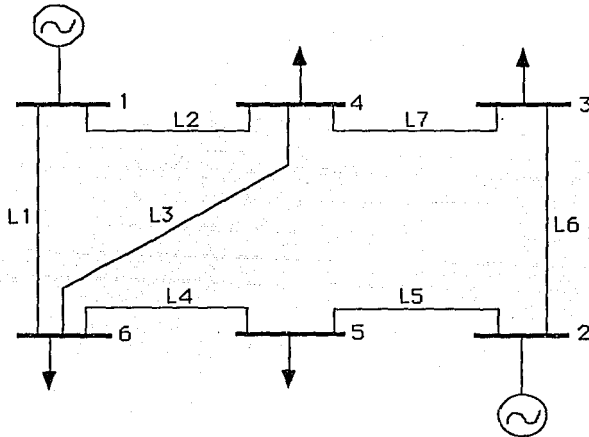


Figura 4.2

Línea No.	Entre buses	Impedancia de la línea
1	1-6	$0.123 + j0.518$
2	1-4	$0.08 + j0.370$
3	4-6	$0.097 + j0.407$
4	6-5	$0 + j0.3$
5	5-2	$0.282 + j0.640$
6	2-3	$0.723 + j1.050$
7	4-3	$0 + j0.133$

Tabla 4.3

Bus No.	Tipo	Generador		Carga		Magnitud de Voltaje
		P	Q	P	Q	
1	Slack					1.05
2	P-V	0.90	0.4			1.1
3	P-Q			0.35	0.13	1.0
4	P-Q					1.0
5	P-Q			0.10	0.18	1.0
6	P-Q			0.20	0.05	1.0

Tabla 4.4

A continuación se muestran los resultados obtenidos con el algoritmo para el caso 2:

 LOS DATOS QUE A CONTINUACION APARECEN, SON LOS DATOS DE LA RED CUYAS RESISTENCIAS DE CADA LINEA, SON MULTIPLICADAS POR EL FACTOR ESPECIFICADO

NUMERO DE BUSES: 6 NUMERO DE LINEAS: 7

LINEA No.	ENTRE BUSES	IMPEDANCIA DE LA LINEA
1	1- 6	0.123 J0.518
2	1- 4	0.080 J0.370
3	4- 6	0.097 J0.407
4	6- 5	0.000 J0.300
5	5- 2	0.282 J0.640
6	2- 3	0.723 J1.050
7	4- 3	0.000 J0.133

DATOS DE LOS BUSES :

BUS No.	TIPO	GENERACION		CARGA		VOLTAJE	
		P	Q	P	Q	V	ang.
1	SLACK	0.0000	0.0000	0.0000	0.0000	1.0500	0.0000
2	P-V	0.9000	0.4000	0.0000	0.0000	1.1000	0.0000
3	P-Q	0.0000	0.0000	0.3500	0.1300	1.0000	0.0000
4	P-Q	0.0000	0.0000	0.0000	0.0000	1.0000	0.0000
5	P-Q	0.0000	0.0000	0.1000	0.1800	1.0000	0.0000
6	P-Q	0.0000	0.0000	0.2000	0.0500	1.0000	0.0000

DATOS DE LA RED:

FACTOR DE MULTIPLICACION : 0.50

LINEA No.	ENTRE BUSES	IMPEDANCIA DE LA LINEA
1	1- 6	0.062 J0.518
2	1- 4	0.040 J0.370
3	4- 6	0.049 J0.407
4	6- 5	0.000 J0.300
5	5- 2	0.141 J0.640
6	2- 3	0.362 J1.050
7	4- 3	0.000 J0.133

EL SISTEMA DESACOPLADO RAPIDO, CON FACTOR 0.5, CONVERGE EN LA 3a. ITERACION

LOS RESULTADOS DESPUES DE 3 ITERACIONES SON:

BUS	VOLTAJE	ANGULO(rad)
1	1.0500	0.0000
2	1.1000	0.4398
3	0.9446	0.0363
4	0.9713	0.0338
5	0.9328	0.1715
6	0.9672	0.0574

EL SISTEMA CON EL METODO PROPUESTO, CON FACTOR 0.5, CONVERGE EN LA 4a. ITERACION

LOS RESULTADOS DESPUES DE 4 ITERACIONES SON:

BUS	VOLTAJE	ANGULO(rad)
1	1.0500	0.0000
2	1.1000	0.4378
3	0.9399	0.0341
4	0.9671	0.0318
5	0.9290	0.1700
6	0.9612	0.0548

DATOS DE LA RED:

FACTOR DE MULTIPLICACION : 1.00

LINEA No.	ENTRE BUSES	IMPEDANCIA DE LA LINEA
1	1- 6	0.123 J0.518
2	1- 4	0.080 J0.370
3	4- 6	0.097 J0.407
4	6- 5	0.000 J0.300
5	5- 2	0.282 J0.640
6	2- 3	0.723 J1.050
7	4- 3	0.000 J0.133

EL SISTEMA DESACOPLADO RAPIDO, CON FACTOR 1.0, CONVERGE EN LA 4a. ITERACION

LOS RESULTADOS DESPUES DE 4 ITERACIONES SON:

BUS	VOLTAJE	ANGULO(rad)
1	1.0500	0.0000
2	1.1000	0.4517
3	0.9017	0.0240
4	0.9395	0.0310
5	0.8880	0.1821
6	0.9391	0.0622

EL SISTEMA CON EL METODO PROPUESTO, CON FACTOR 1.0, CONVERGE EN LA 6a. ITERACION

LOS RESULTADOS DESPUES DE 6 ITERACIONES SON:

BUS	VOLTAJE	ANGULO(rad)
1	1.0500	0.0000
2	1.1000	0.4510
3	0.9011	0.0227
4	0.9388	0.0296
5	0.8880	0.1798
6	0.9378	0.0606

DATOS DE LA RED:

FACTOR DE MULTIPLICACION : 1.50

LINEA No.	ENTRE BUSES	IMPEDANCIA DE LA LINEA
1	1- 6	0.185 J0.518
2	1- 4	0.120 J0.370
3	4- 6	0.146 J0.407
4	6- 5	0.000 J0.300
5	5- 2	0.423 J0.640
6	2- 3	1.085 J1.050
7	4- 3	0.000 J0.133

EL SISTEMA DESACOPLADO RAPIDO, CON FACTOR 1.5, CONVERGE EN LA 5a. ITERACION

LOS RESULTADOS DESPUES DE 5 ITERACIONES SON:

BUS	VOLTAJE	ANGULO(rad)
1	1.0500	0.0000
2	1.1000	0.4929
3	0.8664	0.0142
4	0.9112	0.0313
5	0.8438	0.1986
6	0.9123	0.0726

EL SISTEMA CON EL METODO PROPUESTO, CON FACTOR 1.5, CONVERGE EN LA 6a. ITERACION

LOS RESULTADOS DESPUES DE 6 ITERACIONES SON:

BUS	VOLTAJE	ANGULO(rad)
1	1.0500	0.0000
2	1.1000	0.4928
3	0.8627	0.0138
4	0.9079	0.0310
5	0.8374	0.1999
6	0.9072	0.0729

DATOS DE LA RED:

FACTOR DE MULTIPLICACION : 2.00

LINEA No.	ENTRE BUSES	IMPEDANCIA DE LA LINEA
1	1- 6	0.246 J0.518
2	1- 4	0.160 J0.370
3	4- 6	0.194 J0.407
4	6- 5	0.000 J0.300
5	5- 2	0.564 J0.640
6	2- 3	1.446 J1.050
7	4- 3	0.000 J0.133

EL SISTEMA DESACOPLADO RAPIDO, CON FACTOR 2.0, CONVERGE EN LA 7a. ITERACION

LOS RESULTADOS DESPUES DE 7 ITERACIONES SON:

BUS	VOLTAJE	ANGULO(rad)
1	1.0500	0.0000
2	1.1000	0.5493
3	0.8227	0.0087
4	0.8734	0.0370
5	0.7772	0.2221
6	0.8704	0.0908

EL SISTEMA CON EL METODO PROPUESTO, CON FACTOR 2.0, CONVERGE EN LA 6a. ITERACION

LOS RESULTADOS DESPUES DE 6 ITERACIONES SON:

BUS	VOLTAJE	ANGULO(rad)
1	1.0500	0.0000
2	1.1000	0.5524
3	0.8258	0.0089
4	0.8760	0.0365
5	0.7775	0.2269
6	0.8719	0.0922

DATOS DE LA RED:

FACTOR DE MULTIPLICACION : 2.50

LINEA No.	ENTRE BUSES	IMPEDANCIA DE LA LINEA
1	1- 6	0.308 J0.518
2	1- 4	0.200 J0.370
3	4- 6	0.243 J0.407
4	6- 5	0.000 J0.300
5	5- 2	0.705 J0.640
6	2- 3	1.808 J1.050
7	4- 3	0.000 J0.133

EL SISTEMA DESACOPLADO RAPIDO, CON FACTOR 2.5, CONVERGE EN LA 10a. ITERACION

LOS RESULTADOS DESPUES DE 10 ITERACIONES SON:

BUS	VOLTAJE	ANGULO(rad)
1	1.0500	0.0000
2	1.1000	0.6248
3	0.7879	0.0034
4	0.8421	0.0419
5	0.7253	0.2406
6	0.8380	0.1058

EL SISTEMA CON EL METODO PROPUESTO, CON FACTOR 2.5, CONVERGE EN LA 6a. ITERACION

LOS RESULTADOS DESPUES DE 6 ITERACIONES SON:

BUS	VOLTAJE	ANGULO(rad)
1	1.0500	0.0000
2	1.1000	0.6181
3	0.7882	0.0063
4	0.8414	0.0445
5	0.7065	0.2549
6	0.8298	0.1164

DATOS DE LA RED:

FACTOR DE MULTIPLICACION : 3.00

LINEA No.	ENTRE BUSES	IMPEDANCIA DE LA LINEA
1	1- 6	0.369 J0.518
2	1- 4	0.240 J0.370
3	4- 6	0.291 J0.407
4	6- 5	0.000 J0.300
5	5- 2	0.846 J0.640
6	2- 3	2.169 J1.050
7	4- 3	0.000 J0.133

EL SISTEMA DESACOPLADO RAPIDO, CON FACTOR 3.0, CONVERGE EN LA 16a. ITERACION

LOS RESULTADOS DESPUES DE 16 ITERACIONES SON:

BUS	VOLTAJE	ANGULO(rad)
1	1.0500	0.0000
2	1.1000	0.7017
3	0.7455	0.0038
4	0.8027	0.0546
5	0.6517	0.2740
6	0.7881	0.1359

EL SISTEMA CON EL METODO PROPUESTO, CON FACTOR 3.0, CONVERGE EN LA 6a. ITERACION

LOS RESULTADOS DESPUES DE 6 ITERACIONES SON:

BUS	VOLTAJE	ANGULO(rad)
1	1.0500	0.0000
2	1.1000	0.6837
3	0.7477	0.0029
4	0.8024	0.0522
5	0.6243	0.2764
6	0.7791	0.1416

DATOS DE LA RED:

FACTOR DE MULTIPLICACION : 3.50

LINEA No.	ENTRE BUSES	IMPEDANCIA DE LA LINEA
1	1- 6	0.431 J0.518
2	1- 4	0.280 J0.370
3	4- 6	0.340 J0.407
4	6- 5	0.000 J0.300
5	5- 2	0.987 J0.640
6	2- 3	2.531 J1.050
7	4- 3	0.000 J0.133

EL SISTEMA DESACOPLADO RAPIDO, CON FACTOR 3.5, CONVERGE EN LA 24a. ITERACION

LOS RESULTADOS DESPUES DE 24 ITERACIONES SON:

BUS	VOLTAJE	ANGULO(rad)
1	1.0500	0.0000
2	1.1000	0.7918
3	0.7019	-0.0013
4	0.7617	0.0625
5	0.5937	0.3039
6	0.7487	0.1626

EL SISTEMA CON EL METODO PROPUESTO, CON FACTOR 3.5, CONVERGE EN LA 6a. ITERACION

LOS RESULTADOS DESPUES DE 6 ITERACIONES SON:

BUS	VOLTAJE	ANGULO(rad)
1	1.0500	0.0000
2	1.1000	0.7505
3	0.7036	-0.0046
4	0.7588	0.0568
5	0.5347	0.2844
6	0.7202	0.1634

DATOS DE LA RED:

FACTOR DE MULTIPLICACION : 4.00

LINEA No.	ENTRE BUSES	IMPEDANCIA DE LA LINEA
1	1- 6	0.492 J0.518
2	1- 4	0.320 J0.370
3	4- 6	0.388 J0.407
4	6- 5	0.000 J0.300
5	5- 2	1.128 J0.640
6	2- 3	2.892 J1.050
7	4- 3	0.000 J0.133

EL SISTEMA CON EL METODO DESACOPLADO RAPIDO Diverge DESPUES DE 100 ITERACIONES

EL SISTEMA CON EL METODO PROPUESTO, CON FACTOR 4.0, CONVERGE EN LA 8a. ITERACION

LOS RESULTADOS DESPUES DE 8 ITERACIONES SON:

BUS	VOLTAJE	ANGULO(rad)
1	1.0500	0.0000
2	1.1000	0.8274
3	0.6460	-0.0132
4	0.7031	0.0646
5	0.4403	0.3151
6	0.6477	0.1962

DATOS DE LA RED:

FACTOR DE MULTIPLICACION : 4.50

LINEA No.	ENTRE BUSES	IMPEDANCIA DE LA LINEA
1	1- 6	0.554 J0.518
2	1- 4	0.360 J0.370
3	4- 6	0.437 J0.407
4	6- 5	0.000 J0.300
5	5- 2	1.269 J0.640
6	2- 3	3.254 J1.050
7	4- 3	0.000 J0.133

EL SISTEMA CON EL METODO DESACOPLADO RAPIDO Diverge DESPUES DE 100 ITERACIONES

EL SISTEMA CON EL METODO PROPUESTO, CON FACTOR 4.5, CONVERGE EN LA 9a. ITERACION

LOS RESULTADOS DESPUES DE 9 ITERACIONES SON:

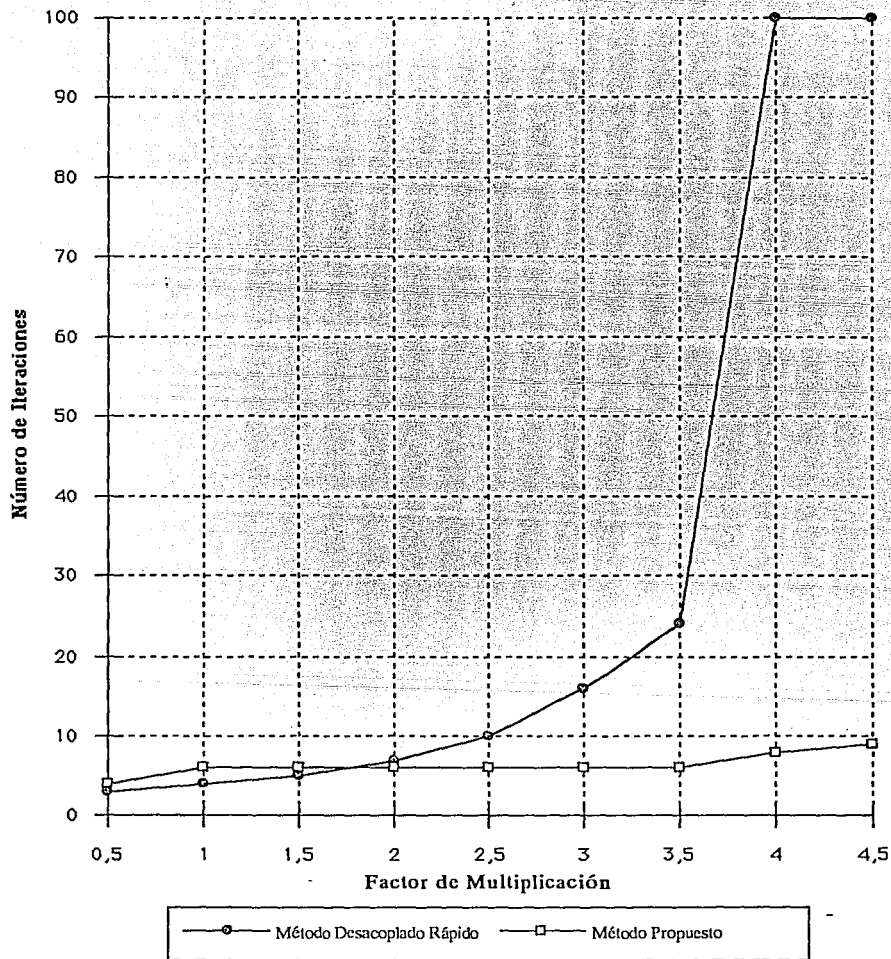
BUS	VOLTAJE	ANGULO(rad)
1	1.0500	0.0000
2	1.1000	0.8691
3	0.5774	-0.0432
4	0.6345	0.0561
5	0.3112	0.2967
6	0.5487	0.2082

Número de iteraciones cuando todas las resistencias de las líneas son multiplicadas por el factor :

factor	0.50	1.00	1.50	2.00	2.50	3.00	3.50	4.00	4.5
Des. Rápido	3.0	4.0	5.0	7.0	10.0	16.0	24.0	div	div
Propuesto	4.0	6.0	6.0	6.0	6.0	6.0	6.0	8.0	9.0

Los resultados se muestran graficados en la siguiente página.

RESULTADOS DEL CASO 2



CONCLUSIONES

Para evaluar, a manera de conclusiones, los resultados obtenidos por el algoritmo al ser aplicado éste a los dos modelos de redes eléctricas utilizadas, es necesario enmarcar el análisis en los objetivos globales que persigue el planteamiento de dicho algoritmo:

1. En primer lugar los valores de las magnitudes de Angulo y Voltaje obtenidos deberán tener una variación máxima del 5% con relación a los valores obtenidos por el Método Desacoplado Rápido. De esta forma la precisión se mantiene al usar este nuevo Método.

Para evaluar este objetivo basta con encontrar el porcentaje de variación entre cada uno de las magnitudes de angulo y voltaje de los dos métodos. Por ejemplo para el caso I, los resultados en la convergencia de los dos métodos para el factor de multiplicación de 0.5, los resultados obtenidos fueron (Ver Pág. 51):

Bus	Desacoplado Rápido		Método Propuesto	
	Voltaje	Angulo	Voltaje	Angulo
1	1.000	0.0000	1.0000	0.0000
2	1.100	0.7232	1.1000	0.7258
3	0.9057	0.3830	0.9082	0.3859

Si tomamos como el 100% de precisión al Método Desacoplado Rápido, tendríamos la siguiente tabla de variación de precisión del Método propuesto, con relación al Desacoplado Rápido:

Bus	% de Variación en la magnitud de Voltaje	% de Variación en la magnitud del Angulo
1	0.00%	0.00%
2	0.00%	0.35%
3	0.27%	0.75%

Como se puede observar ningún porcentaje de variación excede al 5%. Esto se puede comprobar en los dos casos de estudio, con sus respectivos resultados con cada factor de multiplicación.

2. El número de Iteraciones debe disminuir con relación al Método Desacoplado Rápido, a medida que el cociente R/X de alguna de las líneas de transmisión se hace mayor o igual a 0.9. Esta constante no ha sido reportada en ningún artículo hasta el año de 1989, pero es el resultado de varias pruebas hechas a diferentes redes eléctricas.²

Debido a que los modelos de redes son reales y están diseñados para operar bajo ciertas características, el hecho de multiplicar las resistencias de cada línea de transmisión, altera sus características de ampicidad, de capacidad de transmisión de potencia eléctrica, de límite térmico y su relación R/X. Es por esto que algunos resultados se apartan considerablemente de la solución real de flujos de carga, en estos casos se considerará que el sistema con tales parámetros eléctricos diverge cuando los resultados obtenidos en la iteración sean muy lejanos a resultados reales.

² Anjan Bose. "A modification to the fast decoupled power flow for networks with high R/X ratios". IEEE Power Appl. Syst., Vol. 3 No. 2 May 1988, pag. 743.

Ahora analizaremos los resultados obtenidos en los dos casos en estudio.

El primer caso es un sistema sencillo de tres buses y tres líneas de transmisión, con el primer bus funcionando como Bus de Referencia y el segundo bus como bus de control de voltaje. Según la tabla de la Página 54 y la gráfica de la Página 55, el Método propuesto conserva un comportamiento similar al del Método Desacoplado Rápido hasta el caso en que las resistencias de las líneas de transmisión son multiplicadas por el factor 3.5. A partir del factor 4.0 el número de iteraciones es menor al del Desacoplado Rápido y con el factor de 4.5 la diferencia es del doble.

El Segundo caso es un sistema de seis buses y siete líneas de transmisión, con el primer bus funcionando como Bus de Referencia y el segundo bus como bus de control de voltaje. Según la tabla de la Página 66 y la gráfica de la Página 67, el Método propuesto conserva un comportamiento similar al del Método Desacoplado Rápido hasta el caso en que las resistencias de las líneas de transmisión son multiplicadas por el factor 2.0. Para el factor 2.5 y 3.0 la diferencia es considerable. Para el factor 3.5 la diferencia es de cuatro veces menor el número de iteraciones en el Método propuesto y a partir del factor 4.00 en adelante la diferencia ya es tan considerable que el Método Desacoplado Rápido ni siquiera alcanza a converger.

Todo este análisis nos permite concluir que:

- El Método Propuesto mantiene una exactitud con variación máxima del 5% con relación al Método Desacoplado Rápido, en las magnitudes de Voltaje y Angulo.
- A medida que el cociente R/X aumenta, el número de iteraciones en el Método Propuesto Disminuyen considerablemente.

- Para sistemas más complejos la diferencia en el número de iteraciones es más notoria, debido a que existen mayor número de líneas donde el cociente R/X alcanza el valor frontera de 0.9, punto en el cual el Método Desacoplado Rápido comienza a tener problemas.
- El Método Propuesto tiene un mayor rango de aplicación en Redes Eléctricas de Distribución, en las cuales las posibilidades de encontrar cocientes R/X es mayor que en las redes de Potencia.
- Debido a que el Método desacoplado Rápido es un algoritmo muy aplicado en el cálculo de Flujos de Potencia, se puede implementar este nuevo algoritmo en forma de rutina externa a éste, de tal manera que cuando una red Eléctrica contenga cocientes altos de R/X , el Método Desacoplado Rápido derive en forma lógica su control para que el cálculo sea realizado por el algoritmo Propuesto.

BIBLIOGRAFIA

Olle I. Elgerd. *"Electric Energy System Theory. An Introduction"*.

McGraw Hill Series in Electrical Engineering. 1983

B. Stott and O. Alsac. *"Fast Decoupled Load Flow"*. IEEE Trans. Power Appl. Syst., vol. PAS-93. May/June 1974, pp. 859-869

J. Hassall. *"Comparison of modern Load Flow Methods"* B. Sc. Project Report, UMIST, April 1972.

F.F. Wu, *"Theoretical Study of the Convergence of the Fast Decoupled Load Flow"*. IEEE Trans. Power Appl. Syst., vol. PAS-96. Jan/Feb 1977, pp. 268-275.

Anjan Bose. *"A modification to the fast decoupled power flow for networks with high R/X ratios"*. IEEE Power Appl. Syst., Vol. 3 No. 2 May 1988, pag. 743.

ANEXO 1

LISTADO DEL CODIGO DEL PROGRAMA

```
#define MAXIMO 50
#define ITERACION_MAX 100
#include <stdio.h>
#include <malloc.h>
#include <string.h>
#include <math.h>
/* -----
DEFINICION DE VARIABLES GLOBALES DEL PROGRAMA PRINCIPAL
----- */
struct complejo { double x[50];
                  double y[50];
                  };
int      n,n1,n2,no_lin;
char     TIPO[MAXIMO][6],lectura[15],divide;
float    cv;
double   B1[MAXIMO][MAXIMO];
double   B2[MAXIMO][MAXIMO];
double   B3[MAXIMO][MAXIMO];
double   B4[MAXIMO][MAXIMO];
struct complex R[MAXIMO][MAXIMO];
struct complex G[MAXIMO][MAXIMO];
struct complex Sp[MAXIMO];
struct complex Sr[MAXIMO];
struct complejo deltaS;
struct complejo deltaV;
struct complejo Vnuevo;
struct complejo Vviejo;
struct complejo VOLTAJE;
FILE *escribir,*fopen();
/* -----
DEFINICION Y SIGNIFICADO DE VARIABLES GLOBALES
----- */
n      = Numero de buses del SEP
n1     = Numero de buses de control de voltaje del SEP
n2     = Numero de buses de Carga del SEP
no_lin = Numero de Lineas del SEP
R      = estructura compleja formada por:
        Rbus -> Matriz de las resistencias de las lineas.
        Xbus -> Matriz de las reactancias de las lineas.
G      = Estructura compleja formada por:
        Gbus -> Matriz de conductancias de las lineas
        Bbus -> Matriz de Susceptancias de las lineas.
Sp     = Estructura compleja formada por:
        Pp -> Vector de valor calculado de la potencia real en el nodo p
        Qp -> Vector de valor calculado de la potencia reactiva en el nodo p
deltaS = Estructura Compleja de Incrementos de Potencia formada por:
        deltaP -> Incrementos de potencia Real
        deltaQ -> Incrementos de Potencia Reactiva.
deltaV = Estructura Compleja de Incrementos de Voltaje formada por:
```


deltaV -> Incrementos de Voltaje.
 deltaANG -> Incrementos de Angulo de voltaje.
 Sr = Estructura Compleja formada por:
 Pr -> valor inicial de la potencia real en el nodo p
 Qr -> valor inicial de la potencia reactiva en el nodo p
 Sg = Estructura Compleja para la Potencia aparente de "GENERACION" formada por:
 Pg -> Potencia real de Generacion
 Qg -> Potencia reactiva de Generacion
 Sc = Estructura Compleja para la Potencia aparente de "CARGA" formada por:
 Pc -> Potencia real de Carga
 Qc -> Potencia reactiva de Carga
 Vnuevo = Estructura compleja de Calculo de voltaje en cada bus formada por:
 Vnuevo -> Ultimo calculo de Voltaje en cada bus.
 ANGnuevo -> Ultimo calculo del angulo de voltaje en cada bus.
 Vviejo = Estructura compleja de Calculo de voltaje en cada bus formada por:
 Vviejo -> Calculo anterior del Voltaje en cada bus.
 ANGViejo -> Calculo anterior del Angulo de voltaje en cada bus.
 VOLTAJE= Estructura Compleja para leer los datos de voltaje y angulo en los buses
 VOLTAJE -> Contiene el Voltaje Inicial en cada Bus.
 ANG -> Contiene el angulo inicial en cada Bus.

=====*/

```

main(argc,argv)
int argc;
char *argv[ ];
{
  int i,j,n3,n4,l = 1,iteracion = 1,p;
  short salida,adicionales = 0;
  char imprime;
  float factor ,min,max;
  float output[10][4];
  double sumaP_Q[MAXIMO];
  struct complex Z[MAXIMO];
  struct complex Sg[MAXIMO];
  struct complex Sc[MAXIMO];
  void imp_voltajes(short);
/* int solucion(int,char);
void compensa(void);
void resultado(float); */

```

/* -----NOTA -----
 Se verifica que se hayan pasado el numero correcto de argumentos al programa principal :
 argc -> indica el numero de argumentos
 argv[] -> arreglo de apuntadores tipo caracter donde se guardan los argumentos que se pasan.
 -----*/

```

if(argc != 5) /* Se toma en cuenta que el nombre del ejecutable es un argumento */
  printf("\n\n Numero de Argumentos Invalido"),exit(0);
imprime = *(argv[1]);
min = atof(argv[2]); /* Valor Minimo para factor */
max = atof(argv[3]); /* Valor Maximo para factor */
cv = atof(argv[4]);

```

```

/*-----
                          Inicializacion de Arreglos y estructuras
-----*/

```

```

if ((escribir = fopen("salida.dat","w"))==NULL)
{
    printf("\n El archivo de salida no se pudo abrir. ");
    exit(0);
}

```

```

inicializa(Z,Sg,Sc,sumaP_Q);

```

```

for(i=0;i<10;i++)
    for(j=0;j<4;j++)
        output[i][j] = 0;

```

```

/*-----
                          A continuacion se Leen los datos de las lineas
-----*/

```

```

lee_datos(Z,Sg,Sc,'S'); /* lee los datos de lineas y buses */
fprintf(escribir,"\n\n-----\n");
fprintf(escribir," LOS DATOS QUE A CONTINUACION APARECEN, SON LOS
DATOS DE LA RED");
fprintf(escribir,"\n CUYAS RESISTENCIAS DE CADA LINEA, SON
MULTIPLICADAS");
fprintf(escribir,"\n POR EL FACTOR ESPECIFICADO");
fprintf(escribir,"\n-----\n");
esc_datos(Z,Sg,Sc); /* escribe los datos leidos */

```

```

/*-----
                          Inicializacion de las Estructuras Complejas:
                          Vnuevo,Vviejo,Sr
                          Los voltajes se inicializan al valor inicial del voltaje
                          leído de los buses, por lo que al arranque del programa
                          la diferencia Vnuevo - Vviejo = 0
-----*/

```

```

for(i = 1; i <= n; i++)
{
    Vnuevo.x[i] = VOLTAJE.x[i]; Vviejo.x[i] = VOLTAJE.x[i];
    Vnuevo.y[i] = VOLTAJE.y[i]; Vviejo.y[i] = VOLTAJE.y[i];
    Sr[i].x = Sg[i].x - Sc[i].x ;
    Sr[i].y = Sg[i].y - Sc[i].y ;
    /* Potencia reactiva = Potencia reactiva de generacion -
    Potencia real de carga */
}
factor = min;
do
{
    for(i=1;j<=n;i++)
        Z[i].x = Z[i].x * factor;
    output[1][1] = factor;
    fprintf(escribir,"\n\n-----\n");
    fprintf(escribir,"                          DATOS DE LA RED:");
    fprintf(escribir,"\n");
    fprintf(escribir,"\n FACTOR DE MULTIPLICACION : %.2f",factor);
    fprintf(escribir,"\n-----\n");
}

```

```

esc_datos(Z,Sg,Sc); /* escribe los datos leidos */
Zbus(Z,imprime);
Ybus(Z,imprime);
flujos(imprime,'S');
do
{
n4 = 2;
n3 = 2;
gauss(B1,deltaS.x,Vnuevo.y,Vviejo.y,deltaV.y,n3,n4,imprime);
n3 = n - n2 + 1;
n4 = n - n2 + 1;
gauss(B2,deltaS.y,Vnuevo.x,Vviejo.x,deltaV.x,n3,n4,imprime);
salida = solucion(&iteracion,"DESACOPLADO RAPIDO",factor);
if(salida == 1 || salida == -2)
break;
flujos(imprime,'S');
}while(++iteracion) < ITERACION_MAX);
imp_voltajes(iteracion);
output[1][2] = iteracion;
iteracion = 1;
for(i = 1; i <= n; i++)
{
Vnuevo.x[i] = VOLTAJE.x[i]; Vviejo.x[i] = VOLTAJE.x[i];
Vnuevo.y[i] = VOLTAJE.y[i]; Vviejo.y[i] = VOLTAJE.y[i];
Sr[i].x = Sg[i].x - Sc[i].x ;
Sr[i].y = Sg[i].y - Sc[i].y ;
}
flujos(imprime,'N');
do
{
n4 = 2;
n3 = 2;
gauss(B3,deltaS.x,Vnuevo.y,Vviejo.y,deltaV.y,n3,n4,imprime);
n3 = n - n2 + 1;
n4 = n - n2 + 1;
for(p=(n-n2+1);p<=n;p++)
sumaP_Q[p] = deltaS.x[p] + deltaS.y[p];
gauss(B4,sumaP_Q,Vnuevo.x,Vviejo.x,deltaV.x,n3,n4,imprime);
salida = solucion(&iteracion,"CON EL METODO PROPUESTO",factor);
if(salida == 1 || salida == -2)
break;
flujos(imprime,'N');
}while(++iteracion) < ITERACION_MAX);
imp_voltajes(iteracion);

for(i=1;i<=n;i++)
{
Vnuevo.x[i] = VOLTAJE.x[i]; Vviejo.x[i] = VOLTAJE.x[i];
Vnuevo.y[i] = VOLTAJE.y[i]; Vviejo.y[i] = VOLTAJE.y[i];
Sr[i].x = Sg[i].x - Sc[i].x; Sr[i].y = Sg[i].y - Sc[i].y;
}
for(i=1;i<=n;i++)
Z[i].x = Z[i].x / factor;
output[1][3] = iteracion;
factor = factor + 0.5;
I++;
iteracion = 1;
}while(factor <= max);

```

```

/* -----
SE IMPRIME A CONTINUACION LA TABLA COMPARATIVA DE RESULTADOS
----- */

resultado(output);

fclose(escribir); /* Cierra el archivo de salida de datos */
}

/* -----
Funcion esc_datos()
----- */

extern int a[];
extern int b[];
esc_datos(Z,Sg,Sc)
struct complex *Z;
struct complex *Sg;
struct complex *Sc;
{
int i,j;

/* ----- Definicion de variables: -----
n = Numero de buses en el sistema en estudio
no_lin = Numero de lineas del sistema en estudio
a[MAXIMO] = vector para guardar el numero de linea.
b[MAXIMO] = vector que indica el punto inicial de cada linea
c[MAXIMO] = vector que indica el punto final de cada linea
NOTA: LOS VECTORES A,B y C ESTAN RELACIONADOS ENTRE SI.
Zreal[maximo][maximo] = Matriz de la parte real de las impedancias de cada linea.
Zimag[maximo][maximo] = Matriz de la parte imaginaria de las impedancias de cada linea.
----- */

fprintf(escribir, "\n NUMERO DE BUSES: %2d NUMERO DE LINEAS: %2d ", n, no_lin);
fprintf(escribir, "\n\n LINEA No. ENTRE BUSES IMPEDANCIA DE LA LINEA");
for(i=1; i<=no_lin; i++)
{
fprintf(escribir, "\n %d \t %2d-%2d", i, a[i], b[i]);
fprintf(escribir, "\t %.3f J%.3f ", Z[i].x, Z[i].y);
}

fprintf(escribir, "\n\n DATOS DE LOS BUSES : \n");
fprintf(escribir, "\n\n BUS No. TIPO GENERACION CARGA VOLTAJE ");
fprintf(escribir, "\n\n ----- ");
fprintf(escribir, "\n P Q P Q |V| ang. \n ");
for(i=1; i<=n; i++)
{
fprintf(escribir, "\n %2d %s %.4lf %.4lf", i, TIPO[i], Sg[i].x, Sg[i].y);
fprintf(escribir, " %.4lf %.4lf", Sc[i].x, Sc[i].y);
fprintf(escribir, " %.4lf %.4lf", VOLTAJE.x[i], VOLTAJE.y[i]);
}
}
}

```

```
/* -----  
Funcion Inicializa. Se utiliza para Inicializar todas las estructuras, vectores y Matrices Usadas en la  
solucion de Sistema  
----- */
```

```
inicializa(Z,Sg,Sc,sumaP_Q)  
struct complex *Z,*Sg,*Sc;  
double sumaP_Q[];  
{  
  int i,j;  
  for(i=0;i<MAXIMO;i++)  
  {  
    Z[i].x = 0;  
    Z[i].y = 0;  
    Sg[i].x = 0;  
    Sg[i].y = 0;  
    Sc[i].x = 0;  
    Sc[i].y = 0;  
    VOLTAJE.x[i] = 0; VOLTAJE.y[i] = 0;  
    TIPO[i][1] = '\0' ;  
    Sp[i].x = 0; Sp[i].y = 0;  
    deltaS.x[i] = 0; deltaS.y[i] = 0;  
    deltaV.x[i] = 0; deltaV.y[i] = 0;  
    Vnuevo.x[i] = 0; Vnuevo.y[i] = 0;  
    Vviejo.x[i] = 0; Vviejo.y[i] = 0;  
    Sr[i].x = 0; Sr[i].y = 0;  
    sumaP_Q[i] = 0;  
    for( j = 0; j<MAXIMO ;j++)  
    {  
      R[i][j].x = 0; R[i][j].y = 0;  
      G[i][j].x = 0; G[i][j].y = 0;  
      B1[i][j] = 0;  
      B2[i][j] = 0;  
      B3[i][j] = 0;  
      B4[i][j] = 0;  
    }  
  }  
}
```

```

/* -----
Funcion Solucion. Va calculando la diferencia entre el valor de voltaje calculado y el valor anterior,
lo mismo hace para los valores de angulos.
Cuando la diferencia de cada elemento del arreglo de la estructura Vnuevo - Vviejo = 0, la funcion
regresa el valor de 1 (Solucion encontrada), de lo contrario regresa el valor de 0 (No hay solucion en
la iteracion).
** Si el numero de iteraciones alcanzo el numero maximo de iteraciones
regresa el valor -1 (El sistema diverge)
----- */

```

```

float cv;
int solucion(iteracion,sistema,factor)
int *iteracion;
float factor;
char sistema[ ];
{
    double sol1,sol2;
    int i,p,solucion = 0;
    if(*iteracion == ITERACION_MAX - 1)
    {
        fprintf(escribir,"\n\n\n El SISTEMA %s, CON FACTOR %1.1f, DIVERGE DESPUES DE %d
ITERACIONES",sistema,factor,*iteracion);
        return(-1);
    }
    for(i = 1;i <= n; i++)
    {
        sol1 = fabs((fabs(Vnuevo.x[i]) - fabs(Vviejo.x[i])));
        sol2 = fabs((fabs(Vnuevo.y[i]) - fabs(Vviejo.y[i])));
        if(sol1 <= cv && sol2 <= cv && i == n)
        {
            fprintf(escribir,"\n\n\n EL SISTEMA %s, CON FACTOR %1.1f, CONVERGE EN LA %da.
ITERACION",sistema,factor,*iteracion);
            solucion = 1;
            break;
        }
        else
            if((fabs(Vnuevo.x[i] > 10.0) || fabs(Vnuevo.y[i] > 1000000))
            {
                fprintf(escribir,"\n\n\n El PROGRAMA se revienta DESPUES DE %d
ITERACIONES",*iteracion);
                solucion = -2;
                *iteracion = 100;
                break;
            }
        else
            solucion = 0;
    }
    for(i = 1;i <= n;i++)
    {
        Vviejo.x[i] = Vnuevo.x[i];
        Vviejo.y[i] = Vnuevo.y[i];
    }
    return(solucion);
}

```

```

/* -----
Funcion Void que Imprime los resultados de Voltaje y Angulo despues de Iterar hasta obtener la
diferencia entre los valores viejos y nuevos de cv (0.001).
----- */

```

```

void imp_voltajes(iteracion)
short iteracion;
{
    int p;
    fprintf(escribir, "\n\n");
    fprintf(escribir, "\n LOS RESULTADOS DESPUES DE %d ITERACIONES SON:\n", iteracion);
    fprintf(escribir, "\n BUS          VOLTAJE          ANGULO(rad) \n");
    for(p=1; p<=n; p++)
        fprintf(escribir, "\n %d          %4f          %4f", p, Vnuevo.x[p], Vnuevo.y[p]);
}

```

```

/* -----
                                FUNCION RESULTADOS
Función que imprime la tabla donde se guardaron los resultados del número de iteraciones de los dos
mÉtodos:
- Desacoplado Rápido
- Algoritmo Propuesto.
----- */

```

```

resultado(output)
float output[ ][4];
{
    int j,i;
}
/* -----
                                encabezados de salida
----- */

```

```

fprintf(escribir, "\n\n");
fprintf(escribir, "\n\n Número de iteraciones cuando todas las resistencias \n");
fprintf(escribir, "\n de las lineas son multiplicadas por el factor : \n");
fprintf(escribir, "\n");

```

```

/* -----
                                impresion de la matriz de resultados
----- */

```

```

fprintf(escribir, "factor ");
for(i=1; i<=8; i++)
    fprintf(escribir, " %2f", output[i][1]);
fprintf(escribir, "\n");
fprintf(escribir, "Des. Rápido ");
for(i=1; i<=8; i++)
    if(output[i][2]>=99)
        fprintf(escribir, " div");
    else
        fprintf(escribir, " %.1f", output[i][2]);
fprintf(escribir, "\n");
fprintf(escribir, "Propuesto ");
for(i=1; i<=8; i++)
    if(output[i][3]>=50)
        fprintf(escribir, " div");
    else
        fprintf(escribir, " %.1f", output[i][3]);

```

```

printf(escribir, "\n _____ \n");
}

```

FUNCION lee_datos()

estructuras definidas:

Z-> Estructura compleja para los parametros electricos de la linea,

 Z.x-> parte real e indica el valor de la resistencia

 Z.y-> parte imaginaria e indica el valor de la reactancia.

Sg-> Potencia aparente generada,

 Sg.x-> indica la potencia real generada

 Sg.y-> indica la potencia reactiva generada.

Sc-> Potencia aparente de carga,

 Sc.x-> indica la potencia real de carga

 Sc.y-> indica la potencia reactiva de carga.

VOLTAJE-> Estructura compleja para los voltajes en los buses,

 VOLTAJE.x-> Indica el modulo del voltaje

 VOLTAJE.y-> Indica el angulo del voltaje

```

#define MAXIMO 50
#include <math.h>
#include <stdio.h>
struct complejo { double x[50];
                 double y[50];
};
char TIPO[6];
int a[MAXIMO];
int b[MAXIMO];
struct complejo VOLTAJE;
int n,n1,n2,no_lin;
char lectura[15];
lee_datos(Z,Sg,Sc,abre)
struct complex *Z,*Sg,*Sc;
char abre;
{
  FILE *leer,*fopen(); /* Declara las variables que abren el archivo de datos*/
  int k,i;

```

Lectura del archivo de Datos

```

if(abre == 'S')
{
  printf("\n Teclee el nombre del archivo de datos (Maximo 15 caracteres) ");
  do
  {
    scanf("%s",lectura);
    if ((leer=fopen(lectura,"r"))==NULL)
    {
      printf("\n El archivo de datos %s no existe. ",lectura);
      printf(" Intente de Nuevo");
    }
  }while ((leer=fopen(lectura,"r"))==NULL);
}
else
  leer = fopen(lectura,"r");
fscanf(leer,"%d",&n);
fscanf(leer,"%d",&n1);
fscanf(leer,"%d",&n2);

```



```

fscanf(Leer, "%d", &no_lin);

/*-----
A continuacion se leen los datos de las lineas
-----*/
for(i=1; i<=no_lin; i++)
fscanf(Leer, "%d %d %lf %lf", &a[i], &b[i], &(Z[i].x), &(Z[i].y));
/*-----
A continuacion se leen los datos de los Buses
-----*/
for(i=1; i<=n; i++)
{
fscanf(Leer, "%s %lf %lf %lf ", &TIPO[i], &Sg[i].x, &Sg[i].y, &Sc[i].x);
fscanf(Leer, "%lf %lf %lf", &Sc[i].y, &VOLTAJE.x[i], &VOLTAJE.y[i]);
}
fclose(Leer);
}

```

```

/*-----
RUTINA PARA LA FORMACION DE YBUS A PARTIR DE LOS PARAMETROS ELECTRICOS
DE LAS LINEAS
-----*/

```

```

#include <math.h>
int n;
int no_lin;
int n1;
int n2;
int a[50],b[50];
struct complex R[50][50];
struct complex G[50][50];
Zbus(Z,imprime)
struct complex *Z;
char imprime;
{
    double r;
    double x;
    int i,j,k,l;

```

```

/*-----
INICIALIZACION DE LAS MATRICES RBUS, XBUS, GBUS, BBUS. ESTO ES PARA
QUE SUS VALORES NO SE UTILICEN EN LA RUTINA, YA QUE SON ACUMULATIVOS
-----*/

```

```

for(i=1;i<=n;i++)
    for(j=1;j<=n;j++)
    {
        R[i][j].x=0;R[i][j].y=0;
        G[i][j].x=0;G[i][j].y=0;
    }
for(i=1;i<=n;i++)
    {
        for(j=1;j<=n;j++)
        {
            if(i==j)
            {
                for(k=1;k<=no_lin;k++)
                    if((a[k]==i) || (b[k]==j))
                    {
                        R[i][j].x = R[i][j].x + Z[k].x;
                        R[i][j].y = R[i][j].y + Z[k].y;
                        invierte(&r,&x,&k,Z);
                        if(Z[k].x == 0)
                            r = 0;
                        if(Z[k].y == 0)
                            x = 0;
                        G[i][j].x = G[i][j].x + r;
                        G[i][j].y = G[i][j].y + x;
                    }
            }
        }
    }
else
    {
        if(i != j)
            for(k = 1; k <= no_lin; k++)
                if(((a[k] == i) || (a[k] == j)) && ((b[k] == j) || (b[k] == i)))
                {
                    R[i][j].x = Z[k].x;

```

```

        R[i][j].y = Z[k].y;
        invierte(&r,&x,k,Z);
        if(Z[k].y == 0)
            r = 0;
        if(Z[k].y == 0)
            x = 0;
        G[i][j].x = -r;
        G[i][j].y = -x;
        break;
    }
}
}
}
if(imprime == 'S')
{
    printf("\n\n LA MATRIZ ZBUS ES:\n");
    for(i=1;i<=n;i++)
    {
        for(j=1;j<=n;j++)
            printf(" %%.3f  j%.3f ",R[i][j].x,R[i][j].y);
        printf("\n");
    }
}
}
}

```

FUNCION PARA ENCONTRAR EL INVERSO DE UN NUMERO COMPLEJO, EL CUAL SERA UTILIZADO PARA FORMAR YBUS

```

/*-----*/
#define pi 3.1416
invierter(r,x,i,Z)
struct complex *Z;
double *r ;
double *x ;
int i;
{
    double y,h,z;
    double cos(double x);
    double sin(double x);
    double atan(double x);
    *r = 0; *x = 0;
    if(Z[i].x == 0 && Z[i].y == 0)
    {
        y = 0;
        h = 0;
    }
    else
    if(Z[i].x==0 && Z[i].y < 0)
    {
        y = -pi / 2;
        h = -90;
    }
    else
    if(Z[i].x==0 && Z[i].y > 0)
    {
        y = pi / 2;
        h = 90;
    }
}

```

```

    }
else
{
    y = atan(Z[i].y / Z[i].x);
    h = (y * 180) / pi;
    if(Z[i].x < 0 && Z[i].y > 0)
    {
        y = ((pi / 2) + y) + pi / 2;
        h = 180 - h;
    }
else
    if(Z[i].x < 0 && Z[i].y < 0)
    {
        y = pi + y;
        h = h + 180;
    }
}
z = sqrt(Z[i].y * Z[i].y + Z[i].x * Z[i].x);
if(z > 0)
{
    z = 1 / z;
    y = -y;
    *r = z * cos(y);
    *x = z * sin(y);
}
else
    if(z == 0)
    {
        *r = 0;
        *x = 0;
    }
}
}

```

```

/*-----
  RUTINA PARA GENERACION DE LAS MATRICES B" Y B', LAS CUALES RESULTAN DE
  MANIPULAR LA ESTRUCTURA Gbus
  -----*/

```

```

#include<stdio.h>
#include<math.h>
int n1;
int n2;
no_lin;
int n;
struct complex R[50][50];
struct complex G[50][50];
double B1[50][50];
double B2[50][50];
double B3[50][50];
double B4[50][50];
Ybus(Z,imprime)
struct complex *Z;
char imprime;
{
  int i,j,k=1,l=1;

```

```

/*-----
  INICIA FORMACION DE YBUS A PARTIR DE ZBUS, CON SIMPLE INVERSION DE CADA
  ELEMENTO DE ESTA.
  -----*/

```

```

if(imprime == 'S')
{
  printf("\n La matriz Ybus es: \n\n");
  for(i=1;i<=n;i++)
  {
    for(j=1;j<=n;j++)
      printf(" %.2f %.2f",G[i][j].x,G[i][j].y);
    printf("\n");
  }
}
for(i=1;i<=n;i++)
for(j=1;j<=n;j++)
{
  B1[i][j]=0;B2[i][j]=0;
  B3[i][j]=0;B4[i][j]=0;
}
if(imprime == 'S')
{
  printf("\n");
  printf("\n La matriz [-Bbus] es :\n");
  for(i=1;i<=n;i++)
  {
    for(j=1;j<=n;j++)
      printf(" %.4f",-G[i][j].y);
    printf("\n");
  }
}
for(i=1;i<=n;i++)
{
  for(j=1;j<=n;j++)

```

```

if(i==j)
    continue;
if (G[i][j].y == 0)
{
    B1[i][j] = - G[i][j].y;
    B3[i][j] = - G[i][j].y;
}
else
{
    B1[i][j] = - G[i][j].y;
    B3[i][j] = - G[i][j].y-0.4*G[i][j].x-
        0.3*(G[i][j].x*G[i][j].x)/G[i][j].y;
}
}
}
/* i = n - n1 - n2 + 1*/
for(i=1;i<=n;i++)
{
    for(j=1;j<=n;j++)
    {
        if(i == j)
            continue;
        B1[i][i] = - G[i][i].y;
        B3[i][i] = B3[i][i] - B3[i][j];
    }
}
for(i=1;i<=n;i++)
{
    for(j=1;j<=n;j++)
    {
        B2[i][j] = -G[i][j].y;
        B4[i][j] = G[i][j].x - G[i][j].y;
    }
}
if(imprime == 'S')
{
    printf("\n\n La matriz B' Para MDR modificada es :\n");
    for(j=(n-n1-n2+1);j<=n;j++){
        for(i=(n-n1-n2+1);i<=n;i++)
            printf(" %.4f ",B1[j][i]);
        printf("\n");}
    printf("\n\n La matriz B' Para el algoritmo propuesto es :\n");
    for(j=(n-n1-n2+1);j<=n;j++){
        for(i=(n-n1-n2+1);i<=n;i++)
            printf(" %.4f ",B3[j][i]);
        printf("\n");}
    printf("\n\n La matriz B'' Para MDR modificada es :\n");
    for(i=(n-n2+1);i<=n;i++)
    {
        for(j=(n-n2+1);j<=n;j++)
            printf(" %.4f",B2[i][j]);
        printf("\n");
    }
    printf("\n\n La matriz B''' Para el algoritmo propuesto es :\n ");
    for(i=(n-n2+1);i<=n;i++){
        for(j=(n-n2+1);j<=n;j++)
            printf(" %.4f",B4[i][j]);
        printf("\n");}
}

```

**FUNCION FLUJOS, PARA ENCONTRAR LOS FLUJOS DE CARGA EN UN SISTEMA
 ELECTRICO DE POTENCIA CON ALTOS COCIENTES R/X**

```

}
}
/*
#include <stdio.h>
#include <math.h>
struct complejo { double x[50];
                 double y[50];
                 };
int n,n2;
struct complejo Vvicio;
struct complex Sp[50];
struct complejo deltaS;
struct complex G[50][50];
struct complex Sr[50];
flujos(imprime,divide)
char imprime,divide;
{
    int i,j,p,q,k=0,temp;
    double sumatoriap=0,sumatoriapq=0;
    double y,z,x;
    double sin(double x);
    double cos(double x);
    /*=====
    n1=# de buses P-V
    n2=# de buses P-Q
    n=# total de buses (n=n1+n2+SLACK)
    Pp=vector para guardar los calculos de las potencias reales calculadas
    deltaP=vector de incrementos de potencia
    ANGnuevo=vector que contiene los angulos
    deltaANG=vector con los incrementos de los angulos
    =====*/
    k=0;
    for(p=2;p<=n;p++)
    {
        for(q=1;q<=n;q++)
        {
            x=Vvicio.y[p] - Vvicio.y[q];
            sumatoriap=sumatoriap+(G[p][q].x*cos(x)+G[p][q].y*sin(x))*Vvicio.x[q] ;
            if(imprime == 'S')
            {
                printf("\n x=%f",x);
                printf("\n Gbus[%d][%d]=%f cos(x)=%f\n",p,q,G[p][q].x,cos(x));
                printf("\n sin(x)=%f Vvicio[%d]=%f sumatoriap=%f",sin(x),q,Vvicio.x[q],sumatoriap);
            }
        }
        Sp[p].x = Vvicio.x[p] * sumatoriap;
        sumatoriap = 0;
        k++;
        if (divide == 'S')
            deltaS.x[p] = (Sr[p].x - Sp[p].x)/Vvicio.x[p];
        else
            deltaS.x[p] = Sr[p].x - Sp[p].x;
        if(imprime == 'S')
        {
    
```

```

printf("\n Vviejo[%d]=%f Pp[%d]=%f",p,Vviejo.x[p],p,Sp[p].x,p,Sr[p].x);
printf(" Pr[%d]=%f deltaP[%d]=%f \n\n",p,Sr[p].x,p,deltaS.x[p]);
}

```

```

k=0;

```

```

/*-----
CALCULO DE LOS VALORES INICIALES DE Q PARA EL CALCULO DE LOS VOLTAJES EN
LOS BUSES. SOLO SE USAN LOS BUSES DE CARGA.
-----*/

```

```

/*-----
INICIA SOLUCION DEL SISTEMA PARA ENCONTRAR LOS ANGULOS Y VOLTAJES EN
CADA BUS.
-----*/

```

```

for(p=(n-n2+1);p<=n;p++)
{
for(q=1;q<=n;q++)
{
x = Vviejo.y[p] - Vviejo.y[q];
sumatoriaq=sumatoriaq+(G[p][q].x*sin(x)-G[p][q].y*cos(x))*Vviejo.x[q];
if(imprime == 'S')
{
printf("\nGbus[%d][%d]=%f sin(x)=%f Bbus[%d][%d]=%f",p,q,G[p][q].x,sin(x),p,q,
G[p][q].y);
printf(" \n cos(x)=%f Vviejo[%d]=%f sumatoriaq=%f",cos(x),q,Vviejo.x[q],sumatoriaq);
}
}
if(imprime == 'S')
printf("\n\n");
Sp[p].y = Vviejo.x[p] * sumatoriaq;
sumatoriaq=0;
k=k+1;
if (divide == 'S')
deltaS.y[p] = (Sr[p].y - Sp[p].y)/Vviejo.x[p];
else
deltaS.y[p] = Sr[p].y - Sp[p].y;
if(imprime == 'S')
{
printf("\n Qr[%d]=%f Qp[%d]=%f \n",p,Sr[p].y,p,Sp[p].y);
printf(" Vviejo[%d]=%f deltaQ[%d]=%f\n",p,Vviejo.x[p],p,deltaS.y[p]);
}
}
sumatoriaq=0;
}

```



```
-----  
/*-----  
FUNCION GAUSS.C PARA ENCONTRAR LA SOLUCION DE UN SISTEMA DE  
ECUACIONES, MEDIANTE EL METODO DE ELIMINACION DE GAUSS-SEIDEL  
-----*/
```

```
int n1;  
int n2;  
int n;  
gauss(a,b,xnuevo,xviejo,deltax,n3,n4,imprime)  
double a[ ][50],b[ ],xnucvo[ ],xviejo[ ],deltax[ ];  
int n3,n4;  
char imprime;  
{  
float c[30][30],temp,pivote=0;  
float x[50],suma=0;  
int j,i,k=1,l,z;  
-----*/
```

```
-----  
INICIALIZACION DE LA MATRIZ DE PASO C  
-----*/
```

```
for(i=0;i<=n;i++)  
for(j=0;j<=n;j++)  
c[i][j]=0;
```

```
-----  
SE PASA LA MATRIZ A LA MATRIZ C, DEBIDO A QUE A SE MODIFICA AL APLICAR LA  
ELIMINACION DE GAUSS  
-----*/
```

```
for(i=n4;i<=n;i++)  
for(j=n4;j<=n;j++)  
c[i][j]=a[i][j];  
if(imprime == 'S')  
{  
printf( "\n la lista original copiada es:\n");  
for (i=n4;i<=n;i++)  
{  
printf(" %.3f ",b[i]);  
for(j=n4;j<=n;j++)  
printf(" %.3f ",c[i][j]);  
printf(" %f",deltax[i]);  
printf("\n");  
}  
}  
-----*/
```

```
-----  
PASAR LA MATRIZ A A LA MATRIZ B, DEBIDO A QUE LOS RENGLONES DE A NO  
COINCIDEN CON EL VECTOR DELTAX  
-----*/
```

```
for(i=n4;i<=n;i++)  
{  
for(j=n4;j<=n;j++)  
{  
if(i==j)  
{  
if(c[i][i]==0)  
{  
k=i+1;  
while(c[k][i]==0)  
k++;  
-----*/
```

```

temp=b[k];
b[k]=b[i];
b[i]=temp;
for(l=1;l<=n3;l++)
{
temp=c[i][l];
c[i][l]=c[k][l];
c[k][l]=temp;
} /* fin del for que intercambia renglones de la matriz */
} /* fin del for a[i][i]==0 */
break;
} /* fin del if i==j */
}
*/

```

COMIENZA LA ELIMINACION DE GAUSS

```

for(k=i+1;k<=n;k++)
{
pivote=c[k][j]/c[i][j]; b[k]=pivote*b[i] - b[k];
for(l=j;l<=n;l++)
{
c[k][l]=pivote*c[i][l] - c[k][l];
}
} /* fin del for principal (i) */
/*=====TERMINA ELIMINACION DE GAUSS=====*/
if(imprime == 'S')
{
printf("\n la matriz reducida es: \n");
for(i=n4;i<=n;i++)
{
printf("%f ",b[i]);
for(j=n4;j<=n;j++)
printf(" %f",c[i][j]);
printf("\n");
}
printf("\n\n");
}
for(i=n4;i<=n;i++)
{
for(j=n4;j<=n;j++)
deltax[i]=0;
}
l=n;
*/

```

A CONTINUACION SE OBTIENEN LOS RESULTADOS DE LAS VARIABLE NUEVAS

```

for(i=n;i>=n3;i--)
{
for(j=n;j>n3;j--)
{
suma = suma + deltax[j] * c[i][j];
}
deltax[i]=(-suma + b[i])/c[i][i];
xnuevo[i]= xviejo[i] + deltax[i];
suma=0; l--;
}
}

```