



INSTITUTO POLITÉCNICO NACIONAL

ESCUELA SUPERIOR DE INGENIERÍA MECÁNICA Y ELÉCTRICA
SECCIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN

“SOLUCIÓN COORDINADA DE ESTUDIO DE FLUJOS DE POTENCIA ENTRE REDES ELÉCTRICAS DE ALTA Y MEDIA TENSIÓN”

TESIS

QUE PARA OBTENER EL GRADO DE:
MAESTRO EN CIENCIAS EN INGENIERÍA ELÉCTRICA

PRESENTA:
DANIEL MARTÍNEZ PÉREZ.



MEXICO D.F.

2015



INSTITUTO POLITÉCNICO NACIONAL
SECRETARÍA DE INVESTIGACIÓN Y POSGRADO
ACTA DE REVISIÓN DE TESIS

En la Ciudad de México, D. F. siendo las 14:00 horas del día 8 del mes de Junio del 2015 se reunieron los miembros de la Comisión Revisora de la Tesis, designada por el Colegio de Profesores de Estudios de Posgrado e Investigación de E.S.I.M.E. - Zacatenco para examinar la tesis titulada:

SOLUCIÓN COORDINADA DE ESTUDIOS DE FLUJOS DE POTENCIA
ENTRE REDES ELÉCTRICAS DE ALTA Y MEDIA TENSIÓN

Presentada por el alumno:

MARTÍNEZ
Apellido paterno

PÉREZ
Apellido materno

DANIEL

Nombre(s)

Con registro:

A	1	3	0	4	7	3
---	---	---	---	---	---	---

aspirante de:

MAESTRO EN CIENCIAS EN INGENIERÍA ELÉCTRICA

Después de intercambiar opiniones, los miembros de la Comisión manifestaron **APROBAR LA TESIS**, en virtud de que satisface los requisitos señalados por las disposiciones reglamentarias vigentes.

LA COMISIÓN REVISORA

Director(a) de tesis


DR. RICARDO OCTAVIO MOTA PALOMINO

Presidente


DR. DANIEL OLGUÍN SALINAS

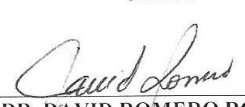
Tercer Vocal


DR. GERMAN ROSAS ORTIZ

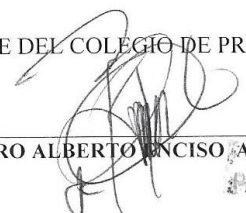
Segundo Vocal

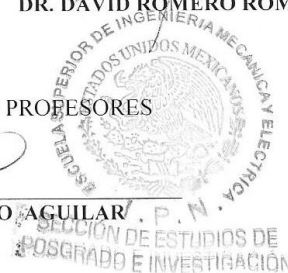

DR. DAVID SEBASTIÁN BALTAZAR

Secretario


DR. DAVID ROMERO ROMERO

PRESIDENTE DEL COLEGIO DE PROFESORES


DR. MAURO ALBERTO ENCISO AGUILAR





INSTITUTO POLITÉCNICO NACIONAL

SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

CARTA CESIÓN DE DERECHOS

En la Ciudad de México, D.F. el día 08 del mes de Junio del año 2015, el que suscribe **Daniel Martínez Pérez**, alumno del Programa de Maestría en Ciencias en Ingeniería Eléctrica, con número de registro A130473, adscrito a la Sección de Estudios de Posgrado e Investigación de la ESIME-Zacatenco del IPN, manifiesta que es el autor intelectual del presente trabajo de Tesis bajo la dirección del Doctor Ricardo Octavio Mota Palomino y cede los derechos del trabajo titulado Solución coordinada de estudios de flujos de potencia entre redes eléctricas de alta y media tensión, al Instituto Politécnico Nacional para su difusión, con fines académicos y de investigación.

Los usuarios de la información no deben reproducir el contenido textual, gráficas o datos del trabajo sin el permiso expreso del autor y/o directores del trabajo. Este puede ser obtenido escribiendo a las siguientes direcciones daniel-martinez-p@hotmail.com y/o rmotap@ipn.mx Si el permiso se otorga, el usuario deberá dar el agradecimiento correspondiente y citar la fuente del mismo.

Daniel Martínez Pérez

RESUMEN

En este trabajo se realiza el acoplamiento de dos sistemas eléctricos diferentes desde el punto de vista del modelaje de elementos y método de solución de flujos de potencia, siendo este uno de los estudios de mayor importancia para el análisis del sistema en operación y planeación de la expansión del sistema, ya que este estudio permite obtener voltajes de barra, flujos de potencia activa y reactiva y corriente en elementos, pérdidas en líneas y transformadores, y en general el estado en el que se encuentra la red. El principal problema para poder generar un algoritmo para la solución de flujos de potencia para redes de alta y media tensión fue, que tanto el sistema de alta como el de media tensión se deben modelar de diferente forma debido a las características que presenta cada uno de ellos, ya que por un lado el sistema de alta tensión es modelado como un sistema balanceado, y es posible hacer una representación monofásica de secuencia positiva del sistema, mientras que por otro lado el sistema de media tensión es modelado como un sistema desbalanceado, debido a las diferentes configuraciones (topología de la red) y de la carga y por lo cual se hace necesario una representación en coordenadas de fase.

La solución a dicho problema fue el desarrollar un algoritmo implementado en lenguaje Fortran 90 capaz de conservar las características de cada sistema (modelado y método numérico para la solución de flujos de potencia). El criterio de convergencia establecido fue la igualdad del voltaje del nodo raíz (voltaje del lado de baja tensión del transformador de Alta/Media tensión), ya que ante cualquier cambio de magnitud del voltaje del nodo raíz, la red de media tensión sufre un cambio en inyección de potencia activa y reactiva, debido a esto, en cada iteración el voltaje del nodo raíz de la red de media tensión es actualizado con base a los resultados de flujos de potencia de la red de alta tensión, pero también de forma recíproca es necesario conocer la carga que se encuentra conectada en los nodos de alta tensión, para lo cual la solución de flujos de potencia del sistema de media tensión proporcionan dichos valores, los cuales son actualizados en cada iteración para la solución de flujos de potencia del sistema de alta tensión.

ABSTRACT

In this thesis work is the coupling of two different electrical systems from the point of view of modeling elements and method of power flow solution, being one of the studies of greater importance for the analysis of the system operation and planning of the expansion of the system, since this study allows obtaining bar voltages , flows of active and reactive and running power elements, losses in lines and transformers, and in general in which the status of the network. The main problem to be able to generate an algorithm for the solution of flows of power for networks of high and medium voltage was, that both the system of high and medium voltage should be modeled differently because of characteristics which presents each one of them, because on the one hand the system is high voltage is modelled as a balanced system , and it is possible to make a single phase of positive sequence representation, while on the other hand medium voltage system is modeled as a system unbalanced, due to the different configurations (network topology) and the load and so it is necessary a representation in phase coordinates.

The solution to this problem was to develop an algorithm implemented in language Fortran 90 able to conserve the features of each system (modeling and numerical method for the solution of power flows). The convergence criteria established was the equality of the node voltage root (voltage high/medium voltage transformer low voltage side), since before any change of the magnitude of the voltage of the root node, the medium voltage network undergoes a change in power injection active and reactive, because of this, at each iteration the root node of the medium voltage network voltage is updated based the results of power flows the network of high voltage, but also mutual form, it is necessary to know the load which is connected in high voltage nodes, for which system of medium voltage power flow solution provide these values, which are updated at each iteration for solving system of high-voltage power flows.

ÍNDICE GENERAL

RESUMEN	IV
ABSTRACT	V
ÍNDICE GENERAL	VI
ÍNDICE DE FIGURAS	VIII
ÍNDICE DE TABLAS	IX
ÍNDICE DE TÉRMINOS	X
CAPÍTULO 1 INTRODUCCIÓN	1
1.1 OBJETIVO	1
1.2 JUSTIFICACIÓN	1
1.3 ESTADO DEL ARTE	2
1.4 APORTACIONES	3
1.5 CONTENIDO DE TESIS	4
CAPÍTULO 2 MODELADO DE REDES DE ALTA Y MEDIA TENSIÓN	5
2.1 MODELO DE REDES DE MEDIA TENSIÓN	5
2.1.1 MODELO DE LA LÍNEA DE DISTRIBUCIÓN	5
2.1.2 MODELOS DE CARGA	7
2.1.2.1 CARGAS CONECTADAS EN ESTRELLA	7
2.1.2.1.1 CARGAS CONECTADAS EN ESTRELLA A POTENCIA ACTIVA Y REACTIVA	8
CONSTANTE	
2.1.2.1.2 CARGAS CONECTADAS EN ESTRELLA A IMPEDANCIA CONSTANTE	8
2.1.2.1.3 CARGAS CONECTADAS EN ESTRELLA A CORRIENTE CONSTANTE	9
2.1.2.2 CARGAS CONECTADAS EN DELTA	9
2.1.2.2.1 CARGAS CONECTADAS EN DELTA A POTENCIA ACTIVA Y REACTIVA	10
CONSTANTE	
2.1.2.2.2 CARGAS CONECTADAS EN DELTA A IMPEDANCIA CONSTANTE	10
2.1.2.2.3 CARGAS CONECTADAS EN DELTA A CORRIENTE CONSTANTE	11
2.1.3 MODELOS DEL TRANSFORMADOR TRIFÁSICO	12
2.1.4 MODELO DE BANCO DE CAPACITORES	13
2.1.5 MODELO DE REGULADORES DE VOLTAJE	14
2.2 SOLUCIÓN DE FLUJOS DE POTENCIA EN SISTEMAS DE MEDIA TENSIÓN	16
2.2.1 SOLUCIÓN DE FLUJOS DE POTENCIA PARA REDES DE DISTRIBUCIÓN RADIAL	17
2.3 MODELO DE REDES DE ALTA TENSIÓN	20
2.3.1 REPRESENTACIÓN DE ELEMENTOS DE RED AL ALTA TENSIÓN	21
2.4 MODELO MATEMÁTICO PARA FLUJOS DE POTENCIA EN REDES DE ALTA TENSIÓN	22
2.4.1 ALGORITMO DE NEWTON-RAPHSON DESACOPLOADO RÁPIDO PARA LA SOLUCIÓN	24
DE FLUJOS DE POTENCIA	
CAPÍTULO 3 ESTUDIO DE FLUJOS DE POTENCIA	27
3.1 ETAPAS DE UN ESTUDIO DE FLUJOS DE POTENCIA	27

3.2 CARACTERÍSTICAS DEL SISTEMA DE ALTA Y MEDIA TENSIÓN	28
3.3 DESVENTAJAS DEL MÉTODO DE SOLUCIÓN DE FLUJOS DE POTENCIA DE SISTEMA DE ALTA TENSIÓN SOBRE SISTEMAS DE MEDIA TENSIÓN	29
3.4 METODOLOGÍA PARA EL ACOPLAMIENTO DEL SISTEMA DE MEDIA Y ALTA TENSIÓN	29
3.5 CRITERIOS PARA ENLACE DE SISTEMAS DE ALTA Y MEDIA TENSIÓN	30
3.5.1 COMPONENTES SIMÉTRICAS	30
3.5.2 TRANSFORMADOR DE DISTRIBUCIÓN	35
3.5.3 DESBALANCE DE TENSIÓN	39
CAPÍTULO 4 RESULTADOS	45
4.1 PRUEBA Y VALIDACIÓN DE FLUJOS DE POTENCIA DE REDES DE MEDIA TENSIÓN	45
4.1.1 VALIDACIÓN DE RED DE MEDIA TENSIÓN BALANCEADA	45
4.1.2 VALIDACIÓN DE RED DE MEDIA TENSIÓN DESBALANCEADA	47
4.2 PRUEBA Y VALIDACIÓN DE FLUJOS DE POTENCIA DE REDES DE ALTA TENSIÓN	49
4.3 SOLUCIÓN DE FLUJOS DE POTENCIA DE REDES DE DISTRIBUCIÓN DE ALTA Y MEDIA TENSIÓN A TRAVÉS DEL PROGRAMA COMPUTACIONAL ACOPLAMIENTO.FOR	50
4.3.1 CASO DE ESTUDIO 1: ACOPLAMIENTO DE UNA RED DE DISTRIBUCIÓN A LA RED DE TRANSMISIÓN	50
4.3.2 CASO DE ESTUDIO 2: ACOPLAMIENTO DE UNA RED DE DISTRIBUCIÓN CON UN 1.5% DE DESBALANCE A LA RED DE TRANSMISIÓN	52
4.4 DISCUSIÓN DE LOS RESULTADOS	53
CAPÍTULO 5 CONCLUSIONES	54
REFERENCIAS	55
APÉNDICE A DATOS DE SISTEMAS DE PRUEBA	57
A1. RED DE DISTRIBUCIÓN DE 28 NODOS	57
A2. RED DE DISTRIBUCIÓN DE 13 NODOS	58
A3. RED DE TRANSMISIÓN DE 4 NODOS	61
APÉNDICE B	62
B1 CÓDIGO FUENTE DEL PROGRAMA RADIFLUX.FOR	62
B2. CODIGO FUENTE DEL PROGRAMA DR.FOR	97
B2. CODIGO FUENTE DEL PROGRAMA ACOPLAMIENTO.FOR	118

ÍNDICE DE FIGURAS

Figura 2.1	Modelo pi de una línea de distribución	6
Figura 2.2	Cargas en conexión estrella	9
Figura 2.3	Cargas en conexión delta	11
Figura 2.4	Modelo de transformador trifásico	12
Figura 2.5	Regulador tipo A en conexión elevadora	14
Figura 2.6	Regulador tipo B en conexión elevadora	15
Figura 2.7	Red de distribución radial	17
Figura 2.8	Diagrama de flujo para la solución de flujos de potencia para una red radial	18
Figura 2.9	Modelo pi de secuencia positiva de línea de transmisión	22
Figura 2.10	Diagrama de flujo de método desacoplado rápido	26
Figura 3.1	Componentes simétricas	31
Figura 3.2	Representación grafica para el enlace de ambos sistemas	33
Figura 3.3	Diagrama de flujo para análisis de componentes simétricas	34
Figura 3.4	Equivalente de secuencia cero para conexión estrella aterrizada-delta	36
Figura 3.5	Equivalente de secuencia cero para conexión estrella -delta	36
Figura 3.6	Diagrama unifilar para el análisis de enlace entre sistema al media y alta tensión	38
Figura 3.7	Diagrama de flujos para solución de flujos de potencia para sistema de transmisión y distribución (a)	41
Figura 3.8	Diagrama de flujos para solución de flujos de potencia para sistema de transmisión y distribución (b)	42
Figura 3.9	Diagrama de flujos para solución de flujos de potencia para sistema de transmisión y distribución (c)	43
Figura 4.1	Red de distribución radial balanceada de 28 nodos	45
Figura 4.2	Red radial de distribución desbalanceada de 13 nodos	47
Figura 4.3	Diagrama unifilar de sistema de transmisión	49
figura 4.4	Diagrama unifilar del enlace entre red de distribución y red de transmisión	

ÍNDICE DE TABLAS

Tabla 2.1	Matrices constantes para cada tipo de conexión del transformador trifásico	12
Tabla 2.2.	Matrices constantes para tipo y conexión del regulador de voltaje.	16
Tabla 4.1	Comparación de voltajes nodales de red de distribución balanceada	46
Tabla 4.2	Comparación de voltajes nodales de red de distribución desbalanceada	48
Tabla 4.3	Validación de solución de flujos de potencia para redes de transmisión	49
Tabla 4.4	Comparación de voltajes del nodo raíz	51
Tabla 4.5	Comparación del nodo raíz incluyendo desbalance de tensiones	53
Tabla A1.1	Datos de impedancia de la red de 28 nodos	57
Tabla A1.2	Datos de carga de red de distribución de 28 nodos	58
Tabla A2.1	Datos de líneas aéreas de red de 13 nodos	58
Tabla A2.2	Datos de líneas subterráneas de red de 13 nodos	58
Tabla A2.3	Datos de transformadores de red de 13 nodos	59
Tabla A2.4	Datos de capacitores de red de 13 nodos	59
Tabla A2.5	Datos de secciones de alimentadores de red de 13 nodos	58
Tabla A2.6	Impedancia serie de alimentadores de red de 13 nodos	59
Tabla A2.7	Admitancia en derivación de red de 13 nodos	60
Tabla A2.8	Datos de carga spot de red de 13 nodos	60
Tabla A2.9	Datos de carga distribuida de red de 13 nodos	60
Tabla A3.1	Datos de líneas de red de 4 nodos	60
Tabla A3.2	Datos de generación y carga	60

SIMBOLOGIA

$[z_{abc}]$	Matriz de impedancias de fase
$[y_{abc}]$	Matriz de admitancias de fase
P_a, P_b, P_c	Potencia real de fases a, b, c
P_{ab}, P_{bc}, P_{ca}	Potencia real de fases ab, bc, ca
Q_a, Q_b, Q_c	Potencia reactiva de fases a, b, c
Q_{ab}, Q_{bc}, Q_{ca}	Potencia reactiva de fases ab, bc, ca
$\theta_a, \theta_b, \theta_c$	Factor de potencia de las cargas de las fases a, b, c
$\theta_{ab}, \theta_{bc}, \theta_{ca}$	Factor de potencia de las cargas de las fases ab, bc, ca
$\delta_a, \delta_b, \delta_c$	Angulo de voltaje de las fases a, b, c
V_{an}, V_{bn}, V_{cn}	Fasores de voltaje línea-neutro de las fases a, b, c
V_{ab}, V_{bc}, V_{ca}	Fasores de voltaje línea-línea entre las fases ab, bc, ca
I_a, I_b, I_c	Fasores de corrientes línea-neutro de las fases a, b, c
I_{ab}, I_{bc}, I_{ca}	Fasores de corrientes línea-línea entre las fases ab, bc, ca
$[V_{LGabc}]$	Vector de voltajes de fase línea-tierra
$[A_T], [B_T]$	Matrices constantes generalizadas para transformadores
$[d_T]$	Matriz constantes que relaciona corrientes de línea del lado de baja con el lado de alta del transformador
Z_{ta}, Z_{tb}, Z_{tc}	Impedancias de transformador monofásico referido al lado de baja
$Z_{tab}, Z_{tbc}, Z_{tca}$	Impedancias de transformador entre las fases ab, bc, ca
B_{an}, B_{bn}, B_{cn}	Susceptancias capacitivas de las fases a, b, c
B_{ab}, B_{bc}, B_{ca}	Susceptancias capacitivas de las fases ab, bc, ca
S_i	Potencia aparente inyectada en el nodo i
P_i	Potencia real inyectada en el nodo i
Q_i	Potencia reactiva inyectada en el nodo i
g_{kj}	Parte real de la admitancia de los elementos de transmisión que conectan a los nodos kj
b_{kj}	Parte imaginaria de la admitancia de los elementos de transmisión que conectan a los nodos kj
V_k	Magnitud de voltaje de nodo k
δ_k	Angulo de voltaje de nodo k
V_j	Magnitud de voltaje de nodo j
δ_j	Angulo de voltaje de nodo j
Y_{ks}	Suma de todas las admitancias en derivación conectadas al nodo k
B'	Matriz de potencia real
B''	Matriz de potencia reactiva
$[\Delta\delta^k]$	Vector de corrección de los ángulos de fase de los nodos de carga
$[\Delta V^k]$	Vector de corrección de voltajes en nodos de carga de red

TOLP	Tolerancia de potencia real
KP	Bandera para potencia real
KQ	Bandera para potencia reactiva
MAX ΔP	Máxima variación de potencia real
TOLQ	Tolerancia de potencia reactiva
V_{a0}, V_{a1}, V_{a2}	Componentes simétricas de voltajes de fase a
V_{b0}, V_{b1}, V_{b2}	Componentes simétricas de voltajes de fase b
V_{c0}, V_{c1}, V_{c2}	Componentes simétricas de voltajes de fase c
a	Operador a
V_a, V_b, V_c	Voltajes de fase a, b, c
A	Matriz de transformación
A^{-1}	Matriz inversa de transformación
$[V_{abc}]$	Matriz de voltajes de fase
$[V_{012}]$	Matriz de componentes simétricas
V_{DA}, V_{DB}, V_{DC}	Voltajes de sistema de distribución de fase
$\%V_{DES}$	Porcentaje de desbalance
ε	Error de voltaje
Dy	Conexión Delta-estrella
Yy	Conexión Estrella-estrella

CAPITULO 1

INTRODUCCIÓN

Tanto para un sistema de alta como de media tensión es indispensable conocer el estado en régimen permanente, como lo son voltajes nodales, flujos de potencia activa y reactiva y pérdidas, entre otros valores. Estos parámetros son de gran importancia para analizar contingencias o problemas que se puedan presentar tanto en sistemas ya existentes (buscando resolver problemas de operación económica y regulación de voltaje) como en la planeación de nuevos sistemas (verificar el comportamiento de los elementos en distintas alternativas, compensación shunt y derivaciones de los transformadores).

Entre los métodos numéricos utilizados para encontrar la condiciones en estado permanente de un sistema eléctrico, se ha tenido un mayor desarrollo en los sistemas de alta tensión debido a que su modelo es más simple comparado con el sistema de distribución, ya que es suficiente con desarrollar un modelo de secuencia positiva para realizar los análisis necesarios. En caso contrario, en un sistema de media tensión, su modelado se tiene que hacer en coordenadas de fase, debido a sus características como son el desbalance de cargas, baja relación X/R , líneas no transpuestas, presencia de laterales bifásicos ó monofásicos, entre otras.

Debido a las diferentes características entre el sistema de alta y media tensión, no es posible establecer un modelo matemático común, por tal motivo se ha buscado tener una técnica y algoritmo que permitan resolver dichos modelos de una forma coordinada y más eficiente, tratando de optimizar el tiempo de cálculo y uso de memoria del método, pero principalmente obtener soluciones integradas de ambos ambientes.

1.1 Objetivo

Desarrollar un programa computacional para obtener la solución coordinada de estudios de flujos en redes de alta y media tensión mediante la consideración de dos entornos de diferentes tipos de análisis.

1.2 Justificación

Actualmente se cuenta con métodos numéricos y herramientas computacionales [1, 2, 4] que permiten obtener el estado, tanto para sistemas de alta como de media tensión, pero de forma independiente. Para ello cada sistema realiza diferentes suposiciones como en el caso del sistema de alta tensión, en donde todas las cargas que se encuentran en un alimentador de distribución se representan como un nodo de carga, mientras que para un sistema de media tensión, el sistema de alta tensión es simulado como un bus infinito.

Esto implica resolver por separado ambos sistemas componentes de un gran sistema de potencia comercial.

Si bien, los resultados obtenidos de las suposiciones o consideraciones para cada sistema no son erróneos, surge la necesidad de contar con una herramienta que permita obtener en forma coordinada y automática ambas soluciones,

1.3 Estado del arte

A pesar de que se han realizado grandes aportaciones, tanto para sistemas de transmisión como para sistemas de distribución, la necesidad de contar con herramientas computacionales que permitan observar al sistema eléctrico en forma completa, promueve el surgimiento de nuevos algoritmos que hagan posible esto.

En 2008, Francisco Gonzales-Longatt y Erix J. Palencia, presentan el desarrollo de un programa denominado ASPd [27] (Analizador de Sistemas de Potencia y Distribución), el cual cuenta con una interfaz sencilla en ambiente Microsoft Windows y permite evaluar las condiciones de operación en régimen permanente en sistemas de potencia y de distribución, a través de los métodos de Newton-Raphson para el sistema de transmisión, mientras que el sistema de distribución es resuelto con dos métodos diferentes, uno planteado por Dariush-Shirmohammadi y el otro por Ghosh-Das[Referencias].

Entre las características del ASPd, se encuentran las siguientes:

- Efectúa flujos de potencia en redes de distribución radiales empleando dos métodos diferentes uno por Dariush-Shirmohammadi y el segundo por Ghosh-Das.
- En cuanto a redes de transmisión utiliza el método de Newton-Raphson para la solución de flujos de potencia.
- Emplea una interfaz gráfica implementada en Microsoft Windows
- Modela las cargas en función del voltaje como cargas de potencia, impedancia y corriente constante.
- Modela los dispositivos de control de voltaje y potencia reactiva del sistema de potencia.

En 2009, Eric Morales Aguilar presenta la tesis de maestría [3] que lleva por nombre “Desarrollo de un programa para resolver el problema de flujos de potencia en redes aéreas de distribución por el método progresivo regresivo”, mediante la cual analiza el desempeño y rendimiento [5] de cuatro diferentes métodos, basándose en los siguientes factores:

- Numero de iteraciones para alcanzar una tolerancia determinada.

- Factor de convergencia por método.
- Número de operaciones de punto flotante necesarias por cada método.

Después de realizar las pruebas para evaluar el método más idóneo de sistemas de distribución, se encuentra que el método de Newton-Raphson, a pesar de presentar un número menor de iteraciones, arroja más errores y problemas de convergencia, debido a que realiza un mayor número de operaciones en comparación con el método de barrido progresivo-regresivo.

En conclusión se determinó que la prueba que muestra ser indicador relativo del tiempo de ejecución es el número de operaciones de punto flotante, por lo cual el método más idóneo fue el de barrido progresivo-regresivo ya que realiza un número menor de operaciones, sin embargo, el número de iteraciones y convergencia es muy cercana al método de Newton-Raphson [3].

En el mismo año Oscar Miranda Urióstegui presenta la tesis de maestría [25] que lleva por nombre “Minimización de pérdidas en redes de distribución empleando generación dinámica”, la cual tuvo el objetivo de generar una herramienta computacional capaz de obtener la solución de flujos de potencia trifásicos en sistemas puramente radiales y aplicar esta herramienta en un algoritmo de programación dinámica para el despacho de capacitores que permita mantener el voltaje de la red de distribución dentro de un perfil de voltaje normal.

Para el año 2011, Alfonso Jara Morales presenta la tesis de maestría [26], la cual lleva por nombre “Estudio de flujos de potencia y de cortocircuito en sistemas de media tensión incorporando generación distribuida”, en ella se tenía como objetivo estudiar el efecto de la integración de fuentes de generación renovable en redes de distribución de topología radial y en la cual se tenía implícitamente la tarea de generar una herramienta computacional capaz de realizar la solución de flujos de potencia en redes de distribución de topología radial y en donde de igual forma se utiliza el método

En general, los trabajos revisados como las referencias consultadas por [26] se enfocan a resolver el problema de flujos de potencia de forma individual para cada sistema, ya sea para el sistema de alta o de media tensión pero no en forma conjunta.

1.4 Aportaciones

- Implementación de un algoritmo para la solución coordinada de flujos de potencia de los sistemas de alta y media tensión.
- Contar con una herramienta computacional que permita resolver ambos sistemas de forma conjunta.

1.5 Contenido de tesis

Dentro del capítulo 1, se presenta una breve introducción de las diferentes características que presentan los sistemas de alta y media tensión, además del objetivo general, la justificación, el estado del arte, aportaciones y el contenido de la tesis.

En el capítulo 2, se presentan los modelos de elementos y método numérico de solución de flujos de potencia para redes de alta y media tensión.

En el capítulo 3, se presentan las características, ventajas y desventajas de los métodos de solución de flujos de potencia para sistemas de alta y media tensión, así como el algoritmo para la solución coordinada de flujos de potencia entre la red de media y alta tensión.

En el capítulo 4, se presentan los resultados obtenidos del acoplamiento entre una red de media tensión con una pequeña red de alta tensión.

En el capítulo 5, se muestran las conclusiones a las que se llegaron con la realización del presente trabajo de tesis.

CAPITULO 2

MODELADO DE REDES DE ALTA Y MEDIA TENSIÓN

2.1 Modelado de redes de media tensión

Un sistema de media tensión parte de una subestación de distribución, la cual está alimentada por una o más líneas de subtransmisión o transmisión, esta subestación, a su vez cuenta con los alimentadores primarios que por lo regular suelen ser del tipo radiales para finalizar proporcionando la energía necesaria que el usuario necesita.

A pesar de que los sistemas de distribución se habían caracterizado por ser el componente menos importante del sistema eléctrico [4], con el transcurso del tiempo, ha cambiado la necesidad de contar con sistemas más eficientes y poder obtener mejores estimaciones de las condiciones del sistema en régimen permanente, esto ha generado que tanto ingenieros como investigadores hayan puesto atención al comportamiento, capacidad y operación de los sistemas de distribución, siendo necesario contar con un modelo exacto del sistema de distribución y una buena técnica de análisis que nos permita determinar el estado de la red.

Para la correcta solución de flujos de potencia de un sistema de media tensión es necesario contar con los modelos de cada uno de los elementos con los que cuenta la red con la que se desea trabajar, como lo son, los elementos serie (transformadores, segmentos de líneas y reguladores de voltaje) y los elementos en derivación (cargas puntuales, cargas distribuidas y bancos de capacitores).

A continuación se presentan los modelos considerados para la correcta solución de flujos de potencia.

2.1.1 Modelo de la línea de distribución

Éste, se considera como un modelo exacto de la línea de distribución al modelo pi (figura 2.1), debido al amplio rango de distancia con el que cuenta, ya que es posible representar líneas mayores a 80 km con gran exactitud, situación que no se podría realizar con el modelo de línea corta.

El primer elemento del modelo de la línea a considerar es la impedancia serie de las líneas, la cual está formada por una parte resistiva y una parte reactiva, donde se involucran tanto las características físicas como los campos magnéticos producidos por el conductor con corriente.

$$Z = R + jX \quad (2.1)$$

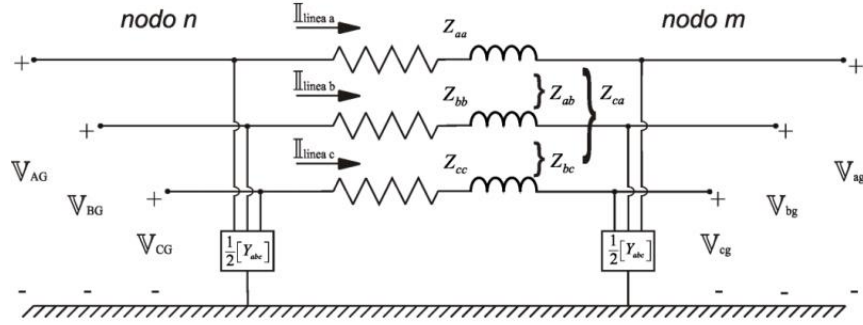


Figura 2.1. Modelo pi de una línea de distribución.

Al tratarse de una red trifásica la ecuación 2.1 se transforma en una matriz de 3x3, la cual se desarrolla en [4].

$$[z_{abc}] = \begin{bmatrix} Z_{aa} & Z_{ab} & Z_{ac} \\ Z_{ba} & Z_{bb} & Z_{bc} \\ Z_{ca} & Z_{cb} & Z_{cc} \end{bmatrix} \frac{\Omega}{\text{milla}} \quad (2.2)$$

Si el segmento de línea fuese bifásico, es decir que solo se incluyen 2 de las 3 fases existentes (a, b y c), se modelaría de la misma forma que en la ecuación 2.2, con la restricción de colocar ceros donde la fase faltante se vea involucrada, por ejemplo, si la fase faltante fuera la fase c, su modelo sería el siguiente:

$$[z_{abc}] = \begin{bmatrix} Z_{aa} & Z_{ab} & 0 \\ Z_{ba} & Z_{bb} & 0 \\ 0 & 0 & 0 \end{bmatrix} \frac{\Omega}{\text{milla}} \quad (2.3)$$

Si el segmento de línea fuera monofásico, siguiendo la forma de la ecuación 2.2, al solo existir una fase se colocarían ceros en las filas y columnas de las fases faltantes, por ejemplo, si solo existiera la fase b, su modelo sería el siguiente:

$$[z_{abc}] = \begin{bmatrix} 0 & 0 & 0 \\ 0 & Z_{bb} & 0 \\ 0 & 0 & 0 \end{bmatrix} \frac{\Omega}{\text{milla}} \quad (2.4)$$

El segundo elemento importante a considerar en la modelación de la línea es la admitancia en derivación compuesta por la conductancia y la susceptancia capacitiva, normalmente la conductancia es despreciada debido a que es muy pequeña comparada con la susceptancia capacitiva, como se hizo para la impedancia serie, el modelo de la admitancia en derivación se presenta de la misma forma.

$$[y_{abc}] = \begin{bmatrix} y_{aa} & y_{ab} & y_{ac} \\ y_{ba} & y_{bb} & y_{bc} \\ y_{ca} & y_{cb} & y_{cc} \end{bmatrix} \frac{\Omega}{milla} \quad (2.5)$$

Como se observa en la ecuación 2.5 la admitancia en derivación está presente en las tres líneas, para un sistema trifásico pero en líneas bifásicas ó monofásicas, las admitancias correspondientes a las fases faltantes serán iguales a cero.

Finalmente obtenidas las impedancias y admitancias que describen la figura 2.1 se multiplican por la longitud de la línea.

$$Z_{abc} = \text{longitud de la línea} * [z_{abc}] = \begin{bmatrix} Z_{aa} & Z_{ab} & Z_{ac} \\ Z_{ba} & Z_{bb} & Z_{bc} \\ Z_{ca} & Z_{cb} & Z_{cc} \end{bmatrix} \Omega \quad (2.6)$$

$$Y_{abc} = \text{longitud de la línea} * [y_{abc}] = \begin{bmatrix} Y_{aa} & Y_{ab} & Y_{ac} \\ Y_{ba} & Y_{bb} & Y_{bc} \\ Y_{ca} & Y_{cb} & Y_{cc} \end{bmatrix} \Omega \quad (2.7)$$

2.1.2 Modelos de carga

El modelo y análisis en estado estable de un sistema de potencia depende de la carga, ésta la podemos definir como la potencia aparente consumida en un nodo. Se detalla, ya sea como potencia activa y reactiva o potencia activa con su respectivo factor de potencia. Esta carga es modelada tanto en conexión delta como en conexión estrella, además, dichas cargas son de naturaleza trifásica, bifásica y/o monofásica, sin importar el grado de desbalance y puede modelarse como potencia real y activa constante, corriente constante e impedancia constante; todas estas formas de modelado se pueden asignar como cargas puntuales (cargas que se encuentran conectadas en un solo nodo) o cargas distribuidas (cargas asignadas uniformemente sobre un segmento de la línea).

2.1.2.1 Cargas conectadas en estrella

El modelo para cargas conectadas en estrella se muestra en la figura 2.2 de donde se obtiene lo siguiente por fase:

Para la fase a tenemos:

$$|S_a| \angle \theta_a = P_a + jQ_a \quad y \quad |V_{an}| \angle \delta_a \quad (2.8)$$

Para la fase b tenemos:

$$|S_b| \angle \theta_b = P_b + jQ_b \quad y \quad |V_{bn}| \angle \delta_b \quad (2.9)$$

Para la fase c tenemos:

$$|S_c| \angle \theta_c = P_c + jQ_c \quad y \quad |V_{cn}| \angle \delta_c \quad (2.10)$$

2.1.2.1.1 Cargas conectadas en estrella a potencia activa y reactiva constante

Las ecuaciones que definen las corrientes de línea para cargas de potencia real y reactiva constantes en configuración estrella son las siguientes:

$$I_a = \left(\frac{S_a}{V_{an}} \right)^* = \left(\frac{S_a}{V_{an}} \right) \angle (\delta_a - \theta_a) = |I_a| \angle \alpha_a \quad (2.11)$$

$$I_b = \left(\frac{S_b}{V_{bn}} \right)^* = \left(\frac{S_b}{V_{bn}} \right) \angle (\delta_b - \theta_b) = |I_b| \angle \alpha_b \quad (2.12)$$

$$I_c = \left(\frac{S_c}{V_{cn}} \right)^* = \left(\frac{S_c}{V_{cn}} \right) \angle (\delta_c - \theta_c) = |I_c| \angle \alpha_c \quad (2.13)$$

2.1.2.1.2 Cargas conectadas en estrella a impedancia constante

En este modelo, los voltajes de línea a neutro cambiarán en cada iteración mientras que la impedancia calculada permanecerá constante, esta se calcula de la siguiente manera:

$$Z_a = \frac{|V_{an}|^2}{S_a^*} = \frac{|V_{an}|^2}{|S_a|} \angle \theta_a = |Z_a| \angle \theta_a \quad (2.14)$$

$$Z_b = \frac{|V_{bn}|^2}{S_b^*} = \frac{|V_{bn}|^2}{|S_b|} \angle \theta_b = |Z_b| \angle \theta_b \quad (2.15)$$

$$Z_c = \frac{|V_{cn}|^2}{S_c^*} = \frac{|V_{cn}|^2}{|S_c|} \angle \theta_c = |Z_c| \angle \theta_c \quad (2.16)$$

Una vez calculadas las potencias se calculan las corrientes de la siguiente manera:

$$I_a = \frac{V_{an}}{Z_a} = \frac{|V_{an}|}{|Z_a|} \angle (\delta_a - \theta_a) = |I_a| \angle \alpha_a \quad (2.17)$$

$$I_b = \frac{V_{bn}}{Z_b} = \frac{|V_{bn}|}{|Z_b|} \angle (\delta_b - \theta_b) = |I_b| \angle \alpha_b \quad (2.18)$$

$$\mathbb{I}_c = \frac{\mathbb{V}_{cn}}{Z_c} = \frac{|\mathbb{V}_{cn}|}{|Z_c|} \angle(\delta_c - \theta_c) = |\mathbb{I}_c| \angle \alpha_c \quad (2.19)$$

2.1.2.1.3 Cargas conectadas en estrella a corriente constante

Para este modelo el ángulo del fasor de voltaje (δ) cambiara en cada iteración, permaneciendo el factor de potencia de la carga constante (θ), resultando en un ángulo de la corriente cambiante pero permaneciendo la magnitud de la corriente constante:

$$\mathbb{I}_a = |\mathbb{I}_a| \angle(\delta_a - \theta_a) \quad (2.20)$$

$$\mathbb{I}_b = |\mathbb{I}_b| \angle(\delta_b - \theta_b) \quad (2.21)$$

$$\mathbb{I}_c = |\mathbb{I}_c| \angle(\delta_c - \theta_c) \quad (2.22)$$

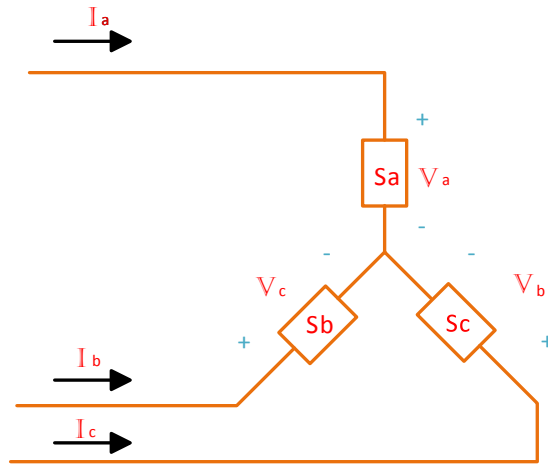


Figura 2.2 Cargas en conexión estrella.

2.1.2.2 Cargas conectadas en delta

El modelo para cargas conectadas en delta se muestra en la figura 2.3 de donde se obtiene lo siguiente por fase:

Para las fases ab tenemos:

$$|S_{ab}| \angle \theta_{ab} = P_{ab} + jQ_{ab} \quad y \quad |\mathbb{V}_{ab}| \angle \delta_{ab} \quad (2.23)$$

Para las fases bc tenemos:

$$|S_{bc}| \angle \theta_{bc} = P_{bc} + jQ_{bc} \quad y \quad |\mathbb{V}_{bc}| \angle \delta_{bc} \quad (2.24)$$

Para las fases ca tenemos:

$$|S_{ca}| \angle \theta_{ca} = P_{ca} + jQ_{ca} \quad y \quad |V_{ca}| \angle \delta_{ca} \quad (2.25)$$

2.1.2.2.1 Cargas conectadas en delta a potencia activa y reactiva constante

En este modelo los voltajes de línea a línea cambiarán en cada iteración resultando nuevas magnitudes y ángulos de corriente, estas se obtienen de la siguiente manera:

$$I_{ab} = \left(\frac{S_{ab}}{V_{ab}} \right)^* = \left(\frac{S_{ab}}{V_{ab}} \right) \angle (\delta_{ab} - \theta_{ab}) = |I_{ab}| \angle \alpha_{ab} \quad (2.26)$$

$$I_{bc} = \left(\frac{S_{bc}}{V_{bc}} \right)^* = \left(\frac{S_{bc}}{V_{bc}} \right) \angle (\delta_{bc} - \theta_{bc}) = |I_{bc}| \angle \alpha_{bc} \quad (2.27)$$

$$I_{ca} = \left(\frac{S_{ca}}{V_{ca}} \right)^* = \left(\frac{S_{ca}}{V_{ca}} \right) \angle (\delta_{ca} - \theta_{ca}) = |I_{ca}| \angle \alpha_{ca} \quad (2.28)$$

2.1.2.2.2 Cargas conectadas en delta a impedancia constante

La carga a impedancia constante se determina a través de la potencia aparente consumida y el voltaje nominal de línea a línea.

$$Z_{ab} = \frac{|V_{ab}|^2}{S_{ab}^*} = \frac{|V_{ab}|^2}{|S_{ab}|} \angle \theta_{ab} = |Z_{ab}| \angle \theta_{ab} \quad (2.29)$$

$$Z_{bc} = \frac{|V_{bc}|^2}{S_{bc}^*} = \frac{|V_{bc}|^2}{|S_{bc}|} \angle \theta_{bc} = |Z_{bc}| \angle \theta_{bc} \quad (2.30)$$

$$Z_{ca} = \frac{|V_{ca}|^2}{S_{ca}^*} = \frac{|V_{ca}|^2}{|S_{ca}|} \angle \theta_{ca} = |Z_{ca}| \angle \theta_{ca} \quad (2.31)$$

Una vez calculadas las impedancias de línea a línea, las corrientes se obtienen de la siguiente forma:

$$I_{ab} = \frac{V_{ab}}{Z_{ab}} = \frac{|V_{ab}|}{|Z_{ab}|} \angle (\delta_{ab} - \theta_{ab}) = |I_{ab}| \angle \alpha_{ab} \quad (2.32)$$

$$I_{bc} = \frac{V_{bc}}{Z_{bc}} = \frac{|V_{bc}|}{|Z_{bc}|} \angle (\delta_{bc} - \theta_{bc}) = |I_{bc}| \angle \alpha_{bc} \quad (2.33)$$

$$I_{ca} = \frac{V_{ca}}{Z_{ca}} = \frac{|V_{ca}|}{|Z_{ca}|} \angle (\delta_{ca} - \theta_{ca}) = |I_{ca}| \angle \alpha_{ca} \quad (2.34)$$

En este modelo los voltajes de línea a línea cambiarán en cada iteración hasta lograr la convergencia.

2.1.2.2.3 Cargas conectadas en delta a corriente constante

Para este modelo el ángulo del fasor de voltaje (δ) cambiara en cada iteración, permaneciendo el factor de potencia de la carga constante (θ), resultando en un ángulo de la corriente cambiante pero permaneciendo la magnitud de la corriente constante:

$$\mathbb{I}_{ab} = |\mathbb{I}_{ab}| \angle (\delta_{ab} - \theta_{ab}) \quad (2.35)$$

$$\mathbb{I}_{bc} = |\mathbb{I}_{bc}| \angle (\delta_{bc} - \theta_{bc}) \quad (2.36)$$

$$\mathbb{I}_{ca} = |\mathbb{I}_{ca}| \angle (\delta_{ca} - \theta_{ca}) \quad (2.37)$$

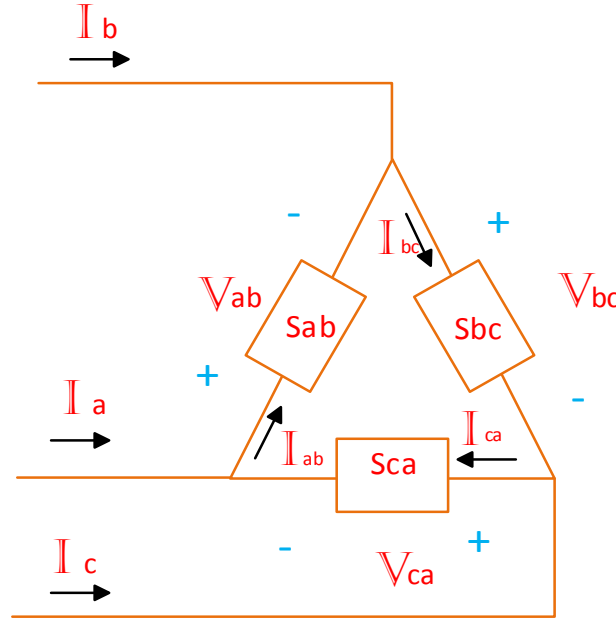


Figura 2.3 Cargas en conexión delta.

Como resultado de aplicar la ley de Kirchhoff de corrientes en cada nodo de la delta (figura 2.3) se obtienen las corrientes de línea.

$$\begin{bmatrix} \mathbb{I}_a \\ \mathbb{I}_b \\ \mathbb{I}_c \end{bmatrix} = \begin{bmatrix} 1 & 0 & -1 \\ -1 & 1 & 0 \\ 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} \mathbb{I}_{ab} \\ \mathbb{I}_{bc} \\ \mathbb{I}_{ca} \end{bmatrix} \quad (2.38)$$

2.1.3 Modelos de transformador trifásico

El elemento más esencial de una subestación de distribución es el transformador, donde en la mayoría de los casos suele ser trifásico con cambiadores de derivaciones bajo carga en el lado de baja tensión, este es el encargado de transformar el voltaje de transmisión o subtransmisión a un nivel bajo de distribución.

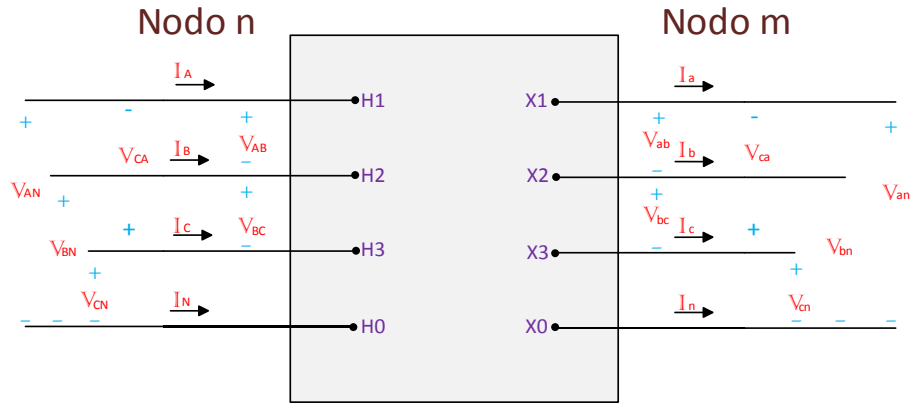


Figura 2.4 Modelo de transformador trifásico.

La figura 2.4 muestra el modelo de un transformador trifásico con entradas y salidas, donde los parámetros del lado de alta tensión (nodo n) estarán en función del lado de baja tensión (nodo m), para el caso de las tensiones (ecuación 2.39), mientras que para las corrientes del nodo n se obtienen en función del nodo m (ecuación 2.40). Para ambos casos el tipo de conexión que se presenta definirá el valor que tomen las matrices constantes (A_T , B_T , d_T) como se indica en la tabla 2.1, esta relación está dada por:

$$[\mathbb{V}_{LG_{abc}}] = [A_T] * [\mathbb{V}_{LG_{ABC}}] - [B_T] * [\mathbb{I}_{abc}] \quad (2.39)$$

$$[\mathbb{I}_{ABC}] = [d_T][\mathbb{I}_{abc}] \quad (2.40)$$

Para el análisis del transformador es necesario modelar correctamente las conexiones más comunes que se presentan en las subestaciones de distribución como son: delta-estrella aterrizada, estrella-delta, estrella aterrizada-estrella aterrizada, delta-delta y estrella abierta-delta abierta, y las cuales se exponen de manera más detallada en [4].

Tabla 2.1 Matrices constantes para cada tipo de conexión del transformador trifásico.

Devanado primario	Devanado secundario	Matriz constante $[A_T]$	Matriz constante $[B_T]$	Matriz constante $[d_T]$
Delta	Estrella aterrizada	$\frac{1}{n_t} \begin{bmatrix} 1 & 0 & -1 \\ -1 & 1 & 0 \\ 0 & -1 & 1 \end{bmatrix}$	$\begin{bmatrix} Z_{ta} & 0 & 0 \\ 0 & Z_{tb} & 0 \\ 0 & 0 & Z_{tc} \end{bmatrix}$	$\frac{1}{n_t} \begin{bmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \\ -1 & 0 & 1 \end{bmatrix}$

Estrella	Delta	$\frac{1}{3n_t} \begin{bmatrix} 2 & 1 & 0 \\ 0 & 2 & 1 \\ 1 & 0 & 2 \end{bmatrix}$	$\frac{1}{9} \begin{bmatrix} 2Z_{tab} + Z_{tbc} & 2Z_{tbc} - 2Z_{tab} & 0 \\ 2Z_{tbc} - 2Z_{tca} & 4Z_{tbc} - Z_{tca} & 0 \\ 2Z_{tab} - 4Z_{tca} & -Z_{tab} - 2Z_{tca} & 0 \end{bmatrix}$	$\frac{1}{3n_t} \begin{bmatrix} 1 & -1 & 0 \\ 1 & 2 & 0 \\ -2 & -1 & 0 \end{bmatrix}$
Estrella aterrizada	Delta aterrizada	$\frac{1}{n_t} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} Z_{ta} & 0 & 0 \\ 0 & Z_{tb} & 0 \\ 0 & 0 & Z_{tc} \end{bmatrix}$	$\frac{1}{n_t} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$
Delta	Delta	$\frac{1}{3n_t} \begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix}$	$\frac{1}{3(Z_{tab} + Z_{tbc} + Z_{tca})} \begin{bmatrix} 2Z_{tab} & Z_{tbc} & 0 \\ 0 & 2Z_{tbc} & Z_{tca} \\ Z_{tab} & 0 & 2Z_{tca} \end{bmatrix} \begin{bmatrix} Z_{tca} & -Z_{tbc} & 0 \\ Z_{tca} & Z_{tab} + Z_{tca} & 0 \\ -(Z_{tab} + Z_{tbc}) & -Z_{tbc} & 0 \end{bmatrix}$	$\frac{1}{n_t} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$
Estrella abierta	Delta abierta	$\frac{1}{3n_t} \begin{bmatrix} 2 & 1 & 0 \\ -1 & 1 & 0 \\ -1 & -2 & 0 \end{bmatrix}$	$\frac{1}{3} \begin{bmatrix} 2Z_{tab} & 0 & -Z_{tbc} \\ -Z_{tab} & 0 & -Z_{tbc} \\ -Z_{tab} & 0 & 2Z_{tbc} \end{bmatrix}$	$\frac{1}{n_t} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 0 & 0 \end{bmatrix}$

Donde n_t representa la relación de espiras entre el primario y secundario del transformador, esta se da con el cociente entre tensiones (primario/secundario) y las impedancias $Z_{ta}, Z_{tb}, Z_{tc}, Z_{tab}, Z_{tbc}, Z_{tca}$; son las impedancias de cada transformador monofásico referidas al lado de baja.

2.1.4 Modelo de banco de capacitores

La aplicación de los bancos de capacitores permite compensar la potencia reactiva en el punto de la red donde se coloque (subestaciones o alimentadores de distribución), esto permite la corrección del factor de potencia y da un soporte a la regulación de voltaje. Los bancos de capacitores se modelan como susceptancias constantes, conectadas en delta o en estrella.

Para el caso de la conexión estrella la susceptancia constante se obtiene en Siemens para posteriormente obtener las corrientes que fluyen hacia el banco de capacitores, las cuales se obtiene de la siguiente forma:

$$B_c = \frac{KVAR_{nom}}{KV^2_{LN} * 1000} \quad \text{Siemens} \quad (2.41)$$

$$\begin{bmatrix} \mathbb{I}_a \\ \mathbb{I}_b \\ \mathbb{I}_c \end{bmatrix} = \begin{bmatrix} -B_{an} & 0 & 0 \\ 0 & -B_{bn} & 0 \\ 0 & 0 & -B_{cn} \end{bmatrix} \begin{bmatrix} \mathbb{V}_{an} \\ \mathbb{V}_{bn} \\ \mathbb{V}_{cn} \end{bmatrix} \quad (2.42)$$

Para la conexión delta la susceptancia y corrientes que fluyen al banco de capacitores están dadas por:

$$B_c = \frac{KVAR_{nom}}{KV^2_{LL} * 1000} \quad \text{Siemens} \quad (2.43)$$

$$\begin{bmatrix} \mathbb{I}_a \\ \mathbb{I}_b \\ \mathbb{I}_c \end{bmatrix} = \begin{bmatrix} 1 & 0 & -1 \\ -1 & 1 & 0 \\ 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} -B_{ca} & 0 & 0 \\ 0 & -B_{cb} & 0 \\ 0 & 0 & -B_{ca} \end{bmatrix} \begin{bmatrix} \mathbb{V}_{ab} \\ \mathbb{V}_{bc} \\ \mathbb{V}_{ca} \end{bmatrix} \quad (2.44)$$

2.1.5 Modelo de regulador de voltaje

Una de las características de un circuito de distribución es que las cargas se encuentran en constante cambio, por lo que la regulación de voltaje se convierte en una función de gran importancia en un alimentador de distribución, el cual debe de mantener el voltaje en niveles aceptables. Dentro de los métodos más comunes se encuentra la aplicación de reguladores a pasos, transformadores con cambio de derivaciones bajo carga y condensadores en paralelo.

El modelo utilizado para el desarrollo del programa de cálculo de flujos de potencia es el modelo del regulador de voltaje a pasos, este modelo consta básicamente de un autotransformador y un mecanismo de cambio de derivaciones bajo carga, donde el cambio de voltaje se obtiene cambiando las derivaciones del devanado serie del autotransformador.

El regulador a paso estándar contiene un regulador reversible que permite un rango de $\pm 10\%$ usualmente en 32 pasos, lo cual significa que por cada paso habrá un cambio de 0.625% con respecto al voltaje base, por otro lado, el regulador de voltaje puede ser de tipo A o de tipo B de acuerdo a la norma ANSI/IEEE C57.15-1986, aunque la conexión externa y el uso en la red es el mismo, suele ser el regulador tipo B el más utilizado debido a que el circuito primario se encuentra conectado vía taps al devanado serie del transformador, en la figura 2.5 se muestra el regulador tipo A en posición elevadora y en la figura 2.6 se muestra el regulador tipo B en posición elevadora, la única variante para la posición reductora es el cambio en interruptor reversible para realizar el cambio a la posición reductora.

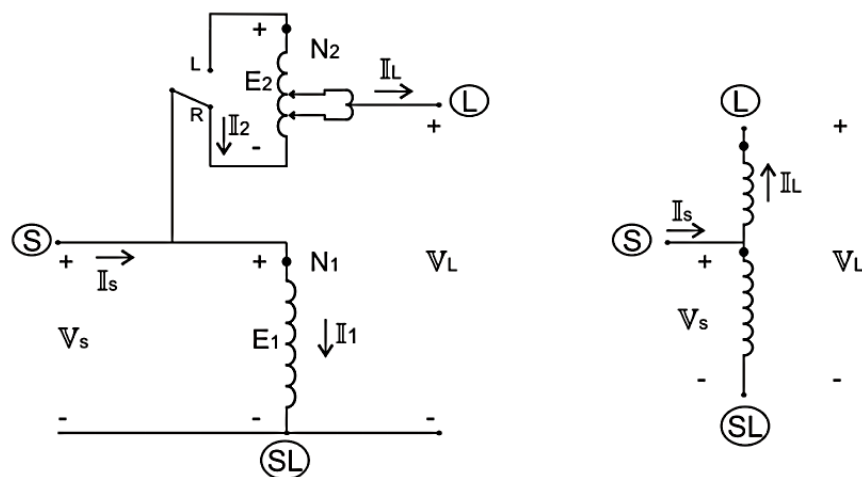


Figura 2.5 Regulador tipo A en conexión elevadora.

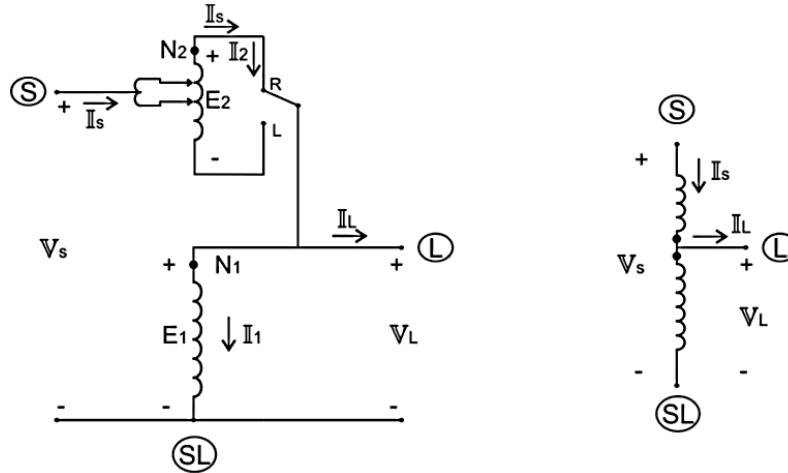


Figura 2.6 Regulador tipo B en conexión elevadora.

El regulador a pasos trifásicos puede estar formado por tres reguladores monofásicos, si cada regulador cuenta con su propio circuito compensador permite que cada regulador actúe de forma separada, pero si solo se cuenta con un circuito compensador, las derivaciones en todos los compensadores actúan al mismo tiempo, dentro de las conexiones más comunes para reguladores monofásicos se encuentran las siguientes:

- Monofásicos entre línea-tierra
- Estrella abierta (dos reguladores monofásicos entre línea-tierra)
- Estrella (tres reguladores monofásicos entre línea-tierra)
- Delta abierta (dos reguladores monofásicos entre línea-línea)
- Delta abierta (tres reguladores monofásicos entre línea-línea)

De los tipos de conexión anteriores solo se incorporaron la conexión estrella y delta para el análisis del alimentador de distribución y desarrollo del método regresivo-progresivo planteado en [2], por otro lado el voltaje del lado de carga se da en función de los voltajes y corrientes del lado primario del regulador (ecuación 2.44) mientras que la corriente del lado primario se da en función del lado de carga del regulador (ecuación 2.45).

$$[\mathbb{V}_{LG_{abc}}] = [A_R] * [\mathbb{V}_{LG_{ABC}}] - [B_R] * [\mathbb{I}_{abc}] \quad (2.39)$$

$$[\mathbb{I}_{ABC}] = [d_R][\mathbb{I}_{abc}] \quad (2.40)$$

Al igual que para el modelo del transformador las matrices constantes dependen del tipo de regulador que se utilice conforme a la tabla 2.2.

Tabla 2.2. Matrices constantes para tipo y conexión del regulador de voltaje.

Tipo de regulador	Matriz constante $[A_R]$	Matriz constante $[B_T]$	Matriz constante $[d_T]$
Tipo A en conexión estrella	$\begin{bmatrix} a_{Ra} & 0 & 0 \\ 0 & a_{Rb} & 0 \\ 0 & 0 & a_{Rc} \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} a_{Ra} & 0 & 0 \\ 0 & a_{Rb} & 0 \\ 0 & 0 & a_{Rc} \end{bmatrix}$
Tipo B en conexión estrella	$\begin{bmatrix} 1/a_{Ra} & 0 & 0 \\ 0 & 1/a_{Rb} & 0 \\ 0 & 0 & 1/a_{Rc} \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 1/a_{Ra} & 0 & 0 \\ 0 & 1/a_{Rb} & 0 \\ 0 & 0 & 1/a_{Rc} \end{bmatrix}$
Tipo A en conexión delta abierta	$\begin{bmatrix} a_{Rab} & 0 & 0 \\ 0 & a_{Rcb} & 0 \\ -a_{Rab} & -a_{Rcb} & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} a_{Rab} & 0 & 0 \\ -a_{Rab} & 0 & -a_{Rcb} \\ 0 & 0 & a_{Rcb} \end{bmatrix}$
Tipo B en conexión delta abierta	$\begin{bmatrix} 1/a_{Rab} & 0 & 0 \\ 0 & 1/a_{Rcb} & 0 \\ -1/a_{Rab} & -1/a_{Rcb} & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 1/a_{Rab} & 0 & 0 \\ -1/a_{Rab} & 0 & -1/a_{Rcb} \\ 0 & 0 & 1/a_{Rcb} \end{bmatrix}$

Donde a_{Rab} , a_{Rab} y a_{Rab} son las relaciones de transformación de cada regulador monofásico.

2.2 Solución de flujos de potencia en sistemas de media tensión

La solución de flujos de potencia en sistemas de distribución nos permite determinar las condiciones en las que se encuentra la red, como lo son: magnitudes y ángulos de voltaje en cada nodo, flujos de potencia activa y reactiva en cada segmento de la línea de distribución, pérdidas totales o parciales (en cada sección del alimentador) y la entrada total de potencia activa y reactiva a la red; para lograr una correcta estimación de estas condiciones, es necesario que tanto los elementos en serie como en derivación sean correctamente modelados, así como contar con un correcto método de análisis, ya que según [2] se necesita de un método de análisis robusto y eficiente para lograr resultados satisfactorios.

La aplicación de métodos de solución de redes de transmisión a sistemas de distribución ocasionan serios problemas de convergencia para la solución de flujos de potencia, esto se debe a la topología radial y amplio rango de valores de resistencia y reactancia de la red de distribución para poder ser resueltos con métodos de solución, como son: Gauss-Seidel y Newton-Raphson, los cuales al aplicarlos a sistemas de distribución presentan problemas derivado a una mala inicialización y gran uso de memoria.

Debido a las diferencias que existen entre ambos sistemas es necesario contar con un método que analice el sistema trifásico por completo, sin hacer ningún tipo de simplificación de la red, lo cual nos permita solucionar problemas específicos de la red de distribución, como pueden ser desbalance de carga, caída de tensión y compensación capacitiva.

2.2.1 Solución de flujos de potencia para redes de distribución radial

El método para la solución de flujos de potencia por el método de barrido regresivo-progresivo propuesto en [2], se basa en la aplicación directa de la ley de corrientes y voltajes de Kirchhoff y en un esquema eficiente de numeración de las ramas, lo cual mejora el rendimiento del método de solución. La numeración se realiza de forma ascendente y formando niveles partiendo del nodo raíz hasta la última rama del circuito, en la figura 2.7 se muestra el esquema de numeración de una red típica radial.

Una vez numeradas las ramas y nodos de la red de distribución y conocida la tensión del nodo raíz, se debe de asumir inicialmente un perfil plano de voltajes para todos los nodos de la red, el algoritmo iterativo de solución se presenta en el siguiente diagrama de la figura 2.8.

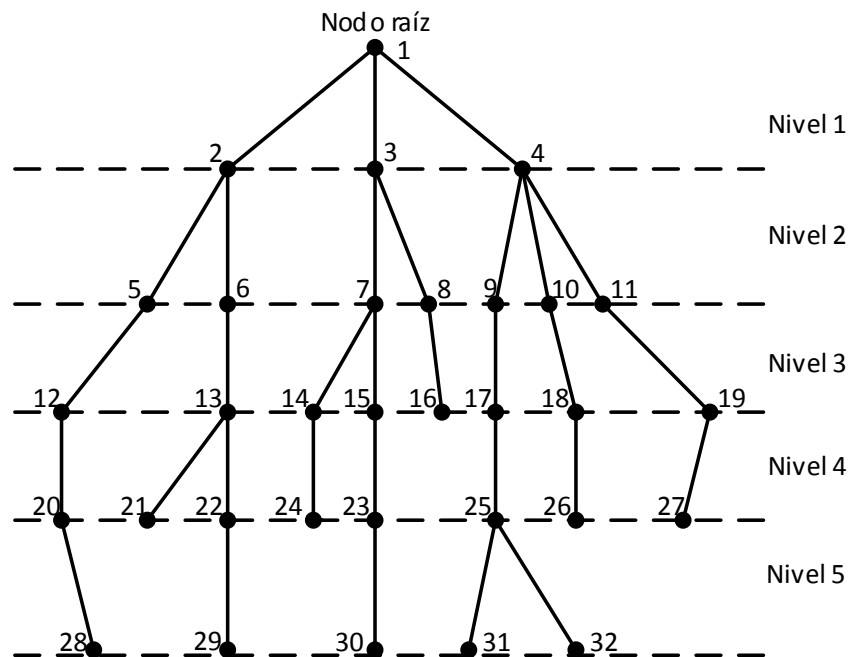


Figura 2.7 Red de distribución radial.

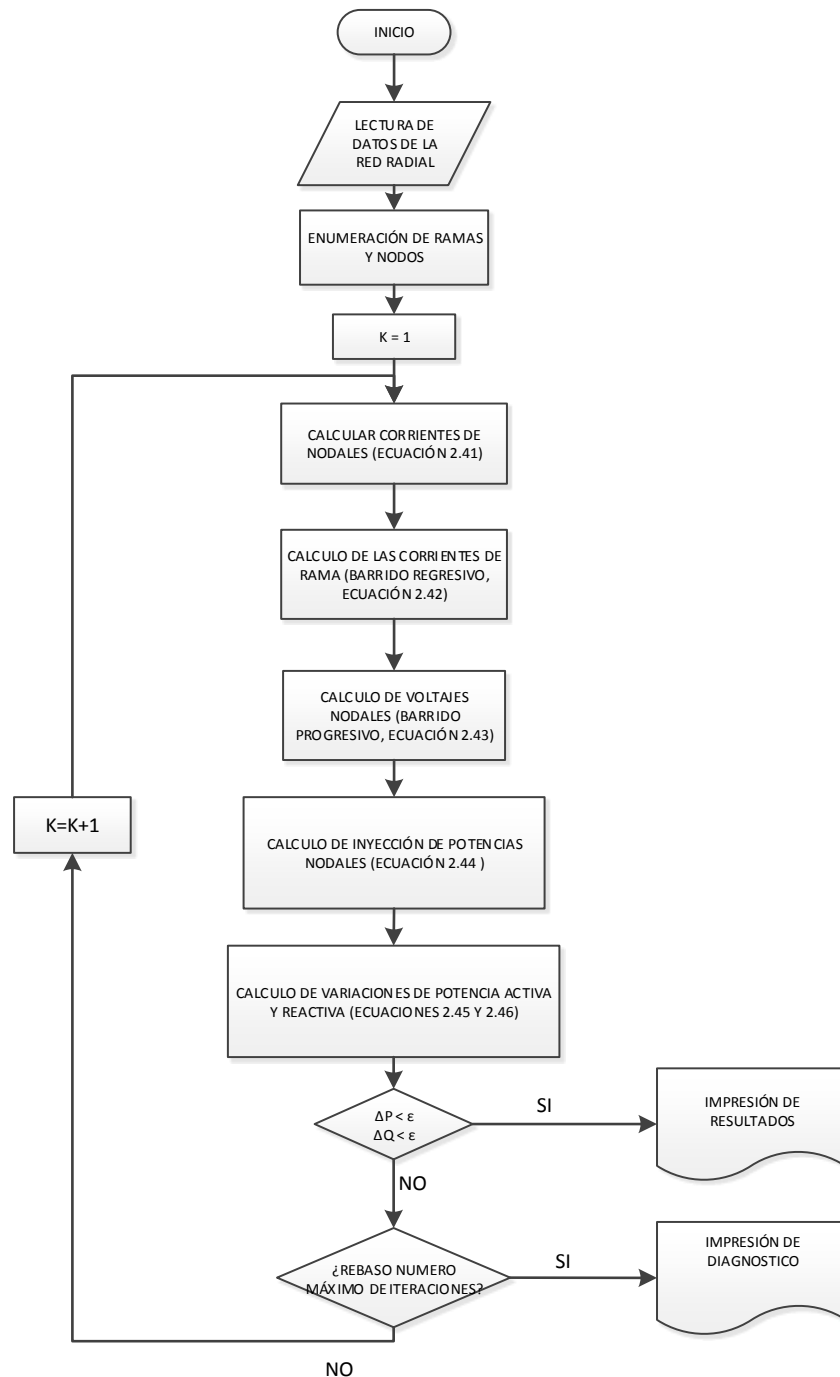


Figura 2.8 Diagrama de flujo para la solución de flujos de potencia para una red radial.

La figura 2.8 muestra el diagrama de flujo del método de solución para la obtención de flujos de potencia que consta básicamente de tres pasos. Partiendo de un perfil plano de voltajes, inicialmente se obtienen las corrientes nodales debido a las cargas y admitancias a tierra que se presente en cada uno de los nodos, posteriormente se obtienen las corrientes de rama mediante un barrido regresivo y finalmente se obtienen los voltajes nodales actualizados mediante un barrido progresivo partiendo del nodo raíz hasta la ultimo nivel de la red.

Calculo de corrientes nodales.

$$[\mathbb{I}_{abc}]_i^{(k)} = [\mathbb{I}_{abc}]_{carga\ i}^{(k)} + [\mathbb{I}_{abc}]_{derivacion\ i}^{(k)} \quad (2.41)$$

Donde:

$[\mathbb{I}_{abc}]_n^{(k)}$: Vector de corrientes totales inyectadas en el nodo n en la iteración k

$[\mathbb{I}_{abc}]_{carga\ n}^{(k)}$: Vector de corrientes que aporta la carga en el nodo n en la iteración k

$[\mathbb{I}_{abc}]_{derivacion\ n}^{(k)}$: Vector de corrientes que aportan los elementos en derivación en el nodo n en la iteración k

Barrido regresivo

$$[\mathbb{I}_{abc}]_{nm}^{(k)} = [d] \left([\mathbb{I}_{abc}]_m^{(k)} + \sum [\mathbb{I}_{abc}]_{mp}^{(k)} \right) \quad (2.42)$$

Donde:

$[\mathbb{I}_{abc}]_{nm}^{(k)}$: Vector de corrientes de la rama de los nodos n y m en la iteración k

$[d]$: Matriz constante, dependiendo del elemento que se encuentre conectado en los nodos n y m (línea, transformador o regulador de voltaje)

$[\mathbb{I}_{abc}]_m^{(k)}$: Vector de corrientes inyectadas en el nodo m (ecuación 2.48)

$\sum [\mathbb{I}_{abc}]_{mp}^{(k)}$: Sumatoria de los vectores de corrientes que se encuentran un nivel por debajo del nivel del nodo m

Barrido progresivo

$$[\mathbb{V}_{abc}]_m^{(k)} = [A][\mathbb{V}_{ABC}]_n^{(k)} - [B][\mathbb{I}_{abc}]_{nm}^{(k)} \quad (2.43)$$

Donde:

$[\mathbb{V}_{abc}]_m^{(k)}$: Vector de voltajes del nodo m en la iteración k

$[A]$, $[B]$: Matrices constantes de acuerdo al elemento conectado en los nodos n y m

$[\mathbb{V}_{ABC}]_n^{(k)}$: Vector de voltajes del nodo n en la iteración k

$[\mathbb{I}_{abc}]_{nm}^{(k)}$: Vector de corrientes de rama en la iteración k (ecuación 2.42)

Como se observa en la figura 2.8, el cálculo de corrientes nodales, el barrido regresivo y progresivo se repetirá hasta lograr la convergencia siempre y cuando no exceda el número de iteraciones máximas. Para lograr que el criterio de convergencia se cumpla, se utilizan las variaciones de la potencia activa y reactiva en los nodos de la red.

$$[\Delta S_{abc}]_m^{(k)} = \sum_{n=1}^n [\mathbb{V}_{abc}]_n^{(k)} [\mathbb{I}_{abc}]_m^{(k)*} - \sum_{n=1}^n [\mathbb{V}_{abc}]_n^{(k-1)} [\mathbb{I}_{abc}]_m^{(k-1)*} \quad (2.44)$$

Una vez conocida la potencia inyectada al nodo m, las variaciones de potencia se obtienen de la siguiente forma:

$$\Delta P_m^{(k)} = \text{Re}[S_m^{(k)} - S_m^{(k-1)}] \quad (2.45)$$

$$\Delta Q_m^{(k)} = \text{Im}[S_m^{(k)} - S_m^{(k-1)}] \quad (2.46)$$

2.3 Modelado de redes de alta tensión

El sistema eléctrico de potencia es un conjunto de elementos interconectados con el fin de transportar energía eléctrica desde su punto de generación, pasando por las etapas de transmisión y subtransmisión con niveles de tensión de 400KV – 230KV y 161KV – 69KV, para finalmente llegar a la etapa de distribución que cuenta con niveles de media tensión de entre 60KV – 2.4KV y de baja tensión con niveles de 0.220KV – 0.120KV.

La continuidad y calidad del servicio dependen directamente de la magnitud de voltaje y el valor de la frecuencia, los cuales se deben mantener dentro de una tolerancia del $\pm 10\%$ para la regulación del voltaje y de un $\pm 1\%$ para el control de frecuencia, además de que el suministro debe de ser confiable y lo más económico posible. Para poder cumplir con los requerimientos de un buen sistema eléctrico de potencia es indispensable contar con herramientas que nos permitan predecir el comportamiento bajo diferentes estados de operación, esto, con la finalidad de contar con estrategias y alternativas de operación que garanticen el menor número de interrupciones a usuarios.

Para el análisis del sistema eléctrico, se considera que los elementos de la red están operando de forma balanceada, lo cual puede ser una suposición válida solo para la etapa de transmisión, ya que en este caso se cumplen las restricciones de igualdad para voltajes, corrientes e impedancias:

$$\tilde{V}_A + \tilde{V}_B + \tilde{V}_C = 0 \quad (2.47)$$

$$\tilde{I}_A + \tilde{I}_B + \tilde{I}_C = 0 \quad (2.48)$$

$$\tilde{Z}_A = \tilde{Z}_B = \tilde{Z}_C \quad (2.49)$$

Suponiendo que el sistema cumple con dichas igualdades, el sistema puede ser modelado y resuelto mediante una representación monofásica (red de secuencia positiva), de no ser así se considera que estará operando en régimen desequilibrado y consecuentemente a esto, se tendrán que emplear métodos de análisis alternos, siendo el método de componentes simétricas el más común. Por otro lado, cerca del 99% del tiempo de operación el sistema de potencia se encuentra operando en un estado normal, significando esto que tanto la frecuencia y la magnitud de voltaje se encuentran oscilando dentro de sus márgenes de operación respectivamente.

El estado normal de operación debe de contar con un nivel de seguridad el cual permita afrontar pequeños disturbios que se pueden generar durante el tiempo de operación sin afectar la calidad y continuidad del servicio, de lo contrario, se deben de contar con acciones de control de acuerdo al tipo de falla que se presente, como pueden ser, desconexión de la sección fallada, reorientación de flujos de carga o incluso la desconexión de la carga por baja frecuencia.

Debido a la gran complejidad y tamaño de los sistemas eléctricos de potencia, este ha sido considerado como uno de los sistemas más grandes donde se pretende tener bajo control, derivado a ello, se hace indispensable contar con metodologías y técnicas computacionales que analicen el comportamiento de estos sistemas. Las mejores aproximaciones a las soluciones de los sistemas de potencia implican contar con un adecuado manejo de modelos matemáticos, así como también de herramientas computacionales, como son: algoritmos, lenguajes de programación, manejo de sistemas operativos, sistemas de adquisición y procesamiento de datos en tiempo real.

2.3.1 Representación de elementos de red de alta tensión

El primer paso para el correcto y confiable análisis de un sistema eléctrico de potencia se encuentra en la formación de un modelo capaz de representar las características de cada uno de los elementos que participan en la red, así como también de representar las relaciones que podrían existir entre cada uno de ellos.

Un modelo matricial de la red ya sea en forma de impedancias o de admitancias es el más conveniente para la obtención de la solución de flujos de potencia el cual nos permitirá obtener una relación voltaje-corriente de la red, este modelo puede obtenerse por simple

inspección de la topología en caso de que no se considere el efecto de las impedancias mutuas, de forma contraria este método no aplica y por lo tanto, será necesario hacer uso de otros, como el método de incorporación de elemento a elemento a partir de una red parcial o el método de transformaciones singulares. El método de transformaciones singulares representa un algoritmo muy sencillo basado en la combinación lineal de matrices de incidencia y primitivas de la red.

La representación de los elementos de la red presenta ser sencilla, aunque cabe mencionar que los modelos exactos y complejos se pueden limitar para objetivos muy especiales, donde la exactitud es clave, como es el caso de estudios de estabilidad. Los estudios de flujos de potencia suelen despreciar la resistencia ya que con ello solo se sacrifica un pequeño nivel de exactitud en la solución, pero se logra ahorro en los cálculos. Para el caso de los generadores, comúnmente en los estudios de flujos de potencia se representan como fuentes de P-V ó P-Q y suelen ser representadas por el voltaje interno en serie con la impedancia. En el caso de las líneas de transmisión, estas dependen de la longitud de la línea, si se trata de una línea corta donde su longitud es menor a 50 millas (80 Km) es posible representarla a través de una reactancia única en serie, más sin embargo si su longitud es mayor a 200 millas (320 Km) es posible modelarla por un circuito pi (figura 2.9). Es posible representar a los transformadores mediante su reactancia de corto circuito en serie con un transformador ideal que tome en cuenta la posición del cambiador de derivaciones.

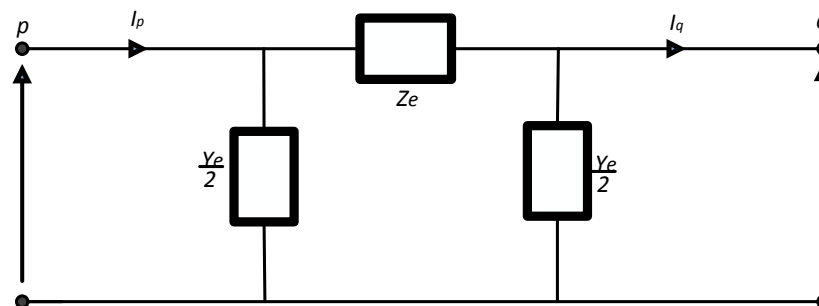


Figura 2.9 Modelo pi de secuencia positiva de línea de transmisión

2.4 Modelo matemático para flujos de potencia en redes de alta tensión

El planteamiento del análisis de flujos de potencia parte de la condición de que el sistema de potencia se encuentra operando normalmente en un estado de balance, cumpliendo con las condiciones de igualdad de las ecuaciones (2.47), (2.48) y (2.49), para lo cual, el problema se resume al análisis de una representación monofásica de secuencia positiva e inicialmente pareciendo ser un problema lineal, los siguientes factores demuestran lo contrario:

- Aunque por un lado se cuenta con una estimación de las potencias netas inyectadas, por el otro, se desconocen las corrientes netas inyectadas.

- Se desconocen las pérdidas del sistema y en consecuencia no se pueden conocer todas las potencias netas inyectadas, para que las ecuaciones de equilibrio se puedan cumplir.
- Puesto que la potencia aparente es el producto del voltaje por el conjugado de la corriente, lo que inicialmente presentaba ser un problema lineal resulta ser un problema no lineal.

Tomando en cuenta lo anterior y las variables especificadas en cada nodo, estos se pueden clasificar en 3 tipos de barras que ayudaran para la formulación y solución del estudio de flujos de potencia:

- Barra PQ: son aquellas barras en las que se especifica la potencia neta tanto activa como reactiva, las cuales son también llamadas como barras de carga.
- Barra PV: son aquellas en las cuales se especifica la potencia activa y la magnitud de voltaje, también llamadas como barra de voltaje controlado.
- Barra VØ: en esta barra se especifica el voltaje en magnitud y ángulo, es la encargada de proveer las potencias activas y reactivas para alimentar las perdidas y balances reactivos de transmisión, también es nombrada como barra compensadora.

En términos generales cualquier método de solución de flujos de potencia se basa en resolver las ecuaciones determinadas a partir de un conjunto de igualdades definidas como ecuaciones de inyección de potencia a la red, en donde para cualquier barra i , estas ecuaciones se pueden escribir como:

$$S_i = P_i + jQ_i = \tilde{V}_i * \tilde{I}_i^* \quad (2.50)$$

Desarrollando la ecuación (2.50), agrupando y separando términos reales e imaginarios se obtienen las siguientes ecuaciones para la potencia activa y reactiva de la red de alta tensión.

$$P_i = \sum_{jk} g_{kj} (V_k^2 - V_k V_j \cos(\delta_k - \delta_j)) - \sum_{jk} V_k V_j b_{kj} \sin(\delta_k - \delta_j) \quad (2.51)$$

$$Q_i = -V_k^2 Y_{ks} - \sum_{JK} V_k V_j g_{kj} \sin(\delta_k - \delta_j) - \sum_{JK} b_{kj} (V_k^2 - V_k V_j \cos(\delta_k - \delta_j)) \quad (2.52)$$

2.4.1 Algoritmo de Newton-Raphson desacoplado rápido para la solución de flujos de potencia

El crecimiento de los sistemas eléctricos de potencia trae consigo nuevos problemas para la solución de flujos de potencia, en cuanto al análisis, planificación y seguridad de las nuevas redes de transmisión, así como, en el aumento y lentitud para la solución computacional de flujos de potencia, generando que los tiempos de procesamiento sean inaceptables. Por tales motivos, con el fin de contar con nuevos algoritmos de solución, y considerando las características que presenta el sistema eléctrico en estado estacionario, como lo son, la gran dependencia que existe entre las potencias activas con los ángulos de fase (acoplamiento $P\theta$) y las potencias reactivas con la magnitud de voltaje (acoplamiento QV). Mientras que por otro lado, los acoplamientos PV y $Q\theta$ son bastante débiles, es por ello, que es posible contar con un desacople de las submatrices del jacobiano.

Reescribiendo las ecuaciones (2.51) y (2.52) de la siguiente forma:

$$\frac{P_i}{V_k} = V_k \sum_{jk} g_{kj} - \sum_{jk} V_j g_{kj} \cos(\delta_k - \delta_j) - \sum_{jk} V_j b_{kj} \sin(\delta_k - \delta_j) \quad (2.53)$$

$$\frac{Q_i}{V_k} = -V_k \left(\sum_{jk} b_{kj} + Y_{ks} \right) - \sum_{jk} V_j g_{kj} \sin(\delta_k - \delta_j) + \sum_{jk} V_j b_{kj} \cos(\delta_k - \delta_j) \quad (2.54)$$

De la ecuación (2.53) se tiene que el término dominante es el siguiente

$$- \sum_{jk} V_j b_{kj} \sin(\delta_k - \delta_j) \quad (2.55)$$

Este término puede ser simplificado a consideraciones prácticas de operación de un sistema eléctrico de potencia [11].

$$V_j = 1.0 \quad (2.56)$$

$$\sin(\delta_k - \delta_j) = (\delta_k - \delta_j) \quad (2.57)$$

De modo que el término dominante de la ecuación de potencia activa queda de la siguiente forma:

$$- \sum_{jk} b_{kj} (\delta_k - \delta_j) \quad (2.58)$$

Para la ecuación (2.54) se tiene que el término dominante es el siguiente:

$$-V_k Y_{ks} \sum_{jk} b_{kj} (V_k - V_j \cos(\delta_k - \delta_j)) \quad (2.59)$$

De igual forma considerando la siguiente aproximación:

$$\cos(\delta_k - \delta_j) = 1.0 \quad (2.60)$$

El termino dominante queda de la siguiente forma:

$$-V_k Y_{ks} \sum_{jk} b_{kj} (V_k - V_j) \quad (2.61)$$

Colocando los términos dominantes de las ecuaciones (2.58) y (2.61) de potencia activa y reactiva, se obtiene la siguiente expresión:

$$\frac{P_i}{V_k} = - \sum_{jk} b_{kj} (\delta_k - \delta_j) \quad (2.62)$$

$$\frac{Q_i}{V_k} = -V_k Y_{ks} \sum_{jk} b_{kj} (V_k - V_j) \quad (2.63)$$

Tomando en cuenta los supuestos anteriores es posible reescribir el sistema de ecuaciones (2.62) y (2.63) de la siguiente forma

$$[\Delta P^k / V^k] = [B'] \cdot [\Delta \delta^k] \quad (2.64)$$

$$[\Delta Q^k / V^k] = [B''] \cdot [\Delta V^k] \quad (2.65)$$

El modelo B' está formado por la parte imaginaria de la matriz de admitancias nodales de la red, sin considerar conexiones a nodos compensadores, excepto en los elementos diagonales. Generalmente, se simplifica la construcción de estos modelos despreciando las resistencias de los elementos de transmisión. Los elementos de B'' son elementos de la parte imaginaria de la matriz de admitancias nodal, despreciando las filas y columnas correspondientes a compensadores y nodos generadores. Demostrando finalmente, que los modelos B' y B'' son una aproximación a las submatrices diagonales H y L usados en el método de Newton Raphson [1]. Por lo que, se puede entender que el método Desacoplado Rápido es una aproximación al método de Newton Raphson considerando un Jacobiano Desacoplado y Constante.

La figura 2.10 muestra el diagrama de flujo para la solución de flujos de potencia por el método desacoplado rápido, en él se observa, que durante el proceso de solución los modelos de B' y B'' , permanecen constantes durante todo el proceso, al igual que la matriz de impedancias (Y_{bus}), por lo que durante todo el proceso iterativo, las ecuaciones (2.64) y

(2.65), se actualizarán en cada vuelta siempre y cuando no se rebase la tolerancia de variaciones máximas de potencia activa y reactiva.

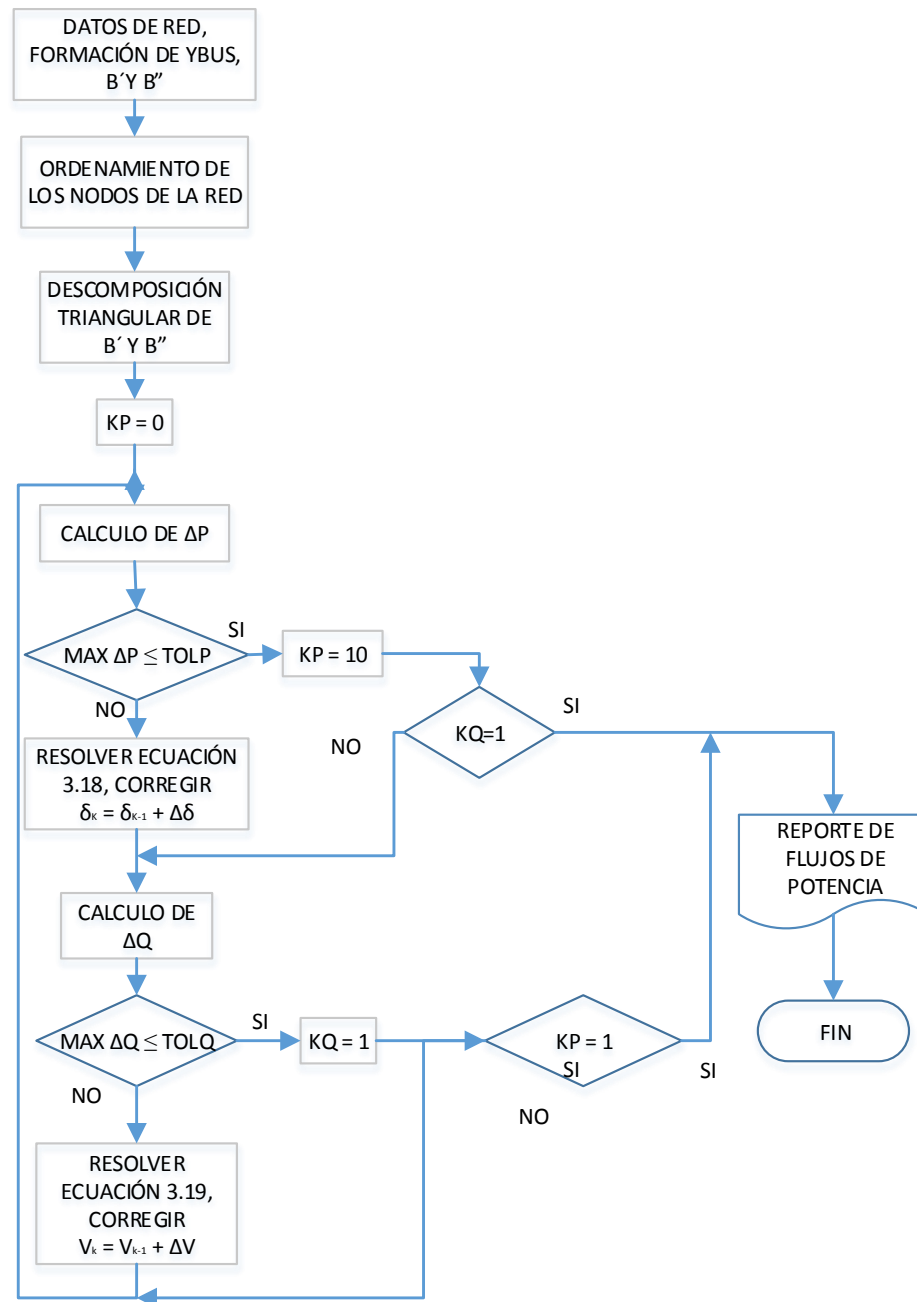


Figura 2.10 Diagrama de flujo de método desacoplado rápido [28]

CAPITULO 3

ESTUDIO DE FLUJOS DE POTENCIA

El estudio más frecuente en un sistema eléctrico, ya sea de transmisión o distribución, lo constituye el cálculo de las condiciones de operación en régimen permanente. En estos cálculos interesa determinar los voltajes en las distintas barras de la red, flujo de potencia activa y reactiva y las pérdidas en los transformadores y líneas de transmisión y distribución.

Los estudios de flujos de potencia son llevados a cabo para verificar la operación de un sistema ya existente y comparar nuevas alternativas de adición de fuentes y cargas al sistema, y comprobar el rendimiento del mismo. Generalmente los estudios realizados en la ingeniería de potencia poseen un conjunto de criterios que deben ser cumplidos, como lo son:

- La generación debe de ser igual a la demanda en cada momento y esta potencia debe de ser dividida entre los generadores en un promedio único, a fin de mantener una operación económica, se concluye entonces que la aportación de generación de cada unidad, debe ser mantenida en límites muy restringidos. Es importante recordar que la demanda del sistema presenta cambios permanentes a lo largo del día, de modo que la generación debe de adaptarse continuamente, prediciendo a corto plazo la variación de la carga y cambiando consecuentemente la configuración de los flujos de potencia.
- Los enlaces de líneas de transmisión tienen límites de cargabilidad en cuanto a los aumentos de potencia transmitida, debiendo tener cuidado de operar dentro de los límites térmicos o de estabilidad.
- Es necesario mantener los niveles de voltaje de la barras del sistema eléctrico dentro de rangos reducidos.
- Si el sistema bajo análisis es solo una parte de un gran sistema, se debe tener un estricto control en los enlaces sobre los niveles de intercambio de energía entre las áreas.
- Los disturbios posteriores a la presencia de una falla severa en la red, pueden ocasionar salida de sectores importantes, este efecto puede minimizarse con una adecuada estrategia de flujos de prefalla.
- Los análisis de flujos de potencia son vitales en la determinación de estados de operación, en la planeación de nuevos sistemas o para el crecimiento y fortalecimiento de los ya existentes.

3.1 Etapas de un estudio de flujos de potencia

El problema de flujos de potencia puede ser dividido en los siguientes sub problemas, los cuales pueden ser tratados en forma separada.

- Formulación de un modelo matemático confiable del sistema. El modelo empleado para la solución de flujos de potencia debe describir adecuadamente las relaciones entre voltajes y potencias en el sistema.
- Especificaciones de las restricciones de potencia y voltaje que deben de ser aplicadas a los nodos del sistema.
- Cálculo numérico de las ecuaciones de voltaje sujetas a restricciones establecidas. Estos cálculos deben de proporcionar con suficiente exactitud los valores de voltaje en todos los nodos.
- Una vez que todos los voltajes han sido determinados, se debe finalmente de calcular los flujos de potencia en todos los nodos de la red.

3.2 Características del sistema de alta y media tensión

Los sistemas de distribución son típicamente radiales, lo que significa que el flujo de potencia proviene de un solo nodo, esto es donde se encuentra ubicada la subestación de distribución, esta subestación se encarga de reducir el voltaje de un nivel de alta tensión a un nivel de media tensión. Generalmente, en el lado secundario del transformador de distribución se encuentra el cambiador de derivaciones o en su defecto un banco de capacitores, los cual es de gran ayuda en las horas pico, ya que es ahí donde fluye una mayor cantidad de corriente por la líneas, provocando una mayor caída de tensión y es ahí donde tanto el cambiador de derivaciones o el banco de capacitores atenúan dicho problema.

La distribución se hace a nivel de media o baja tensión, logrando encontrar diferentes tipos de conexiones, pero no es hasta en las cargas residenciales donde existe un mayor problema, debido en su mayoría a las cargas residenciales de naturaleza monofásica, la cual trata de amortiguarse repartiendo equitativamente las cargas en la tres fases del sistema.

Contrario a lo que sucede en sistemas de transmisión, en distribución, la resistencia de las líneas es comparable a su reactancia, generalmente la razón X/R tiene un amplio rango de variaciones, pudiendo llegar a ser bastante menor que uno. Por otro lado, en distribución no existe transposiciones, debido a que las líneas son cortas (menores a 50 Km), esto motiva a que las caídas de tensión, debido a los acoplamientos entre las fases, sean desequilibradas.

La naturaleza desbalanceada del sistema de distribución no hace atractiva la transformación a componentes simétricas, la posibilidad de desacoplar y modelar el sistema como una red de secuencia hace más complejo el análisis de flujo de potencia. En [29] se emplea un método para tratar los efectos mutuos, en este método la línea trifásica se reduce de cuatro conductores a uno, suponiendo retornos por tierra, en donde los efectos mutuos han sido incorporados a las admitancias propias de la línea, con este modelo reducido se obtienen cuatro hilos independientes entre si. El problema de este método, son las muchas aproximaciones que se deben hacer para llegar al modelo reducido, además de ser un trabajo antiguo y no contar con las tendencias de los trabajo actuales. Por lo tanto, es

necesario resolver este problema sobre una base trifásica, con los componentes modelados en forma exacta por fase.

3.3 Desventajas de métodos de solución de flujos de potencia de sistemas de alta tensión sobre sistemas de media tensión

Como ya se ha mencionado en capítulos anteriores, los métodos tradicionales (Gauss-Seidel, Newton-Raphson y versiones desacopladas) para la solución de flujos de potencia no arrojan resultados satisfactorios al ser aplicados en sistemas de media tensión.

El método Gauss-Seidel indirecto se caracteriza por ser relativamente insensible a las tensiones iniciales estimadas, a su simple programación, su poco requerimiento de memoria debido a la característica radial de la red y por la matriz de admitancias nodales que es muy dispersa, así como, por su lenta convergencia debida a la gran dispersión de la matriz de admitancias nodales, haciéndolo poco atractivo para la solución de flujos de potencia. La característica radial de los sistemas de media y baja tensión, incide en que los valores de la diagonal de la matriz de admitancias nodales sean pequeñas y de que el perfil de tensiones en cada iteración sea inversamente proporcional, generando grandes oscilaciones de la tensión y haciendo lenta la convergencia.

Por otro lado, se tiene el método de Newton-Raphson completo y sus versiones desacopladas, estos métodos son ampliamente conocidos por sus excelentes características de convergencia para sistemas de alta tensión. La mayor desventaja que se tiene del método Newton-Raphson completo sobre redes de media tensión, es el cálculo e inversión para cada iteración de la matriz Jacobiana, que es aproximadamente igual a cuatro veces el tamaño de la matriz de admitancias nodales, aunque para la característica de dispersión de la matriz jacobiana pueden ser aplicadas técnicas de bifactorización, en su inversión para reducir tiempos de procesamiento resulta ser poco práctico, ya que en sistemas radiales los tiempos de solución resultan ser excesivos [2]. Para las versiones desacopladas se contemplan una serie de aproximaciones que simplifican la matriz jacobiana, logrando con esto que los tiempos de solución se reduzcan, sin embargo estas aproximaciones consideran un alto valor de X/R , lo cual no es efectivo para sistemas de media tensión, por lo tanto, el método de Newton-Raphson como sus versiones desacopladas resultan poco atractivos para la solución de flujos de potencia para redes de media tensión.

De acuerdo a lo anterior y a lo expuesto en [3] se opta por emplear el algoritmo de barrido regresivo-progresivo [2] para la solución de flujos de potencia en sistemas de media tensión.

3.4 Metodología para el acoplamiento del sistema de media y alta tensión

Durante años, el sistema de gestión de energía y el sistema de gestión de distribución se han desarrollado como dos sistemas eléctricos distintos, por un lado las funciones del sistema de gestión de energía implica la supervisión y control de plantas de energía y transmisión de alta/media tensión, mientras que por otro lado las funciones del sistema de

gestión de distribución se concentran en las inyecciones de potencia a niveles de tensión medio/bajo.

Un factor importante a considerar es el tamaño del sistema eléctrico completo, el cual es un factor importante para los tiempos de respuesta ante algún tipo de acción de posibles contingencias, ya que para un sistema típico de medio tamaño, el número de nodos de un sistema de transmisión oscila entre los 200 y 500 nodos, con la inclusión del sistema de distribución el número de nodos aumentan significativamente. En general, por naturaleza el sistema de control y técnicas de optimización de las redes de transmisión y distribución son diferentes pero van enfocadas a un objetivo común que es la minimización de pérdidas y costos de producción, aunque el sistema de transmisión cuenta con diversas formas de control remoto, en sistema de distribución, los dispositivos de control rara vez son directamente controlables, haciendo necesario que el control se aplique de forma física en el punto indicado. Como resultado del gran tamaño, es necesario establecer aplicaciones para la comunicación y el almacenamiento de parámetros, mediciones y datos necesarios para otras aplicaciones

Para lograr el enlace entre ambos sistemas, se debe contar con un correcto modelo de la red, considerando que el sistema de transmisión hace uso de transformadores de distribución con cambiadores de tap bajo carga [15], con el fin de obtener una red balanceada y conseguir una representación monofásica, por tal motivo es que se debe de dar gran importancia a los elementos comunes entre el sistema de transmisión y el sistema de distribución, esto es, el transformador de la subestación de distribución que juega un papel importante para el enlace entre ambos sistemas ya que por el lado de distribución es el encargado de proporcionar un control para la tensión del alimentador primario

3.5 Criterios para enlace de sistemas de alta y media tensión

3.5.1 Componentes simétricas

Los sistemas de transmisión, por razones económicas y técnicas son considerados como sistemas trifásicos simétricos, ya que en condiciones balanceadas de operación son sistemas trifásicos balanceados, mientras que el sistema de distribución debido a constantes cambios de las cargas, se puede considerar como un sistema asimétrico, por tal motivo, se hace necesario el empleo de componentes simétricas para el análisis del enlace entre ambos sistemas.

Considerando un sistema trifásico y aplicando lo establecido en [15] se tiene que los tres fasores de un sistema desequilibrado (V_a , V_b , V_c de secuencia abc) puede descomponerse en los tres fasores equilibrados siguientes:

- Componentes de secuencia positiva. Están compuestos por tres fasores de igual magnitud, desplazados un ángulo de 120° entre si y poseen la misma secuencia de los fasores originales (figura 3.1A)
- Componentes de secuencia negativa. Formado por tres fasores de igual magnitud, desfasados 120° entre si y con secuencia de fase opuesta a los fasores originales (figura 3.1B)
- Componentes de secuencia cero. Está formado por tres fasores de igual magnitud y sin ninguna diferencia angular (figura 3.1C)

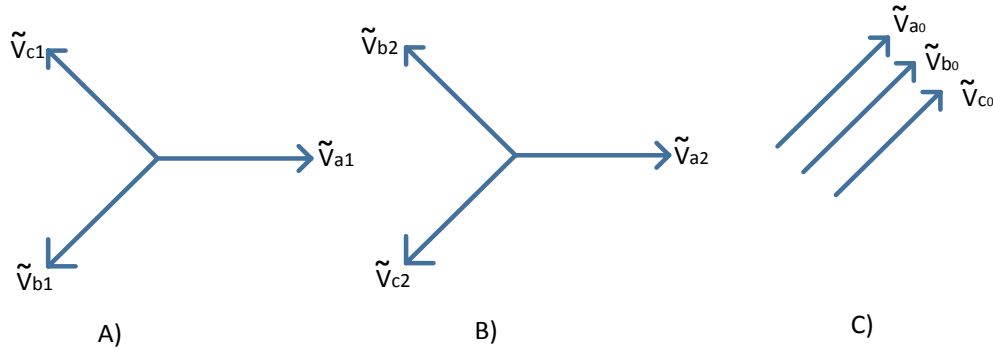


Figura 3.1 Componentes simétricas

Al tener un conjunto de fasores asimétricos (sistema trifásico) es posible descomponer en sus componentes simétricos, como un producto del operador a [1] y un componente V_a , de esta forma las componentes de secuencia positiva pueden expresarse de la siguiente manera:

$$\begin{aligned} V_{a1} &= V_{a1} \\ V_{b1} &= a^2 \cdot V_{a1} \\ V_{c1} &= a \cdot V_{a1} \end{aligned} \quad (3.1)$$

En el caso de las componentes de secuencia negativa, sus componentes se pueden expresar de la siguiente forma:

$$\begin{aligned} V_{a2} &= V_{a2} \\ V_{b2} &= a \cdot V_{a2} \\ V_{c2} &= a^2 \cdot V_{a2} \end{aligned} \quad (3.2)$$

Y finalmente las componentes de secuencia negativa es posible expresarlas de la siguiente forma:

$$\begin{aligned} V_{a0} &= V_{a0} \\ V_{b0} &= V_{a0} \end{aligned} \quad (3.3)$$

$$V_{c0} = V_{a0}$$

De acuerdo con el teorema de Fortescue la suma de las componentes simétricas de cada fase da como resultado el fasor desequilibrado original:

$$\begin{aligned} V_a &= V_{a0} + V_{a1} + V_{a2} \\ V_b &= V_{b0} + V_{b1} + V_{b2} \\ V_c &= V_{c0} + V_{c1} + V_{c2} \end{aligned} \quad (3.4)$$

Sustituyendo (3.1), (3.2), (3.3) y (3.4) queda el siguiente sistema:

$$\begin{aligned} V_a &= V_{a0} + V_{a1} + V_{a2} \\ V_b &= V_{a0} + a^2 V_{a1} + a V_{a2} \\ V_c &= V_{a0} + a V_{a1} + a^2 V_{a2} \end{aligned} \quad (3.5)$$

Reescribiendo el sistema de ecuaciones (3.5) en forma matricial, queda de la siguiente manera:

$$\begin{bmatrix} V_a \\ V_b \\ V_c \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & a^2 & a \\ 1 & a & a^2 \end{bmatrix} \begin{bmatrix} V_0 \\ V_1 \\ V_2 \end{bmatrix} \quad (3.6)$$

De la ecuación (3.6) se puede denotar a los fasores desequilibrados originales, a la matriz de transformación y a sus componentes simétricas de la siguiente forma:

$$V_{abc} = A \cdot V_{012} \quad (3.7)$$

De la ecuación (3.7) se pueden conocer los fasores simétricos de un sistema trifásico desequilibrado, si se multiplica por la matriz inversa de transformación de ambos lados es posible saber los fasores desequilibrados originales de un conjunto de fasores equilibrados.

$$V_{012} = A^{-1} \cdot V_{abc} \quad (3.8)$$

Donde:

$$A^{-1} = \frac{1}{3} \begin{bmatrix} 1 & 1 & 1 \\ 1 & a & a^2 \\ 1 & a^2 & a \end{bmatrix}$$

El uso de componentes simétricas es una gran herramienta para el análisis y enlace de ambos sistemas, ya que parte del objetivo no solo consistía en obtener el enlace entre una red de distribución balanceada con el sistema de transmisión, sino también contar con sistemas de distribución desbalanceados, ya que teniendo un sistema balanceado, su representación puede ser sencilla, sin importar el punto desde donde se mire, ya sea solo representando una fase del sistema trifásico o transformado a sus componentes simétricas

la representación es monofásica, caso contrario con un sistema de distribución desbalanceado que al tener originalmente un sistema asimétrico es mucho más conveniente trabajar con las componentes simétricas de este. En la figura 3.2 se puede tener una mejor perspectiva del problema.

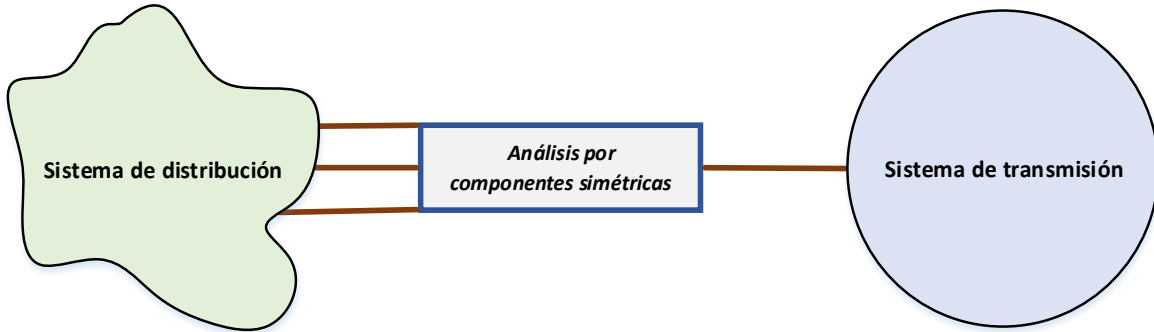


Figura 3.2 Representación gráfica para el enlace de ambos sistemas

Como se observa en la figura 3.2 del lado izquierdo se tiene representado un sistema asimétrico, del cual se tiene, como dato inicial las lecturas de tensiones del lado secundario del transformador de distribución, potencias activas y reactivas en los nodos de carga, por el contrario, del lado derecho se tiene representado un sistema completamente simétrico, del cual se cuenta, con datos iniciales las potencias de carga y generación de acuerdo al tipo de nodo con que se cuenta. Cada sistema obtiene los resultados de flujos de potencia por métodos numéricos diferentes expuestos en capítulos anteriores, por lo que es importante recalcar, que el sistema de distribución es resuelto en coordenadas de fase (V_{abc}) mientras que el sistema de transmisión es resuelto en componentes simétricas (V_{012}) debido a las justificaciones antes mencionadas.

El bloque de análisis por componentes simétricas de la figura 3.2 se presenta en forma de diagrama de flujo y el cual es implementado en el programa ACOPLAMIENTO.90, el análisis se realiza una vez que se han obtenido los primeros flujos de potencia del sistema de distribución, que servirán como una estimación de la carga ante la condición inicial de tensión del nodo raíz.

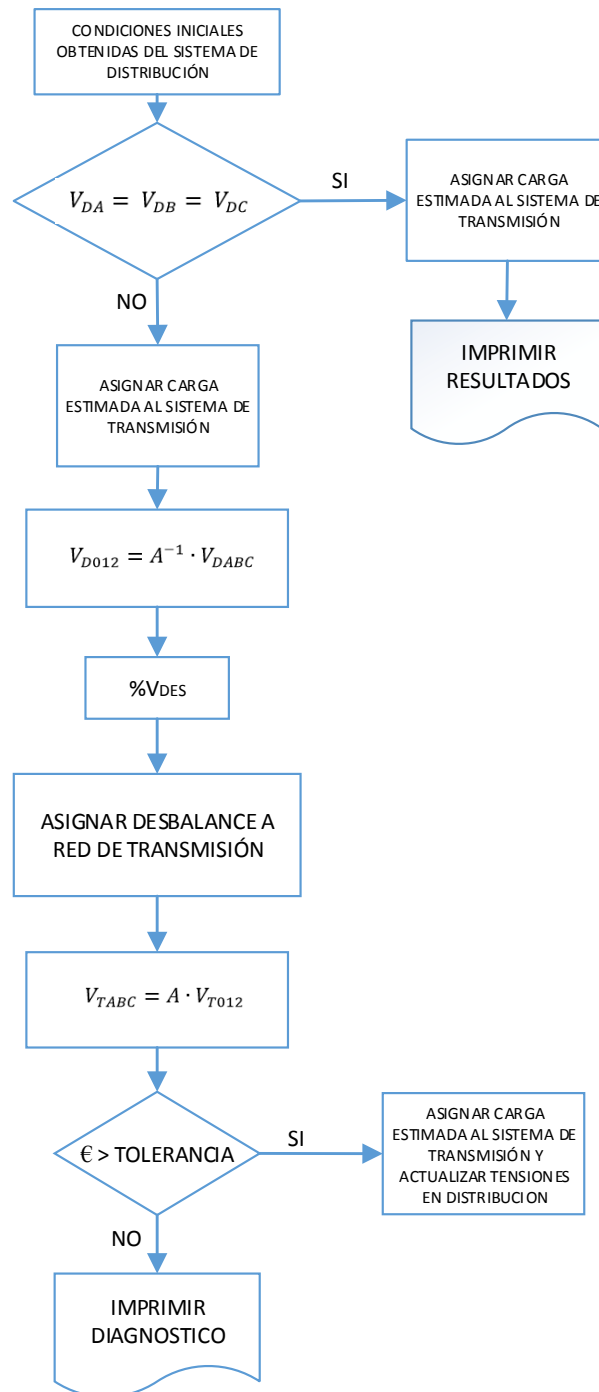


Figura 3.3 Diagrama de flujos para análisis de componentes simétricas.

La figura 3.3 presenta el diagrama de flujos para el análisis de componentes simétricas para el acoplamiento del sistema de alta con el de media tensión, como primer paso se tienen que obtener las condiciones iniciales de la red de media tensión, una vez obtenidas existen dos casos. El primero de ellos, donde los voltajes del nodo raíz son balanceados y de ser así el proceso de acoplamiento resulta ser algo sencillo, ya que las condiciones iniciales de

potencia activa y reactiva suministradas a la red de media tensión pasan directamente al nodo de carga donde se encuentra ubicada la red de media tensión y de esta forma obtener la solución de ambos sistemas de forma conjunta. El segundo caso, se da cuando se presentan voltajes de nodo raíz desbalanceados, de esta forma el proceso iterativo para la solución del acoplamiento de ambos sistemas resulta no ser tan sencillo, ya que primero se deben de obtener sus componentes simétricas de los voltajes de nodo raíz y de esta forma obtener el porcentaje de desbalance, este desbalance es asignado a los voltajes obtenidos de la solución de flujos de potencia de la red de alta tensión y de nuevo son transformados a voltajes de fase, de esta forma son actualizados los voltajes del nodo raíz para la segunda iteración para la solución del acoplamiento de los sistemas de alta y media tensión.

3.5.2 Transformador de distribución

En un sistema trifásico de distribución, los devanados que se encuentran conectados a las líneas a, b y c pueden presentar tanto una configuración Δ (Delta) o Y (estrella), de lo cual se puede establecer de la conexión Δ lo siguiente en base a las corrientes de línea, como se observa en la figura 2.3.

$$I_a = I_{ab} - I_{ca} \quad (3.9)$$

$$I_b = I_{bc} - I_{ab} \quad (3.10)$$

$$I_c = I_{ca} - I_{bc} \quad (3.11)$$

Sumando las ecuaciones (3.9), (3.10), (3.11) y obteniendo el valor para de la corriente de secuencia cero de la misma forma que en la ecuación 3.8 se obtiene lo siguiente:

$$I_{a0} = \frac{1}{3}(I_a + I_b + I_c) \quad (3.12)$$

Sustituyendo el valor de la suma de corrientes de línea en la ecuación (3.12) se obtiene:

$$I_{a0} = \frac{1}{3}((I_{ab} - I_{ca}) + (I_{bc} - I_{ab}) + (I_{ca} - I_{bc})) = 0 \quad (3.13)$$

Lo anterior indica que las corrientes de línea de secuencia cero para una conexión Δ no existen, sabiendo esto se puede establecer lo siguiente para las componentes simétricas de la corriente de línea de la fase a.

$$I_{a1} + I_{a2} = (I_{ab0} + I_{ab1} + I_{ab2}) - (I_{ca0} + I_{ca1} + I_{ca2}) \quad (3.14)$$

Agrupando términos semejantes y poniendo en función de las corrientes de fase ab las corrientes de fase ca se puede establecer:

$$I_{a1} + I_{a2} = (1 - a)I_{ab1} + (1 - a^2)I_{ab2} \quad (3.15)$$

Igualando términos semejantes se obtiene las corrientes de línea en función de las corrientes de fase para la conexión delta.

$$I_{a1} = (\sqrt{3} \angle -30^\circ) I_{ab1} \quad (3.16)$$

$$I_{a2} = (\sqrt{3} \angle 30^\circ) I_{ab1} \quad (3.17)$$

De igual forma, se pueden establecer las mismas condiciones para los voltajes de fase para la conexión delta, donde se puede concluir que los voltajes de línea a línea no cuentan con componentes de secuencia cero.

Dentro de los diferentes tipos de configuraciones que se tiene para las conexiones estrella y delta en los transformadores trifásicos de distribución, el análisis de los circuitos equivalentes dependen de las conexiones de las diferentes combinaciones de las conexiones de sus devanados, los cuales determinan las configuraciones de los circuitos de secuencia cero y el defasamiento entre los circuitos de secuencia positiva y negativa del transformador. Se consideran entre las conexiones más usuales para transformadores de distribución la configuración estrella-delta (figura 3.4) y estrella delta aterrizada (figura 3.5).

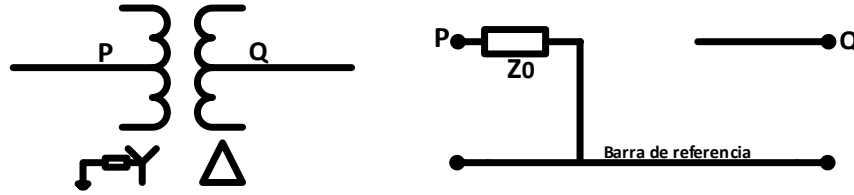


Figura 3.4 Equivalente de secuencia cero para conexión estrella aterrizada-delta

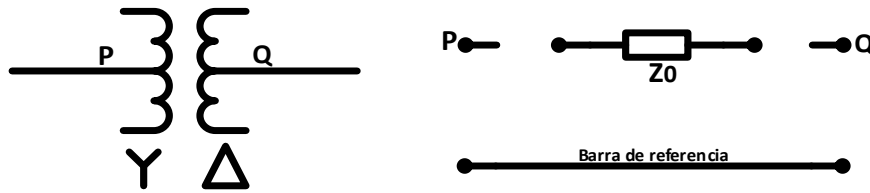


Figura 3.5 Equivalente de secuencia cero para conexión estrella-delta

Para el caso de la figura 3.4 cuando el neutro se encuentra aterrizado, las corrientes de secuencia cero tiene una trayectoria a tierra a través de la estrella, caso contrario para el lado de la conexión delta, aunque existen corrientes de secuencia cero en los devanados, estas se encuentran equilibradas magnéticamente, lo que hace que estas no circulen en las líneas conectadas a ella. Se pueden expresar los voltajes del lado de la conexión estrella en función de la delta de la siguiente manera.

$$V_{A0} + V_{A1} + V_{A2} = \frac{N_1}{N_2} V_{ab0} + \frac{N_1}{N_2} V_{ab1} + \frac{N_1}{N_2} V_{ab2} + 3Z_N I_{A0} \quad (3.18)$$

Agrupando términos semejantes, se obtiene los siguientes resultados.

$$V_{A0} - 3Z_N I_{A0} = \frac{N_1}{N_2} V_{ab0} = 0 \quad (3.19)$$

$$V_{A1} = \frac{N_1}{N_2} V_{ab1} = \left(\frac{N_1 \sqrt{3}}{N_2} \angle 30^\circ \right) V_{a1} \quad (3.20)$$

$$V_{A2} = \frac{N_1}{N_2} V_{ab2} = \left(\frac{N_1 \sqrt{3}}{N_2} \angle 30^\circ \right) V_{a2} \quad (3.21)$$

En el caso especial donde no se encuentra aterrizada la estrella (figura 3.5), se considera que se tiene una impedancia que tiende a infinito, generando que las corrientes de secuencia cero no puedan fluir en los devanados del transformador, mientras que los voltajes de secuencia positiva y negativa se siguen comportando de la misma forma que en las ecuaciones (3.20) y (3.21).

Tomando en cuenta que en los sistemas de distribución las cargas conectadas en el secundario del transformador de distribución presentan ser desequilibradas, en [17] se analizan los comportamientos de cada tipo de conexión normalizada de un transformador de distribución ante cargas desequilibradas entre fase/neutro y entre fases, en donde se tienen importantes conclusiones como son:

Para desequilibrio entre fases se tiene:

- Ante desequilibrio entre fases, todas las configuraciones entre las conexiones Δ/Y de los devanados del transformador tiene un comportamiento adecuado ante cargas desequilibradas.
- Para conexiones estrella-estrella con desfase de 0° y 180° , cuando se presentan cargas desequilibradas en el secundario del transformador, el desequilibrio se presenta sobre dos de los tres conductores del lado primario.
- Para conexiones delta-estrella con desfase de 150° y 330° , el desequilibrio se presenta, los tres conductores del lado primario se encuentran desequilibrados.

Para desequilibrio entre fase/neutro se tiene:

- El comportamiento ante la configuración Yy es inadecuado, esto debido a la existencia de componentes de secuencia cero y a la disposición de las corrientes en el lado primario.
- En caso de hacer uno de la conexión Yy es necesario incluir un devanado terciario, de esta forma si este devanado es conectado en delta eliminara o anulara la presencia de las componentes de secuencia cero y de esta forma mejorar su comportamiento ante cargas desequilibradas.
- La conexión Dy presenta ser la conexión más adecuada ante esta condición de desequilibrio.

Tomando en cuenta las conclusiones obtenidas de [17] se puede decir que las configuraciones Yy reflejan por completo el desbalance al lado primario del transformador, además de presentar un mal comportamiento tanto para desequilibrios entre fases como entre fase y neutro, cosa contraria con la configuración Dy en donde el desequilibrio presentado por las cargas disminuye en gran sentido del lado primario del transformador.

Uno de los criterios para realizar el enlace entre el sistema de distribución y el sistema de transmisión es considerar el promedio de las tensiones de fase del lado primario del transformador y poder ser comparado con las tensiones de fase del sistema de transmisión, ya que es ahí hasta donde el sistema de transmisión se considera como balanceado.

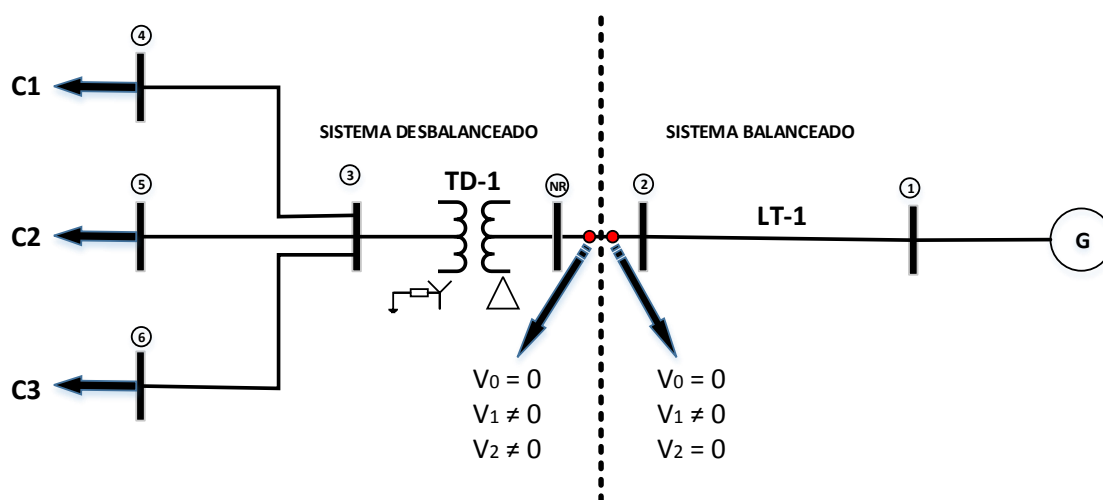


Figura 3.6 Diagrama unifilar para el análisis de enlace entre sistema de media y alta tensión

Como se observa en la figura 3.6 el criterio utilizado para realizar la comparación entre las tensiones de fase entre ambos sistema se considera como correcta, ya que se basa en los siguientes puntos.

- El tipo de conexión de los devanados del transformador de distribución hacen que ante el desbalance de las cargas presentes en el lado secundario del transformador, los devanados primarios disminuyan en gran medida el desbalance de tensiones presentado por las cargas en el lado secundario.
- Tomando en cuenta que es poco probable que exista un porcentaje de desbalance de voltajes elevado en el lado secundario del transformador, se puede pensar que el transformador de distribución estará operando con desbalance de tensiones menores al 2%, de esta forma este desbalance se reducirá en el lado primario del transformador.
- Como se observa en la figura 3.6 el nodo raíz (NR) es un nodo ficticio, el cual no pertenece físicamente a la red, al observar las componentes simétricas de las tensiones se observa que al hacer la transición al análisis de un sistema de

transmisión solo eliminan la componente de secuencia negativa, ya que relativamente es pequeña.

De esta forma, al resolver el sistema de distribución, realizar el promedio de las tensiones del lado primario de transformador y hacer la comparación con las tensiones del sistema de transmisión, la componente de secuencia negativa no se elimina ya que al realizar la actualización de las tensiones del nodo raíz con las tensiones resultantes del sistema de transmisión se considera el desbalance constante durante todo el proceso iterativo.

3.5.3 Desbalance de tensión

El desbalance de tensiones se presenta cuando las magnitudes de los fasores de tensión y/o ángulos entre fases no son iguales, esto es generado por el continuo cambio de las cargas y a la existencia de cargas monofásicas presentes a lo largo de la red de distribución, por lo que es prácticamente imposible obtener un balance perfecto de las tensiones de fase.

Como ya se mencionó en secciones anteriores, un sistema simétrico o balanceado es aquel donde las tensiones tienen el mismo valor en magnitud y sus fasores se encuentran separados 120° entre sí, por otro lado los sistemas desbalanceados no cumplen con estas condiciones por lo que el análisis de este tipo de sistemas se realiza en componentes simétricas (figura 3.1).

El análisis en componentes simétricas nos permite determinar el flujo de potencia que proviene de la red hacia la carga (componente de secuencia positiva), el desbalance existente en el sistema, o sea, la falta de simetría entre los fasores de tensión en un punto determinado (componente de secuencia negativa) y las conexiones respecto a tierra (componente de secuencia cero).

Aunque existen diversas formas de obtener el desbalance de tensiones, en general, expresa la relación que existe entre la componente de secuencia positiva y la componente de secuencia negativa.

$$\%V_{DES} = \frac{|V_2|}{|V_1|} * 100 \quad (3.22)$$

Por otro lado en [18], [21] y [22] se establecen formas alternas de obtener el desbalance de tensión en base a mediciones de tensión de línea de la siguiente manera:

$$\%V_{DES} = \frac{\sqrt{1-\sqrt{3-(6*\beta)}}}{\sqrt{1+\sqrt{3-(6*\beta)}}} * 100 \quad (3.23)$$

Donde:

$$\beta = \frac{V_{AB}^4 + V_{BC}^4 + V_{CA}^4}{(V_{AB}^2 + V_{BC}^2 + V_{CA}^2)^2}$$

Conforme a [18], ante la presencia de un desbalance de tensiones, además de generar componentes de secuencia negativa, también es posible que genere componentes de secuencia positiva, para lo cual se hace necesario obtener el factor de desbalance de secuencia negativa con respecto a la componente de secuencia positiva de la misma forma que se obtiene en la ecuación (3.22), por otro lado, esta misma normatividad indica que el coeficiente de desbalance puede ser obtenido en proporción a la potencia suministrada a la carga con respecto a la potencia de falla en un punto (potencia de corto circuito) para lo cual se puede establecer la siguiente relación:

$$\frac{V_2}{V_1} = \frac{S_C}{S_F} \quad (3.24)$$

En [21] el desbalance se establece como el cociente entre la máxima diferencia entre las tensiones de fase y el promedio de las tensiones.

$$\%V_{DES} = \frac{\Delta V_{MAX}}{V_{PROM}} \quad (3.25)$$

Donde:

$$\Delta V_{MAX} = MAX(V_{AB} - V_{PROM}, V_{BC} - V_{PROM}, V_{CA} - V_{PROM})$$

$$V_{PROM} = \frac{V_{AB} + V_{BC} + V_{CA}}{3}$$

En [22] definen el desbalance de tensiones de la siguiente manera:

$$\%V_{DES} = \sqrt{\frac{6*(V_{AB}^2 + V_{BC}^2 + V_{CA}^2)}{V_{AB} + V_{BC} + V_{CA}}} - 2 \quad (3.26)$$

Como se indica en la ecuación (3.22) la aparición de componentes de secuencia negativa indican un grado de desbalance, lo cual da como resultados efectos negativos sobre el sistema de distribución, como son:

- Perdidas adicionales de potencia.
- Calentamiento adicional en máquinas giratorias, limitándose de esta forma la capacidad nominal de esta.
- Propagación de desbalance a otros nodos de conexión a la red.

En base a lo anterior, es necesario obtener el desbalance de tensión como se indica en el diagrama de flujo de la figura 3.2 y para lo cual se hace uso de la ecuación (3.22) ya que se pueden obtener de forma indirecta pero sencilla a través de la transformación de las tensiones de fase del nodo raíz del sistema de distribución y así obtener las componentes de secuencia positiva como con los componentes de secuencia negativa y de esta forma poder incluir el desbalance a la red de distribuciones en cada iteración para la solución de flujos de potencia de ambos sistemas.

3.6 Proceso iterativo para la solución de flujos de potencia de sistemas de distribución de alta y media tensión

Considerando que para la solución de flujos de potencia para sistemas de transmisión, el sistema de distribución es incorporado a este como nodo de carga, se parte de la idea que en cada nodo de carga existe una subestación de distribución y por ende un transformador de distribución (Figura 3.6), de modo que el proceso iterativo para la solución de flujos de potencia se basa en las tensiones del lado primario del transformador de distribución y desbalance que se puede presentar entre ellas. El siguiente diagrama de flujos muestra el algoritmo para la solución, para lo cual fue desarrollado un programa en fortran.

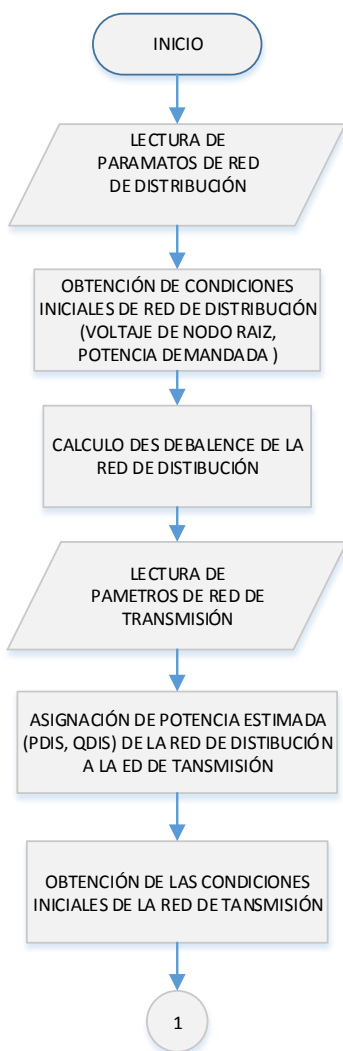


Figura 3.7 Diagrama de flujos para solución de flujos de potencia para sistema de transmisión y distribución (a)

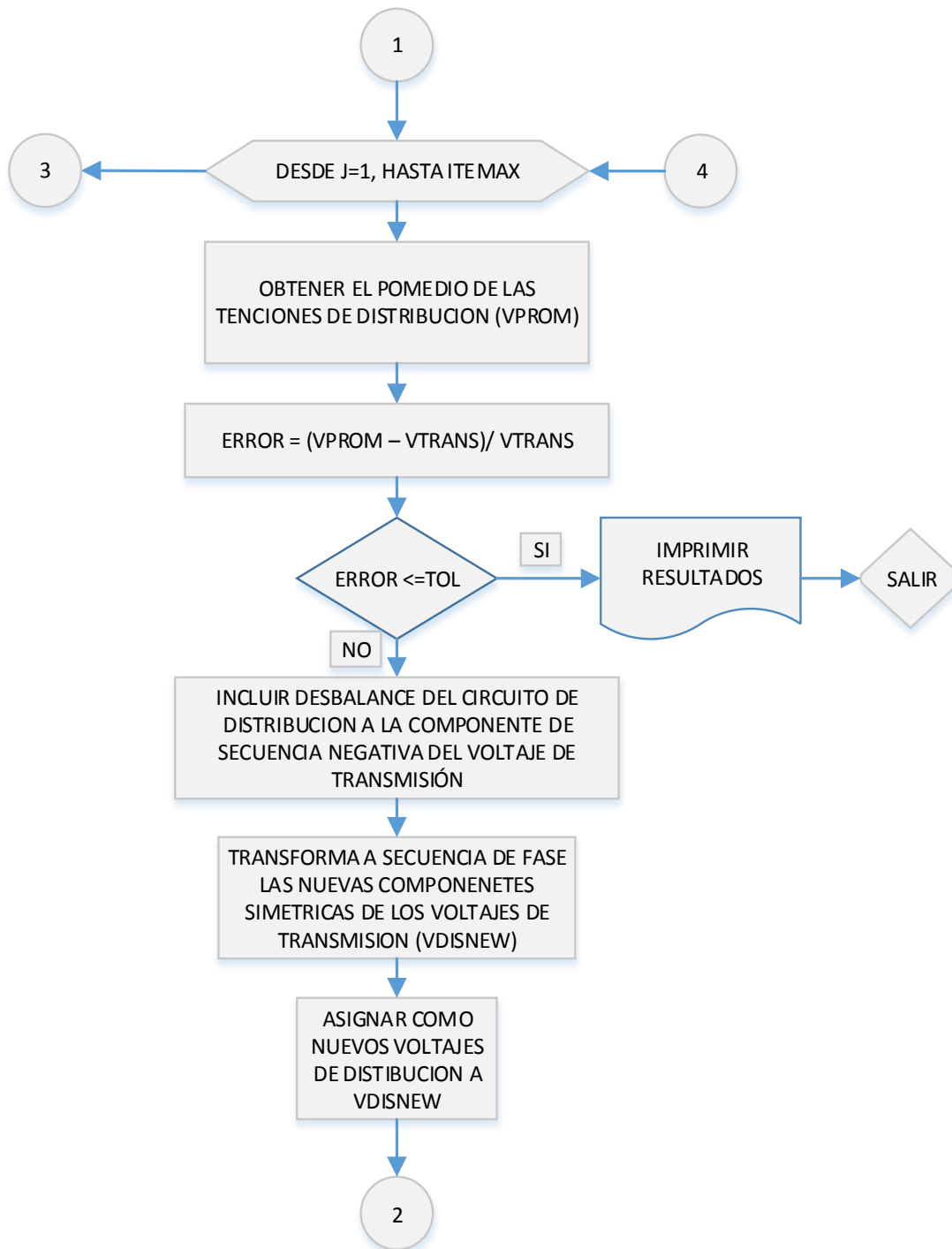


Figura 3.8 Diagrama de flujos para solución de flujos de potencia para sistema de transmisión y distribución (b)

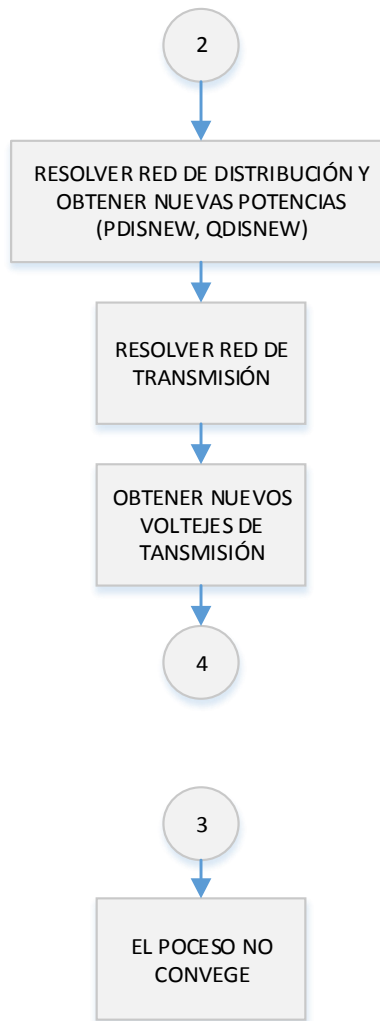


Figura 3.9 Diagrama de flujos para solución de flujos de potencia para sistema de transmisión y distribución (c)

Como se observa en el diagrama de flujo, el programa parte con el cálculo de flujos de potencia de la red de distribución con lo cual se tendría estimada la potencia inyectada a esta y de esta forma poder ser asignada a los nodos de carga de la red de transmisión, posteriormente una vez asignadas las potencias de carga se obtiene la solución de flujos de potencia de la red de transmisión, de esta forma se obtienen los voltajes de los nodos de carga y ser comparados con los voltajes del nodo raíz del sistema de distribución.

El criterio de convergencia para la solución de flujos de potencia de ambos sistemas se basa en la comparación entre tensiones del sistema de transmisión con el sistema de distribución, si el error obtenido entre la comparación es menor a una tolerancia, el programa converge, de no ser así, se asignan como nuevas tensiones las obtenidas de la solución del sistema de transmisión incluyendo el desbalance inicial de la red de distribución. Si el desbalance original es igual a cero se trata de una red de distribución simétrica o balanceada, de lo contrario, el desbalance será diferente de cero y se deberá incluir a la componente de secuencia negativa de la tensión del sistema de transmisión.

El diagrama de flujo mostrado solo muestra el proceso para cuando se trata de una subestación de distribución conectada a uno de los nodos de carga del sistema de transmisión, el programa implementado puede tomar en cuenta el mismo número de subestaciones de distribución (redes de distribución) que el mismo número de nodos de carga y de esta poder obtener la solución de un sistema eléctrico más grande y completo.

CAPITULO 4

RESULTADOS

4.1 Pruebas y validación de solución de flujos de potencia de redes de media tensión

Con el fin de validar el programa computacional implementado en [10] y el cual servirá como subrutina para el enlace de ambos sistemas, se aplicaron redes de distribución de prueba con diferentes características, esto con el fin de observar y calificar su comportamiento ante cada escenario diferente.

4.1.1 Validación de red de media tensión balanceada

Para la validación del estudio de flujos de potencia por el método regresivo-, se emplea una primera red de distribución balanceada de 28 nodos obtenida de [8], la cual se presenta en la figura 4.1.

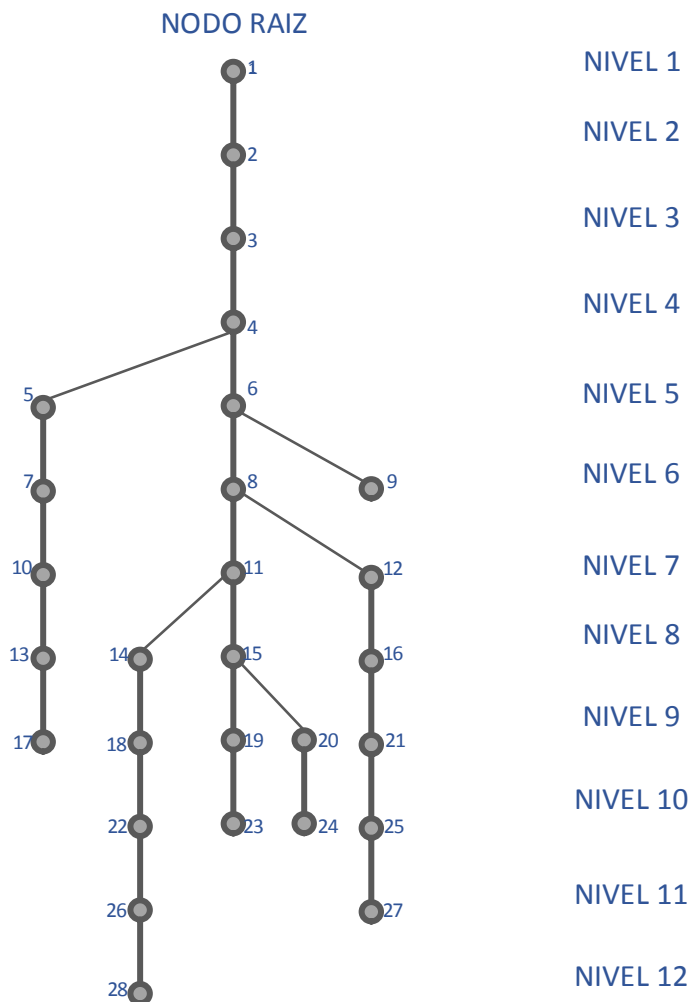


Figura 4.1 Red de distribución radial balanceada de 28 nodos.

Como se observa en la figura 4.1 se cuenta con una red totalmente radial y numerada de acuerdo al criterio planteado en secciones anteriores, de esta forma resulta fácil la aplicación del método planteado en la figura 2.8, en el cual se parte con un perfil plano de voltajes, comenzando con los nodos ubicados en el último nivel, se obtienen las inyecciones de corrientes por cada nivel, hasta llegar al nodo raíz, una vez que se llega al nodo raíz, el voltaje calculado hasta ese punto se sustituye por el voltaje conocido del nodo raíz, dando comienzo con el barrido progresivo, el cual actualiza los voltajes nodales, aplicando la ley de voltajes de Kirchhoff, el barrido se realizara tantas veces sea necesario siempre y cuando no se exceda el número máximo de iteraciones y se cumpla con el criterio de convergencia. Los datos de la red se pueden encontrar en el apéndice A1.

En la siguiente tabla se muestran los resultados obtenidos de la solución de flujos de potencia obtenidos del programa computacional [10] comparados con los resultados de la referencia [8].

Tabla 4.1 Comparación de voltajes nodales de red de distribución balanceada

Numero de nodo	Resultados Obtenidos de referencia A	Resultados Obtenidos de referencia B	%Error
1	1	1	0
2	0.98621	0.98694	0.073966
3	0.96644	0.96674	0.03103213
4	0.95233	0.95234	0.00105005
5	0.94614	0.9461	0.00422788
6	0.93817	0.93875	0.06178429
7	0.94437	0.9443	0.0074129
8	0.92763	0.92788	0.02694314
9	0.93704	0.93762	0.06185875
10	0.94331	0.94332	0.00106009
11	0.91846	0.91869	0.02503565
12	0.92585	0.92591	0.00648011
13	0.94303	0.943	0.00318134
14	0.91557	0.9158	0.02511465
15	0.916	0.91623	0.02510287
16	0.92487	0.92483	0.00432512
17	0.94279	0.94179	0.10618078
18	0.91403	0.91426	0.02515696
19	0.91547	0.91595	0.05240461
20	0.9155	0.91373	0.19371149
21	0.9232	0.92396	0.08225464
22	0.912286	0.91208	0.02258574
23	0.91547	0.9157	0.0251174
24	0.91538	0.91561	0.02511987
25	0.92234	0.922	0.03687636
26	0.9126	0.91182	0.0855432
27	0.9217	0.9218	0.01084834
28	0.91538	0.91466	0.07871777

Donde:

Referencia A: programa implementado en fortran 90 [10]

Referencia B: resultados obtenidos de referencia [8]

En la tabla 4.1 se muestran los resultados obtenidos del programa implementado en la referencia [10], al cual se le realizaron mejoras, como incluir el modelo estrella-delta al modelo del transformador, reasignación de variables y minimización del uso de memoria. Por otro lado, se toman los resultados de magnitud de voltaje obtenidos para realizar la comparación y adquirir un error relativo, tomando como referencia [8] finalmente con base a los resultados conseguidos, se puede concluir que el programa implementado en fortran 90 arroja resultados satisfactorios y presenta un adecuado comportamiento ante redes balanceadas.

4.1.2 Validación de red de media tensión desbalanceada

Con la finalidad de comprobar una de las herramientas computacionales que servirán para el enlace entre sistemas, se utiliza una segunda red para la solución de flujos de potencia, con la diferencia de tratarse de una red trifásica desbalanceada de 13 nodos [12], con un voltaje en nodo raíz de 4.16 KV, esta red está compuesta por diferentes configuraciones ya que cuenta con segmentos de línea trifásicos, bifásicos y monofásicos, bancos de capacitores, modelos de carga concentradas y distribuidas, por lo cual se considera como un buen ejemplo de comparación para la validación del programa implementado. Los datos de la red se pueden encontrar en el apéndice A2.

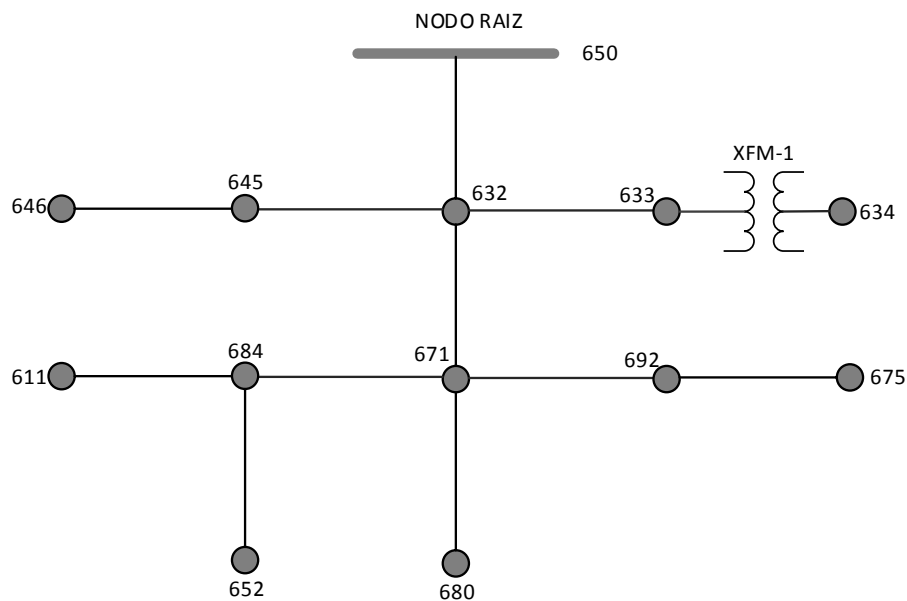


Figura 4.2 Red radial de distribución desbalanceada de 13 nodos

La siguiente tabla muestra el error de voltajes derivado de la solución de flujos de potencia, comparado con el obtenido por la referencia [14] y tomando como referencia la magnitud de voltajes obtenida por este.

Tabla 4.2 Comparación de voltajes nodales de red de distribución desbalanceada

Nodo	Referencia	Fase A-N	Fase B-N	Fase C-N	Error A-N en %	Error B-N en %	Error C-N en %
650	A	1	1	1	0	0	0
	B	1	1	1			
632	A	0.956	0.99101	0.9461	0.06272214	0.08483392	0.08462922
	B	0.9566	0.99017	0.9453			
645	A	0	0.9814	0.9458	0	0.03057792	0.36078098
	B	0	0.9811	0.9424			
671	A	0.92452	1.0066	0.9018	0.08877341	0.54939566	0.0999001
	B	0.9237	1.0011	0.9009			
633	A	0.95455	0.99035	0.94373	0.13112347	0.16688581	0.21556759
	B	0.9533	0.9887	0.9417			
646	A	0	0.97452	0.9412	0	0.48810375	0.09571413
	B	0	0.9793	0.9403			
684	A	0.92386	0	0.89704	0.2126044	0	0.19581664
	B	0.9219	0	0.8988			
680	A	0.92452	0.99866	0.9018	0.07793895	0.24373189	0.0999001
	B	0.9238	1.0011	0.9009			
692	A	0.92452	0.99866	0.9018	0.05627706	0.24373189	0.0999001
	B	0.924	1.0011	0.9009			
634	A	0.9251	0.96797	0.92134	0.28026302	0.14751393	0.00651183
	B	0.9277	0.9694	0.9214			
652	A	0.91191	0	0	0.44650655	0	0
	B	0.916	0	0			0
611	A	0	0	0.89659	0	0	0.0122672
	B	0	0	0.8967			0
675	A	0.9119	1.002	0.8967	0.53446771	0.14947683	0.21144002
	B	0.9168	1.0035	0.8986			

Como se observa en la tabla 4.2 el error derivado de la comparación en la magnitud de voltajes nodales no son tan significativas en la mayoría de los nodos de la red, por lo tanto, se puede afirmar que la herramienta para el cálculo de flujos de potencia para redes desbalanceadas presenta un comportamiento aceptable ante este tipo de redes.

4.2 Pruebas y validación de solución de flujos de potencia de redes de alta tensión

Con el fin de validar el programa computacional implementado en [11] y el cual servirá como subrutina para el enlace de ambos sistemas, se analizaran los resultados de flujos de potencia de redes de transmisión obtenidos de [1] y [23], lo anterior, con el fin de comprobar los ajustes realizados para la correcta solución de flujos de potencia.

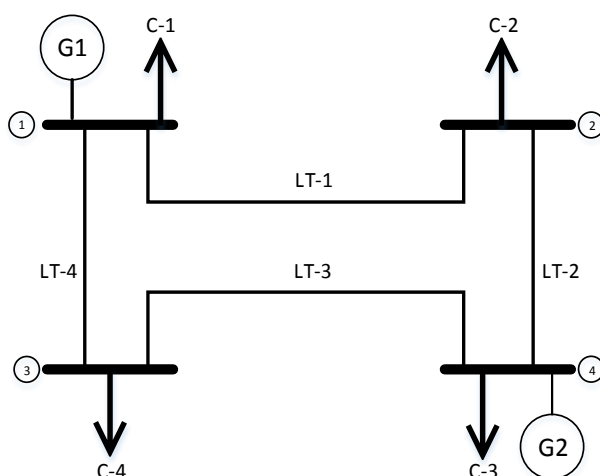


Figura 4.3 Diagrama unifilar de sistema de transmisión

Los datos del sistema se presentan en el apéndice A3 con valores base de 100 MVA y 230 KV, cabe señalar que el método numérico utilizado para la solución de flujos de potencia en las referencias, es el método de Newton-Raphson, la siguiente tabla muestra los resultados obtenidos de la solución de flujos de potencia y el obtenido.

Tabla 4.3 Validación de solución de flujos de potencia para redes de transmisión

<i>Numero de nodo</i>	<i>Referencia</i>	<i>Magnitud de voltajes</i>	<i>% Error</i>	<i>Angulo de voltajes</i>	<i>% Error</i>
1	A	1	0	0	0
	B	1		0	
2	A	0.98242104	0.00117542	0.97612197	0.00079844
	B	0.9835758		0.9753426	
3	A	0.9690048	0.0002741	-1.87217671	0.00040647
	B	0.9687392		-1.8729377	
4	A	1.02	0.00012451	1.52305529	0.00373178
	B	1.019873		1.528739	

Donde

Referencia A: Resultados obtenidos de [23]

Referencia B: Resultados de programa implementado en [11]

Con el propósito de validar la solución de flujos de potencia para redes de transmisión, se presentan los resultados mostrados en la tabla 4-3, donde se puede observar que el programa implementa en [11] tiene un comportamiento adecuado a pesar de utilizar métodos de solución diferentes. Considerando que los resultados mostrados en [23] son idénticos que en [1] se toman como referencia para obtener el error entre la magnitud y ángulo de voltaje

4.3 Solución de flujos de potencia de redes de distribución de alta y media tensión a través del programa computacional ACOPLAMIENTO.FOR

Con base a los criterios expuestos para la realización entre el sistema de distribución y el sistema de transmisión se implementó un programa computacional en FORTRAN 90 denominado ACOPLAMIENTO.FOR, del cual se muestra su código fuente en el apéndice B1 para la solución de solo el acoplamiento de una red de distribución al sistema de transmisión, esta herramienta computacional sigue la lógica del diagrama de flujos de la figura 3.7, 3.8 y 3.9. A continuación se muestra ejemplos de acoplamiento de redes de distribución y transmisión para observar el comportamiento de ambos sistemas.

4.3.1 Caso de estudio 1: Acoplamiento de una red de distribución a la red de transmisión.

Para este caso se toma la red de transmisión del apéndice A3 con una red de distribución desbalanceada como la que se muestre en el apéndice A2. Aunque la red de distribución presenta ser desbalanceada a lo largo de sus líneas, el nodo raíz de la subestación de distribución se presenta como balanceado.

Como se observa en la figura 4.2, la carga 4 (C4) es sustituida por la red de distribución del apéndice A2. Si se observan los voltajes del nodo raíz de los resultados obtenidos de flujos de potencia de la tabla 4.2, estos muestran ser balanceados en el lado secundario de la subestación de distribución, debido a esto y tomando en cuenta las características de la conexión Delta(devanados primarios)-Estrella(devanados secundarios) para un transformador de distribución, ya que en caso de presentarse un desbalance en el lado secundario esta conexión elimina un cierto porcentaje de desbalance de voltajes en las terminales de los devanados primarios pero sigue existiendo un mínimo de porcentaje de desbalance de voltaje, sin embargo, como el desbalance de tensiones en el lado secundario del transformador es nulo, por consiguiente el lado primario se encuentra balanceado pero desfasado 30°.

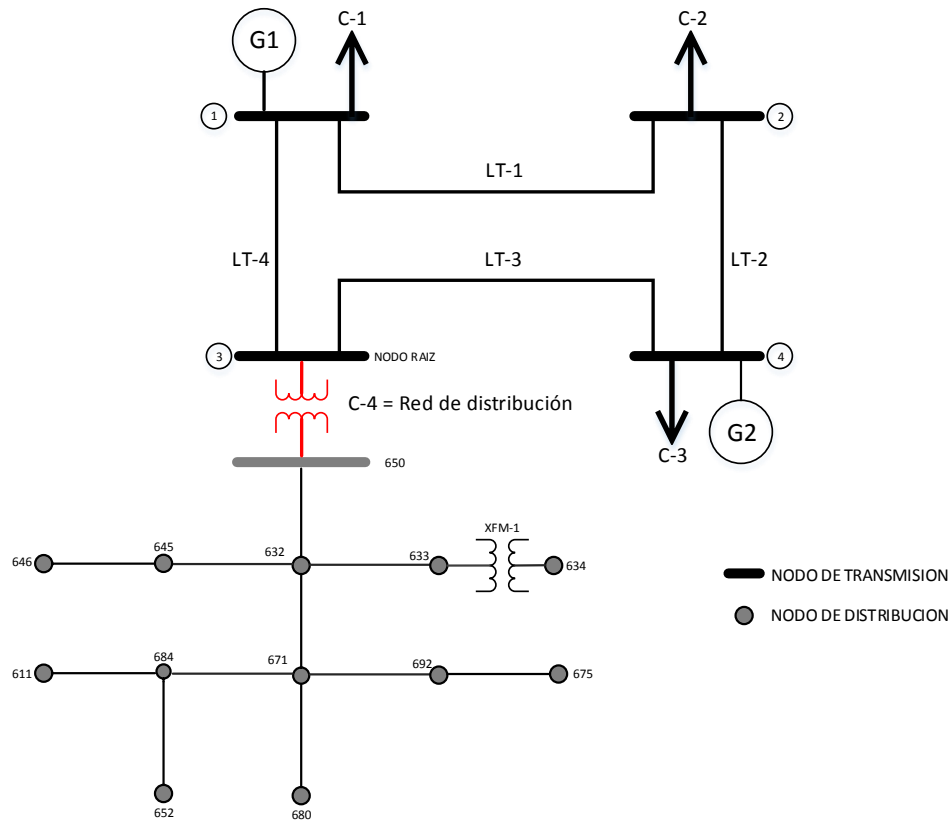


Figura 4.4 Diagrama unifilar del enlace entre red de distribución y red de transmisión

Con el fin de analizar los resultados obtenidos del programa implementado se obtuvieron los siguientes resultados:

Tabla 4.4 Comparación de voltajes del nodo raíz

<i>Iteración</i>	<i>Promedio de Magnitud de voltaje de nodo raíz de distribución (KV)</i>	<i>Potencia activa trifásica de distribución (KW)</i>	<i>Potencia reactiva trifásica de distribución (KVAR)</i>	<i>Magnitud de voltaje de nodo 3 de transmisión (KV)</i>
0	115	3333.128	2055.447	111.435552
1	111.435552	3335.625	2062.754	111.435552

Como se observa en la tabla 4.4, en la iteración cero son obtenidas las condiciones iniciales de cada sistema, para conseguir los voltajes actualizados (iteración uno) y realizar la comparación entre ellos, tomando en cuenta que la tolerancia para el error fue de 0.00001, al momento de calcular el error relativo entre los voltajes de la primera iteración se obtiene un error de 0.0, lo cual es menor a la tolerancia, alcanzando de esta forma la solución de ambos sistemas.

El número de iteraciones para la solución de ambos sistemas es de una, esto se debe a que el sistemas de distribución actualiza los voltajes en función a los resultados que se obtienen en la solución del sistema de transmisión, por otro lado como se observa en la tabla 4.4 ante cambios de potencia activa y reactiva en el sistema de transmisión el voltaje de transmisión permanece constante

4.3.2 Caso de estudio 2: Acoplamiento de una red de distribución con un 1.5 % de desbalance a la red de transmisión

En este caso se plantean los mismos sistemas de transmisión y distribución del caso uno, pero con la diferencia, que al nodo raíz se le incluye un desbalance de 1.5% entre fases como se muestra a continuación:

$$\begin{bmatrix} V_A \\ V_B \\ V_C \end{bmatrix} = \begin{bmatrix} 115\angle 30 \\ 115\angle -90 \\ 115\angle 150 \end{bmatrix} \quad (5.27)(4.1)$$

Con ayuda de la ecuación (3.8) se transforman a componentes simétricas el sistema de (4.1) y se obtienen los siguientes resultados:

$$V_{012} = \begin{bmatrix} 0 \\ 115\angle 30 \\ 0 \end{bmatrix} \quad (4.2)$$

De la ecuación (3.22) se despeja el voltaje de secuencia negativa y se obtiene lo siguiente:

$$|V_2| = \frac{\%V_{DES} \cdot |V_1|}{100} \quad (4.3)$$

Sustituyendo valores se obtienen:

$$|V_2| = \frac{1.5 \cdot |115|}{100} = 1.725 \quad (4.3)$$

Por lo tanto las componentes simétricas de los voltajes de fase con un 1.5% de desbalance son los siguientes:

$$V_{012} = \begin{bmatrix} 0\angle 150 \\ 115\angle 30 \\ 1.725\angle -90 \end{bmatrix}$$

Con ayuda de la ecuación (3.7) se transforman a secuencia de fase los voltajes anteriores y se obtienen los siguientes resultados:

$$V_{ABC} = \begin{bmatrix} 114.14\angle 30 \\ 114.14\angle -90 \\ 116.725\angle 150 \end{bmatrix}$$

Una vez obtenido el desbalance en los voltajes de fase se analizara el comportamiento de la herramienta computacional ante este tipo de escenarios.

Tabla 4.5 Comparación de voltajes del nodo raíz incluyendo desbalance de tensiones

<i>Iteración</i>	<i>Promedio de Magnitud de voltaje de nodo raíz de distribución (KV)</i>	<i>Potencia activa trifásica de distribución (KW)</i>	<i>Potencia reactiva trifásica de distribución (KVAR)</i>	<i>Magnitud de voltaje de nodo 3 de transmisión (KV)</i>
0	115.0016667	3333.154	2055.3888	111.43555
1	111.4318167	3336.487	2064.9450	111.43501
2	111.4330930	3334.145	2062.030	111.43502
3	111.4345678	3334.098	2060.017	111.43502
4	111.4349576	3333.904	2058.856	111.43503
5	111.4350378	3332.302	2055.1094	111.43503

Como se observa en la tabla 4.5 el número de iteraciones aumenta, esto debido a que en cada iteración el voltaje calculado de la solución de flujos de potencia en transmisión no es asignado de forma directa al nodo raíz de la red de distribución, ya que en cada iteración es incluido el desbalance. Otro factor que genera el aumento de iteraciones, es la comparación de los voltajes del nodo raíz con el nodo de carga, ya que se obtiene el promedio de voltajes del nodo raíz, provocando que se aleje del voltaje de nodo de carga de transmisión.

4.4 Discusión de los resultados

El análisis de ambos casos de estudio fue enfocado al estado en el que se encontraban los voltajes del nodo raíz, ya que se puede decir que cualquier alteración o cambio en el nodo raíz repercute en toda la red de distribución, generando cambios en los voltajes nodales y aumento o disminución de pérdidas de potencia activa y reactiva.

En base a los resultados obtenidos del primer caso de estudio, se obtiene que teniendo un nodo raíz balanceado, el proceso presenta ser sencillo, esto debido a que al momento de realizar la actualización de voltajes de distribución y ser estimada la potencia de carga de la red de transmisión, los voltajes nodales de transmisión no presentan ningún cambio, esto debido a que la variación de potencia de carga es mínima.

Tomando en cuenta que las condiciones iniciales son obtenidas a partir de la solución de flujos de potencia de la red de distribución y teniendo como dato inicial los voltajes del nodo raíz, se podría tener un menor número de iteraciones, si se tiene una medición más exacta de los voltajes de fase, y siendo así, nos acercaría más rápido a la convergencia.

CAPITULO 5

CONCLUSIONES

- La información presentada sobre el modelado de la red y solución numérica de flujos de potencia de sistemas de alta y media tensión permitieron contar con herramientas necesarias para la correcta elaboración de herramientas computacionales para la solución de flujos de potencias.
- Los métodos numéricos empleados para la solución de flujos de potencia, tanto para sistemas de alta y media tensión fueron seleccionados de los métodos existentes debido a que cuentan con mejores características de exactitud y velocidad de cálculo para cada ambiente, aunque con el fin comparación para trabajos futuros se pueden hacer variaciones o combinaciones de distintos métodos de solución para comprobar si los métodos seleccionados fueron los correctos.
- El comportamiento de los cambios de voltajes de alta tensión ante los cambios de potencia activa y reactiva generados por la actualización de los voltajes del nodo raíz en cada iteración son mínimos, a comparación de los cambios de potencia activa y reactiva debido a las actualizaciones de voltajes de fase del nodo raíz. Esto permite establecer que ante pequeños cambios en la demanda de potencia de los circuitos de distribución el sistema de alta tensión permanece estable, a menos que se presente un gran cambio en la potencia demandada por el circuito de distribución, debido a esto es posible que para futuros trabajo se utilice el algoritmo de acoplamiento de sistemas de alta y media tensión para el análisis de fallas, ya que es aquí donde se presentan grandes cambios de potencia.

REFERENCIAS

- [1] Grainger J.J,Stevenson W. D. Jr, análisis de sistemas eléctricos de potencia, libro, Mc Graw Hill
- [2] D. shirmohammadi, H. W. Hong, A. Semlyen, G. X. Luo, A Compensation-Based Power Flow Method For Weakly Meshed Distribution And Transmission Networks, IEEE Tansactions on Power systems, Vol.3, No. 2, May 1998.
- [3] Eric Morales Aguilar, "Desarrollo de un programa para resolver el problema de flujos de potencia en redes aéreas de distribución por el método progresivo regresivo", Tesis de Maestría, Instituto Politécnico Nacional, SEPI ESIME Zacatenco, junio 2009
- [4] William H. Kersting, Distribution System Modeling and Analysis, Libro, CRC Press LLC
- [5] Ray Daniel Zimmerman, Comprehensive Distribution Power Flow Modeling, Formulation, Solution Algorithms And Analysis, Tesis, Enero 1995
- [6] W. H. Kersting and W. Howard Phillips, "Distribution Feeder Line Models", IEEE Transactions on Industry Application, Vol. 31, No. 4, July/August 1995
- [7] M.T. Bishop, J. D. Foster and D. A. Down, "The Application of Single-Phase Voltage Regulators on Three-Phase Distribution Systems", IEEE Industry Applications Magazine, July/August 1996
- [8] D.Das, HS Nagi and D.P. Kothari. "Novel Method for Solving Radial Distribution Networks". Proceedings IEEE Part C, Vol.141, No.4, pp.291-298, 1991
- [9] C. S. Cheng and D. Shirmohammadi, " A Three Phase Power Method for Real Time Distribution System Analysis", IEEE Transactions on Power Systems, Vol. 10, No. 2, May 1995
- [10] Fondo sectorial para investigación y desarrollo tecnológico en energía CFE-CONACYT; PROYECTO 48545, Simulador De Redes De Distribución Para Centros Regionales De Control De Distribución De CFE; Marzo 2010
- [11] Fondo Sectorial Para Investigación Y Desarrollo Tecnológico En Energía CFE-CONACYT; Proyecto 48545, Manual Técnico Del Programa De Flujos De Potencia Por El Método Desacoplado Rápido Para Redes De Subtransmisión; Agosto 2009
- [12] Radial Distribution Test Feeders. [online].
Disponible en : <http://www.ewh.ieee.org/soc/pes/dsacom/testfeeders.html>
- [13] S. Khushalani and M. Solanki. "Development of three-Phase unbalanced Power Flow using PV and PQ Models for Distributed Generation and Study of Impact of DG Models". IEEE Transactions on power system. Vol. 22, No.3, pp. 1019-1025. 2007.
- [14] S. Khushalani and Schulz. "Unbalanced Distribution Power Flow with Distributed Generacion". Proceeding of the IEEE power Engineering Society (PES) Power Systems Conference and Exhibition, Atlanta, Georgia, October 2006

- [15] K. Imhof, F. Oesch and I. Nordanlycke, "Modelling of tap-charger transformers in an energy management system", Proc. IEEE PICA conf., Salt Lake City, pp. 378-384
- [16] C. L. Fortescue, "Method of Symmetrical Coordinates Applied to the Solution of Polyphase Networks", Transactions AIEE, vol 37, pag. 1027-1140, 1918
- [17] E. Ras, "Transformadores de potencia de medida y de protección", libro, Marcombo Boixareu Editores
- [18] IEC 61000-4-30, "Electromagnetic compatibility (EMC)-Part 4-30: Testing and measurement techniques-power quality measurement methods", Basic EMC publication
- [19] Math Bollen and Irene GU, "Signal Processing of Power Quality Disturbance", IEEE Press Series, Wiley-Interscience
- [20] Henryk Markiewicz and Antoni Klajn, "Voltage Disturbance Standard EN 50160 Voltage Characteristics in Public Distribution Systems", Power Quality Application Guide, Leonardo Power Quality Initiative, July 2004
- [21] IEC 61000-2-1, "Electromagnetic compatibility. Part 2: Environment. Section 1: Description of the environment-Electromagnetic environment for low-frequency conducted disturbances and signal-lin in public power supply systems", Geneva. 1990
- [22] IEC 61000-2-12, "Electromagnetic compatibility. Part 2: Environment. Section 12: Description of the environment-Electromagnetic environment for low-frequency conducted disturbances and signal-lin in public power supply systems", Genève, April 1999
- [23] Guzmán Dolores Misael, "Flujos de Potencia con Matlab", Trabajo de licenciatura, Universidad Veracruzana, junio 2012
- [24] Wildi Theodore, "Maquinas Eléctricas y Sistemas de Potencia", libro, Person Educación, México 2007
- [25] Oscar Miranda Uriostegui, "Minimización de pérdidas en redes de 56 distribución empleando programación dinámica", Tesis de Maestría, Instituto Politécnico Nacional, SEPI ESIME Zacatenco, junio 2009
- [26] Alfonso Jara Morales, "Estudio de flujos de potencia y de corto circuito en sistemas de distribución de media tensión incorporando generación distribuida", Tesis de Maestría, Instituto Politécnico Nacional, SEPI ESIME Zacatenco, Diciembre 2011
- [27] Palencia Ruiz Erix Jorel, "Diseño e implementación de una herramienta computacional para estudios de flujos de potencia en sistemas de distribución", tesis de licenciatura, Universidad Experimental Politécnica de la Fuerza Armada Nacional, Maracay, Junio 2007
- [28] Mota Palomino R., "Programa para solución de estudios de flujos por el método desacoplado rápido". Manual de programa, 1989
- [29] Berg R. Jr., Hamkins E. S., Pleines W. W., "Mechanized calculation of unbalanced load flow in radial distribution circuits". IEEE Trans. On PAS, Vol. PAS-86, 415-421, April 1967

APÉNDICE A

DATOS DE SISTEMAS DE PRUEBA

A1. RED DE DISTRIBUCIÓN DE 28 NODOS

Los datos de este sistema fueron obtenidos de [8]. El sistema presenta un voltaje nominal en su nodo raíz de 11 KV, a continuación se muestran sus valores de impedancia y carga.

Tabla A1.1 Datos de impedancia de la red de 28 nodos

<i>Numero de rama</i>	<i>Nodo de envió</i>	<i>Nodo de recepción</i>	<i>R(Ω)</i>	<i>X(Ω)</i>
1	1	2	1.197	0.82
2	2	3	1.796	1.231
3	3	4	1.306	0.895
4	4	5	1.851	1.268
5	5	6	1.524	1.044
6	6	7	1.905	1.305
7	7	8	1.197	0.82
8	8	9	0.653	0.447
9	9	10	1.143	0.783
10	4	11	2.823	1.172
11	11	12	1.184	0.491
12	12	13	1.002	0.416
13	13	14	0.455	0.189
14	14	15	0.546	0.227
15	5	16	2.550	1.058
16	6	17	1-366	0.567
17	17	18	0.819	0.34
18	18	19	1.548	0.642
19	19	20	1.366	0.567
20	20	21	3.552	1.474
21	7	22	1.548	0.642
22	22	23	1.092	0.453
23	23	24	0.91	0.378
24	24	25	0.455	0.189
25	25	26	0.364	0.151
26	8	27	0.546	0.226
27	27	28	0.273	0.113

Tabla A1.2 Datos de carga de red de distribución de 28 nodos

<i>Numero de nodo</i>	<i>P_L (KW)</i>	<i>Numero de nodo</i>	<i>P_L (KW)</i>
1	0.0	2	35.28
2	35.28	3	35.28
3	14.0	4	8.96
4	35.28	5	8.96
5	14.0	6	35.28
6	35.28	7	35.28
7	35.28	8	14.0
8	35.28	9	35.28
9	14.0	10	8.96
10	14.0	11	56.0
11	56.0	12	8.96
12	35.28	13	35.28
13	35.28	14	35.28
14	14,0	15	35.28

El factor de potencia en todas las cargas es de $\cos \theta = 0.7$, por lo que la potencia reactiva se puede calcular como $Q_L = P_L * \tan \theta$

A2. RED DE DISTRIBUCIÓN DE 13 NODOS

Los datos de la red fueron obtenidos de [12], a continuación se muestran los valores de carga y sus diferentes configuraciones de líneas.

Tabla A2.1 Datos de líneas aéreas de red de 13 nodos

<i>Configuración</i>	<i>Faseo</i>	<i>Conductor de fase</i>	<i>Conductor neutro</i>	<i>Identificador</i>
601	BACN	556/500 26/7	4/0 6/1	500
602	CABN	4/0 6/1	4/0 6/1	500
603	CBN	1/0	1/0	505
604	ACN	1/0	1/0	505
605	CN	1/0	1/0	510

Tabla A2.2 Datos de líneas subterráneas de red de 13 nodos

<i>Configuración</i>	<i>Faseo</i>	<i>Conductor de fase</i>	<i>Conductor neutro</i>	<i>Identificador</i>
601	ABCN	250,000 AA,CN	Ninguno	515
602	AN	1/0 AA,TS	1/0 Cu	520

Tabla A2.3 Datos de transformadores de red de 13 nodos

	<i>KVA</i>	<i>KV lado de alta</i>	<i>KV lado de baja</i>	<i>R(%)</i>	<i>X(%)</i>
Subestación	5000.0	115-D	4.16 Yg	1.0	8.0
XFM-1	500.0	4.16 YG	0.48 Yg	1.1	2.0

Tabla A2.4 Datos de capacitores de red de 13 nodos

<i>Nodo</i>	<i>Fase A(KVAR)</i>	<i>Fase B(KVAR)</i>	<i>Fase C (KVAR)</i>
675	200.0	200.0	200.0
611	-----	-----	100.0
Total	200.0	200.0	300.0

Tabla A2. 5 Datos de secciones de alimentadores de red de 13 nodos

<i>Elemento</i>	<i>Nodo i</i>	<i>Nodo j</i>	<i>Longitud (ft)</i>	<i>Configuración</i>
Alimentador 1	632	645	500.0	603
Alimentador 2	632	633	500.0	602
Alimentador 3	645	646	300.0	603
Alimentador 4	650	632	2000.0	601
Alimentador 5	684	652	800.0	607
Alimentador 6	632	671	2000.0	601
Alimentador 7	671	684	300.0	604
Alimentador 8	671	680	1000.0	601
Alimentador 9	984	611	300.0	605
Alimentador 10	692	675	500.0	606

Tabla A2.6 Impedancia serie de alimentadores de red de 13 nodos

<i>Configuración</i>	<i>Z_{abc} [Ω/mi]</i>		
601	$\begin{bmatrix} 0.3465 + j1.0179 & 0.1560 + j0.5017 & 0.1580 + j0.4236 \\ 0.1560 + j0.5017 & 0.3375 + j1.0478 & 0.1535 + j0.3849 \\ 0.1580 + j0.4236 & 0.1535 + j0.3849 & 0.3414 + j1.0348 \end{bmatrix}$		
602	$\begin{bmatrix} 0.7526 + j1.1814 & 0.1580 + j0.4236 & 0.1560 + j0.5017 \\ 0.1580 + j0.4236 & 0.7475 + j1.1983 & 0.1535 + j0.3849 \\ 0.1560 + j0.5017 & 0.1535 + j0.3849 & 0.7436 + j1.2112 \end{bmatrix}$		
603	$\begin{bmatrix} 0.0 & 0.0 & 0.0 \\ 0.0 & 1.3294 + j1.3471 & 0.2066 + j0.4591 \\ 0.0 & 0.2066 + j0.4591 & 1.3138 + j1.3589 \end{bmatrix}$		
604	$\begin{bmatrix} 1.3238 + j1.3568 & 0.0 & 0.2066 + j0.4591 \\ 0.0 & 0.0 & 0.0 \\ 0.2066 + j0.4591 & 0.0 & 1.3294 + j1.3471 \end{bmatrix}$		

605	$\begin{bmatrix} 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 1.3292 + j1.3475 \end{bmatrix}$
606	$\begin{bmatrix} 0.7982 + j0.4463 & 0.3192 + j0.0328 & 0.2849 - j0.0143 \\ 0.3192 + j0.0328 & 0.7891 + j0.4041 & 0.3192 + j0.0328 \\ 0.2849 - j0.0143 & 0.3192 + j0.0328 & 0.7982 + j0.4463 \end{bmatrix}$
607	$\begin{bmatrix} 1.3425 + j0.5124 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 \end{bmatrix}$

Tabla A2.7 Admitancia en derivación de red de 13 nodos

Configuración	$Z_{abc} [\Omega/mi]$
601	$\begin{bmatrix} 6.2998 & -1.9958 & -1.2595 \\ -1.9958 & 5.9597 & -0.7417 \\ -1.2595 & -0.7417 & 5.6386 \end{bmatrix}$
602	$\begin{bmatrix} 5.6990 & -1.0817 & -1.6905 \\ -1.0817 & 5.1795 & -0.6588 \\ -1.6905 & -0.6588 & 5.4246 \end{bmatrix}$
603	$\begin{bmatrix} 0.0 & 0.0 & 0.0 \\ 0.0 & 4.7097 & -0.8999 \\ 0.0 & -0.8999 & 4.6658 \end{bmatrix}$
604	$\begin{bmatrix} 4.6658 & 0.0 & -0.8999 \\ 0.0 & 0.0 & 0.0 \\ -0.8999 & 0.0 & 4.7097 \end{bmatrix}$
605	$\begin{bmatrix} 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 4.5193 \end{bmatrix}$
606	$\begin{bmatrix} 96.8897 & 0.0 & 0.0 \\ 0.0 & 96.8897 & 0.0 \\ 0.0 & 0.0 & 96.8897 \end{bmatrix}$
607	$\begin{bmatrix} 88.9912 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 \end{bmatrix}$

Tabla A2.8 Datos de carga spot de red de 13 nodos

Nodo	Modelo de carga	Fase A	Fase A	Fase B	Fase B	Fase C	Fase C
		KW	KVAR	KW	KVAR	KW	KVAR
634	Y-PQ	160.0	110.0	120.0	90.0	120.0	90.0
645	Y-PQ	0.0	0.0	170.0	125.0	0.0	0.0
646	D-Z	0.0	0.0	230.0	132.0	0.0	0.0
652	Y-Z	128.0	86.0	0.0	0.0	0.0	0.0
671	D-PQ	385.0	220.0	385.0	220.0	385.0	220.0
675	Y-PQ	485.0	190.0	68.0	60.0	290.0	212.0

692	D-I	0.0	0.0	0.0	0.0	170.0	151.0
611	Y-I	0.0	0.0	0.0	0.0	170.0	80.0

Tabla A2.9 Datos de carga distribuida de red de 13 nodos

<i>Nodo i</i>	<i>Nodo j</i>	<i>Modelo de carga</i>	<i>Fase A</i>	<i>Fase A</i>	<i>Fase B</i>	<i>Fase B</i>	<i>Fase C</i>	<i>Fase C</i>
			KW	KVAR	KW	KVAR	KW	KVAR
632	671	Y-PQ	17.0	10.0	66.0	38.0	117.0	68.0

A3. RED DE TRANSMISIÓN DE 4 NODOS

La red de prueba de transmisión de obtuvo de [1], la cual cuenta con valores base de 100 MVA y 230 KV. A continuación se muestran datos del sistema.

Tabla A3.1 Datos de líneas de red de 4 nodos

		<i>Impedancia serie</i>		<i>Admitancia en derivación</i>
<i>Nodo i</i>	<i>Nodo j</i>	R(pu)	X(pu)	Y/2 (pu)
1	2	0.01008	0.05040	0.05125
1	3	0.00744	0,03720	0.03875
2	4	0.00744	0.03720	0.03875
3	4	0.01272	0.06360	0.06375

Tabla A3.2 Datos de generación y carga

<i>Barra</i>	<i>Generación</i>		<i>Carga</i>		<i>Voltaje de barra</i>
	P(KW)	Q(KVAR)	P(KW)	Q(KVAR)	V(pu)
1	-----	-----	50.0	30.99	1.0
2	0.0	0.0	170.0	105.35	1.0
3	0.0	0.0	200	123.94	1.0
4	318.0	-----	80	49.58	1-02

APÉNDICE B

B1. CODIGO FUENTE DEL PROGRAMA RADIFLUX.FOR

A continuación se muestra el código fuente del programa RADIFLUX:FOR para la solución de flujos de potencia de redes de media y baja tension, el cual fue reportado inicialmente en [10] y al cual fue modificado con el fin de realizar el acoplamiento de ambos sistemas.

```
SUBROUTINE RADIFLUX
!-----
use estructuras
use estructuras2
use estructuras3

use rutinas_de_asignacion

use ConstantesG14
use ConstantesG2
use ConstantesG3
use Mcons

use PrintEntradaDetallada
use PrintSalidaDetallada
use PrintSalidaDetallada_2
use PrintSalidaRR

use BackForw
USE MDATOS
USE MDATOS2
USE MDATOS3

!-----
IMPLICIT NONE

!Declaracion de Variables
!.....
character(20)      :: filename !Nombre del archivo de entrada
character(len=60)  :: header   !Encabezado del archivo de entrada
!integer           :: n2       !Número de nodos de la red (incluido en nodo raiz)
integer           :: m2       !Numero de mallas
real, dimension(6) :: Vrootr   !Voltaje del nodo Raiz (kV-LN para todos los casos) vector de
lectUra
complex, dimension(3) :: Vrootc !Procesa el anterior para hacerlo complejo y asignarlo al nodo 1 (raiz)

type(lectura)::temp              !Contenedor de Lectura de nodo
type(lista), allocatable, dimension(:)::nodo      !Datos de la Red para Lista de Barrido

type(lectura_mallas)::temp2      !Contenedor de lectura de malla
type(lista_mallas), allocatable, dimension(:)::malla !Datos de estructura de malla

type(THV_est)::thv              !*****
integer, allocatable, dimension(:):: VISIT !auxiliar para marcar nodos visitados (puede estar el la
estructura nodal)

complex, allocatable, dimension(:,:):: Zthevenin
complex, allocatable, dimension(:,:):: Zthevenin_KPZ

complex, allocatable, dimension(:,:):: Ythevenin_KPZ
complex, allocatable, dimension(:,:):: Ythevenin

real, allocatable, dimension(:,:):: KPZ !Matriz de Compactacion

complex, allocatable, dimension(:)::Vthevenin
complex, allocatable, dimension(:)::Ithevenin

!.....
integer :: maxIter = 3000 !No. Máximo de Iteraciones Permitidas (Cambiar desde aqui)
```

```

real      :: tolerancia = 0.01      !Tolerancia de error del proceso iterativo ((Cambiar desde aqui)

integer :: iteraciones =0           !Iteraciones del algoritmo de Barrido (algoritmo de flujos o iterativo)
integer :: converge      =1         !numero acumulativo de convergencia de nodos (para converger debe alcanzar
el máximo n2)
!.....

integer :: iteracionesMacro=0
integer :: convergeMacro= -1

!.....
integer::k,m, p, t , q ,r, xx, yy, zz      !contador (es)- variable (s) auxiliares (no usar "i" ni
"j")
integer:: extremo1=0, extremo2=0           !15feb2010

!-----
!-----INICIA CODIFICACION DE PROGRAMA PRINCIPAL-----
!-----
!-----

!Introduccion y Apertura del Archivo de Entrada
IF (ACTUALIZAVOL .EQ. 0) THEN
READ(*,*)      fileName
END IF
open(1,file=fileName,action='read')
!-----

!lectura y asignación
read(1,*)header
READ(1,*)NODOPOT
read(1,*)n2
IF (ACTUALIZAVOL .EQ. 0) THEN
allocate(nodo(n2))
END IF
READ (1,*)Vrootr
call conversor_voltaje( Vrootr, Vrootc) !Se convierte en vector complejo (rectangular) la entrada de
voltaje Vroot

IF (ACTUALIZAVOL .EQ. 1) THEN
Vrootc = VDISNEW
PRINT*, 'ACTUALIZO VOLTAJES'
PRINT*, Vrootc
END IF

VDIS(1,1) = Vrootc(1)
VDIS(2,1) = Vrootc(2)
VDIS(3,1) = Vrootc(3)

!+++++

LECTURA_Y_ASIGNACION: do k=1,n2

read(1,*) temp%busName

read(1,*) temp%n, temp%pre, temp%niv, temp%tybr, temp%nt

read(1,*) temp%Z
read(1,*) temp%Y

read(1,*) temp%cx_Sc, temp%md_Sc, temp%Sc
read(1,*) temp%cx_Sd, temp%md_Sd, temp%Sd

read(1,*) temp%cx_Bc, temp%SBc

read(1,*) temp%lc_Reg, temp%ty_Reg,temp%cx_Reg, temp%Tap

!+++++

nodo(k)%busName=temp%busName

nodo(k)%n      =temp%n
nodo(k)%pre     =temp%pre
nodo(k)%niv     =temp%niv
nodo(k)%tybr    =temp%tybr
nodo(k)%nt      =temp%nt

```

```

call conversor_Z(temp%Z, nodo(k)%Z, nodo(k)%Z1, nodo(k)%Z2, nodo(k)%Z3, nodo(k)%Z4)

call conversor_Y(temp%Y, nodo(k)%Y)

nodo(k)%cx_Sc =temp%cx_Sc
nodo(k)%md_Sc =temp%md_Sc

nodo(k)%cx_Sd =temp%cx_Sd
nodo(k)%md_Sd =temp%md_Sd

nodo(k)%cx_Bc =temp%cx_Bc

call conversor_carga (nodo(k)%cx_Sc, nodo(k)%md_Sc, temp%Sc, nodo(k)%cx_Sd, nodo(k)%md_Sd, temp%Sd,
nodo(k)%cx_Bc, temp%SBc,nodo(k)%S1YS, nodo(k)%S2YS, nodo(k)%S3YS, nodo(k)%S4YS,
nodo(k)%S1YI, nodo(k)%S2YI, nodo(k)%S3YI, nodo(k)%S4YI, nodo(k)%S1YZ, nodo(k)%S2YZ, nodo(k)%S3YZ,
nodo(k)%S4YZ,nodo(k)%S1DS, nodo(k)%S2DS, nodo(k)%S3DS, nodo(k)%S4DS,
nodo(k)%S1DI, nodo(k)%S2DI, nodo(k)%S3DI, nodo(k)%S4DI, nodo(k)%S1DZ, nodo(k)%S2DZ, nodo(k)%S3DZ,
nodo(k)%S4DZ, nodo(k)%Sc, nodo(k)%Sd, nodo(k)%SBc )

nodo(k)%lc_Reg = temp%lc_Reg
nodo(k)%ty_Reg = temp%ty_Reg
nodo(k)%cx_Reg = temp%cx_Reg
nodo(k)%Tap = temp%Tap

call arReg (nodo(k)%Tap, nodo(k)%ar)

call dAB14( nodo(k)%lc_Reg, nodo(k)%ty_Reg, nodo(k)%cx_Reg, nodo(k)%ar, nodo(k)%Z1, nodo(k)%Z4,
&
nodo(k)%d1, nodo(k)%A1, nodo(k)%B1, &
nodo(k)%d4, nodo(k)%A4, nodo(k)%B4
)

call dAB2 (nodo(k)%Z2, nodo(k)%d2, nodo(k)%A2, nodo(k)%B2)
call dAB3 (nodo(k)%tybr,nodo(k)%nt, nodo(k)%Z3, nodo(k)%d3, nodo(k)%A3, nodo(k)%B3)

!Inicializacion voltajes nodales , corrientes nodales (normales y primadas) , corrientes de incoming
branch
!y admitancia total de linea NODAL Y2 (charge admittance)

nodo(k)%Y2=CMPLX(0.0, 0.0)

nodo(k)%V1=CMPLX(0.0, 0.0)
nodo(k)%V2=CMPLX(0.0, 0.0)
nodo(k)%V3=CMPLX(0.0, 0.0)
nodo(k)%V4=CMPLX(0.0, 0.0)

nodo(k)%I1=CMPLX(0.0, 0.0)
nodo(k)%I2=CMPLX(0.0, 0.0)
nodo(k)%I3=CMPLX(0.0, 0.0)
nodo(k)%I4=CMPLX(0.0, 0.0)

nodo(k)%Ibreak=CMPLX(0.0, 0.0) !23feb2010 -----"/&%/&%/&

nodo(k)%J1=CMPLX(0.0, 0.0)
nodo(k)%J2=CMPLX(0.0, 0.0)
nodo(k)%J3=CMPLX(0.0, 0.0)
nodo(k)%J4=CMPLX(0.0, 0.0)

nodo(k)%JP1=CMPLX(0.0, 0.0)
nodo(k)%JP2=CMPLX(0.0, 0.0)
nodo(k)%JP3=CMPLX(0.0, 0.0)
nodo(k)%JP4=CMPLX(0.0, 0.0)

enddo LECTURA_Y_ASIGNACION

!-----
CALCULO_Y2: DO k=1,n2

call calculoY2 (nodo(k)%Y, nodo(nodo(k)%pre)%Y2, nodo(k)%Y2)

ENDDO CALCULO_Y2

```

```

!-----
VOLTAJES_INICIALES: do k=1, n2

!Establecimiento de voltajes nodales iniciales
!Se calculan con un barrido progresivo en vacio tomando en cuenta los reguladores
!(Aqui se aprovecha que las condiciones iniciales de corriente son cero, porque asi fueron
!inicializadas y en este caso permite hacer el barrido en vacio)

IF (k==1) then

nodo(k)%V1= Vrootc
nodo(k)%V2= Vrootc
nodo(k)%V3= Vrootc
nodo(k)%V4= Vrootc

ELSE

nodo(k)%V1= MATMUL(nodo(k)%A1, nodo(nodo(k)%pre)%V4)
nodo(k)%V2= MATMUL(nodo(k)%A2, nodo(k)%V1)
nodo(k)%V3= MATMUL(nodo(k)%A3, nodo(k)%V2)
nodo(k)%V4= MATMUL(nodo(k)%A4, nodo(k)%V3)

ENDIF

ENDDO VOLTAJES_INICIALES

!-----

VOLTAJES_REFERENCIA: do k=1, n2
!Establecimiento de voltajes nodales de referencia
!Se calculan con un barrido progresivo en vacio tomando SIN TOMAR en cuenta los reguladores
!Estos voltajes se supone serviran para calcular los voltajes en pu para cada nodo.
!Se supone que se aprovecha el hecho de que la fuente es simetrica y balanceada)-ojo-- yuo know!
!(Aqui se aprovecha que las condiciones iniciales de corriente son cero, porque asi fueron
!inicializadas y en este caso permite hacer el barrido en vacio)

IF (k==1) then

nodo(k)%VR1= Vrootc
nodo(k)%VR2= Vrootc
nodo(k)%VR3= Vrootc
nodo(k)%VR4= Vrootc

ELSE

nodo(k)%VR1= MATMUL(U3C, nodo(nodo(k)%pre)%VR4)
nodo(k)%VR2= MATMUL(nodo(k)%A2, nodo(k)%VR1)
nodo(k)%VR3= MATMUL(nodo(k)%A3, nodo(k)%VR2)
nodo(k)%VR4= MATMUL(U3C, nodo(k)%VR3)

ENDIF

ENDDO VOLTAJES_REFERENCIA

!-----
!Imprime Entrada

!call EntradaDetallada (filename,header, Vrootr,Vrootc, n2, nodo)

!-----
!Algoritmo de Barrido
#####
!05 FEBRERO 2010

!Anexión para MAYATOR

read(1,*)m2

CONDITION1: if(m2>0) then

allocate(malla(m2))

LecturaAsignacionMayas:do k=1, m2
read(1,*) temp2%in
read(1,*) temp2%from

```

```

        read(1,*) temp2%to
        read(1,*) temp2%Z
        read(1,*) temp2%Y

        malla(k)%m      = temp2%m
        malla(k)%from   = temp2%from
        malla(k)%to     = temp2%to
        call conversor_Zm(temp2%Z, malla(k)%Z)
        call conversor_Y (temp2%Y, malla(k)%Y )

    enddo LecturaAsignacionMayas

!suma la capacitancia de las mallas a los nodos de la red radial
do      zz=1, m2

nodo(malla(zz)%from)%Y2=  nodo(malla(zz)%from)%Y2 + (1.0 / 2.0)*malla(zz)%Y
nodo(malla(zz)%to  )%Y2=  nodo(malla(zz)%to  )%Y2 +      (1.0 / 2.0)*malla(zz)%Y

enddo

allocate(   thv%I(m2), thv%V(m2), thv%Z(m2,m2)      )

!Se inicializan las corrientes thv y los voltajes thv con cero

do k=1, m2

thv%I(k)%v31=cplx(0.0, 0.0)    !corregido 25feb2010
thv%V(k)%v31=cplx(0.0, 0.0)

enddo

!Se inicializa las impedancias thv%Z con cero por si las moscas (por si no hay mallas)

do k=1, m2
    do m=1, m2
        thv%Z(k,m)%m33=cplx(0.0, 0.0)
    enddo
enddo

!Se forma ahora la matriz thv%Z, compound con el algoritmo monofásico
!-----

!Se mete a la diagonal el valor de las impedancias de rama

do k=1, m2
    thv%Z(k,k)%m33 = malla(k)%Z
enddo

!asigna memoria e inicializa el vector de visitas de nodo

allocate(VISIT(n2))
VISIT=0

!-----

EXTERIOR: DO p=1, m2

VISIT=0

!Empieza con la ruta del "FROM"

t= malla(p)%from

do while(t>1) !Haga mientras no se ha alcanzado el nodo raiz

    thv%Z(p,p)%m33=thv%Z(p,p)%m33 + nodo(t)%Z
    visit(t)=1
    t=nodo(t)%pre

enddo

```

```

!y se sigue con la ruta del "TO"

t= malla(p)%to

do while(t>1) !Haga mientras no se ha alcanzado el nodo raiz

  IF(visit(t)==0) then

    thv%Z(p,p)%m33=thv%Z(p,p)%m33 + nodo(t)%Z
    visit(t)=1
    t=nodo(t)%pre

  ELSE
    !el nodo ya fue visitao
    thv%Z(p,p)%m33=thv%Z(p,p)%m33 - nodo(t)%Z
    visit(t)=0
    t=nodo(t)%pre
  ENDIF
enddo

      INTERIOR:DO q=p+1, m2

      extremo1=0
      extremo2=0

      !**
      !determinación de mallas internas
      !-----
      t= malla(p)%from
      do while(t>1)
      if (t==malla(q)%from .OR. t==malla(q)%to) extremo1=t
      t=nodo(t)%pre
      enddo

      t= malla(p)%to
      do while(t>1)
      if (t==malla(q)%from .OR. t==malla(q)%to) extremo2=t
      t=nodo(t)%pre
      enddo
      !-----

      t= malla(q)%from
      do while(t>1)
      if (t==malla(p)%from .OR. t==malla(p)%to) extremo1=t
      t=nodo(t)%pre
      enddo

      t= malla(q)%to
      do while(t>1)
      if (t==malla(p)%from .OR. t==malla(p)%to) extremo2=t
      t=nodo(t)%pre
      enddo
      !**

      t= malla(q)%from
      do while(t>1)
        if (visit(t)==1) thv%Z(p,q)%m33= thv%Z(p,q)%m33 + nodo(t)%Z
        t=nodo(t)%pre
      enddo

      t= malla(q)%to
      do while(t>1)

        if(extremo1>0 .AND. extremo2>0 .AND. extremo1/=extremo2) then

          if(visit(t)==1) thv%Z(p,q)%m33= thv%Z(p,q)%m33 + nodo(t)%Z
          t=nodo(t)%pre

        else

          if(visit(t)==1) thv%Z(p,q)%m33= thv%Z(p,q)%m33 - nodo(t)%Z
          t=nodo(t)%pre

        endif

      enddo

```

```

!CORRECCION PARA Numeros NEGATIVOS

do xx=1, 3
do yy=1, 3

if ( REAL( thv%Z(p,q)%m33(xx, yy)) < 0.0 ) thv%Z(p,q)%m33(xx, yy)= -
thv%Z(p,q)%m33(xx, yy) !ojo aca, no es tan directo

enddo
enddo
thv%Z(q,p)%m33 =thv%Z(p,q)%m33
ENDDO INTERIOR
ENDDO EXTERIOR

!!print*, 'matriz z'
!!print*, thv%Z

!Se asigna espacio para la matriz thevenin bruta y para los vectores de voltaje y corriente thevenin
allocate(Zthevenin (3*m2,3*m2))

allocate( Vthevenin (3*m2) )
allocate( Ithevenin (3*m2) )

!Se inicializan los vectores de voltaje/corriente Thevenin

Vthevenin = cmplx(0.0, 0.0)
Ithevenin = cmplx(0.0, 0.0)

!y se copian valores a Zthevenin (desde la matriz compound)

Do p=1, m2
Do q=1, m2

Zthevenin(3*p-2:3*p , 3*q-2:3*q) = thv%Z(p,q)%m33

Enddo
Enddo

!!call WRCRN('Zthevenin',3*m2, 3*m2, Zthevenin, 3*m2, 0) !quitar, solo prueba

!-----
!PROCESO DE KOMPACTAZION*****

!Se cuentan los elementos no nulos en la diagonal de Zth

r=0
do k=1,3*m2
IF( Zthevenin(k,k) /= cmplx(0.0, 0.0) ) r=r+1
enddo

!!print*, 'valor de R-----'
!!print*, r

!Se asigna memoria para la matriz de compactacion y las matrices compactadas de Z e Y

allocate(KPZ(r,3*m2))
allocate(Zthevenin_KPZ(r,r))
allocate(Ythevenin_KPZ(r,r))
allocate(Ythevenin (3*m2,3*m2))

!se llama a rutina de Kompactación para conocer la matriz KPZ
KPZ=0.0 !INICIALIZACION DE KOMPACTAZION
call KOMPACTAZION(Zthevenin,3*m2,r,KPZ)

! y se hacen las transformaciones

Zthevenin_KPZ = matmul(matmul(KPZ,Zthevenin),transpose(KPZ))
call LINCg(r,Zthevenin_KPZ,r,Ythevenin_KPZ,r) !obtiene la matriz Ythevenin compactada invirtiendo
Zthevenin

```

```

Ythevenin = matmul(matmul(transpose(KPZ),Ythevenin_KPZ),KPZ)    !and ahora se descompacta para Y
endif CONDITION1

#####
!call BackwardForward(nodo, n2, maxIter, iteraciones, converge, tolerancia )
call BackwardForward(nodo, n2, maxIter, iteraciones, converge, tolerancia, m2, Ithevenin, Ythevenin,
Vthevenin, malla, thv, iteracionesMacro, convergeMacro )    !23feb2010
#####

!-----
!Imprime Salida

!PRINT *, '>>LLEGO A RUTINAS DE IMPRESION'
!PAUSE '>>ENTER PARA CONTINUAR'

!call SalidaDetallada (filename, n2,m2, nodo, iteraciones, iteracionesMacro, converge, header)

!PRINT *, '>>TERMINO SALIDA DETALLADA'
!PAUSE '>>ENTER PARA CONTINUAR'
call SalidaDetallada_2 (filename, n2,m2, nodo, iteraciones, iteracionesMacro, converge, header)

!PRINT *, '>>TERMINO SALIDA DETALLADA2'
!PAUSE '>>ENTER PARA CONTINUAR'

!call SalidaRR (filename, n2, nodo, iteraciones, converge, header)    !salida para archivo de Reconf &
Restaur.

maxIter      = 3000                      !No. Máximo de Iteraciones Permitidas      (Cambiar desde aqui)
tolerancia    = 0.01                     !Tolerancia de error del proceso iterativo ((Cambiar desde aqui)

iteraciones =0                           !Iteraciones del algoritmo de Barrido (algoritmo de flujos o iterativo)
converge     =1                           !numero acumulativo de convergencia de nodos (para converger debe
alcanzar el máximo n2)
extremo1=0
extremo2=0
IMPRIMIR = 0
iteracionesMacro=0
convergeMacro=-1

CLOSE (UNIT =1)

PRINT*,'>>TERMINO RADIOFLUX'
PAUSE '>> ENTER PARA CONTINUAR'

!-----
END SUBROUTINE RADIFLUX

SUBROUTINE KOMPACTAZION(M,n,r,KPZ)

!Argumentos de Entrada Salida
complex, intent(in ), dimension(n,n)::M          !Matriz a compactar
integer, intent(in )      ::n                    !Orden de la matriz "M"
integer, intent(in )      ::r                    !No. Renglones de la Matriz KPZ
real , intent(out), dimension(r,n)::KPZ          !Matriz de Compactación
!Variable(s) internas
integer:: i,j
!-----*-----*-----*-----*-----*-----*-----*-----*-----*-----*-----*-----*-----*-----*-----*-----*

i=1
do j=1, n
if(M(j,j) /= cmplx(0.0, 0.0)) then

        KPZ(i,j)= 1.0          !cmplx(1.0, 0.0)
        i=i+1
endif
enddo

END SUBROUTINE KOMPACTAZION

MODULE BackForw                                !Algoritmo BackwardForward Trifásico
!+++++
CONTAINS

```

```

Subroutine BackwardForward(nodo, n2, maxIter, iteraciones, converge, tolerancia, m2, Ithevenin,
Ythevenin, Vthevenin, malla, thv, iteracionesMacro, convergeMacro )      !23feb2010

USE MSIMSL
USE estructuras
USE estructuras2
USE estructuras3
implicit none

!Declaracion de Variables I/O
integer, intent(in)::n2
type(lista), dimension(n2),intent(inout) :: nodo      !INOUT para meter todo y sacar voltaje

integer, intent(in)      :: maxIter
integer, intent(inout)   :: iteraciones

integer, intent(inout)   :: converge
real      , intent(in)   :: tolerancia

integer, intent(in)                                     :: m2
complex, dimension(3*m2)      , intent(inout):: Ithevenin
complex, dimension(3*m2, 3*m2), intent(in ):: Ythevenin
complex, dimension(3*m2)      , intent(inout):: Vthevenin

type(lista_mallas), dimension(m2), intent(in):: malla      !24feb2010
type(THV_est), intent(inout)      :: thv                  !24feb2010--!03mar2010

integer, intent(inout)::iteracionesMacro
integer, intent(inout)::convergeMacro
!Declaracion de Variables internas
integer ::k, t      !Contador(es)
complex, dimension(3):: V4antes
complex, dimension(3):: ERROR

!normalizacion a cantidaes reales  (de kV a V  y de kW a W)

NORMALIZACION: do k=1,n2

nodo(k)%V1=      nodo(k)%V1      * 1000.00
nodo(k)%V2=      nodo(k)%V2      * 1000.00
nodo(k)%V3=      nodo(k)%V3      * 1000.00
nodo(k)%V4=      nodo(k)%V4      * 1000.00

nodo(k)%VR1=      nodo(k)%VR1      * 1000.00
nodo(k)%VR2=      nodo(k)%VR2      * 1000.00
nodo(k)%VR3=      nodo(k)%VR3      * 1000.00
nodo(k)%VR4=      nodo(k)%VR4      * 1000.00

nodo(k)%S1YS =      nodo(k)%S1YS      * 1000.00
nodo(k)%S1YI =      nodo(k)%S1YI      * 1000.00
nodo(k)%S1YZ =      nodo(k)%S1YZ      * 1000.00
nodo(k)%S1DS =      nodo(k)%S1DS      * 1000.00
nodo(k)%S1DI =      nodo(k)%S1DI      * 1000.00
nodo(k)%S1DZ =      nodo(k)%S1DZ      * 1000.00

nodo(k)%S2YS =      nodo(k)%S2YS      * 1000.00
nodo(k)%S2YI =      nodo(k)%S2YI      * 1000.00
nodo(k)%S2YZ =      nodo(k)%S2YZ      * 1000.00
nodo(k)%S2DS =      nodo(k)%S2DS      * 1000.00
nodo(k)%S2DI =      nodo(k)%S2DI      * 1000.00
nodo(k)%S2DZ =      nodo(k)%S2DZ      * 1000.00

nodo(k)%S3YS =      nodo(k)%S3YS      * 1000.00
nodo(k)%S3YI =      nodo(k)%S3YI      * 1000.00
nodo(k)%S3YZ =      nodo(k)%S3YZ      * 1000.00
nodo(k)%S3DS =      nodo(k)%S3DS      * 1000.00
nodo(k)%S3DI =      nodo(k)%S3DI      * 1000.00
nodo(k)%S3DZ =      nodo(k)%S3DZ      * 1000.00

nodo(k)%S4YS =      nodo(k)%S4YS      * 1000.00
nodo(k)%S4YI =      nodo(k)%S4YI      * 1000.00
nodo(k)%S4YZ =      nodo(k)%S4YZ      * 1000.00
nodo(k)%S4DS =      nodo(k)%S4DS      * 1000.00

```

```

nodo(k)%S4DI =      nodo(k)%S4DI      * 1000.00
nodo(k)%S4DZ =      nodo(k)%S4DZ      * 1000.00

nodo(k)%Y2 =        nodo(k)%Y2        * 1.0E-6      !Se normaliza a Siemens por que estaba en micro-siemens
enddo NORMALIZACION

!-----

MACROITERACIONES: do while(convergeMacro<m2 .AND. iteracionesMacro<MaxIter)

convergeMacro=0

converge=1      !se pone aquí provisionalmente porque converge a la primera MacroIteracion &
                !y no deja que el ciclo de Macroiteraciones vuelva a entrar, !pudiera
                !inicializarse también con cero.

!-----

!-----
CICLO_DE_ITERACIONES: DO WHILE ( converge < n2      .and.      iteraciones < maxIter)

converge=1

!=====
!=====
Inyecciones_Nodales: DO k=1,n2

!!!Contribucion de las cargas conectadas en Estrella en todos los modelos (ZIP)

nodo(k)%I2= CONJG (nodo(k)% S2YS / nodo(k)% V2) + &

                CONJG (nodo(k)% S2YI / nodo(k)% VR2) * &

                EXP (CMPLX(0.0,  ATAN2( IMAG(nodo(k)% VR2),REAL(nodo(k)% VR2))-ATAN2( IMAG(nodo(k)%
V2),REAL(nodo(k)% V2)) ) )      +      &

                CONJG (nodo(k)% S2YZ / nodo(k)% V2) * (ABS( nodo(k)% V2 / nodo(k)% VR2)) ** 2.0

!!!Contribucion de la delta con potencia constante

nodo(k)%I2(1) = nodo(k)%I2 (1) + &
                CONJG(nodo(k)% S2DS(1) / (nodo(k)% V2(1)-nodo(k)% V2(2))) - CONJG(nodo(k)% S2DS(3) /
(nodo(k)% V2(3)-nodo(k)% V2(1)))

nodo(k)%I2(2) = nodo(k)%I2 (2) + &
                CONJG(nodo(k)% S2DS(2) / (nodo(k)% V2(2)-nodo(k)% V2(3))) - CONJG(nodo(k)% S2DS(1) /
(nodo(k)% V2(1)-nodo(k)% V2(2)))

nodo(k)%I2(3) = nodo(k)%I2 (3) + &
                CONJG(nodo(k)% S2DS(3) / (nodo(k)% V2(3)-nodo(k)% V2(1))) - CONJG(nodo(k)% S2DS(2) /
(nodo(k)% V2(2)-nodo(k)% V2(3)))

!!!Contribucion de la delta con corriente constante

nodo(k)%I2(1) = nodo(k)%I2 (1) + &

CONJG(nodo(k)% S2DI(1) / (nodo(k)% VR2(1)-nodo(k)% VR2(2)))* &

EXP (CMPLX( 0.0,  atan2(IMAG(nodo(k)% VR2(1)-nodo(k)% VR2(2)), REAL(nodo(k)% VR2(1)-nodo(k)% VR2(2)))-
atan2(IMAG(nodo(k)% V2(1)-nodo(k)% V2(2)), REAL(nodo(k)% V2(1)-nodo(k)% V2(2)))) - &

CONJG(nodo(k)% S2DI(3) / (nodo(k)% VR2(3)-nodo(k)% VR2(1)))      * &

EXP (CMPLX( 0.0,  atan2(IMAG(nodo(k)% VR2(3)-nodo(k)% VR2(1)), REAL(nodo(k)% VR2(3)-nodo(k)% VR2(1)))-
atan2(IMAG(nodo(k)% V2(3)-nodo(k)% V2(1)), REAL(nodo(k)% V2(3)-nodo(k)% V2(1)))) - &

nodo(k)%I2(2) = nodo(k)%I2 (2) + &

CONJG(nodo(k)% S2DI(2) / (nodo(k)% VR2(2)-nodo(k)% VR2(3)))* &

EXP (CMPLX( 0.0,  atan2(IMAG(nodo(k)% VR2(2)-nodo(k)% VR2(3)), REAL(nodo(k)% VR2(2)-nodo(k)% VR2(3)))-
atan2(IMAG(nodo(k)% V2(2)-nodo(k)% V2(3)), REAL(nodo(k)% V2(2)-nodo(k)% V2(3)))) - &

```

```

CONJG(nodo(k)% S2DI(1) / (nodo(k)% VR2(1)-nodo(k)% VR2(2))) * &

EXP (CMPLX( 0.0, atan2(IMAG(nodo(k)% VR2(1)-nodo(k)% VR2(2)), REAL(nodo(k)% VR2(1)-nodo(k)% VR2(2)))-
atan2(IMAG(nodo(k)% V2(1)-nodo(k)% V2(2)), REAL(nodo(k)% V2(1)-nodo(k)% V2(2)))))

nodo(k)%I2(3) = nodo(k)%I2 (3) + &

CONJG(nodo(k)% S2DI(3) / (nodo(k)% VR2(3)-nodo(k)% VR2(1)))* &

EXP (CMPLX( 0.0, atan2(IMAG(nodo(k)% VR2(3)-nodo(k)% VR2(1)), REAL(nodo(k)% VR2(3)-nodo(k)% VR2(1)))-
atan2(IMAG(nodo(k)% V2(3)-nodo(k)% V2(1)), REAL(nodo(k)% V2(3)-nodo(k)% V2(1))))) - &

CONJG(nodo(k)% S2DI(2) / (nodo(k)% VR2(2)-nodo(k)% VR2(3))) * &

EXP (CMPLX( 0.0, atan2(IMAG(nodo(k)% VR2(2)-nodo(k)% VR2(3)), REAL(nodo(k)% VR2(2)-nodo(k)% VR2(3)))-
atan2(IMAG(nodo(k)% V2(2)-nodo(k)% V2(3)), REAL(nodo(k)% V2(2)-nodo(k)% V2(3)))))

!nodo(k)%I2(2) = nodo(k)%I2 (2) + &
!
!      CONJG(nodo(k)% S2DI(2) / (nodo(k)% VR2(2)-nodo(k)% VR2(3))) - CONJG(nodo(k)% S2DI(1) /
(nodo(k)% VR2(1)-nodo(k)% VR2(2)))

!nodo(k)%I2(3) = nodo(k)%I2 (3) + &
!
!      CONJG(nodo(k)% S2DI(3) / (nodo(k)% VR2(3)-nodo(k)% VR2(1))) - CONJG(nodo(k)% S2DI(2) /
(nodo(k)% VR2(2)-nodo(k)% VR2(3)))

!!!Contribucion de la delta con impedancia constante

nodo(k)%I2(1) = nodo(k)%I2 (1) + &
      CONJG(nodo(k)% S2DZ(1) / (nodo(k)% V2(1)-nodo(k)% V2(2))) *
      (ABS((nodo(k)% V2(1)-nodo(k)% V2(2)) / (nodo(k)% VR2(1)-nodo(k)%
VR2(2))))**2.0      &

      -CONJG(nodo(k)% S2DZ(3) / (nodo(k)% V2(3)-nodo(k)% V2(1))) *
&
      (ABS((nodo(k)% V2(3)-nodo(k)% V2(1)) / (nodo(k)% VR2(3)-nodo(k)%
VR2(1))))**2.0

nodo(k)%I2(2) = nodo(k)%I2 (2) + &
      CONJG(nodo(k)% S2DZ(2) / (nodo(k)% V2(2)-nodo(k)% V2(3))) *
      (ABS((nodo(k)% V2(2)-nodo(k)% V2(3)) / (nodo(k)% VR2(2)-nodo(k)%
VR2(3))))**2.0      &

      -CONJG(nodo(k)% S2DZ(1) / (nodo(k)% V2(1)-nodo(k)% V2(2))) *
&
      (ABS((nodo(k)% V2(1)-nodo(k)% V2(2)) / (nodo(k)% VR2(1)-nodo(k)%
VR2(2))))**2.0

nodo(k)%I2(3) = nodo(k)%I2 (3) + &
      CONJG(nodo(k)% S2DZ(3) / (nodo(k)% V2(3)-nodo(k)% V2(1))) *
      (ABS((nodo(k)% V2(3)-nodo(k)% V2(1)) / (nodo(k)% VR2(3)-nodo(k)%
VR2(1))))**2.0      &

      -CONJG(nodo(k)% S2DZ(2) / (nodo(k)% V2(2)-nodo(k)% V2(3))) *
&
      (ABS((nodo(k)% V2(2)-nodo(k)% V2(3)) / (nodo(k)% VR2(2)-nodo(k)%
VR2(3))))**2.0

!%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
!%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
!%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

!!!Contribucion de las cargas conectadas en Estrella en todos los modelos (ZIP)

nodo(k)%I4= CONJG (nodo(k)% S4YS / nodo(k)% V4) + &

      CONJG (nodo(k)% S4YI / nodo(k)% VR4) * &

      EXP (CMPLX(0.0, ATAN2( IMAG(nodo(k)% VR4),REAL(nodo(k)% VR4))-ATAN2( IMAG(nodo(k)%
V4),REAL(nodo(k)% V4)) ) ) ) + &

```

```

CONJG (nodo(k)% S4YZ / nodo(k)% V4) * (ABS( nodo(k)% V4 / nodo(k)% VR4)) ** 2.0

!!!Contribucion de la delta con potencia constante

nodo(k)%I4(1) = nodo(k)%I4 (1) + &
                CONJG(nodo(k)% S4DS(1) / (nodo(k)% V4(1)-nodo(k)% V4(2))) - CONJG(nodo(k)% S4DS(3) /
(nodo(k)% V4(3)-nodo(k)% V4(1)))

nodo(k)%I4(2) = nodo(k)%I4 (2) + &
                CONJG(nodo(k)% S4DS(2) / (nodo(k)% V4(2)-nodo(k)% V4(3))) - CONJG(nodo(k)% S4DS(1) /
(nodo(k)% V4(1)-nodo(k)% V4(2)))

nodo(k)%I4(3) = nodo(k)%I4 (3) + &
                CONJG(nodo(k)% S4DS(3) / (nodo(k)% V4(3)-nodo(k)% V4(1))) - CONJG(nodo(k)% S4DS(2) /
(nodo(k)% V4(2)-nodo(k)% V4(3)))

!!!Contribucion de la delta con corriente constante

nodo(k)%I4(1) = nodo(k)%I4 (1) + &

CONJG(nodo(k)% S4DI(1) / (nodo(k)% VR4(1)-nodo(k)% VR4(2)))* &

EXP (CMPLX( 0.0, atan2(IMAG(nodo(k)% VR4(1)-nodo(k)% VR4(2)), REAL(nodo(k)% VR4(1)-nodo(k)% VR4(2)))-
atan2(IMAG(nodo(k)% V4(1)-nodo(k)% V4(2)), REAL(nodo(k)% V4(1)-nodo(k)% V4(2)))) - &

CONJG(nodo(k)% S4DI(3) / (nodo(k)% VR4(3)-nodo(k)% VR4(1))) * &

EXP (CMPLX( 0.0, atan2(IMAG(nodo(k)% VR4(3)-nodo(k)% VR4(1)), REAL(nodo(k)% VR4(3)-nodo(k)% VR4(1)))-
atan2(IMAG(nodo(k)% V4(3)-nodo(k)% V4(1)), REAL(nodo(k)% V4(3)-nodo(k)% V4(1)))) - &

nodo(k)%I4(2) = nodo(k)%I4 (2) + &

CONJG(nodo(k)% S4DI(2) / (nodo(k)% VR4(2)-nodo(k)% VR4(3)))* &

EXP (CMPLX( 0.0, atan2(IMAG(nodo(k)% VR4(2)-nodo(k)% VR4(3)), REAL(nodo(k)% VR4(2)-nodo(k)% VR4(3)))-
atan2(IMAG(nodo(k)% V4(2)-nodo(k)% V4(3)), REAL(nodo(k)% V4(2)-nodo(k)% V4(3)))) - &

CONJG(nodo(k)% S4DI(1) / (nodo(k)% VR4(1)-nodo(k)% VR4(2))) * &

EXP (CMPLX( 0.0, atan2(IMAG(nodo(k)% VR4(1)-nodo(k)% VR4(2)), REAL(nodo(k)% VR4(1)-nodo(k)% VR4(2)))-
atan2(IMAG(nodo(k)% V4(1)-nodo(k)% V4(2)), REAL(nodo(k)% V4(1)-nodo(k)% V4(2)))) - &

nodo(k)%I4(3) = nodo(k)%I4 (3) + &

CONJG(nodo(k)% S4DI(3) / (nodo(k)% VR4(3)-nodo(k)% VR4(1)))* &

EXP (CMPLX( 0.0, atan2(IMAG(nodo(k)% VR4(3)-nodo(k)% VR4(1)), REAL(nodo(k)% VR4(3)-nodo(k)% VR4(1)))-
atan2(IMAG(nodo(k)% V4(3)-nodo(k)% V4(1)), REAL(nodo(k)% V4(3)-nodo(k)% V4(1)))) - &

CONJG(nodo(k)% S4DI(2) / (nodo(k)% VR4(2)-nodo(k)% VR4(3))) * &

EXP (CMPLX( 0.0, atan2(IMAG(nodo(k)% VR4(2)-nodo(k)% VR4(3)), REAL(nodo(k)% VR4(2)-nodo(k)% VR4(3)))-
atan2(IMAG(nodo(k)% V4(2)-nodo(k)% V4(3)), REAL(nodo(k)% V4(2)-nodo(k)% V4(3)))) - &

!nodo(k)%I4(1) = nodo(k)%I4 (1) + &
!
                CONJG(nodo(k)% S4DI(1) / (nodo(k)% VR4(1)-nodo(k)% VR4(2))) - CONJG(nodo(k)% S4DI(3) /
(nodo(k)% VR4(3)-nodo(k)% VR4(1)))

!nodo(k)%I4(2) = nodo(k)%I4 (2) + &
!
                CONJG(nodo(k)% S4DI(2) / (nodo(k)% VR4(2)-nodo(k)% VR4(3))) - CONJG(nodo(k)% S4DI(1) /
(nodo(k)% VR4(1)-nodo(k)% VR4(2)))

!nodo(k)%I4(3) = nodo(k)%I4 (3) + &
!
                CONJG(nodo(k)% S4DI(3) / (nodo(k)% VR4(3)-nodo(k)% VR4(1))) - CONJG(nodo(k)% S4DI(2) /
(nodo(k)% VR4(2)-nodo(k)% VR4(3)))

!!!Contribucion de la delta con impedancia constante

nodo(k)%I4(1) = nodo(k)%I4 (1) + &

```

```

CONJG(nodo(k)% S4DZ(1) / (nodo(k)% V4(1)-nodo(k)% V4(2))) *
VR4(2)))**2.0      &
                    (ABS((nodo(k)% V4(1)-nodo(k)% V4(2)) / (nodo(k)% VR4(1)-nodo(k)%
&
                    -CONJG(nodo(k)% S4DZ(3) / (nodo(k)% V4(3)-nodo(k)% V4(1))) *
&
                    (ABS((nodo(k)% V4(3)-nodo(k)% V4(1)) / (nodo(k)% VR4(3)-nodo(k)%
VR4(1)))**2.0

nodo(k)%I4(2) = nodo(k)%I4 (2) + &
CONJG(nodo(k)% S4DZ(2) / (nodo(k)% V4(2)-nodo(k)% V4(3))) *
VR4(3)))**2.0      &
                    (ABS((nodo(k)% V4(2)-nodo(k)% V4(3)) / (nodo(k)% VR4(2)-nodo(k)%
&
                    -CONJG(nodo(k)% S4DZ(1) / (nodo(k)% V4(1)-nodo(k)% V4(2))) *
&
                    (ABS((nodo(k)% V4(1)-nodo(k)% V4(2)) / (nodo(k)% VR4(1)-nodo(k)%
VR4(2)))**2.0

nodo(k)%I4(3) = nodo(k)%I4 (3) + &
CONJG(nodo(k)% S4DZ(3) / (nodo(k)% V4(3)-nodo(k)% V4(1))) *
VR4(1)))**2.0      &
                    (ABS((nodo(k)% V4(3)-nodo(k)% V4(1)) / (nodo(k)% VR4(3)-nodo(k)%
&
                    -CONJG(nodo(k)% S4DZ(2) / (nodo(k)% V4(2)-nodo(k)% V4(3))) *
&
                    (ABS((nodo(k)% V4(2)-nodo(k)% V4(3)) / (nodo(k)% VR4(2)-nodo(k)%
VR4(3)))**2.0

!!!Contribucion de la Admitancia de Carga

nodo(k)%I4 = nodo(k)%I4 + MATMUL(nodo(k)%Y2 , nodo(k)%V4)

!contribucion de la corriente de los break points (mayator) 23feb2010
nodo(k)%I4 = nodo(k)%I4 + nodo(k)%Ibreak

END DO Inyecciones_Nodales

!=====
!=====
!Inicializacion de corrientes de incoming branch (amperes)

DO k=1, n2

nodo(k)%JP4 = nodo(k)%I4
nodo(k)%J4 = MATMUL(nodo(k)%d4, nodo(k)%JP4)

nodo(k)%JP3 = nodo(k)%I3 + nodo(k)%J4
nodo(k)%J3 = MATMUL(nodo(k)%d3, nodo(k)%JP3)

nodo(k)%JP2 = nodo(k)%I2 + nodo(k)%J3
nodo(k)%J2 = MATMUL(nodo(k)%d2, nodo(k)%JP2)

nodo(k)%JP1 = nodo(k)%I1 + nodo(k)%J2
nodo(k)%J1 = MATMUL(nodo(k)%d1, nodo(k)%JP1)

enddo

!=====
!=====
!Barrido Regresivo

Backward_Sweep: DO k=n2, 2, -1

nodo(nodo(k)%pre)%JP4 = nodo(nodo(k)%pre)%JP4 + nodo(k)%J1
nodo(nodo(k)%pre)%J4 = nodo(nodo(k)%pre)%J4 + MATMUL(nodo(nodo(k)%pre)%d4 , nodo(k)%J1)

nodo(nodo(k)%pre)%JP3 = nodo(nodo(k)%pre)%JP3 + MATMUL(nodo(nodo(k)%pre)%d4 , nodo(k)%J1)

```

```

        nodo(nodo(k)%pre)%J3 = nodo(nodo(k)%pre)%J3 + MATMUL(nodo(nodo(k)%pre)%d3,
MATMUL(nodo(nodo(k)%pre)%d4 , nodo(k)%J1))

        nodo(nodo(k)%pre)%JP2 = nodo(nodo(k)%pre)%JP2 + MATMUL(nodo(nodo(k)%pre)%d3,
MATMUL(nodo(nodo(k)%pre)%d4 , nodo(k)%J1))
        nodo(nodo(k)%pre)%J2 = nodo(nodo(k)%pre)%J2 + MATMUL(nodo(nodo(k)%pre)%d2,
MATMUL(nodo(nodo(k)%pre)%d3, MATMUL(nodo(nodo(k)%pre)%d4 , nodo(k)%J1)))

        nodo(nodo(k)%pre)%JP1 = nodo(nodo(k)%pre)%JP1 + MATMUL(nodo(nodo(k)%pre)%d2,
MATMUL(nodo(nodo(k)%pre)%d3, MATMUL(nodo(nodo(k)%pre)%d4 , nodo(k)%J1)))
        nodo(nodo(k)%pre)%J1 = nodo(nodo(k)%pre)%J1 + MATMUL(nodo(nodo(k)%pre)%d1,
MATMUL(nodo(nodo(k)%pre)%d2, MATMUL(nodo(nodo(k)%pre)%d3, MATMUL(nodo(nodo(k)%pre)%d4 , nodo(k)%J1))))

ENDDO Backward_Sweep

!=====
!=====
!Barrido Progresivo

Forward_Sweep: DO k=2,n2

!-----
V4antes=nodo(k)%V4
!-----
        nodo(k)%V1= MATMUL(nodo(k)%A1, nodo(nodo(k)%pre)%V4) - MATMUL(nodo(k)%B1, nodo(k)%JP1)
        nodo(k)%V2= MATMUL(nodo(k)%A2, nodo(k)%V1) - MATMUL(nodo(k)%B2, nodo(k)%JP2)
        nodo(k)%V3= MATMUL(nodo(k)%A3, nodo(k)%V2) - MATMUL(nodo(k)%B3, nodo(k)%JP3)
        nodo(k)%V4= MATMUL(nodo(k)%A4, nodo(k)%V3) - MATMUL(nodo(k)%B4, nodo(k)%JP4)
!-----

ERROR = V4antes - nodo(k)%V4

        if( abs(ERROR(1))<= tolerancia .AND. abs(ERROR(2)) <= tolerancia .AND. abs(ERROR(3)) <= tolerancia)
converge=converge+1

ENDDO Forward_Sweep

!=====
!=====

iteraciones=iteraciones + 1

ENDDO CICLO_DE_ITERACIONES

!=====
!=====

iBREAK_CONDITION: if(m2>0) then

!Update of Brakpoint Currents--- 23feb2010*****

UPDATE_BKP_CURRENTS: do k=1, m2

        thv%V(k)%v31 = ( nodo(malla(k)%from)%V4 - MATMUL (malla(k)%Z , thv%I(k)%v31)) -
nodo(malla(k)%to)%V4

end do UPDATE_BKP_CURRENTS

do k=1, m2

        Vthevenin(3*k-2:3*k) = thv%V(k)%v31

enddo

Ithevenin = Ithevenin + MATMUL ( Ythevenin, Vthevenin )
do k=1, m2

nodo(malla(k)%from)%Ibreak = Ithevenin(3*k-2:3*k)
nodo(malla(k)%to)%Ibreak = - Ithevenin(3*k-2:3*k)

enddo

parche: do k=1, m2

        thv%I(k)%v31 = Ithevenin(3*k-2:3*k)

end do parche

```

```

DO k=1, m2

if( abs(thv%V(k)%v31(1))<= tolerancia .AND. abs(thv%V(k)%v31(2)) <= tolerancia .AND.
abs(thv%V(k)%v31(3)) <= tolerancia) convergeMacro=convergeMacro+1

ENDDO

iteracionesMacro=iteracionesMacro+1

ENDIF iBREAK_CONDITION

ENDDO MACROITERACIONES

!Calculo de Flujos en Lineas y Cargas Totales- Una vez conocidos los voltajes nodales y corrientes de
rama y carga

!calculo de potencias de carga

do k=1, n2
do t=1, 3
nodo(k)%SI1(t)= nodo(k)%V1(t)* CONJG (nodo(k)%I1(t))
nodo(k)%SI2(t)= nodo(k)%V2(t)* CONJG (nodo(k)%I2(t))
nodo(k)%SI3(t)= nodo(k)%V3(t)* CONJG (nodo(k)%I3(t))
nodo(k)%SI4(t)= nodo(k)%V4(t)* CONJG (nodo(k)%I4(t))
enddo
enddo

!calculo de caidas de tension en branches!

do k=2, n2
nodo(k)%e1= MATMUL(nodo(k)%B1, nodo(k)%JP1)
nodo(k)%e2= MATMUL(nodo(k)%B2, nodo(k)%JP2)
nodo(k)%e3= MATMUL(nodo(k)%B3, nodo(k)%JP3)
nodo(k)%e4= MATMUL(nodo(k)%B4, nodo(k)%JP4)
enddo

!calculo de potencias de perdida en branches
do k=2, n2
do t=1, 3
nodo(k)%SJP1(t)= nodo(k)%e1(t)* CONJG (nodo(k)%JP1(t))
nodo(k)%SJP2(t)= nodo(k)%e2(t)* CONJG (nodo(k)%JP2(t))
nodo(k)%SJP3(t)= nodo(k)%e3(t)* CONJG (nodo(k)%JP3(t))
nodo(k)%SJP4(t)= nodo(k)%e4(t)* CONJG (nodo(k)%JP4(t))

enddo
enddo

!Se suma la carga total por rama-nodo

do k=1, n2
nodo(k)%SI= nodo(k)%SI1 + nodo(k)%SI2 + nodo(k)%SI3 + nodo(k)%SI4
nodo(k)%SJP= nodo(k)%SJP1 + nodo(k)%SJP2 + nodo(k)%SJP3 + nodo(k)%SJP4
enddo

do k=2, n2
do t=1,3

if( nodo(k)%Z3(t,t) == CMPLX(0.0, 0.0) .AND. nodo(k)%tybr /= -1 ) nodo(k)%V4(t)=
CMPLX(0.00, 0.00)

enddo
enddo
END Subroutine BackwardForward

END MODULE BackForw

```



```

CASE(2) !El regulador es tipo B

      Conetsion: SELECT CASE (cx_Reg) !Modifica de acuerdo a la conexion

            CASE(1) !El regulador esta conectado en estrella

                  d(1,1)= (1.0 / ar(1) ) * d(1,1)
                  d(2,2)= (1.0 / ar(2) ) * d(2,2)
                  d(3,3)= (1.0 / ar(3) ) * d(3,3)

                  A(1,1)= (1.0 / ar(1) ) * A(1,1)
                  A(2,2)= (1.0 / ar(2) ) * A(2,2)
                  A(3,3)= (1.0 / ar(3) ) * A(3,3)

            CASE(2) !El regulador esta conectado en delta

                  !Todavia no esta definifo!!!!!!
            END SELECT Conetsion

            END SELECT Tipo

end subroutine modify_dAB14
END subroutine dAB14
End Module ConstantesG14

Module ConstantesG2
CONTAINS
subroutine dAB2 (Z2, d2, A2, B2)

USE MCons

implicit none

!Declaracion de Variables I/O

complex, dimension(3,3), intent(in):: Z2

complex, dimension(3,3), intent(out):: d2
complex, dimension(3,3), intent(out):: A2
complex, dimension(3,3), intent(out):: B2

!El elemento 2 siempre será una linea sin importar si hay o no carga distribuida

d2= U3C

A2=d2

B2=Z2
End subroutine dAB2
End Module ConstantesG2
Module ConstantesG3

subroutine dAB3 (tybr,nt, Z3, d3, A3, B3)

use MCons

implicit none

!Declaracion de Variables I/O

integer, intent(in):: tybr
real, intent(in):: nt
complex, dimension(3,3), intent(in):: Z3

complex, dimension(3,3), intent(out):: d3
complex, dimension(3,3), intent(out):: A3
complex, dimension(3,3), intent(out):: B3

Selector_Constantes: SELECT CASE (tybr)

CASE(1)      !Linea Aérea, Linea Subterranea -----

d3=U3C
A3=d3
B3=Z3

```

```

CASE(2)      !Transformador D-Yg-----

d3=CMPLX(0.0, 0.0)

d3(1,1)=CMPLX( 1.0, 0.0)
d3(1,2)=CMPLX(-1.0, 0.0)
d3(2,2)=CMPLX( 1.0, 0.0)
d3(2,3)=CMPLX(-1.0, 0.0)
d3(3,1)=CMPLX(-1.0, 0.0)
d3(3,3)=CMPLX( 1.0, 0.0)

d3=(1.0 / nt) * d3

A3=CMPLX(0.0, 0.0)

A3(1,1)=CMPLX( 1.0, 0.0)
A3(1,3)=CMPLX(-1.0, 0.0)
A3(2,1)=CMPLX(-1.0, 0.0)
A3(2,2)=CMPLX( 1.0, 0.0)
A3(3,2)=CMPLX(-1.0, 0.0)
A3(3,3)=CMPLX( 1.0, 0.0)

A3=(1.0 / nt) * A3

B3=Z3

CASE(3)      !Transformador Yg-Yg-----

d3=U3C

d3=(1.0 / nt) * d3

A3=d3

B3=Z3

CASE DEFAULT

d3=U3C
A3=U3C
B3=CMPLX(0.0, 0.0)

END SELECT Selector_Constantes

End subroutine dAB3
End Module ConstantesG3

Module estructuras

implicit none

!*****
type::lectura
!-----
character(10)::      busName  !nombre del nodo

integer::              n          !numero de nodo
integer::              pre        !predecesor
integer::              niv        !nivel
integer::              tybr       !tipo de rama (line, transformer...)
real::                 nt         !relacion de espiras (de primario a secundario)

real, dimension(12)::Z          !Impedancia of incoming branch
real, dimension(6):: Y          !Admitancia of incoming branch microSiemens

integer::              cx_Sc      !conexion de la carga concentrada(1:estrella, 2:delta)
integer::              md_Sc      !modelo de la carga concentrada(1:S , 2: I,  3: Z)  Potencia,
Corriente o Impedancia constante
real, dimension(6):: Sc          !Carga concentrada del nodo en kW

integer::              cx_Sd      !conexion de la carga distribuida(1:estrella, 2:delta)
integer::              md_Sd      !modelo de la carga distribuida(1:S , 2: I,  3: Z)  Potencia,
Corriente o Impedancia constante

```

```

real, dimension(6):: Sd          !Carga distribuida del nodo en kW

integer::          cx_Bc        !Conexion del Banco de capacitores; 1:estrella, 2: delta (no
acoplamiento)
real, dimension(3):: SBc        !Potencia reactiva del banco de capacitores en kVAr

integer::          lc_Reg        !Ubicacion del Regulador      (1: principio, 2: Final)
integer::          ty_Reg        !Tipo de Regulador      (1: Tipo A, 2: Tipo B. See Kersting)
integer::          cx_Reg        !Conexion del Regulador      (1: Estrella, 2: Delta)
real, dimension(3):: Tap        !Ajuste de Tap de regulador para fases a-b-c.

end type lectura

type::lista
character(10)::      busName      !nombre del nodo

integer::n
integer::pre
integer::niv
integer::tybr
real::      nt

complex, dimension(3,3):: Z
complex, dimension(3,3):: Y

integer::          cx_Sc
integer::          md_Sc
complex, dimension(3):: Sc

integer::          cx_Sd
integer::          md_Sd
complex, dimension(3):: Sd

integer::          cx_Bc
complex, dimension(3):: SBc

integer::          lc_Reg
integer::          ty_Reg
integer::          cx_Reg
real, dimension(3):: Tap
!-----
complex, dimension(3)      :: V1, V2, V3, V4
complex, dimension(3)      :: VR1, VR2, VR3, VR4
complex, dimension(3,3)    :: Y2          !Admitancia total de carga, en el nodo

complex, dimension(3,3)    :: Z1, Z2, Z3, Z4

complex, dimension(3)      :: S1YS, S2YS, S3YS, S4YS
complex, dimension(3)      :: S1YI, S2YI, S3YI, S4YI
complex, dimension(3)      :: S1YZ, S2YZ, S3YZ, S4YZ

complex, dimension(3)      :: S1DS, S2DS, S3DS, S4DS
complex, dimension(3)      :: S1DI, S2DI, S3DI, S4DI
complex, dimension(3)      :: S1DZ, S2DZ, S3DZ, S4DZ

!-----
complex, dimension(3,3)::d1, d2, d3, d4          !for backward sweep
complex, dimension(3,3)::A1, A2, A3, A4          !for forward sweep
complex, dimension(3,3)::B1, B2, B3, B4          !for forward sweep

COMPLEX, dimension(3):: I1, I2, I3, I4          !Inyeccion de corrientes nodal (total)
COMPLEX, dimension(3):: J1, J2, J3, J4          !Corriente de incoming BRANCH
COMPLEX, dimension(3):: JP1, JP2, JP3, JP4      !Corriente de incoming BRANCH primadas

real, dimension(3):: ar          !ar del regulador (see Kersting)

complex, dimension(3)::Ibreak          !Inyeccion de corrientes de Brakpoints
complex, dimension(3):: SI1, SI2, SI3, SI4
complex, dimension(3):: SJP1, SJP2, SJP3, SJP4
complex, dimension(3):: e1, e2, e3, e4

complex, dimension(3):: SI
complex, dimension(3):: SJP

end type lista

```

```

end module estructuras

Module Estructuras2

implicit none
type::lectura_mallas

integer::
m
from !FROM
to !TO
!numero de malla

real, dimension(12)::Z
real, dimension(6)::Y
!Impedancia of incoming branch
!Admitancia of incoming branch microSiemens

end type lectura_mallas
type::lista_mallas
integer::m
integer::from
integer::to

complex, dimension(3,3):: Z
complex, dimension(3,3):: Y
end type lista_mallas

END Module Estructuras2

Module Estructuras3 !15 feb 2010

implicit none
type:: vec31
complex, dimension(3)::v31
end type vec31

type:: mat33
complex, dimension(3,3)::m33
end type mat33

type::THV_est
!thevenin_estructura
type(vec31), pointer, dimension(:):: I
type(vec31), pointer, dimension(:):: V
type(mat33), pointer, dimension(:,:):: Z
!type(mat33), pointer, dimension(:,:):: Y
end type THV_est
End Module Estructuras3

Module Estructuras4

implicit none

type:: line

!*****
integer::
n
from !número de linea
character(10)::
from !self explained
character(10)::
to !self explained
!*****

complex, dimension(3)::
Jfrom !self explained (rectangular)
complex, dimension(3)::
Jto !self explained (rectangular)

real, dimension(3)::
Jfrom_mag !idem, pero magnitud (polar)
real, dimension(3)::
Jfrom_ang !idem, pero angulo (polar)

real, dimension(3)::
Jto_mag !idem, pero magnitud (polar)
real, dimension(3)::
Jto_ang !idem, pero angulo (polar)

!*****
complex, dimension(3)::
Sfrom !Self explained
complex, dimension(3)::
Sto !Self explained

end type line
End Module Estructuras4

```

```

module MCons
!Matrices constantes
IMPLICIT NONE
!Matriz unidad de 3x3 compleja
complex, dimension(3,3), parameter:: &
U3C=reshape((/(1.0, 0.0), (0.0, 0.0), (0.0, 0.0), (0.0, 0.0), (1.0, 0.0), (0.0, 0.0), (0.0, 0.0), (0.0,
0.0), (1.0, 0.0)/), (/3,3/))
end module MCons

MODULE MDATOS

IMPLICIT NONE

!se agrega como variable global n2(NUMERO DE NODOS)
INTEGER :: n2 ! NUMERO DE NODOS
INTEGER :: IMPRIMIR
INTEGER :: NODOPOT
INTEGER :: ACTUALIZAVOL
INTEGER :: ACTUALIZAPOT
COMPLEX*8, dimension(3) :: VDISNEW

END MODULE MDATOS

MODULE MDATOS2

IMPLICIT NONE

! MODULO PARA VARIABLES GLOBALES DE CONTROL DEL CIRCUITO DE DISTRIBUCION

REAL*8 :: PDIS(3), QDIS(3)
COMPLEX*8 :: VDIS(3,1) ! VECTOR DE VOLTAJES INICIALES DE FASE EN FORMA RECTAGULAR
REAL*8 :: P3D, Q3D

END MODULE MDATOS2

MODULE MDATOS3
IMPLICIT NONE

INTEGER :: NONODOS, REGRESA
REAL*8, ALLOCATABLE :: MAGV(:), ANGV(:), ANGRADOS(:), VBASE(:)
REAL*8, ALLOCATABLE :: VTA(:), ANGTA(:), VTB(:), ANGTB(:), VTC(:), ANGTC(:) ! MANGITUD Y
ANGULO DE MEDIA TENSION &
COMPLEX*8, PARAMETER :: a = (-0.5,0.866025403)

END MODULE MDATOS3

module PrintSalidaDetallada_2
!
contains

Subroutine SalidaDetallada_2 (n_archivo ,n2,m2, nodo, iteraciones, iteracionesMacro, converge, header)

USE MSIMSL

use estructuras
use UtileriasComplejas2

use Estructuras4
USE MDATOS2

Implicit none

!Declaracion de variables de I/O
character(20), intent(in) :: n_archivo
integer, intent(in) :: n2
integer, intent(in) :: m2
type(lista), dimension(:), intent(in) :: nodo

integer, intent(in):: iteraciones

```

```

integer, intent(in):: iteracionesMacro
integer, intent(in):: converge
character(len=60), intent (in)          :: header

!Declaracion de variables internas
integer:: k, q !contador (es)

complex, dimension(3):: Sint = (0.0 , 0.0)
complex, dimension(3):: Sout = (0.0 , 0.0)
complex, dimension(3):: Slos = (0.0 , 0.0)

type(line), dimension(n2-1):: linea          !solo incluye lineas radiales
!Apertura de archivo de Resultados de Flujos de Potencia
open(22, file='SalidaD2.txt')

!Asignación Estructura de flujo de corrientes de RAMA (Linea)

QWE: do k=1, n2-1

linea(k)%n=k

linea(k)%from= nodo( nodo(k+1)% pre ) % busName
linea(k)%to = nodo(k+1)% busName

linea(k)%Jfrom = nodo(k+1)%J1
linea(k)%Jto = nodo(k+1)%JP4

linea(k)%Jfrom_mag=abs(linea(k)%Jfrom)
do q=1, 3
  IF ( Imag( linea(k)%Jfrom(q) ) /= 0.0 .OR. Real( linea(k)%Jfrom(q) ) /= 0.0 ) then
    linea(k)%Jfrom_ang(q) = ATAN2D( Imag(linea(k)%Jfrom(q)), Real (linea(k)%Jfrom(q)) )
  ELSE
    linea(k)%Jfrom_ang(q)= 0.0
  ENDIF
enddo

linea(k)%Jto_mag=abs(linea(k)%Jto)
do q=1, 3
  IF ( Imag( linea(k)%Jto(q) ) /= 0.0 .OR. Real( linea(k)%Jto(q) ) /= 0.0 ) then
    linea(k)%Jto_ang(q) = ATAN2D( Imag(linea(k)%Jto(q)), Real (linea(k)%Jto(q)) )
  ELSE
    linea(k)%Jto_ang(q)= 0.0
  ENDIF
enddo

enddo QWE

!Asignación Estructura de flujo de potencias de RAMA (Linea)                                     30-03-2011

QWEr: do k=1, n2-1
linea(k)%Sfrom= nodo(k+1)%SI + nodo(k+1)%SJP
linea(k)%Sto = nodo(k+1)%SI
enddo QWEr

RRR: do k=n2-1, 1, -1

!linea(nodo(k)%n)%Sfrom = linea(nodo(k)%n)%Sfrom + linea(nodo(k+1)%pre)%Sfrom
!linea(k)%Sto = linea(k)%Sto + nodo(k+1)%SJP

linea( (nodo(k+1)%pre)-1 )%Sto = linea( (nodo(k+1)%pre)-1 )%Sto + linea( k )%Sfrom

linea( (nodo(k+1)%pre)-1 )%Sfrom = linea( (nodo(k+1)%pre)-1 )%Sto + nodo(nodo(k+1)% pre)%SJP

enddo RRR

!Comienza la Impresion.....
!=====
!Impresion de Encabezado de Reporte

```

```

!-----
write(22, '(/')
write(22, '("=====")')
write(22, '("                                RESULTADOS DE FLUJOS DE POTENCIA                                ")')
write(22, '("                                radiflux v2.1                                ")')
write(22, '("=====")')
!-----
write(22, '(/')

write(22, '("RED                                :", A60)') header
write(22, '("ARCHIVO DE ENTRADA:", A)')n_archivo
write(22, '(/')
!=====
!=====

IMPRIMIR: IF (converge<n2) then

write(22, '("RED                                :", A60)') header
write(22, '("ARCHIVO DE ENTRADA:", A)')n_archivo

write(22, '(/')

    if (m2==0) then
        write(22, '("NO SE ALCANZO LA CONVERGENCIA EN", 2X, I3, 2X, "ITERACIONES" )')ITERACIONES
    else
        write(22, '("NO SE ALCANZO LA CONVERGENCIA EN", 2X, I3, 2X, "ITERACIONES" )')iteracionesMacro
    endif

!write(22, '("Nodos en los que se alcanza la convergencia:", 2X, I3 )')CONVERGE

ELSE

    if (m2==0) then
        write(22, '("SE ALCANZO LA CONVERGENCIA EN", 2X, I3, 2X, "ITERACIONES" )')ITERACIONES
    else
        write(22, '("SE ALCANZO LA CONVERGENCIA EN", 2X, I3, 2X, "ITERACIONES" )')iteracionesMacro
    endif

!write(22, '("Nodos en los que se alcanza la convergencia:", 2X, I3 )')CONVERGE

write(22, '(/')
!=====
!=====

Sint=  nodo(1)%V1 * CONJG (nodo(1)%J1)

do k=2, n2

Sout= Sout  +  nodo(k)%SI

Slos= Slos  +  nodo(k)%SJP

enddo

!Normalizacion
Sint  = Sint / 1000.00          !Se normaliza a KVA
Sout  = Sout / 1000.00
Slos  = Slos / 1000.00

!POTENCIAS ACTIVAS Y REACTIVAS DE FASE EN MW Y MVAR

PDIS(1) = REAL(Sint(1))/1000.00
PDIS(2) = REAL(Sint(2))/1000.00
PDIS(3) = REAL(Sint(3))/1000.00

QDIS(1) = IMAG(Sint(1))/1000.00
QDIS(2) = IMAG(Sint(2))/1000.00
QDIS(3) = IMAG(Sint(3))/1000.00

```

```

=====
!IMPRESION DE RESUMEN TOTAL DEL SISTEMA

!-----
write(22, '("=====")')
write(22, '("
                                R E S U M E N   G E N E R A L
                                ")')
write(22, '("=====")')
!-----
!write(22, '()')
!write(22, '("-----")')
WRITE(22, ' (1X, "
                                ", T22, "kW", T34, "kVA", T47, "kVA")')
write(22, '("-----")')
write(22, ' (2X, "ENTRADA :", T18, F9.3, T30, F9.3, T43, F9.3)') Real(Sint(1)+Sint(2)+Sint(3)),
Imag(Sint(1)+Sint(2)+Sint(3)), Abs(Sint(1)+Sint(2)+Sint(3))
write(22, ' (2X, "CARGA   :", T18, F9.3, T30, F9.3, T43, F9.3)') Real(Sout(1)+Sout(2)+Sout(3)),
Imag(Sout(1)+Sout(2)+Sout(3)), Abs(Sout(1)+Sout(2)+Sout(3))
write(22, ' (2X, "PERDIDAS:", T18, F9.3, T30, F9.3, T43, F9.3)') Real(Slos(1)+Slos(2)+Slos(3)),
Imag(Slos(1)+Slos(2)+Slos(3)), Abs(Slos(1)+Slos(2)+Slos(3))
write(22, '("-----")')
write(22, '()')

!-----
write(22, '("=====")')
write(22, '("
                                PERFIL DE VOLTAJES
                                ")')
write(22, '("=====")')
!-----
write(22, '()')

!IMPRESION DE PERFIL DE VOLTAJE EN FORMA POLAR, CON UNIDADES REALES
write(22, '("PERFIL DE VOLTAJE -- (Forma Polar, kV @ °) ")')

write(22, '("-----")')
write(22, ' (2X, "-----N O D O-----", 4X, "----F A S E   A---", 4X, "----F A S E   B---", 4X, "----F A S E   C---")')
write(22, ' (3X, "NO.", 5X, "NOMBRE", 7X, "kV", 6X, "GRADOS", 6X, "kV", 6X, "GRADOS", 6X, "kV", 6X, "GRADOS",
6X )')
write(22, '("-----")')

do k=1, n2
CALL RecPol (22, nodo(k)%n, nodo(k)%busName, nodo(k)%V4 / 1000.00)      !lo divido entre 1000.00 para
que sea kV
enddo
write(22, '("-----")')
write(22, '()')

!IMPRESION DE PERFIL DE VOLTAJE EN FORMA POLAR, CON VALORES DE MAGNITUD EN P.U. Y ANGULO EN GRADOS
write(22, '("PERFIL DE VOLTAJE -- (Forma Polar, PU @ °) ")')

write(22, '("-----")')
write(22, ' (2X, "-----N O D O-----", 4X, "----F A S E   A---", 4X, "----F A S E   B---", 4X, "----F A S E   C---")')
write(22, ' (3X, "NO.", 5X, "NOMBRE", 7X, "PU", 6X, "GRADOS", 6X, "PU", 6X, "GRADOS", 6X, "PU", 6X, "GRADOS",
6X )')
write(22, '("-----")')

do k=1, n2
CALL RecPol_pu (22, nodo(k)%n, nodo(k)%busName, nodo(k)%V4 / abs(nodo(k)%VR4))
enddo
write(22, '("-----")')
write(22, '()')

```

```

!-----

```

```

write(22,'("=====")')
write(22,'("
                                FLUJO DE CORRIENTES
                                ")')
write(22,'("=====")')
write(22,'(2x,"-----NODO-----", 16x,"-----CORRIENTE-----")')
write(22,'(2x,"ENVIO",8x,"RECEPCION",5x,"FASE", 10x,"ENVIO",17x,"RECEPCIÓN")')
write(22,'(37x,"AMPERES", 4x,"GRADOS", 8x,"AMPERES", 4x,"GRADOS")')
write(22,'("=====")')
!-----

do k=1, n2-1
write(22,'(2x , A10,5x,A10, 3x, "A",3x, F10.3, 1x, "@",1x, F7.2, 5x, F10.3, 1x, "@",1x, F7.2 )')
linea(k)%from, linea(k)%to,linea(k)%Jfrom_mag(1), linea(k)%Jfrom_ang(1),linea(k)%Jto_mag(1),
linea(k)%Jto_ang(1)
write(22,'(30x,"B",3x, F10.3, 1x, "@",1x, F7.2, 5x, F10.3, 1x, "@",1x, F7.2 )')
linea(k)%Jfrom_mag(2), linea(k)%Jfrom_ang(2),linea(k)%Jto_mag(2), linea(k)%Jto_ang(2)
write(22,'(30x"C",3x, F10.3, 1x, "@",1x, F7.2, 5x, F10.3, 1x, "@",1x, F7.2 )')
linea(k)%Jfrom_mag(3), linea(k)%Jfrom_ang(3),linea(k)%Jto_mag(3), linea(k)%Jto_ang(3)
write(22,'(//)')
enddo
write(22,'("=====")')

write(22,'(///)')

write(22,'("=====")')
write(22,'("
                                FLUJO DE POTENCIAS
                                ")')
write(22,'("=====")')
write(22,'(2x,"-----NODO-----", 16x,"-----POTENCIA-----")')
write(22,'(2x,"ENVIO",8x,"RECEPCION",5x,"FASE", 10x,"ENVIO",17x,"RECEPCIÓN")')
!write(22,'(37x,"AMPERES", 4x,"GRADOS", 8x,"AMPERES", 4x,"GRADOS")')
write(22,'(37x," kW ", 4x," kVAr ", 8x," kW ", 4x," kVAr ")')

write(22,'("=====")')

do k=1, n2-1
write(22,'(2x , A10,5x,A10, 3x, "A",3x, F10.3, 3x, F7.2, 5x, F10.3, 3x, F7.2 )') linea(k)%from,
linea(k)%to,real(linea(k)%Sfrom(1))/1000.0, imag(linea(k)%Sfrom(1))/1000.0,real(linea(k)%Sto(1))/1000.0,
imag(linea(k)%Sto(1))/1000.0
write(22,'(30x,
                                "B",3x, F10.3, 3x, F7.2, 5x, F10.3, 3x, F7.2 )')
real(linea(k)%Sfrom(2))/1000.0, imag(linea(k)%Sfrom(2))/1000.0,real(linea(k)%Sto(2))/1000.0,
imag(linea(k)%Sto(2))/1000.0
write(22,'(30x,
                                "C",3x, F10.3, 3x, F7.2, 5x, F10.3, 3x, F7.2 )')
real(linea(k)%Sfrom(3))/1000.0, imag(linea(k)%Sfrom(3))/1000.0,real(linea(k)%Sto(3))/1000.0,
imag(linea(k)%Sto(3))/1000.0
write(22,'(//)')
enddo

write(22,'("=====")')

endif IMPRIMIR
End Subroutine SalidaDetallada_2
end module PrintSalidaDetallada_2

module rutinas_de_asignacion
!rutinas de conversion de datos para lista de barrido
CONTAINS
!*****
!*****
!*****
subroutine conversor_Z(Ztemp, Z, Z1, Z2, Z3, Z4)

implicit none

!Declaracion de variables I/O

real, dimension(12), intent(in)      ::Ztemp

complex, dimension(3,3), intent(out)::Z
complex, dimension(3,3), intent(out)::Z1, Z2, Z3, Z4

!Declaracion de variables internas

real:: FZ2 = 1.0 / 4.0
real:: FZ3 = 3.0 / 4.0

```

```

!Inicializacion de impedancias serie

Z = (0.0, 0.0)

Z1= (0.0, 0.0)    !siempre cero (Regulador sin pérdidas)
Z2= (0.0, 0.0)    !inicializa hasta saber si hay carga distribuida en la linea
Z3= (0.0, 0.0)    !inicializa hasta saber si hay carga distribuida en la linea
Z4= (0.0, 0.0)    !siempre cero (Regulador sin pérdidas)

!Se asigna la entrada de datos al contenedor Z que es una matriz compleja de 3X3

Z(1,1)= cmplx(Ztemp(1) ,  Ztemp(2) )
Z(1,2)= cmplx(Ztemp(3) ,  Ztemp(4) )
Z(1,3)= cmplx(Ztemp(5) ,  Ztemp(6) )
Z(2,2)= cmplx(Ztemp(7) ,  Ztemp(8) )
Z(2,3)= cmplx(Ztemp(9) ,  Ztemp(10) )
Z(3,3)= cmplx(Ztemp(11),  Ztemp(12) )

Z(2,1)= cmplx(Ztemp(3) ,  Ztemp(4) )
Z(3,1)= cmplx(Ztemp(5) ,  Ztemp(6) )
Z(3,2)= cmplx(Ztemp(9) ,  Ztemp(10) )

!Asignacion de la impedancia serie de entrada a Z3, como si "TODA" la carga fuese concentrada

Z3=Z

!Finalmente se refinan las Impedancias Z2 y Z3 con los factores auxiliares, repartiendo la impedancia
entre los elementos 2 y 3,
!Incluso si no hubiese carga distribuida (realmente los elementos estan en serie)

!Z2= FZ2 * Z3
!Z3= FZ3 * Z3

end subroutine conversor_Z
!*****
subroutine conversor_Y(Ytemp, Y)

implicit none

real, dimension(6), intent(in)::Ytemp
complex, dimension(3,3), intent(out)::Y

Y=cmplx(0.0, 0.0)

Y(1,1)=cmplx(0.0, 1.0*Ytemp(1))
Y(1,2)=cmplx(0.0, 1.0*Ytemp(2))
Y(1,3)=cmplx(0.0, 1.0*Ytemp(3))
Y(2,2)=cmplx(0.0, 1.0*Ytemp(4))
Y(2,3)=cmplx(0.0, 1.0*Ytemp(5))
Y(3,3)=cmplx(0.0, 1.0*Ytemp(6))

Y(2,1)=cmplx(0.0, 1.0*Ytemp(2))
Y(3,1)=cmplx(0.0, 1.0*Ytemp(3))
Y(3,2)=cmplx(0.0, 1.0*Ytemp(5))

end subroutine conversor_Y
!*****
SUBROUTINE conversor_carga( cx_Sc, md_Sc, Sc, cx_Sd, md_Sd, Sd, cx_Bc, SBc,      &
                           S1YS, S2YS, S3YS, S4YS,
                           &
                           S1YI, S2YI, S3YI, S4YI,
                           &
                           S1YZ, S2YZ, S3YZ, S4YZ,
                           &
                           S1DS, S2DS, S3DS, S4DS,
                           &
                           S1DI, S2DI, S3DI, S4DI,
                           &
                           S1DZ, S2DZ, S3DZ, S4DZ,
                           &
                           VSc, VSd, VSBc )

```

```

implicit none

!Declaracion de variables I/O

integer, intent (in)::      cx_Sc
integer, intent (in)::      md_Sc
real, dimension(6),intent(in)::  Sc

integer, intent (in)::      cx_Sd
integer, intent (in)::      md_Sd
real, dimension(6),intent(in)::  Sd

integer, intent (in)::      cx_Bc
real, dimension(3),intent(in)::  SBc

complex, dimension(3), intent(out):: S1YS, S2YS, S3YS, S4YS
complex, dimension(3), intent(out):: S1YI, S2YI, S3YI, S4YI
complex, dimension(3), intent(out):: S1YZ, S2YZ, S3YZ, S4YZ

complex, dimension(3), intent(out):: S1DS, S2DS, S3DS, S4DS

complex, dimension(3), intent(out):: S1DI, S2DI, S3DI, S4DI

complex, dimension(3), intent(out):: S1DZ, S2DZ, S3DZ, S4DZ

complex, dimension (3), intent(out)  :: VSc      !Vector de carga concentrada
complex, dimension (3), intent(out)  :: VSd      !Vector de carga distribuida
complex, dimension (3), intent(out)  :: VSBc     !Vector de carga de Banco Capacitivo

!Declaracion de variables internas

real:: FS2 = 2.0 / 3.0      !factores auxiliares para modelar lineas con carga distribuida
real:: FS4 = 1.0 / 3.0      !ver Kersting          (estos son para cargas)

!.....
VSc(1)=CMPLX(Sc(1),Sc(2))
VSc(2)=CMPLX(Sc(3),Sc(4))
VSc(3)=CMPLX(Sc(5),Sc(6))

VSd(1)=CMPLX(Sd(1),Sd(2))
VSd(2)=CMPLX(Sd(3),Sd(4))
VSd(3)=CMPLX(Sd(5),Sd(6))

VSBc(1)= CMPLX( 0.0 , -SBc(1) )
VSBc(2)= CMPLX( 0.0 , -SBc(2) )
VSBc(3)= CMPLX( 0.0 , -SBc(3) )

!.....
!Inicializacion de cargas

S1YS = (0.0, 0.0)
S2YS = (0.0, 0.0)
S3YS = (0.0, 0.0)
S4YS = (0.0, 0.0)

S1YI = (0.0, 0.0)
S2YI = (0.0, 0.0)
S3YI = (0.0, 0.0)
S4YI = (0.0, 0.0)

S1YZ = (0.0, 0.0)
S2YZ = (0.0, 0.0)
S3YZ = (0.0, 0.0)
S4YZ = (0.0, 0.0)

S1DS = (0.0, 0.0)
S2DS = (0.0, 0.0)
S3DS = (0.0, 0.0)
S4DS = (0.0, 0.0)

S1DI = (0.0, 0.0)

```

```

S2DI = (0.0, 0.0)
S3DI = (0.0, 0.0)
S4DI = (0.0, 0.0)

S1DZ = (0.0, 0.0)
S2DZ = (0.0, 0.0)
S3DZ = (0.0, 0.0)
S4DZ = (0.0, 0.0)

!.....
IF (cx_Sc == 1 .AND. md_Sc == 1) S4YS = VSc
IF (cx_Sc == 1 .AND. md_Sc == 2) S4YI = VSc
IF (cx_Sc == 1 .AND. md_Sc == 3) S4YZ = VSc

IF (cx_Sc == 2 .AND. md_Sc == 1) S4DS = VSc
IF (cx_Sc == 2 .AND. md_Sc == 2) S4DI = VSc
IF (cx_Sc == 2 .AND. md_Sc == 3) S4DZ = VSc

IF (cx_Sd == 1) THEN

    SELECT CASE (md_Sd)

        CASE (1)

            S2YS = FS2 * VSd
            S4YS = S4YS + FS4 * VSd

        CASE (2)

            S2YI = FS2 * VSd
            S4YI = S4YI + FS4 * VSd

        CASE (3)

            S2YZ = FS2 * VSd
            S4YZ = S4YZ + FS4 * VSd

    ENDSELECT

ENDIF

IF (cx_Sd == 2) THEN

    SELECT CASE (md_Sd)

        CASE (1)

            S2DS = FS2 * VSd
            S4DS = S4DS + FS4 * VSd

        CASE (2)

            S2DI = FS2 * VSd
            S4DI = S4DI + FS4 * VSd

        CASE (3)

            S2DZ = FS2 * VSd
            S4DZ = S4DZ + FS4 * VSd

    ENDSELECT

ENDIF

IF (cx_Bc == 1) S4YZ = S4YZ + VSBc
IF (cx_Bc == 2) S4DZ = S4DZ + VSBc

end subroutine conversor_carga
subroutine arReg (Tap, ar)

implicit none

!Declaracion de variables I/O

real, dimension(3), intent(in) :: Tap
real, dimension(3), intent(out) :: ar

```

```

        ar(1) = 1.0 +    0.00625 * Tap(1)
        ar(2) = 1.0 +    0.00625 * Tap(2)
        ar(3) = 1.0 +    0.00625 * Tap(3)
    end subroutine arReg

subroutine conversor_voltaje(Vrootr, Vrootc)
implicit none

!Declaracion de variables I/O
real, dimension(6), intent(in):: Vrootr      !kV polar
complex, dimension(3), intent(out)::Vrootc    !kV      rectangular
real, parameter::pi=3.141593

!Declaracion de variables internas
real, dimension(3)::mag,tetha                !contenedor de magnitud y angulos en radianes

!::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
!Se inicializa la salida con valor nulo

Vrootc=cplx(0.0, 0.0)

!Se preprocesan los datos de entrada
mag(1)=Vrootr(1)
mag(2)=Vrootr(3)
mag(3)=Vrootr(5)

tetha(1)=Vrootr(2)*(pi/180.0)
tetha(2)=Vrootr(4)*(pi/180.0)
tetha(3)=Vrootr(6)*(pi/180.0)

!Se llena el vector complejo
Vrootc(1)=cplx(mag(1)*cos(tetha(1)), mag(1)*sin(tetha(1)))
Vrootc(2)=cplx(mag(2)*cos(tetha(2)), mag(2)*sin(tetha(2)))
Vrootc(3)=cplx(mag(3)*cos(tetha(3)), mag(3)*sin(tetha(3)))

end subroutine conversor_voltaje

subroutine calculoY2 ( Y, Y2pre, Y2k)

implicit none

!Declaracion de variables I/O
complex, dimension(3,3), intent(in):: Y
complex, dimension(3,3), intent(out):: Y2pre
complex, dimension(3,3), intent(out):: Y2k

Y2k = Y2k + (1.0 / 2.0) * Y
Y2pre= Y2pre + (1.0 / 2.0) * Y

end subroutine calculoY2

subroutine conversor_Zm(Ztemp, Z) !subrutina para convertir la entrada de la impedancia de mallas a
complejos
!05feb2010

implicit none
!Declaracion de variables I/O

real, dimension(12), intent(in)      ::Ztemp

complex, dimension(3,3), intent(out)::Z

!Inicializacion de impedancias serie
Z = (0.0, 0.0)

!Se asigna la entrada de datos al contenedor Z que es una matriz compleja de 3X3
Z(1,1)= cplx(Ztemp(1) , Ztemp(2) )
Z(1,2)= cplx(Ztemp(3) , Ztemp(4) )
Z(1,3)= cplx(Ztemp(5) , Ztemp(6) )

```

```

Z(2,2)= cmplx(Ztemp(7) , Ztemp(8) )
Z(2,3)= cmplx(Ztemp(9) , Ztemp(10) )
Z(3,3)= cmplx(Ztemp(11), Ztemp(12) )

Z(2,1)= cmplx(Ztemp(3) , Ztemp(4) )
Z(3,1)= cmplx(Ztemp(5) , Ztemp(6) )
Z(3,2)= cmplx(Ztemp(9) , Ztemp(10) )

end subroutine conversor_Zm
end module rutinas_de_asignacion

module rutinas_de_asignacion
!rutinas de conversion de datos para lista de barrido
CONTAINS
subroutine conversor_Z(Ztemp, Z, Z1, Z2, Z3, Z4)

implicit none

!Declaracion de variables I/O

real, dimension(12), intent(in)      ::Ztemp

complex, dimension(3,3), intent(out)::Z
complex, dimension(3,3), intent(out)::Z1, Z2, Z3, Z4

!Declaracion de variables internas

real:: FZ2 = 1.0 / 4.0
real:: FZ3 = 3.0 / 4.0

!Inicializacion de impedancias serie

Z = (0.0, 0.0)

Z1= (0.0, 0.0)  !siempre cero (Regulador sin pérdidas)
Z2= (0.0, 0.0)  !inicializa hasta saber si hay carga distribuida en la linea
Z3= (0.0, 0.0)  !inicializa hasta saber si hay carga distribuida en la linea
Z4= (0.0, 0.0)  !siempre cero (Regulador sin pérdidas)

!Se asigna la entrada de datos al contenedor Z que es una matriz compleja de 3X3

Z(1,1)= cmplx(Ztemp(1) , Ztemp(2) )
Z(1,2)= cmplx(Ztemp(3) , Ztemp(4) )
Z(1,3)= cmplx(Ztemp(5) , Ztemp(6) )
Z(2,2)= cmplx(Ztemp(7) , Ztemp(8) )
Z(2,3)= cmplx(Ztemp(9) , Ztemp(10) )
Z(3,3)= cmplx(Ztemp(11), Ztemp(12) )

Z(2,1)= cmplx(Ztemp(3) , Ztemp(4) )
Z(3,1)= cmplx(Ztemp(5) , Ztemp(6) )
Z(3,2)= cmplx(Ztemp(9) , Ztemp(10) )

!Asignacion de la impedancia serie de entrada a Z3, como si "TODA" la carga fuese concentrada

Z3=Z

!Finalmente se refinan las Impedancias Z2 y Z3 con los factores auxiliares, repartiendo la impoedancia
entre los elementos 2 y 3,
!Incluso si no hubiese carga distribuida (realmente los elementos estan en serie)

!Z2= FZ2 * Z3
!Z3= FZ3 * Z3

end subroutine conversor_Z
!*****
subroutine conversor_Y(Ytemp, Y)

implicit none

real, dimension(6), intent(in)::Ytemp
complex, dimension(3,3), intent(out)::Y

```

```

Y=cmplx(0.0, 0.0)

Y(1,1)=cmplx(0.0, 1.0*Ytemp(1))
Y(1,2)=cmplx(0.0, 1.0*Ytemp(2))
Y(1,3)=cmplx(0.0, 1.0*Ytemp(3))
Y(2,2)=cmplx(0.0, 1.0*Ytemp(4))
Y(2,3)=cmplx(0.0, 1.0*Ytemp(5))
Y(3,3)=cmplx(0.0, 1.0*Ytemp(6))

Y(2,1)=cmplx(0.0, 1.0*Ytemp(2))
Y(3,1)=cmplx(0.0, 1.0*Ytemp(3))
Y(3,2)=cmplx(0.0, 1.0*Ytemp(5))

end subroutine conversor_Y
!*****

SUBROUTINE conversor_carga( cx_Sc, md_Sc, Sc, cx_Sd, md_Sd, Sd, cx_Bc, SBc,      &
                           S1YS, S2YS, S3YS, S4YS,
                           &
                           S1YI, S2YI, S3YI, S4YI,
                           &
                           S1YZ, S2YZ, S3YZ, S4YZ,
                           &
                           S1DS, S2DS, S3DS, S4DS,
                           &
                           S1DI, S2DI, S3DI, S4DI,
                           &
                           S1DZ, S2DZ, S3DZ, S4DZ,
                           &
                           VSc, VSd, VSBc )

implicit none

!Declaracion de variables I/O

integer, intent (in)::      cx_Sc
integer, intent (in)::      md_Sc
real, dimension(6),intent(in)::  Sc

integer, intent (in)::      cx_Sd
integer, intent (in)::      md_Sd
real, dimension(6),intent(in)::  Sd

integer, intent (in)::      cx_Bc
real, dimension(3),intent(in)::  SBc

complex, dimension(3), intent(out):: S1YS, S2YS, S3YS, S4YS
complex, dimension(3), intent(out):: S1YI, S2YI, S3YI, S4YI
complex, dimension(3), intent(out):: S1YZ, S2YZ, S3YZ, S4YZ

complex, dimension(3), intent(out):: S1DS, S2DS, S3DS, S4DS

complex, dimension(3), intent(out):: S1DI, S2DI, S3DI, S4DI

complex, dimension(3), intent(out):: S1DZ, S2DZ, S3DZ, S4DZ

complex, dimension (3), intent(out)  :: VSc      !Vector de carga concentrada
complex, dimension (3), intent(out)  :: VSd      !Vector de carga distribuida
complex, dimension (3), intent(out)  :: VSBc     !Vector de carga de Banco Capacitivo

!Declaracion de variables internas

real:: FS2 = 2.0 / 3.0      !factores auxiliares para modelar lineas con carga distribuida
real:: FS4 = 1.0 / 3.0      !ver Kersting          (estos son para cargas)

!.....
VSc(1)=CMPLX(Sc(1),Sc(2))
VSc(2)=CMPLX(Sc(3),Sc(4))
VSc(3)=CMPLX(Sc(5),Sc(6))

VSd(1)=CMPLX(Sd(1),Sd(2))
VSd(2)=CMPLX(Sd(3),Sd(4))
VSd(3)=CMPLX(Sd(5),Sd(6))

VSBc(1)= CMPLX( 0.0 , -SBc(1) )

```

```

VSBc(2)= CMPLX( 0.0 , -SBc(2) )
VSBc(3)= CMPLX( 0.0 , -SBc(3) )

!Inicializacion de cargas

S1YS = (0.0, 0.0)
S2YS = (0.0, 0.0)
S3YS = (0.0, 0.0)
S4YS = (0.0, 0.0)

S1YI = (0.0, 0.0)
S2YI = (0.0, 0.0)
S3YI = (0.0, 0.0)
S4YI = (0.0, 0.0)

S1YZ = (0.0, 0.0)
S2YZ = (0.0, 0.0)
S3YZ = (0.0, 0.0)
S4YZ = (0.0, 0.0)

S1DS = (0.0, 0.0)
S2DS = (0.0, 0.0)
S3DS = (0.0, 0.0)
S4DS = (0.0, 0.0)

S1DI = (0.0, 0.0)
S2DI = (0.0, 0.0)
S3DI = (0.0, 0.0)
S4DI = (0.0, 0.0)

S1DZ = (0.0, 0.0)
S2DZ = (0.0, 0.0)
S3DZ = (0.0, 0.0)
S4DZ = (0.0, 0.0)
IF (cx_Sc == 1 .AND. md_Sc == 1) S4YS = VSc
IF (cx_Sc == 1 .AND. md_Sc == 2) S4YI = VSc
IF (cx_Sc == 1 .AND. md_Sc == 3) S4YZ = VSc

IF (cx_Sc == 2 .AND. md_Sc == 1) S4DS = VSc
IF (cx_Sc == 2 .AND. md_Sc == 2) S4DI = VSc
IF (cx_Sc == 2 .AND. md_Sc == 3) S4DZ = VSc

IF (cx_Sd == 1) THEN
SELECT CASE (md_Sd)
CASE (1)

S2YS = FS2 * VSd
S4YS = S4YS + FS4 * VSd

CASE (2)

S2YI = FS2 * VSd
S4YI = S4YI + FS4 * VSd

CASE (3)

S2YZ = FS2 * VSd
S4YZ = S4YZ + FS4 * VSd

ENDSELECT
ENDIF

IF (cx_Sd == 2) THEN
SELECT CASE (md_Sd)
CASE (1)

S2DS = FS2 * VSd
S4DS = S4DS + FS4 * VSd

CASE (2)

S2DI = FS2 * VSd
S4DI = S4DI + FS4 * VSd

```

```

CASE (3)

S2DZ =          FS2 * VSd
S4DZ = S4DZ +   FS4 * VSd

ENDSELECT

ENDIF
IF (cx_Bc == 1) S4YZ = S4YZ + VSBc
IF (cx_Bc == 2) S4DZ = S4DZ + VSBc
end subroutine conversor_carga
subroutine arReg (Tap, ar)

implicit none

!Declaracion de variables I/O

real, dimension(3), intent(in)  :: Tap
real, dimension(3), intent(out) :: ar
ar(1) = 1.0 + 0.00625 * Tap(1)
ar(2) = 1.0 + 0.00625 * Tap(2)
ar(3) = 1.0 + 0.00625 * Tap(3)

end subroutine arReg
subroutine conversor_voltaje(Vrootr, Vrootc)
implicit none

!Declaracion de variables I/O
real, dimension(6), intent(in):: Vrootr      !kV polar
complex, dimension(3), intent(out)::Vrootc    !kV      rectangular
real, parameter::pi=3.141593

!Declaracion de variables internas
real, dimension(3)::mag,tetha                !contenedor de magnitud y angulos en radianes
!Se inicializa la salida con valor nulo

Vrootc=cplx(0.0, 0.0)

!Se preprocesan los datos de entrada
mag(1)=Vrootr(1)
mag(2)=Vrootr(3)
mag(3)=Vrootr(5)

tetha(1)=Vrootr(2)*(pi/180.0)
tetha(2)=Vrootr(4)*(pi/180.0)
tetha(3)=Vrootr(6)*(pi/180.0)

!Se llena el vector complejo
Vrootc(1)=cplx(mag(1)*cos(tetha(1)), mag(1)*sin(tetha(1)))
Vrootc(2)=cplx(mag(2)*cos(tetha(2)), mag(2)*sin(tetha(2)))
Vrootc(3)=cplx(mag(3)*cos(tetha(3)), mag(3)*sin(tetha(3)))

end subroutine conversor_voltaje
subroutine calculoY2 ( Y, Y2pre, Y2k)

implicit none

!Declaracion de variables I/O

complex, dimension(3,3), intent(in):: Y

complex, dimension(3,3), intent(out):: Y2pre
complex, dimension(3,3), intent(out):: Y2k
Y2k = Y2k + (1.0 / 2.0) * Y
Y2pre= Y2pre + (1.0 / 2.0) * Y

end subroutine calculoY2
subroutine conversor_Zm(Ztemp, Z) !subrutina para convertir la entrada de la impedancia de mallas a
complejos

implicit none
!Declaracion de variables I/O

real, dimension(12), intent(in)      ::Ztemp

```

```

complex, dimension(3,3), intent(out)::Z

!Inicializacion de impedancias serie

Z = (0.0, 0.0)

!Se asigna la entrada de datos al contenedor Z que es una matriz compleja de 3X3

Z(1,1)= cmplx(Ztemp(1) , Ztemp(2) )
Z(1,2)= cmplx(Ztemp(3) , Ztemp(4) )
Z(1,3)= cmplx(Ztemp(5) , Ztemp(6) )
Z(2,2)= cmplx(Ztemp(7) , Ztemp(8) )
Z(2,3)= cmplx(Ztemp(9) , Ztemp(10) )
Z(3,3)= cmplx(Ztemp(11), Ztemp(12) )

Z(2,1)= cmplx(Ztemp(3) , Ztemp(4) )
Z(3,1)= cmplx(Ztemp(5) , Ztemp(6) )
Z(3,2)= cmplx(Ztemp(9) , Ztemp(10) )

end subroutine conversor_Zm
end module rutinas_de_asignacion

module UtileriasComplejas2

contains

subroutine RecPol (unit, n, BusName, V)

!Declaracion de variables I/O

integer, intent(in) :: unit !unidad para impresion
integer, intent(in) :: n !nodo numero...
character(10), intent(in) :: BusName !nombre del nodo

complex,dimension(3),intent(in) :: V !voltaje de entrada en forma rectangular

!Declaracion de variables internas

real, dimension(3):: mag
real, dimension(3):: ang

integer:: k

mag= abs(V)

do k=1, 3

IF ( Imag( V(k) ) /= 0.0 .OR. Real( V(k) ) /= 0.0 ) then

ang(k) = ATAN2D( Imag(V(k)), Real (V(k)) )

ELSE

ang(k)= 0.0

endif

enddo

write(unit,'(2X, I5,2X, A10 ,3X, F7.3, 1X,"@",1X, F7.2, 3X , F7.3, 1X , "@",1X, F7.2, 3X, F7.3,
1X,"@",1X, F7.2 )') &

n, BusName, mag(1), ang(1), mag(2), ang(2), mag(3), ang(3)

end subroutine RecPol
subroutine RecPol_pu (unit, n, BusName, V)

!Declaracion de variables I/O

integer, intent(in) :: unit !unidad para impresion
integer, intent(in) :: n !nodo numero...
character(10), intent(in) :: BusName !nombre del nodo

complex,dimension(3),intent(in) :: V !voltaje de entrada en forma rectangular

```

```

!Declaracion de variables internas

real, dimension(3):: mag
real, dimension(3):: ang

integer:: k
mag= abs(V)

do      k=1, 3

    IF ( Imag( V(k) ) /= 0.0 .OR. Real( V(k) ) /= 0.0 ) then

        ang(k) =          ATAN2D( Imag(V(k)), Real (V(k)) )

    ELSE

        ang(k)= 0.0

    endif

enddo

write(unit,'(2X, I5,2X, A10 ,3X, F7.5, 1X,"@",1X, F7.2, 3X , F7.5, 1X ,"@@",1X, F7.2, 3X, F7.5,
1X,"@",1X, F7.2 )') &

n, BusName, mag(1), ang(1), mag(2), ang(2), mag(3), ang(3)

end subroutine RecPol_pu
subroutine RecPol_1 (unit,V)

!Declaracion de variables I/O

integer, intent(in)  :: unit          !unidad para impresion
integer, intent(in)  :: n             !nodo numero...
character(10), intent(in) :: BusName  !nombre del nodo

complex,dimension(3),intent(in)  :: V !voltaje de entrada en forma rectangular

!Declaracion de variables internas

real, dimension(3):: mag
real, dimension(3):: ang

integer:: k
mag= abs(V)

do      k=1, 3

    IF ( Imag( V(k) ) /= 0.0 .OR. Real( V(k) ) /= 0.0 ) then

        ang(k) =          ATAN2D( Imag(V(k)), Real (V(k)) )

    ELSE

        ang(k)= 0.0

    endif

enddo

write(unit,'(2X, "      ",2X, "      " ,3X, F7.3, 1X,"@",1X, F7.2, 3X , F7.3, 1X ,"@@",1X, F7.2, 3X,
F7.3, 1X,"@",1X, F7.2 )') &

mag(1), ang(1), mag(2), ang(2), mag(3), ang(3)
end subroutine RecPol_1
end module UtileriasComplejas2

```

B2. CODIGO FUENTE DEL PROGRAMA DR.FOR

A continuación se muestra el código fuente del programa DR.FOR para la solución de flujos de potencia en redes de alta tensión, el cual fue reportado inicialmente en [11] y el cual fue modificado con el fin de realizar el acoplamiento de ambos sistemas

```
SUBROUTINE DR
USE MDATOS
USE MDATOS3

IMPLICIT REAL*8 (A-H),INTEGER*2 (I-N),REAL*8 (O-Z)

CHARACTER ARCHI1*12,ARCHI3*12
CHARACTER(len=30) :: ARCHI

! *****
! A APERTURA DE ARCHIVOS DE DATOS
! *****
I=1

IF(ACTUALIZAPOT .EQ.0) THEN
READ(5,20)ARCHI
END IF

20 FORMAT(A30)
OPEN(UNIT=10, FILE = ARCHI)

! REGRESA=0
CALL LECTURA(ARCHI)
ARCHI1 = 'DATSAL.DAT'

ARCHI3 = 'FLUJOS.SAL'
CALL FINALE(ARCHI1,ARCHI3)

CALL VOLTAJESPTOT

END SUBROUTINE DR

SUBROUTINE LECTURA(ARCHI)

USE MDATOS3
USE MDATOS
USE MDATOS2
IMPLICIT REAL*8 (A-H),INTEGER*2 (I-N),REAL*8 (O-Z)
INTEGER*2 NE (1000), NR (1000), NCT (300), NST (300), NU (250), &
NAU (250), NSH (150), TIP, BAN(250), ISLA(400), &
NORD(400), ISCOM (10), M (400), N (2400), ISP(2400), &
IPP(2400), NB1 (400), NB1C(2400), LP (1000), NLIS(400), &
ITRC (300)
REAL*8 R (1000), X (1000), B (1000), PC(400), QC (400), &
T1 (300), PG (250), VG (250), QMA(250), QMI (250), &
VLS (150), VMIN(400), VMAX(400), B1(2400), B2(2400), &
PGIS(10), PCIS(10), PSLA (10), PGC(250,5),AX(250,4)
REAL*8 LMI(1000), LMA(1000), TAFA (1000),TARE (1000)
CHARACTER NOMBRE (400)*10, NOMUNO*10, NOMDOS*10,ARCHI*12
CHARACTER*80 TEXUNO, TEXDOS

LOGICAL IEX

! *****
! INICIACION DE VARIABLES
! *****
!
TOLP = 0.010
TOLQ = 0.01
TINIA = 0.1
```

```

BASE = 100.
NITER = 60
NNOD = 1
NPRO = 1
NLIN = 1
NTUG = 1
NTR = 0
NISLA = 0
LTERM = 6
NSHU = 1

!
! *****
! A APERTURA DE ARCHIVOS DE DATOS
! *****
OPEN (10, FILE = ARCHI, FORM = 'FORMATTED', &
ACCESS = 'SEQUENTIAL' )

!
READ (10,10) TEXUNO
READ (10,10) TEXDOS
10 FORMAT (A80)
!
! *****
! LECTURA DE DATOS NODALES
! *****
!
IF (ACTUALIZAPOT .EQ. 0) THEN
ALLOCATE (VBASE(NNOD))
END IF

VBASE = 0.0
! SE INICIALIZAN POTENCIA DE CARGA PARA POSTERIORMENTE ASIGNARLES VALORES ESTIMADOS DE RADIOFLUJOS
PC = 0.0
QC = 0.0

! ASIGNA VALORES DE POTENCIAS INICIALES AL NODO DONDE SE ENCUENTRA CONECTADA LA RED DE DISTRIBUCION

PC(NODOPOT)=P3D
QC(NODOPOT)=Q3D

20 READ (10,30,END=50) NOMBRE (NNOD), & !PC (NNOD),QC (NNOD)
VMIN (NNOD), VMAX (NNOD), VBASE(NNOD)

30 FORMAT (2X,A8,2X,F8.2,2X,F8.2,F10.2,F10.2,F10.5)
IF ( NOMBRE (NNOD) .EQ. '0' ) GOTO 40
NNOD = NNOD + 1
GOTO 20

40 NNOD = NNOD - 1
WRITE (*,'(A,I5,A)') ' *** SE LEYO INFORMACION DE ', &
NNOD,' NODOS *** '
GOTO 100

50 WRITE (*,'(A)') ' *** FIN ARCHIVO EN LECTURA NODAL '

!
! *****
! LECTURA DE DATOS DE GENERADORES
! *****
!
100 TIP = 1
110 READ (10,120,END=190) NOMUNO,NOMDOS,PG (NTUG),QMA (NTUG), &
QMI (NTUG),VG (NTUG), BAN (NTUG), (PGC (NTUG,I),I=1,2), &
AX (NTUG,1)
120 FORMAT (2 (2X,A8),2X,3 (F8.2,2X),F8.6,1X,I1,2 (1X,F5.1),1X,F6.4)
IF (NOMUNO .EQ. '0' ) GOTO 180
CALL NODOS (NOMBRE,NOMUNO,NOMDOS,NNOD,JX,JY,TIP,IEX)

!
! ***** ALTA EN NOMBRES DE NODOS *****
!
IF (.NOT. IEX ) GOTO 110

!
NU (NTUG) = JX
NAU (NTUG) = JY
QMA (NTUG) = QMA (NTUG)/BASE
QMI (NTUG) = QMI (NTUG)/BASE
NTUG = NTUG + 1
GOTO 110
!

```

```

180  NTUG = NTUG - 1
    WRITE (*,'(A,I5,A)') ' *** SE LEYO INFORMACION DE ', &
        NTUG,' NODOS GENERADORES *** '
    GOTO 300
190  WRITE (*,'(A,I5)') ' *** FIN ARCHIVO LECTURA GENERADORES ',NTUG
    GOTO 1000

!
! *****
!  LECTURA DE LINEAS Y TRANSFORMADORES
! *****
!
300  TIP = 1
310  READ (10,320,END=390) NOMUNO,NOMDOS,R(NLIN),X(NLIN),B(NLIN), &
        T,LMA(NLIN),LMI(NLIN),F,TAF(A(NLIN),TARE(NLIN)
320  FORMAT (2X,A8,2X,A8,4(2X,F7.5),2(1X,F5.1),1X,F10.1,2X,F4.2,1X, &
        F5.1)
!
    IF ( NOMUNO.EQ. '0' ) GOTO 380
    CALL NODOS (NOMBRE,NOMUNO,NOMDOS,NNOD,JX,JY,TIP,IEX)
!
    IF (.NOT. IEX ) GOTO 310
!
    NE (NLIN) = JX
    NR (NLIN) = JY
    IF ( T .LE. 1.0E-08 ) GOTO 370
    IF ( NTR.EQ. 0 ) GOTO 340
    DO 330 K = 1, NTR
        IF (NCT (K) .NE. JX ) GOTO 330
        IF (NST (K) .NE. JY ) GOTO 330
        IF (T1 (K) .EQ. T ) GOTO 330
330  CONTINUE
340  NTR = NTR + 1
    NCT (NTR) = JX
    NST (NTR) = JY
    T1 (NTR) = T
    ITRC (NTR) = ITR
!
370  NLIN = NLIN + 1
    GOTO 310
!
380  NLIN = NLIN - 1
    NL = NLIN - NTR
!
    WRITE (*,'(A,I5,A)') ' *** SE LEYO INFORMACION DE ', &
        NL,' LINEAS *** '
    WRITE (*,'(A,I5,A)') ' *** SE LEYO INFORMACION DE ', &
        NTR,' TRANSFORMADORES *** '
!
    GOTO 400
390  WRITE (*,'(A,I5)') ' *** FIN ARCHIVO LECTURA DE LINEAS ',NLIN
    GOTO 1000

!
! *****
!  LECTURAS DE DATOS DE COMPENSADORES
! *****
!
400  TIP = 2
420  READ (10,430,END=460) NOMUNO,VLS (NSHU)
430  FORMAT (2X,A8,2X,F8.5)
!
    IF (NOMUNO.EQ. '0' ) GOTO 450
    CALL NODOS (NOMBRE,NOMUNO,NOMDOS,NNOD,JX,JY,TIP,IEX)
!
    IF ( .NOT. IEX ) GOTO 420
    NSH (NSHU) = JX
    NSHU = NSHU + 1
    GOTO 420
!
450  NSHU = NSHU - 1
    WRITE (*,'(A,I5,A)') ' *** SE LEYO INFORMACION DE ', &
        NSHU,' COMPENSADORES *** '
    GOTO 500
460  WRITE (*,'(A,I5)') ' *** FIN ARCHIVO LECTURA COMPENSADOR ',NSHU
    GOTO 1000

!
! *****

```

```

!      ESCRITURA DE ARCHIVO DE DATOS
! *****
!
500      OPEN (11, FILE = 'DATSAL.DAT', FORM = 'UNFORMATTED', &
ACCESS = 'SEQUENTIAL' )

      WRITE (11) TEXUNO,TEXDOS
      WRITE (11) LTERM,NISLA,NITER,TOLP,TOLQ,TINIA,NNOD
      WRITE (11) NLIN,NTR,NSHU,NTUG,NPRO,BASE
      DO 510 I = 1, NNOD
        WRITE (11) NOMBRE(I),PC(I),QC(I),VMIN(I),VMAX(I)
510      CONTINUE
      DO 520 I = 1, NTUG
        WRITE (11) NU(I),NAU(I),VG(I),PG(I),QMA(I),QMI(I),BAN(I)
520      CONTINUE
      IF ( NTR .EQ. 0 ) GOTO 540
      DO 530 I = 1, NTR
        WRITE (11) NCT (I),NST(I),ITRC(I),T1(I)
530      CONTINUE
      IF ( NSHU .EQ. 0 ) GOTO 560
      DO 550 I = 1, NSHU
        WRITE (11) NSH(I),VLS(I)
550      CONTINUE
560      DO 570 I = 1, NLIN
        WRITE (11) NE(I),NR(I),R(I),X(I),B(I),LMA(I),LMI(I),Tafa(I), &
TARE(I)
570      CONTINUE
      CLOSE (10)
      CLOSE (11)
! *****
!      CONSTRUCCION DE MODELOS PARA EL DESACOPADO RAPIDO
! *****
      NL2 = 2*NLIN + NNOD
      IEXT = 600
      CALL GRAPH (NNOD,NLIN,NL2,NE,NR,M,N,IPP,ISP)
      CALL PARTS (NNOD,NL2,M,N,ISP,ISLA,NLIS,NISLA)
      CALL ACTIV (NNOD,NISLA,NTUG,ISLA,NU,PG,PC,ISCOM,PGIS,PSLA)
      CALL PREPA (NNOD,NLIN,NL2,IEXT,NISLA,M,N,ISP,IPP,ISLA,ISCOM,LP,&
NORD,N4,IDISP)
      CALL ORDER (NISLA,NNOD,NLIN,NL2,IEXT,N4,M,N,ISP,IPP,NORD,IDISP)
      IFIL = NL2
      CALL FORMA (NNOD,NLIN,NL2,NISLA,IEXT,IFIL,NTUG,NSHU,NTR,LP,M,N,&
NB1,NB1C,B1,B2,IPP,R,X,B,ISP,NORD,NCT,NST,T1,NSH,VLS,NU,ISCOM,&
NE,NR,NLIS)
      CALL FACTA (NNOD,IFIL,NISLA,NB1,NB1C,NORD,B1,B2)
!
! *****
!      A APERTURA DE ARCHIVO DE DATOS PROCESADOS
! *****
!
      OPEN (12, FILE = 'FLUDAT.DAT' , FORM = 'UNFORMATTED',&
ACCESS = 'SEQUENTIAL' )
!
! *****
!      SE ESCRIBE ARCHIVO DE DATOS PROCESADOS
! *****
!
      WRITE (12) IFIL,NISLA,NITER,TOLP,TOLQ,TINIA,NNOD
      DO 140 I = 1, NISLA
        WRITE (12) ISCOM (I)
140      CONTINUE
      DO 150 I = 1, NNOD
        WRITE (12) ISLA (I), NORD(I), NB1(I)
150      CONTINUE
      DO 160 I = 1, IFIL
        WRITE (12) NB1C (I), B1(I), B2(I)
160      CONTINUE
1000     CLOSE (12)
      RETURN

      END SUBROUTINE LECTURA

! *****
!      SUBROUTINA DE VERIFICACION DE LISTA DE NODOS
! *****

```

```

!
SUBROUTINE NODOS (NOMBRE,NOMUNO,NOMDOS,NNOD,JX,JY,TIP, IEX)
IMPLICIT INTEGER*2 (I-N)
CHARACTER*10 NOMBRE(400),NOMUNO,NOMDOS,NOMTRE
INTEGER*2 TIP
LOGICAL IEX
IEX = .TRUE.
J = 1
3040 NOMTRE = NOMBRE (J)
IF (NOMUNO .EQ. NOMTRE) GOTO 3050
J = J + 1
IF (J .LE. NNOD ) GOTO 3040
WRITE (*,'(3(A))') ' *** EL NODO ',NOMUNO,' NO EXISTE '
IEX = .FALSE.
RETURN
3050 JX = J
IF ( TIP .EQ. 2 ) THEN
RETURN
END IF
J = 1
3060 NOMTRE = NOMBRE (J)
IF ( NOMDOS .EQ. NOMTRE) GOTO 3080
J = J + 1
IF ( J .LE. NNOD ) GOTO 3060
WRITE (*,'(3(A))') ' *** EL NODO ',NOMDOS,' NO EXISTE '
IEX = .FALSE.
3070 RETURN
3080 JY = J
RETURN
END

! *****
! SUBROUTINA DE FORMACION DEL GRAFO
! *****
!
SUBROUTINE GRAPH (NNOD,NLIN,NL2,NE,NR,M,N,IPP,ISP)
IMPLICIT INTEGER*2 (I-N)
INTEGER*2 NE(*), NR(*), M(*), N(*), IPP(*), ISP(*)
J2 = 1
J1 = 1
DO 1030 J = 1, NNOD
DO 1020 K = 1, NLIN
IF ( NE (K) .NE. J ) GOTO 1010
J1 = J1 + 1
N (J1) = NR (K)
IPP (J1) = K
1010 IF ( NR (K) .NE. J ) GOTO 1020
J1 = J1 + 1
N(J1) = NE (K)
IPP (J1) = K
1020 CONTINUE
N (J2) = J1 - J2
M (J) = J2
J1 = J1 + 1
J2 = J1
1030 CONTINUE
!
! *****
! SE INICIALIZA EL VECTOR DE ENCADENAMIENTO
! *****
!
L1 = 1
DO 1050 I = 1, NNOD
J3 = N(M(I))
DO 1040 J = 1, J3
INEXT = L1 + 1
ISP (L1) = INEXT
L1 = INEXT
1040 CONTINUE
ISP (L1) = 0
L1 = L1 + 1
1050 CONTINUE
J2 = J2 - 1
RETURN
END

```

```

!
! *****
! CHECA CONECTIVIDAD DEL GRAFO DA NUMERO DE COMPONENTE A CADA NODO
! *****
!
SUBROUTINE PARTS (NNOD,NL2,M,N,ISP,ISLA,NLIS,NISLA)
IMPLICIT INTEGER*2 (I-N)
INTEGER*2 NLIS(*),ISLA(*),M(*),N(*),ISP(*)
NISLA = 0
2000 DO 2010 I = 1, NNOD
      ISLA (I) = 0
      NLIS (I) = 0
2010 CONTINUE
2020 DO 2070 I1 = 1, NNOD
      K11 = ISLA (I1)
      IF ( K11 .NE. 0) GOTO 2070
      I = 1
      J = 1
      NISLA = NISLA + 1
      NLIS (I) = I1
      ISLA (I1) = NISLA
2030 J1 = NLIS (I)
      J1 = M(J1)
2040 J1 = ISP (J1)
      IF ( J1 .NE. 0 ) GOTO 2050
      IF ( J1 .EQ. 0 ) GOTO 2060
2050 K = N(J1)
      K11 = ISLA (K)
      IF ( K11 .NE. 0 ) GOTO 2040
      J = J + 1
      NLIS (J) = K
      ISLA (K) = NISLA
      GOTO 2040
2060 I = I + 1
      IF ( I .GT. J) GOTO 2020
      GOTO 2030
2070 CONTINUE
      RETURN
      END
!
! *****
! VERIFICA QUE EXISTA CARGA Y GENERACION EN CADA ISLA
! ASIGNA NODO COMENSADOR A CADA ISLA
! *****
!
SUBROUTINE ACTIV (NNOD,NISLA,NTUG,ISLA,NU,PG,PC,ISCOM,PGIS, &
PCIS,PSLA)
IMPLICIT REAL*8 (A-H),INTEGER*2 (I-N),REAL*8 (O-Z)
INTEGER*2 ISLA (*), ISCOM (*), NU(*)
REAL*8 PGIS(*), PCIS(*), PSLA (*), PG(*), PC(*)
!
3000 DO 3010 I = 1, NISLA
      PGIS (I) = 0.
      PCIS (I) = 0.
3010 CONTINUE
      DO 3020 I= 1, NTUG
            IP = NU (I)
            IP = ISLA (IP)
            PGIS (IP) = PGIS (IP) + PG (I)
3020 CONTINUE
      DO 3030 I = 1, NNOD
            IP = ISLA (I)
            PCIS (IP) = PCIS (IP) + PC (I)
3030 CONTINUE
!
! *****
! SE ABANDERAN ISLAS SIN CARGA O GENERACION
! *****
!
      DO 3050 I= 1, NISLA
            IF ( PGIS (I) .GT. 1.0E-08) GOTO 3050
            IF ( PCIS (I) .GT. 1.0E-08) GOTO 3050
            DO 3040 J = 1, NNOD
                  IF ( ISLA (J) .EQ. I ) ISLA (J) = - ISLA (J)
3040 CONTINUE
3050 CONTINUE

```

```

!
! *****
! SE ELIGE COMPENSADOR POR ISLA DE ACUERDO A SU PGEN (EL MAYOR)
! *****
!
DO 3070 J = 1, NISLA
  PSLA (J) = 0.
  DO 3060 I = 1, NTUG
    IP = NU (I)
    IP1 = ISLA (IP)
    IF ( IP1 .LE. 0 ) GOTO 3070
    IF ( IP1 .NE. J ) GOTO 3060
    IF ( PG(I) .LT. PSLA (J) ) GOTO 3060
    ISCOM (J) = IP
    PSLA (J) = PG (I)
3060    CONTINUE
3070  CONTINUE
      RETURN
      END
!
! *****
! SUBROUTINA QUE TRANSFORMA EL GRAFO EN SIMPLE Y DESCONECTA
! COMPENSADORES
! *****
!
SUBROUTINE PREPA (NNOD,NLIN,NL2,IEXT,NISLA,M,N,ISP,IPP,&
ISLA,ISCOM,LP,NORD,N4,IDISP)
  IMPLICIT REAL*8 (A-H),INTEGER*2 (I-N),REAL*8 (O-Z)
  INTEGER*2 LP(*), M(*), N(*), ISP(*), IPP(*), ISLA (*)
  INTEGER*2 NORD (*), ISCOM (*)
!
  N3=NL2
  N4 = NNOD
  LX = 1
  DO 4000 I=1,NNOD
    NORD (I) = 0
4000  CONTINUE
  DO 4010 I= 1, NLIN
    LP (I) = 0
4010  CONTINUE
  DO 4030 I= 1, NNOD
    J3 = N (M(I))
    DO 4020 J = 1, J3
      NC5 = LX + 1
      ISP (LX) = NC5
      LX = NC5
4020  CONTINUE
      ISP (LX) = 0
      LX = LX + 1
4030  CONTINUE
      IDISP = N3 + 1
      DO 4040 I= IDISP, IEXT
        ISP (I) = I+ 1
4040  CONTINUE
        ISP (IEXT) = 0
!
! *****
! SE ELIMINAN LINEAS PARALELAS EN EL GRAFO
! *****
!
DO 4140 I= 1, NNOD
4050  II= M (I)
      IF ( N (II) - 1 .NE. 0 ) GOTO 4060
      IF ( N (II) - 1 .EQ. 0 ) GOTO 4140
4060  NC5 = II
      NC6 = N(NC5)
      DO 4130 J = 1,NC6
        NC5 = ISP (NC5)
        NC7 = NC5
        IF (NC5 .NE. 0 ) GOTO 4070
        IF (NC5 .EQ. 0 ) GOTO 4130
4070  NA = N (NC5)
4080  NC8 = ISP (NC5)
        IF ( NC8 .NE. 0 ) GOTO 4100
        IF ( NC8 .EQ. 0 ) GOTO 4090
4090  NC5 = NC7

```

```

      GOTO 4130
4100    NB = N(NC8)
      IF ( NA-NB .NE. 0 ) GOTO 4110
      IF ( NA-NB .EQ. 0 ) GOTO 4120
4110    NC5 = NC8
      GOTO 4080
4120    N(II) = N(II) - 1
      ISP (NC5) = ISP (NC8)
      ISP (NC8) = IDISP
      IP1 = IPP (NC8)
      LP (IP1) = IPP (NC5)
      IDISP = NC6
      GOTO 4050
4130    CONTINUE
4140    CONTINUE
!
! *****
! SE ELIMINAN NODOS COMPENSADORES DEL GRAFO
! *****
!
      DO 4270 I3 = 1, NISLA
        NSLK = ISCOM (I3)
        IF ( ISLA (NSLK) .LT. 0 ) GOTO 4260
        NORD (NSLK) = NNOD
        NC5 = ISP (M (NSLK) )
4200    IF ( NC5 .NE. 0 ) GOTO 4210
        IF ( NC5 .EQ. 0 ) GOTO 4250
4210    I = M (N(NC5))
        N(I) = N(I) - 1
4220    J = ISP (I)
        IF ( N(J) - NSLK .NE. 0 ) GOTO 4240
        IF ( N(J) - NSLK .EQ. 0 ) GOTO 4230
4230    ISP (I) = ISP(J)
        NC5 = ISP (NC5)
        GOTO 4200
4240    I= J
        GOTO 4220
4250    N4 = N4 - 1
4260    NORD (NSLK) = NNOD
4270    CONTINUE
!
! *****
! SE ASIGNAN LOS NUMEROS ALTOS A NODOS INACTIVOS EN EL ORDENAMIENTO
! *****
!
      DO 4280 I= 1, NNOD
        IF ( ISLA (I) .GE. 0 ) GOTO 4280
        NORD (I) = NNOD
        N4 = N4 - 1
4280    CONTINUE
      RETURN
      END
!
! *****
! SUBROUTINA DE ORDENAMIENTO DE GRADO MINIMO
! *****
!
      SUBROUTINE ORDER (NISLA,NNOD,NLIN,NL2,IEXT,N4,M,N,ISP,IPP,NORD, &
      IDISP)
      IMPLICIT REAL*8(A-H),INTEGER*2(I-N),REAL*8(O-Z)
      INTEGER*2 NORD (*), M(*), N(*), ISP(*), IPP(*)
!
5000    N3 = 0
      DO 5180 I= 1, N4
        NC5 = NLIN
        NELI= NNOD + 1
        DO 5030 J = 1, NNOD
          IF (NORD (J) .NE. 0 ) GOTO 5030
          IF (NORD (J) .EQ. 0 ) GOTO 5010
5010    LX = N (M(J))
          IF ( LX-NC5 .LT. 0 ) GOTO 5020
          IF ( LX-NC5 .GE. 0 ) GOTO 5030
5020    NELI= J
          NC5 = LX
5030    CONTINUE
        N12 = NNOD + 1

```

```

      IF ( NELI-N12 .NE. 0 ) GOTO 5040
      IF ( NELI-N12 .EQ. 0 ) GOTO 5180
5040    LX = M(NELI)
      J = M(NELI)
      NORD (NELI) = I
      J1 = ISP(J)
      IF ( J1 .NE. 0 ) GOTO 5050
      IF ( J1 .EQ. 0 ) GOTO 5180
5050    DO 5170 J1 = 1, NC5
      J = ISP (J)
      J7 = N (J)
      J2 = M(J7)
      J3 = N(J2)
      J5 = LX
      DO 5160 J4 = 1, NC5
      J5 = ISP (J5)
      J6 = N(J5)
      I1 = J2
      IF ( J6-J7 .NE. 0 ) GOTO 5090
      IF ( J6-J7 .EQ. 0 ) GOTO 5060
5060    J33 = J3
      DO 5080 J8 = 1, J33
      I3 = I1
      I1 = ISP (I1)
      I2 = N(I1)
      IF ( I2-NELI .NE. 0 ) GOTO 5080
      IF ( I2-NELI .EQ. 0 ) GOTO 5070
5070    ISP (I3) = ISP (I1)
      J3 = J3 - 1
      ISP (I1) = IDISP
      IDISP = I1
      GOTO 5160
5080    CONTINUE
      GOTO 5160
5090    DO 5110 J8 = 1, J3
      I99 = I1
      I1 = ISP (I1)
      IF ( I1 .NE. 0 ) GOTO 5100
      IF ( I1 .EQ. 0 ) GOTO 5120
5100    I2 = N(I1)
      IF ( J6-I2 .NE. 0 ) GOTO 5110
      IF ( J6-I2 .EQ. 0 ) GOTO 5160
5110    CONTINUE
      GOTO 5130
5120    I1 = I99
5130    IF ( IDISP-N3 .GT. 0 ) GOTO 5140
      IF ( IDISP-N3 .LE. 0 ) GOTO 5150
5140    N3 = IDISP
5150    K89 = IDISP
      ISP (I1) = K89
      N(K89) = J6
      IPP(K89) = 0
      IDISP = ISP (K89)
      ISP (K89) = 0
      J3 = J3 + 1
5160    CONTINUE
      N (J2) = J3
5170    CONTINUE
5180    CONTINUE
      RETURN
      END
! *****
!   FORMA Y CONSTRUYE LAS MATRICES  B' Y B"
! *****
!
      SUBROUTINE FORMA (NNOD,NLIN,NL2,NISLA,IEXT,IFIL,NTUG,NSHU,NTR,LP, &
      M,N,NB1,NB1C,B1,B2,IPP,R,X,B,ISP,NORD,NCT,NST,T1,NSH,VLS,NU, &
      ISCOM,NE,NR,NLIS )
!
      IMPLICIT REAL*8 (A-H),INTEGER*2 (I-N),REAL*8 (O-Z)
      INTEGER*2 &
      LP(*),M(*),N(*),NB1(*),NB1C(*),IPP(*),ISP(*),NORD(*), &
      NCT(*),NST(*),NSH(*),NU(*),ISCOM(*),NE(*),NR(*), &
      NLIS(*)
      REAL*8 &
      B1(*), B2(*), R(*), X(*), B(*), T1(*), VLS(*)

```

```

6000 K8 = 0
      K9 = 0
      DO 6010 I=1, NNOD
        NLIS (I) = 0
6010 CONTINUE
      DO 6020 I = 1, NTUG
        J = NU (I)
        NLIS (J) = 1
6020 CONTINUE
      N12 = NNOD - NISLA
      DO 6110 I= 1, N12
        DO 6030 K = 1, NNOD
          IF ( NORD(K) .EQ. I ) GOTO 6050
6030 CONTINUE
          GOTO 6110
6050 J = M(K)
          J1 = N(J)
          K9 = K9 + 1
          K8 = K8 + 1
          NB1 (I) = K9
          NB1C (K8) = J1
          B1 (K9) = 0.
          B2 (K9) = 0.
          IP = ISP (J)
6060 IF ( IP .GT. 0 ) GOTO 6070
          K9 = K8
          GOTO 6110
6070 K8 = K8 + 1
          NB1C (K8) = N (IP)
          IF ( IPP (IP) .NE. 0 ) GOTO 6080
          B1 (K8) = 0.
          B2 (K8) = 0.
          GOTO 6100
6080 IP2 = IPP (IP)
          B1 (K8) = - 1.0 / X(IP2)
          IF ( NLIS (K) .EQ. 0 .AND. NLIS(N(IP)) .EQ. 0 ) GOTO 6090
          B2 (K8) = 0.
          GOTO 6100
6090 B2 (K8) = -X(IP2) / ( R(IP2)*R(IP2) + X(IP2)*X(IP2) )
6100 IP = ISP (IP)
          GOTO 6060
6110 CONTINUE
!
! *****
! SE FORMAN LOS ELEMENTOS DIAGONALES DE B1
! *****
!
      DO 6160 I = 1, N12
        DO 6120 K1 = 1, NNOD
          IF ( NORD (K1) .EQ. I ) GOTO 6130
6120 CONTINUE
          GOTO 6160
6130 N1 = K1
          N2 = NB1 (I)
          DO 6150 K = 1, NLIN
            IF ( NE (K) .EQ. N1 ) GOTO 6140
            IF ( NR (K) .EQ. N1 ) GOTO 6140
            GOTO 6150
6140 Y = 1.0 / X(K)
            Y11 = X(K) / ( X(K)*X(K) + R(K)*R(K) )
            B1 (N2) = B1(N2) + Y
            B2 (N2) = B2(N2) + Y11
            B2 (N2) = B2(N2) - B(K)
6150 CONTINUE
6160 CONTINUE
      DO 6200 I=1, NLIN
        I1 = LP (I)
        IF ( I1.EQ.0 ) GOTO 6200
        N1 = NE (I)
        N2 = NR (I)
        N4 = NB1 (NORD (N1))
        N5 = NB1C (N4)
        Y = -1.0 / X(I)
        Y11 = - X(I) / ( R(I)*R(I) + X(I)*X(I) )
        IF ( N5 .EQ. 0 ) GOTO 6180
        DO 6170 J = 1, N5

```

```

        K1 = N4 + J
        NB = NB1C (K1)
        IF ( NB .NE. N2 ) GOTO 6170
        B1 (K1) = B1 (K1) + Y
        IF ( NLIS (NB) .NE. 0 ) GOTO 6180
        B2 (K1) = B2(K1) + Y11
        GOTO 6180
6170    CONTINUE
6180    NO = NORD (N2)
        N4 = NB1 (NO)
        IF ( N4 .LE. 0 ) GOTO 6200
        N5 = NB1C (N4)
        IF ( N5 .EQ. 0 ) GOTO 6200
        DO 6190 J = 1, N5
            K1 = N4 + J
            NB = NB1C (K1)
            IF ( NB .NE. N1 ) GOTO 6190
            B1 (K1) = B1 (K1) + Y
            IF ( NLIS (NB) .NE. 0 ) GOTO 6190
            B2 (K1) = B2 (K1) + Y11
            GOTO 6200
6190    CONTINUE
6200    CONTINUE
!
! *****
!     SE INCLUYE EFECTO DE TRANSFORMADORES
! *****
!
        IF ( NTR .LE. 0 ) GOTO 6400
        DO 6330 I= 1, NTR
            N1 = NCT (I)
            N2 = NST (I)
            IF ( NLIS (N1).NE.0 .OR. NLIS (N2).NE.0 ) GOTO 6330
            N3 = NORD (N1)
            N4 = NORD (N2)
            IF ( N3 .LT. N4 ) GOTO 6300
            N5 = N2
            NO = N1
            GOTO 6310
6300    N5 = N1
            NO = N2
6310    N6 = NORD (N5)
            N7 = NB1 (N6)
            J33 = NB1C (N7)
            DO 6320 J = 1, J33
                N8 = N7 + J
                N9 = NB1C (N8)
                IF ( N9 .NE. NO ) GOTO 6320
                N10 = NB1 (N3)
                B2 (N10) = B2(N10) + B2 (N8) * &
                    ( ( T1(I)*T1(I)-1.0 ) / ( T1(I)*T1(I)) )
                B2 (N8) = B2(N8)/ T1 (I)
6320    CONTINUE
6330    CONTINUE
!
! *****
!     SE INCLUYEN EFECTOS SHUNTS
! *****
!C
6400    IF ( NSHU .LT. 1 ) GOTO 6420
        DO 6410 I= 1, NSHU
            N1 = NSH (I)
            N2 = NORD (N1)
            N3 = NB1 (N2)
            B2 (N3) = B2(N3) - VLS (I)
6410    CONTINUE
6420    IFIL = K8
        RETURN
        END
!
! *****
!     GENERA LOS FACTORES TRIANGULARES DE UNA MATRIZ SIMETRICA
!     USANDO LOS ARREGLOS QUE RESULTAN
! *****
!
        SUBROUTINE FACTA (NNOD, IFIL, NISLA, NB1, NB1C, NORD, B1, B2)

```

```

      IMPLICIT REAL*8 (A-H),INTEGER*2(I-N),REAL*8(O-Z)
      INTEGER*2 NB1 (*), NB1C(*), NORD(*)
      REAL*8 B1 (*), B2 (*)
!
7000 N12 = NNOD - NISLA
      DO 7080 I= 1, N12
        K = NB1 (I)
        NEL = NB1C (K)
        IF ( NEL .LT. 1 ) GOTO 7080
        DO 7060 J = 1,NEL
          NA = K + J
          E1 = B1 (NA) / B1 (K)
          EG = B1 (NA) * E1
          ET = DABS ( B2 (NA) )
          IF ( ET .GT. 1.0E-08 ) GOTO 7020
          E2 = 0
          EG2 = 0.
          GOTO 7030
7020 E2 = B2 (NA) / B2 (K)
          EG2 = B2 (NA) * E2
7030 NB = NB1C (NA)
          NC = NORD (NB)
          ND = NB1 (NC)
          B1 (ND) = B1 (ND) - EG
          B2 (ND) = B2 (ND) - EG2
!
! *****
! SE MODIFICAN ELEMENTOS BAJO EL PIVOTE SI HAY CORRESPONDENCIA
! *****
!
      IF ( NEL .LE. 1 ) GOTO 7060
      DO 7050 K1 = 1, NEL
        KA = K + K1
        KA1 = NB1C (KA)
        NB11 = NB1C (ND)
        IF ( NB11 .LT. 1 ) GOTO 7050
        DO 7040 K2 = 1, NB11
          KB = ND + K2
          KB1 = NB1C (KB)
          IF ( KA1 .NE. KB1 ) GOTO 7040
          B1 (KB) = B1 (KB) - E1*B1 (KA)
          B2 (KB) = B2 (KB) - E2*B2 (KA)
7040 CONTINUE
7050 CONTINUE
7060 CONTINUE
!
! *****
! SE NORMALIZAN ELEMENTOS EN FILA PIVOTE
! *****
!
      DO 7070 K1 = 1, NEL
        NA = K + K1
        B1 (NA) = B1 (NA) / B1 (K)
        B2 (NA) = B2 (NA) / B2 (K)
7070 CONTINUE
7080 CONTINUE
! WRITE (*,'(A)') ' *** SE FORMO TABLA DE FACTORES *** '
      RETURN
      END
      SUBROUTINE FINALE (ARCHI1,ARCHI3)
!
      USE MDATOS3
      USE MDATOS
      IMPLICIT REAL*8 (A-H),INTEGER*2(I-N),REAL*8(O-Z)
      INTEGER*2 &
        N (500), NB1 (300), NB1C (300), NORD (300), NORD1 (300), &
        NE (100), NR (100), ISLA (300), M (300), NST (150), &
        NU (150), NCT (150), NAU (150), NSH (150), ISCOM (10), &
        NM (50)
      REAL*8 &
        YR (500), YI (500), B1 (500), B2 (500), R (100), &
        X (100), B (100), E (300), F (300), E3 (300), &
        F3 (300), P2 (300), Q2 (300), PC (300), QC (300), &
        T1 (100), QMA (100), QMI (100), PG (100), VG (100), &
        VLS (50), BI (500), PL (40), QL (40), T2 (40), &
        QG (100), MAG1(500), ANG1(500)

```

```

CHARACTER*10 &
FO*2,      NOMBRE (100),  TEXUNO*80,  TEXTDOS*80
CHARACTER*12  ARCHI1, ARCHI3

!
! *****
!   APERTURA DE ARCHIVOS DE DATOS
! *****
!
OPEN (10, FILE =  ARCHI1, FORM = 'UNFORMATTED', &
ACCESS = 'SEQUENTIAL',  STATUS = 'OLD' )
!
READ (10) TEXUNO,TEXTDOS
READ (10) IFIL,NISLA,NITER,TOLP,TOLQ,TINIA,NNOD
READ (10) NLIN,NTR,NSHU,NTUG,NPRO,BASE
!
DO 100 I = 1, NNOD
READ (10) NOMBRE (I),PC(I),QC(I)
100 CONTINUE
DO 110 I = 1,NTUG
READ (10) NU(I),NAU(I),VG(I),PG(I),QMA(I),QMI(I)
110 CONTINUE
IF ( NTR .EQ.0 ) GOTO 130
DO 120 I= 1,NTR
READ (10) NCT(I),NST(I),ITR,T1(I)
120 CONTINUE
130 IF ( NSHU .EQ. 0 ) GOTO 150
DO 140 I = 1,NSHU
READ (10) NSH(I),VLS(I)
140 CONTINUE
150 DO 160 I = 1,NLIN
READ (10) NE(I),NR(I),R(I),X(I),B(I)
160 CONTINUE
! *****
OPEN (11, FILE='FLUDAT.DAT', FORM = 'UNFORMATTED', &
ACCESS = 'SEQUENTIAL',  STATUS = 'OLD' )
!
READ (11) IFIL,NISLA,NITER,TOLP,TOLQ,TINIA,NNOD
!
DO 170 I = 1, NISLA
READ (11) ISCOM (I )
170 CONTINUE
DO 180 I = 1,NNOD
READ (11) ISLA (I),NORD (I),NB1 (I)
180 CONTINUE
DO 190 I = 1 , IFIL
READ (11) NB1C (I),B1(I),B2(I)
190 CONTINUE
!
NITER = 60
NL2 = 2 * NLIN + NNOD
!
! *****
!   ENTRA A SUBROUTINA DE FORMACION DE LA MATRIZ Y DE BUS
! *****
!
CALL YBUSC (NNOD,NLIN,NTR,NSHU,NL2,M,N,YR,YI,BI,NE,NR,R,X,B,NCT, &
NST,T1,NSH,VLS)
!
! *****
!   ENTRA A SUBROUTINA DE SOLUCION DE FLUJOS
! *****
!
CALL LFSOL (NNOD,NTUG,NISLA,NL2,IFIL,ISCOM,BASE,NORD,NORD1,PG,VG, &
NU,E3,F3,P2,Q2,PC,QC,YR,YI,NB1,NB1C,B1,B2,E,F,NITER, &
TOLP,TOLQ,M,N,NOMBRE)

IF (ACTUALIZAPOT .EQ. 0) THEN
ALLOCATE (MAGV (NNOD),ANGV (NNOD),ANGRADOS (NNOD))
END IF
MAGV = 0.0
ANGV = 0.0
ANGRADOS = 0.0
!
! *****

```

```

!      IMPRIME SOLUCION DE FLUJOS
!      *****
!
DO 200 I = 1, NNOD
    FF = F(I )
    E3 ( I ) = E(I ) * COS (FF)
    F3 ( I ) = E(I ) * SIN (FF)
200 CONTINUE
    LDIS = 8
    OPEN (LDIS, FILE = ARCHI3, ACCESS = 'SEQUENTIAL')

210 WRITE (LDIS,'(A)') TEXUNO
220 WRITE (LDIS,'(A)') TEXDOS
    DO 240 MAIN = 1,NNOD
        IF ( ISLA (MAIN ) .GE. 1 ) GOTO 230
        WRITE (LDIS,'(A,I5,2X,A,A)') ' *** NODO ',MAIN , &
            NOMBRE(MAIN),' AISLADO *** '
        GOTO 240
230 CALL FLUCAL (MAIN,NNOD,NL2,NTUG,NTR,NSHU,M,N,YR,YI,BI, &
    NCT,NST,T1,NSH,VLS,BASE,E,F,E3,F3,NOMBRE,PG,QG,PC, &
    QC,NU,QMA,QMI,NM,PL,QL,T2,LDIS)
240 CONTINUE

NONODOS = NNOD
!
1000 CLOSE (10)
    CLOSE (11)

!
    WRITE(*,299)
299 FORMAT(2/,3X,'CONSULTE EL ARCHIVO FLUJOS.SAL...',/)
    END
!
!      *****
!      SUBROUTINA DE CALCULO DE FLUJOS
!      *****
!
SUBROUTINE FLUCAL (MAIN,NNOD,NL2,NTUG,NTR,NSHU,M,N,YR,YI,BI, &
    NCT,NST,T1,NSH,VLS,BASE,E,F,E3,F3,NOMBRE, &
    PG,QG,PC,QC,NU,QMA,QMI,NM,PL,QL,T2,LDIS)

    USE MDATOS3

    IMPLICIT REAL*8 (A-H),INTEGER*2 (I-N),REAL*8 (O-Z)
    INTEGER*2 M(*),N(*),NCT(*),NST(*),NSH(*),NM(*),NU(*)
    REAL*8 E(*),F(*),E3(*),F3(*),PL(*),QL(*),T2(*),VLS(*), &
        T1(*),QMA(*),QMI(*),PG(*),QG(*),PC(*),QC(*),SUM,QUM, &
        EH,AB,CD,YR(*),YI(*),SI,SR,SG,BI(*),SN,SQ,PC1,QC1
    CHARACTER*10 NOMBRE(*),FO*2

1000 NS1 = 0
    IG = 0
    SUM = 0
    QUM = 0
    N1 = M (MAIN )
    N2 = N (N1 )
    KZ = 0
    EH = E3 (MAIN ) * E3 (MAIN ) + F3 (MAIN ) * F3 (MAIN )
    DO 1010 K = 1, N2
        J = N1 + K
        KZ = KZ + 1
        MQ = N (J)
        AB = E3 (MAIN ) * E3 (MQ ) + F3 (MAIN ) * F3 (MQ )
        CD = F3 (MAIN ) * E3 (MQ ) - E3 (MAIN ) * F3 (MQ )
        PL (KZ ) = - EH*YR(J ) + YR(J )*AB + YI(J )*CD
        QL (KZ ) = EH*( YI(J ) - BI(J ) ) - YI(J )*AB &
            + YR(J )*CD
        PL (KZ ) = PL (KZ ) * BASE
        QL (KZ ) = QL (KZ ) * BASE
        NM (KZ ) = MQ
        T2 (KZ ) = 0.
        SUM = PL (KZ ) + SUM
        QUM = QL (KZ ) + QUM
1010 CONTINUE
!

```

```

! *****
! SE CHECA SI HAY TRANSFORMADORES CONECTADOS A ESTE NODO
! *****
!

      IF ( NTR.EQ.0) GOTO 1070
      DO 1050 K1 = 1,NTR
      IF ( NCT (K1) .NE. MAIN ) GOTO 1030
      DO 1020 K2 = 1,KZ
      IF ( NM (K2) .NE. NST (K1) ) GOTO 1020
      T2 (K2) = T1 (K1)
      NM (K2) = - NM (K2)
1020   CONTINUE
      GOTO 1050
1030   IF ( NST (K1) .NE. MAIN ) GOTO 1050
      DO 1040 K2 = 1, KZ
      IF ( NM (K2) .NE. NCT (K1) ) GOTO 1040
      T2 (K2) = 0.
      NM (K2) = - NM (K2)
1040   CONTINUE
1050   CONTINUE
      DO 1060 K2 = 1 , KZ
      NM (K2) = ABS (NM (K2) )
1060   CONTINUE
!
! *****
! CHECA SI HAY ALGUN SHUNT EN EL NODO
! *****
!
1070   IF ( NSHU .EQ. 0 ) GOTO 1100
      DO 1080 K1 = 1,NSHU
      IF ( NSH (K1) .EQ. MAIN ) GOTO 1090
1080   CONTINUE
      GOTO 1100
1090   SI = - EH * VLS (K1 )
      SI = SI * BASE
      SR = 0.
      QUM = QUM + SI
      NS1 = 1
!
! *****
! SE IMPRIMEN FLUJOS DE POTENCIA
! *****
!
1100   AMAG = E (MAIN )
      ANG = F(MAIN ) * 180.0/3.1416

      MAGV(MAIN) = AMAG
      ANGV(MAIN) = F(MAIN)
      ANGRADOS(MAIN) = ANG

! *****
! ASIGNACION DE VARIABLES PARA EL MODULO 1
! *****

1110   WRITE (LDIS,'(/,A,I3,1X,A,A,(F8.4,1X,f8.2,1x),I3)') ' DE ',MAIN, &
      NOMBRE (MAIN ), ' MW MVAR MVA TAP ', &
      AMAG,ANG,MAIN
!
! *****
! SE CHECA SI EL NODO ES GENERADOR
! *****
!
      DO 1130 K1 = 1 , NTUG
      IF ( NU (K1) .EQ. MAIN ) GOTO 1140
1130   CONTINUE
      GOTO 1160
1140   IG = 1
      PG1 = SUM + PC (MAIN )
      QG1 = QUM + QC (MAIN )
      FO = 'R'
      IF ( QG1 .GE. QMA (K1) * BASE ) FO = 'H'

```

```

      IF ( QG1 .LT. QMI (K1) * BASE ) FO = 'L'
      SG = PG1*PG1 + QG1*QG1
      SG = SQRT (SG)
1150  WRITE (LDIS,'(A,7X,2(F8.2,1X),A,F8.2)') ' GENERADOR', &
      PG1,QG1,FO,SG
      QG (K1 ) = QG1
!
! *****
!       SE CHECA SI EXISTE CARGA CONECTADA EN EL NODO
! *****
!
1160  PC1 = ABS (PC (MAIN) )
      QC1 = ABS (QC (MAIN) )
      IF (PC1 .GT. 1.0E-04 ) GOTO 1180
      IF (QC1 .LT. 1.0E-04 ) GOTO 1200
1180  SG = SQRT ( PC(MAIN) * PC(MAIN) + QC(MAIN) * QC(MAIN) )
1190  WRITE (LDIS,'(A,9X,2(F8.2,1X),2X,F8.2)') ' A CARGA', &
      PC (MAIN) ,QC(MAIN) ,SG
!
! *****
!       SE IMPRIME SHUNT
! *****
!
1200  IF ( NS1 .NE. 1 ) GOTO 1220
      SG = 0.
      SN = ABS (SI)
1210  WRITE (LDIS,'(A,8X,2(F8.2,1X),2X,F8.2)') ' AL SHUNT',SG,SI,SN
!
! *****
!       SE IMPRIMEN FLUJOS EN CADA LINEA
! *****
!
1220  DO 1250 K1 = 1, KZ
      SG = SQRT ( PL(K1)*PL(K1) + QL(K1)*QL(K1) )
      FO = ' '
      IF ( T2(K1) .GT. 1.0E-08 ) FO = 'TR'
1230  WRITE (LDIS,'(A,I3,1X,A,2(F8.2,1X),2(2X,F8.2),1X,A)') &
      ' A ',NM (K1),NOMBRE (NM (K1)),PL(K1),QL(K1),SG,T2(K1),FO
1250  CONTINUE
      SI = PC (MAIN) + SUM
      SQ = QC (MAIN) + QUM
      IF (IG .EQ. 0 ) GOTO 1260
      SI = SI - PG1
      SQ = SQ - QG1
1260  IF ( ABS (SQ) .GT. 0.5 ) GOTO 1270
      IF ( ABS (SI) .LT. 0.5 ) GOTO 1290
1270  S = SQRT (SI*SI + SQ*SQ)
1280  WRITE (LDIS,'(A,9X,3(F8.2,1X))') ' DESAJUSTE ',SI,SQ,SG
1290  RETURN
      END

      SUBROUTINE YBUSC (NNOD,NLIN,NTR,NSHU,NL2,M,N,YR,YI,BI,NE,NR,R, &
      X,B,NCT,NST,T1,NSH,VLS)
      IMPLICIT REAL*8 (A-H),INTEGER*2 (I-N),REAL*8 (O-Z)
      INTEGER*2 M(*),N(*),NE(*),NR(*),NCT(*),NST(*),NSH(*)
      REAL*8 YR(*),YI(*),R(*),X(*),B(*),T1(*),VLS(*),A,BI(*)
!
      N3 = NL2
      J2 = 1
      J1 = 1
      DO 8010 I =1, NL2
      YR (I) = 0
      YI (I) = 0
      BI (I) = 0
8010  CONTINUE
      DO 8080 J = 1 , NNOD
      DO 8060 K = 1 , NLIN
      IF ( NE (K) .NE. J ) GOTO 8030
8020  J1 = J1 + 1
      N (J1) = NR (K)
      GOTO 8050
8030  IF ( NR (K) .NE. J ) GOTO 8060
8040  J1 = J1 + 1
      N (J1) = NE (K)
8050  A = R(K)*R(K) + X(K)*X(K)

```

```

        YR (J1) = - R (K) /A
        YI (J1) = X (K) /A
        BI (J1) = B (K)
        YR (J2) = YR (J2) - YR (J1)
        YI (J2) = YI (J2) - YI (J1) + B (K)
8060      CONTINUE
        N (J2) = J1 - J2
        M (J) = J2
        J1 = J1 + 1
8070      J2 = J1
8080      CONTINUE
!
! *****
! SE INCLUYEN BANCOS CON RELACION NOMINAL
! *****
!
      IF ( NTR .EQ. 0 ) GOTO 8200
      DO 8110 I = 1, NTR
        N1 = NCT (I)
        N2 = NST (I)
        N3 = M (N1)
        N4 = N (N3)
        IF ( N4 .LT. 1 ) GOTO 8110
        DO 8090 J = 1, N4
          K = N3 + J
          N5 = N (K)
          IF ( N5 .NE. N2 ) GOTO 8090
          YR(N3) = YR(N3) + YR(K) * &
                ((T1(I)*T1(I)-1.0)/(T1(I)*T1(I)))
          YI(N3) = YI(N3) + YI(K) * &
                ((T1(I)*T1(I)-1.0)/(T1(I)*T1(I)))
          BI(K) = BI(K) - 1.0 / T1(I) * (1.0/T1(I)-1.0)*YI(K)
          YR(K) = YR(K)/T1(I)
          YI(K) = YI(K)/T1(I)
8090      CONTINUE
          N3 = M (N2)
          N4 = N (N3)
          DO 8100 J = 1, N4
            K = N3 + J
            N5 = N (K)
            IF ( N5 .NE. N1 ) GOTO 8100
            YR(K) = YR(K)/T1(I)
            BI(K) = BI(K) - YI(K) * (1.0 - 1.0/T1(I))
            YI(K) = YI(K)/T1(I)
8100      CONTINUE
8110      CONTINUE
!
! *****
! SE INCLUYEN ELEMENTOS EN DERIVACION
! *****
!
8200      IF ( NSHU .EQ. 0 ) GOTO 8230
          DO 8210 I = 1, NSHU
            N1 = NSH (I)
            N2 = M (N1)
            YI (N2) = YI (N2) + VLS (I)
8210      CONTINUE
8230      RETURN
          END

SUBROUTINE LFSOL (NNOD,NTUG,NISLA,NL2,IFIL,ISCOM,BASE,NORD,NORD1, &
                  PG,VG,NU,E3,F3,P2,Q2,PC,QC,YR,YI,NB1,NB1C,B1, &
                  B2,E,F,NITER,TOLP,TOLQ,M,N,NOMBRE)
IMPLICIT REAL*8 (A-H),INTEGER*2 (I-N),REAL*8 (O-Z)
INTEGER*2 &
ISCOM(*),NORD(*),NORD1(*),NB1(*),NB1C(*),M(*),N(*),NU(*)
REAL*8 PG(*),VG(*),E3(*),F3(*),E(*),F(*),P2(*),Q2(*), &
        PC(*),QC(*),YR(*),YI(*),B1(*),B2(*),D,D2
CHARACTER*10 NOMBRE(*)
KIT = 0
!
      CALL INIF (NNOD,NTUG,BASE,NORD,NORD1,PG,VG,NU,E3,F3,P2,Q2,PC,QC)
      NP = 1
      NQ = 1
2040      D = 0.

```

```

!
      CALL DESVP (NNOD,NISLA,NL2,ISCOM,M,N,P2,F,E3,F3,YR,YI,D,L8)
      IF ( D.LE.TOLP ) GOTO 2080
      CALL SOLP (NNOD,NISLA,IFIL,NB1,NB1C,NORD1,F,B1)
      CALL CORRA (NNOD,NISLA,ISCOM,F,F3)
      NQ = 1
2050 D2 = 0.
      CALL DESVQ (NNOD,NTUG,NL2,NU,M,N,Q2,E,E3,F3,YR,YI,D2,L9)
!
      IF ( D2.LE.TOLQ ) GOTO 2090
      CALL SOLQ (NNOD,NISLA,IFIL,NB1,NB1C,NORD1,E,B2)
      CALL CORRV (NNOD,NTUG,NU,E3,E)
      NP = 1
2060 KIT = KIT + 1
      IF ( KIT.GT.NITER ) GOTO 2100
      DO 2070 I = 1, NNOD
         E (I) = 0.
         F (I) = 0.
2070 CONTINUE
      GOTO 2040
2080 NP = 0
      IF ( NQ.EQ.0 ) GOTO 2130
      GOTO 2050
2090 NQ = 0
      IF ( NP.NE.0 ) GOTO 2060
      GOTO 2130
2100 WRITE (*,'(A,I5,A)') ' EL ESTUDIO NO CONVERGE EN ', &
      KIT , ' ITERACIONES'
      GOTO 2200
2130 WRITE (*,'(A,I5,A,I5)') ' EL ESTUDIO CONVERGE EN ', &
      KIT , ' ITERACIONES'
2140 DO 2150 I = 1, NNOD
      E(I) = E3 (I)
      F(I) = F3 (I)
2150 CONTINUE
2200 RETURN
      END
!
! *****
!
      SUBROUTINE INIF (NNOD,NTUG,BASE,NORD,NORD1,PG,VG,NU,E3,F3,P2, &
      Q2,PC,QC)
      IMPLICIT REAL*8 (A-H),INTEGER*2 (I-N),REAL*8 (O,Z)
      INTEGER*2 NORD(*),NORD1(*),NU(*)
      REAL*8 PG(*),VG(*),E3(*),F3(*),P2(*),Q2(*),PC(*),QC(*)
      DO 3010 I = 1, NNOD
         P2(I) = 0.
         Q2(I) = 0.
         E3(I) = 1.0
         F3(I) = 0.0
3010 CONTINUE
      DO 3020 I = 1, NTUG
         N1 = NU (I)
         E3 (N1) = VG (I)
3020 CONTINUE
!
! ***** CALCULO DE INYECCIONES NODALES DE POTENCIA *****
!
      DO 3030 I = 1, NTUG
         N1 = NU (I)
         P2 (N1) = PG (I)
3030 CONTINUE
      DO 3040 I = 1, NNOD
         P2 (I) = ( P2 (I) - PC (I) ) / BASE
         Q2 (I) = ( Q2 (I) - QC (I) ) / BASE
3040 CONTINUE
      DO 3060 I = 1, NNOD
         DO 3050 J = 1, NNOD
            IF ( NORD (J) .NE. I ) GOTO 3050
            NORD1 (I) = J
3050 CONTINUE
3060 CONTINUE
      RETURN
      END
!
! *****

```

```

!
SUBROUTINE DESVP (NNOD,NISLA,NL2,ISCOM,M,N,P2,F,E3,F3,YR,YI,D, &
      L8 )
IMPLICIT REAL*8 (A-H),INTEGER*2 (I-N),REAL*8 (O-Z)
INTEGER*2 ISCOM(*),M(*),N(*)
REAL*8 P2(*),F(*),E3(*),F3(*),YR(*),YI(*),A,B3,E1,F1,D,D1

      DO 4020 KOMUN = 1, NNOD
      DO 4010 J = 1, NISLA
        IF ( ISCOM (J) .EQ. KOMUN ) GOTO 4020
4010    CONTINUE
        N1 = M (KOMUN)
        N2 = N (N1)

!
! *****
!
        CALL IC (KOMUN,N1,N2,A,B3,E1,F1,NL2,E3,F3,YR,YI,NNOD,N)
        D1 = P2 (KOMUN) - (E1*A + F1*B3)
        F (KOMUN) = D1 / E3 (KOMUN)
        D1 = DABS (D1)
        IF ( D1 .LT. D ) GOTO 4020
        D = D1
        L8 = KOMUN
4020    CONTINUE
      RETURN
      END

!
! *****
!
SUBROUTINE SOLP (NNOD,NISLA,IFIL,NB1,NB1C,NORD1,F,B1)
!
IMPLICIT REAL*8 (A-H),INTEGER*2 (I-N),REAL*8 (O-Z)
INTEGER*2 NB1(*),NB1C(*),NORD1(*)
REAL*8 F(*),B1(*)
4500 N3 = NNOD - NISLA
      DO 4520 I = 1, N3
        L1 = NB1 (I)
        L2 = NB1C (L1)
        IF ( L2 .LT. 1 ) GOTO 4520
        DO 4510 J = 1, L2
          L3 = L1 + J
          L4 = NB1C (L3)
          L5 = NORD1 (I)
          F (L4) = F (L4) - F (L5) * B1 (L3)
4510      CONTINUE
4520    CONTINUE

!
! *****
!
      NORMALIZACION CON LOS ELEMENTOS DIAGONALES
!
! *****
!
      DO 4530 I = 1, N3
        L5 = NORD1 (I)
        L6 = NB1 (I)
        F (L5) = F (L5) / B1 (L6)
4530    CONTINUE

!
! *****
!
      SUSTITUCION REGRESIVA
!
! *****
!
      N4 = NISLA + 1
      N5 = NNOD - 1
      DO 4550 I = N4, N5
        K = NNOD - I
        L5 = NORD1 (K)
        L1 = NB1 (K)
        L2 = NB1C (L1)
        IF ( L2 .LT. 1 ) GOTO 4550
        DO 4540 J = 1, L2
          L3 = L1 + J
          L4 = NB1C (L3)
          F (L5) = F (L5) - B1 (L3) * F (L4)
4540      CONTINUE
4550    CONTINUE
      RETURN

```

```

      END
!
! *****
!
      SUBROUTINE DESVQ (NNOD,NTUG,NL2,NU,M,N,Q2,E,E3,F3,YR,YI,D2,L9)
!
      IMPLICIT REAL*8 (A-H),INTEGER*2 (I-N),REAL*8 (O-Z)
      INTEGER*2 M(*),N(*),NU(*)
      REAL*8 Q2(*),E(*),E3(*),F3(*),YR(*),YI(*),D1,D2,A,B3,E1,F1
!
      DO 5020 KOMUN = 1, NNOD
        DO 5010 J = 1, NTUG
          IF ( NU (J) .EQ. KOMUN ) GOTO 5020
5010      CONTINUE
          N1 = M (KOMUN)
          N2 = N (N1)
          CALL IC (KOMUN,N1,N2,A,B3,E1,F1,NL2,E3,F3,YR,YI,NNOD,N)
          D1 = Q2 (KOMUN ) - (F1*A - E1*B3)
          E (KOMUN ) = D1 / E3 (KOMUN )
          D1 = DABS (D1)
          IF ( D1 .LT. D2 ) GOTO 5020
          D2 = D1
          L9 = KOMUN
5020      CONTINUE
          RETURN
          END
!
! *****
!
      SUBROUTINE SOLQ (NNOD,NISLA,IFIL,NB1,NB1C,NORD1,E,B2)
!
      IMPLICIT REAL*8 (A-H),INTEGER*2 (I-N),REAL*8 (O-Z)
      INTEGER*2 NB1(*),NB1C(*),NORD1(*)
      REAL*8 E(*),B2(*)
!
5500      N3 = NNOD - NISLA
      DO 5570 I = 1, N3
        L1 = NB1 (I)
        L2 = NB1C (L1)
        IF ( L2 .LT. 1 ) GOTO 5570
        DO 5560 J = 1, L2
          L3 = L1 + J
          L4 = NB1C (L3)
          L5 = NORD1 (I)
          E(L4) = E(L4) - E(L5) *B2 (L3 )
5560      CONTINUE
5570      CONTINUE
!
! *****
!
      SE NORMALIZAN ELEMENTOS DIAGONALES
! *****
!
      DO 5580 I = 1 , N3
        L5 = NORD1 (I)
        L6 = NB1 (I)
        E (L5) = E(L5) / B2 (L6)
5580      CONTINUE
!
! *****
!
      SUSTITUCION REGRESIVA
! *****
!
      N4 = NISLA + 1
      N5 = NNOD - 1
      DO 5600 I = N4, N5
        K = NNOD - I
        L5 = NORD1 (K)
        L1 = NB1 (K)
        L2 = NB1C (L1)
        IF ( L2 .LT. 1 ) GOTO 5600
        DO 5590 J = 1 , L2
          L3 = L1 + J
          L4 = NB1C (L3)
          E (L5) = E (L5) - B2 (L3)*E(L4)
5590      CONTINUE
5600      CONTINUE

```

```

      RETURN
      END
!
! *****
!
      SUBROUTINE IC (KOMUN,N1,N2,A,B3,E1,F1,NL2,E3,F3,YR,YI,NNOD,N)

      IMPLICIT REAL*8 (A-H),INTEGER*2 (I-N),REAL*8 (O-Z)
      INTEGER*2 N(*)
      REAL*8 YR(*),YI(*),E3(*),F3(*),E1,F1,E2,F2,A,B3,X3,Y3

!
      E1 = E3 (KOMUN) * COS (F3 (KOMUN))
      F1 = E3 (KOMUN) * SIN (F3 (KOMUN))
      E2 = YR (N1)
      F2 = YI (N1)
      A = E2*E1 - F2*F1
      B3 = E1*F2 + F1*E2
      N33 = N1 + 1
      N34 = N1 + N2
      DO 6010 J1 = N33, N34
         N3 = N (J1)
         X3 = E3 (N3)
         Y3 = F3 (N3)
         E2 = X3 * COS (Y3)
         F2 = X3 * SIN (Y3)
         X3 = YR (J1)
         Y3 = YI (J1)
         A = A + E2*X3 - F2*Y3
         B3 = B3 + F2*X3 + E2*Y3
6010 CONTINUE
      RETURN
      END
!
! *****
!
      SUBROUTINE CORRA (NNOD,NISLA,ISCOM,F,F3)

!
      IMPLICIT REAL*8 (A-H),INTEGER*2 (I-N),REAL*8 (O-Z)
      INTEGER*2 ISCOM (*)
      REAL*8 F(*),F3(*)

!
6500 DO 6530 I = 1, NNOD
      DO 6510 J = 1, NISLA
         IF ( ISCOM (J) .EQ. I ) GOTO 6530
6510 CONTINUE
         F3 (I) = F3 (I) + F (I)
6530 CONTINUE
      RETURN
      END
!
! *****
!
      SUBROUTINE CORRV (NNOD,NTUG,NU,E3,E)

!
      IMPLICIT REAL*8 (A-H),INTEGER*2 (I-N),REAL*8 (O-Z)
      INTEGER*2 NU(*)
      REAL*8 E(*),E3(*)

!
      DO 7020 I = 1, NNOD
         DO 7010 J = 1, NTUG
            IF ( NU (J) .EQ. I ) GOTO 7020
7010 CONTINUE
            E3 (I) = E3 (I) + E (I)
7020 CONTINUE
      RETURN
      END
!
! *****
!
      FIN DE SUBROUTINA LFSOL.FOR
! *****

```

B3. CODIGO FUENTE DEL PROGRAMA ACOPLAMIENTO.FOR

A continuación se muestra el código fuente del programa ACOPLAMIENTO.FOR para la solución de flujos de potencia de redes de ALTA Y MEDIA tensión, el cual fue implementado en este presente trabajo.

```
PROGRAM ACOPLAMIENTO

USE MDATOS
USE MDATOS2
USE MDATOS3
USE rutinas_de_asignacion

IMPLICIT NONE

REAL*8 :: VDES, VABCPROM, ERROR
INTEGER :: NUMITER
COMPLEX*8 :: VDIS012(3,1)
INTEGER :: I,J
real, dimension(6) :: VAUX
COMPLEX, DIMENSION(3) :: VAUX2,V012AUX

WRITE(6,*) ' '
WRITE(6,*) ' '
WRITE(6,*) ' INSTITUTO POLITACNICO NACIONAL '
WRITE(6,*) ' ESCUELA SUPERIOR DE INGENIERIA MECANICA Y ELECTRICA'
WRITE(6,*) ' SECCION DE ESTUDIOS DE POSGRADO E INVESTIGACION'
WRITE(6,*) ' '
WRITE(6,*) ' '
WRITE(6,*) ' PROGRAMA PARA EL CALCULO DE FLUJOS DE POTENCIA DE ALTA Y MEDIA TENCION'
WRITE(6,*) ' '
WRITE(6,*) ' '
WRITE(6,*) ' Version 1.0 (2014) '
WRITE(6,*) ' ELABORADO POR:'
WRITE(6,*) ' ING. ELECTRICISTA DANIEL MARTINEZ PEREZ '
WRITE(6,*) ' '

!PERMITE LA LECTURA INICIAL DE LOS DATOS DE LAS REDES DE DISTRIBUCION Y TRANSMISION
ACTUALIZAVOL = 0
ACTUALIZAPOT = 0

WRITE(6,10)
10 FORMAT(3/,5X,'>> NUMERO DE ITERACIONES MAXIMAS PARA EL PROCESO (?) ',/,3X)

READ (5,*)NUMITER

WRITE(6,101)
101 FORMAT(3/,5X,'>> DAME EL NOMBRE DEL ACHIVO DE DATOS ',/,8X,'DE DISTRIBUCION PARA LA SOLUCION DE FLUJOS DE POTENCIA (?) ',/,3X)

CALL RADIFLUX

CALL SIMETRICAS(VDIS,VDIS012)

!POTENCIAS INICIALES DE CARGA

P3D = PDIS(1)+PDIS(2)+PDIS(3) ! POTENCIA REAL EN MW
Q3D = QDIS(1)+QDIS(2)+QDIS(3) ! POTENCIA REACTIVA EN MVAR

VDES = 100*(CABS(VDIS012(3,1))/CABS(VDIS012(2,1))) ! DESVALANCE DE VOLTAJE (VDES) = (VOLTAJE DE SECUENCIA NEGATIVA / VOLTAJE DE SECUENCIA POSITIVA)

PRINT*,'POTENCIAS INICIALES DE DISTRIBUCION'
WRITE(*,1) P3D, Q3D
```

```

1 FORMAT('POTENCIA ACTIVA TRIFASICA', F8.4, ' MW',/, 'POTENCIA ACTIVA TRIFASICA', F8.4, ' MVAR')
PRINT*, 'DESBALANCE DE LA RED DE DISTRIBUCION'
PRINT*, VDES
PAUSE ' >> ENTER PARA CONTINUAR'

WRITE(6,102)
102 FORMAT(3/,5X,'>> DAME EL NOMBRE DEL ACHIVO DE DATOS ',/,8X,'DE TRANSMISION PARA LA SOLUCION DE FLUJOS
DE POTENCIA (?)',/,3X)

CALL DR

PRINT *, 'VOLTAJES DE FASE DE LA RED DE TRANSMISION'
WRITE(6,2)VTA(NODOPOT),ANGTA(NODOPOT),VTB(NODOPOT),ANGTB(NODOPOT),VTC(NODOPOT),ANGTC(NODOPOT)
2 FORMAT(6X,F15.10,2X,F20.15,3X,F20.12,2X,F15.10,8X,F15.10,2X,F15.10)

J= 0

DO I = 1, NUMITER

J=J+1

PRINT *,J

! LLAMADO DE RUTINA RADIFLUX PARA LA SOLUCION DE FLUJOS DE POTENCIA Y OBTENCION DE LOS SIGUINETES PARAMETROS:
! MATRIZ DE VOLTAJES DE FASE                                VDIS(3,1)          MDATOS2
! MATRIZ DE VOLTAJES EN COMPONENTES SIMETRICAS              V012(3,1)          MDATOS2
! POTENCIA ACTIVA POR FASE                                PDIS(3)            MDAOTS2
! POTENCIA REACTIVA POR FASE                                QDIS(3)            MDATOS2

! VDES = 100*(CABS(V012(3,1))/CABS(V012(2,1)))      ! DESVALANCE DE VOLTAJE (VDES) = (VOLTAJE DE SECUENCIA
NEGATIVA / VOLTAJE DE SECUENCIA POSITIVA)

VABCPROM      = (CABS(VDIS(1,1))+CABS(VDIS(2,1))+CABS(VDIS(3,1)))/3

! PRINT *, VDIS
! PRINT *, V012
! PRINT *, VDES
!PRINT *,      VABCPROM

! LLAMADO DE LA SUBROUTINA DR PARA LA SOLUCION DE FLUJOS DE POTENCIA DE TRANSMISION DE LOS SIGUIENTES
PARAMETROS
! VOLTAJES DE SECUENCIA POSITIVA DE CADA UNO DE LOS NODOS DE LA RED DE TRANSMISION
VSP(NONODOS)      MDATOS
! VOLTAJES DE FASE DE CADA UNO DE LOS NODOS DE LA LINEA DE TRANSMISION                                VAT(NONODOS)
MDATOS
!
MDATOS                                ANGAT(NONODOS)
!
MDATOS                                VBT(NONODOS)
!
MDATOS                                ANGBT(NONODOS)
!
MDATOS                                VCT(NONODOS)
!
MDATOS                                ANGCT(NONODOS)
!
MDATOS

PRINT*, 'VOLTAJE PROMEDIO DE DISTRIBUCION'
PRINT*, VABCPROM
PRINT*, 'VOLTAJE DE SECUENCIA POSITIVA'
PRINT*, VTA(NODOPOT)

ERROR = ABS( (VABCPROM-VTA(NODOPOT))/VTA(NODOPOT) )
PRINT*, 'ERROR RELATIVO'
PRINT*, ERROR

IF(ERROR .LT. 0.001) THEN

WRITE(6,11)J
11 FORMAT(3/,5X,'>> EL PROCESO CONVERGIO EN ',I3,' ITERACIONES',/,3X)
ACTUALIZAVOL=1
ACTUALIZAPOT=1
CALL RADIFLUX

```

```

CALL DR

GOTO 100

ELSE

! SE OBTIENEN LOS VOLTAJES ACTUALIZADOS PARA SER INGRESADOS EN LA RED DE DISTRIBUCION

VAUX(1)=VTA(NODOPOT)
VAUX(2)=ANGTA(NODOPOT)
VAUX(3)=VTB(NODOPOT)
VAUX(4)=ANGTB(NODOPOT)
VAUX(5)=VTC(NODOPOT)
VAUX(6)=ANGTC(NODOPOT)

call conversor_voltaje(VAUX,VAUX2)

CALL SIMETRICAS(VAUX2,V012AUX)
V012AUX(1)= (0.0,0.0)
V012AUX(3)= (0.0,0.0)

V012AUX(3)= (VDES*V012AUX(2))/100

CALL COMPFASE(V012AUX,VDISNEW) !VDISNEW ESTAN EN RECTANGULARES

!UNA VEZ AGRADADO EL DESBALANCE DE LA RED DE DISTRIBUCION AL NUEVO VOLTAJE CALCULADO DE LA SOLUCION DE
FLUJOS DE POTENCIA
!SE CONVIERTEN A FORMA POLAR PARA SER INGRESADO COMO NUEVO DATO EN RADIFLUX

ACTUALIZAVOL=1

CALL RADIFLUX

P3D = PDIS(1)+PDIS(2)+PDIS(3)
Q3D = QDIS(1)+QDIS(2)+QDIS(3)

PRINT*, 'POTENCIAS INICIALES DE DISTRIBUCION'
WRITE(*,3) P3D, Q3D
3 FORMAT('POTENCIA ACTIVA TRIFASICA', F8.4, ' MW',/, 'POTENCIA ACTIVA TRIFASICA', F8.4, ' MVAR')
PAUSE 'ENTER PARA CONTINUAR'

ACTUALIZAPOT=1

CALL DR

END IF

END DO

100 PRINT *, ' >> TERMINO LA SOLUCION DE AMBOS SISTEMAS '
PAUSE ' >> ENTER PARA FINALIZAR '

END PROGRAM ACOPLAMIENTO

SUBROUTINE SIMETRICAS(VFASE,VCOMSIM)

IMPLICIT NONE

COMPLEX*8 :: VFASE(3,1), VCOMSIM(3,1), S(3,3)
COMPLEX*8, PARAMETER :: a = (-0.5,0.8660254038)

! ** RUTINA PARA LA TRANSFORMACION DE SECUENCIA DE FASE A COMPONENTES SIMETRICAS
! ** VCOMSIM[3,1] = S[3,3] * VFASE[3,1]
! ** *****
! ** VCOMSIM VECTOR DE COMPONENTES SIMETRICAS **
! ** VFASE VECTOR DE VOLTAJES DE FASE **
! ** S MATRIZ DE TRANSFORMACION **

! MATRIZ DE TRANSFORMACION

```

```

S(1,1) = (1,0.0)
S(2,1) = (1,0.0)
S(3,1) = (1,0.0)
S(1,2) = (1,0.0)
S(1,3) = (1,0.0)
S(2,2) = (a)
S(2,3) = (a**2)
S(3,2) = (a**2)
S(3,3) = (a)

S= S/3

CALL MULMA(S,VFASE,VCOMSIM,3,3,1)

SUBROUTINE MULMA(a,b,c,m,n,p)
IMPLICIT NONE

INTEGER ::i,j,k,m,n,p

COMPLEX::a(m,n),b(n,p),c(m,p)

! ** RUTINA PARA EL PRODUCTO DE MATRICES COMPLEJAS
! ** [c] = [a] [b]
! ** ***** **
! ** m orden de los renglones de la matriz a **
! ** n orden de las columnas de la matriz a **
! ** m orden de las columnas de la matriz a **
! ** a matriz de pre_producto **
! ** b matriz de post_producto **
! ** c matriz producto **
! ** ***** **

c=(0.0,0.0)

DO i=1,m

DO j=1,p

DO k=1,n

c(i,j)=c(i,j)+a(i,k)*b(k,j)

END DO

END DO

END DO

END SUBROUTINE MULMA

SUBROUTINE VOLTAJESPOT
USE MDATOS3
USE MDATOS
IMPLICIT NONE

INTEGER :: nodo, I
COMPLEX*8 :: V012(3,1), VABC(3,1)
CHARACTER(len=30) :: AUX
COMPLEX*8 :: VSP(NONODOS)
REAL*8 :: MAGPU(3),ANG(3)

IF (ACTUALIZAPOT .EQ. 0) THEN
ALLOCATE (VTA(NONODOS),ANGTA(NONODOS),VTB(NONODOS),ANGTB(NONODOS),VTC(NONODOS),ANGTC(NONODOS))
END IF
! LECTURA DE LOS VOLTAJES NODALES DEL ARCHIVO AUXILIAR DE LA UNIDAD

AUX = 'VOLTAJES.SAL'
OPEN(UNIT=20, FILE = AUX, STATUS='UNKNOWN')
20 FORMAT(A30)

! IMPRESION DE VOLTAJES EN FORMA POLAR

```

```

WRITE(20,1)
1 FORMAT(2/,10X,'VOLTAJES DE SECUENCIA POSITIVA',2/,2X,'NODO',8X,'MAGNITUD
(PU)',5X,'ANGULO (GRADOS)',5X,'VOLTAJE BASE')

DO nodo = 1, NONODOS

    WRITE (20,12)nodo,MAGV(nodo),ANGRADOS(nodo), VBASE(nodo)
12    FORMAT(X,I3,8X,F15.13,4X,F13.9,4X,F13.5)

! TRANSFORMACION DE VOLTAJE DE SECUENCIA POSITIVA DE POLAR A RECTANGULAR

VSP(nodo) = CMPLX((MAGV(nodo) * COS(ANGV(nodo))), (MAGV(nodo) * SIN(ANGV(nodo))))

END DO

! TRANSFORMACION DE LOS VOLTAJES DE COMPONENTES SIMETRICAS A COORDENADAS DE FASE

WRITE(20,30)
30    FORMAT(5/,14X,'VOLTAJES EN COORDENADAS DE FASE',2/,2X,'NODO',13X,'VOLTAJE DE FASE A',20X,'VOLTAJE DE
FASE B',23X,'VOLTAJE DE FASE C',2/)

DO nodo =1, NONODOS

V012(1,1) = (0.0,0.0)
V012(2,1) = VSP(nodo)
V012(3,1) = (0.0,0.0)

CALL COMPPHASE(V012,VABC)

    WRITE(20,31)nodo,VABC(1,1),VABC(2,1),VABC(3,1)
31    FORMAT(2X,I2,2X,F15.10,' + J ',F15.12,5X,F15.12,' + J ',F15.10,5X,F15.10,' + J ',F15.10)

DO I =1, 3
    MAGPU(I) = CABS(VABC(I,1))

    IF ( IMAG( VABC(I,1) ) /= 0.0 .OR. REAL(VABC(I,1) ) /= 0.0 ) THEN

        ANG(I) = ATAN2D( IMAG(VABC(I,1)), REAL (VABC(I,1)) )

    ELSE

        ANG(I) = 0.0

    END IF

    VTA(nodo)=MAGPU(1)*VBASE(nodo)
    ANGTA(nodo)=ANG(1)
    VTB(nodo)=MAGPU(2)*VBASE(nodo)
    ANGTB(nodo)=ANG(2)
    VTC(nodo)=MAGPU(3)*VBASE(nodo)
    ANGTC(nodo)=ANG(3)

END DO

WRITE(20,32)MAGPU(1),ANG(1),MAGPU(2),ANG(2),MAGPU(3),ANG(3)
32    FORMAT(6X,F15.10,2X,F20.15,3X,F20.12,2X,F15.10,8X,F15.10,2X,F15.10)

WRITE(20,42)VTA(nodo),ANGTA(nodo),VTB(nodo),ANGTB(nodo),VTC(nodo),ANGTC(nodo)
42    FORMAT(6X,F15.10,2X,F20.15,3X,F20.12,2X,F15.10,8X,F15.10,2X,F15.10)

END DO
END SUBROUTINE VOLTAJESPTOT

```