

85



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE INGENIERIA

DISEÑO DE UN SISTEMA EN COMPUTADORA PARA EL ANALISIS DE SISTEMAS ELECTRICOS DE POTENCIA.

T E S I S
QUE PARA OBTENER EL TITULO DE:
INGENIERO MECANICO ELECTRICISTA
PRESENTAN:
ARMANDO VERGARA AYALA
EMIGDIO ROMERO ROSALES

290984

DIRECTOR DE TESIS: ING. MARTIN BARCENAS ESCOBAR.

Acompañado de un disquette 3 1/2



CIUDAD UNIVERSITARIA

2001



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

DISEÑO DE UN SISTEMA EN COMPUTADORA PARA EL ANÁLISIS DE SISTEMAS ELECTRICOS DE POTENCIA.

Indice.

	Pag.
1. Introducción.	1
1.1 Antecedentes.	1
1.1.1 Interfaz Gráfica de Usuario (GUI)	1
1.1.2 Programación orientada a objetos (POO)	1
1.1.3 Visual Basic	3
1.1.4 Paquetes comerciales de software	5
1.2 Planteamiento del problema.	7
1.3 Planteamiento de solución.	8
1.3.1 Planteamiento general.	8
1.3.2 Alcances de REDELEC.	9
1.4 Objetivos.	11
1.5 Organización del trabajo.	12
2. Representación y modelado de las redes de los Sistemas Eléctricos de Potencia.	14
2.1 Representación de líneas de transmisión.	14
2.1.1 La línea de transmisión corta.	16
2.1.2 La línea de longitud media.	22
2.1.2.1 Modelo Π .	22
2.1.2.2 Modelo T.	28
2.1.3 La línea de transmisión larga.	36
2.2 Diagramas unifilar, de impedancia y de reactancia.	46
2.3 Cantidades en por unidad.	47
2.3.1 Cambio de base de cantidades en por unidad.	49
2.4 Ecuaciones de redes y soluciones. Algoritmos en computadora para la construcción de las matrices de admitancias e impedancias de barra.	49
2.4.1 Cálculo de los voltajes de nodo.	50
2.4.2 Calculo de las corrientes de malla.	56
2.4.3 Método de descomposición LU.	57
3. Flujos de potencia.	71
3.1 Definición del problema de flujos de potencia.	71

3.2	Ecuaciones para el estudio de flujos de carga.	72
3.3	Métodos de solución para los flujos de potencia.	74
3.3.1	Método de Gauss-Seidel.	76
3.3.2	Método de Newton-Rhapson.	78
3.3.3	Método desacoplado rápido.	80
3.4	Cálculo de las pérdidas y flujos de potencia en las líneas de transmisión.	82
3.5	Aplicaciones del estudio de flujos de carga	83
3.6	Ejemplos de solución de flujos de potencia.	83
4.	Programación y elaboración de interfases gráficas. Plataforma Windows.	96
4.1	Interfaz Gráfica del Usuario (GUI).	97
4.1.1	Características de las GUI	97
4.1.2	Ventajas de las GUI.	98
4.2	Programación orientada a objetos (POO).	99
4.2.1	Orígenes de la POO.	100
4.2.2	Mecanismos de la POO.	102
4.2.2.1	Objetos.	102
4.2.2.2	Mensajes.	102
4.2.2.3	Métodos.	103
4.2.2.4	Clases.	103
4.2.3	Características de la POO.	104
4.2.3.1	Abstracción.	104
4.2.3.2	Encapsulamiento.	104
4.2.3.3	Herencia.	104
4.2.3.4	Polimorfismo.	104
4.3	Programación en Visual Basic.	105
4.4	Elaboración de interfases gráficas del usuario para el manejo de los Algoritmos generados en computadora.	107
5.	Manual del usuario.	109
5.1	Instalación de la aplicación.	109
5.2	Entorno visual de REDELEC.	110
5.2.1	Barra de menús.	111
5.2.2	Barra de herramientas horizontal.	112
5.2.3	Barra de herramientas vertical.	112
5.3	Solución de redes eléctricas.	112
5.3.1	Cálculo de corrientes de malla.	113
5.3.1.1	Matriz de conexión rama-malla.	114
5.3.1.2	Matriz de impedancia de rama.	116
5.3.1.3	Vector de voltajes de generación.	117
5.3.1.4	Lectura de resultados	118
5.3.2	Cálculo de los voltajes de nodo.	120
5.3.2.1	Matriz de conexión rama-nodo.	120

Índice.

5.3.2.2	Procedimiento para generar la matriz de conexión ram-nodo.	121
5.3.2.3	Lectura de impedancias de rama.	122
5.3.2.4	Vector de corrientes de generación.	123
5.3.2.5	Lectura de resultados.	123
5.4	Cálculo de líneas de transmisión.	125
5.4.1	Lectura de resultados.	127
5.5	Calculo de flujos de potencia (método de Gauss-Seidel).	128
5.5.1	Lectura de resultados.	130
5.6	Opciones para editar el dibujo de la red eléctrica.	131
	Conclusiones.	132
	Bibliografía.	135
	Anexo.	139

Capítulo 1. Introducción

1.1 Antecedentes

1.1.1 Interfaz Gráfica de Usuario (GUI).

El desarrollo de esta tesis se apoya en la aplicación de las GUI. Estas presentan las siguientes características:

- a) Ventanas que representan gráficamente lo que la computadora está haciendo.
- b) Iconos que representan archivos, directorios y otras aplicaciones del sistema.
- c) Cuadros de diálogo, botones, barras de desplazamiento y otros objetos gráficos que permiten al programador y al usuario indicar a la computadora lo que debe hacer y cómo hacerlo.
- d) Menús en pantalla que aparecen o desaparecen bajo el control del dispositivo señalador, (apuntador del ratón).

Los objetos que se depositan en el formulario y el formulario mismo poseen procedimientos y funciones propios. En éstos se pueden programar acciones definidas por el programador que responden a un suceso, que puede ser un click del ratón, presionar una tecla, etc. Cuando ocurre un suceso, el entorno del sistema determina la naturaleza del suceso y hace que se ejecute la función apropiada. Se pueden utilizar tantos formularios como se requieran, y dependiendo de la utilidad que se les dé, estos serán de diferentes tipos. Así se puede crear un formulario para crear un gráfico, para visualizar información o para aceptar datos. En una GUI los controles depositados en pantalla se programan de tal manera que existe una correspondencia directa entre los objetos de presentación y código.

Con las GUI, los objetos en pantalla le permiten al usuario manejar imágenes más familiares, esto hace que las aplicaciones se parezcan mucho más a las funciones del mundo real y cada vez menos a los procesos de programación tradicionales.

En el caso de esta tesis, se crearon formularios para realizar las lecturas de datos así como para generar los diagramas que representan las redes eléctricas en estudio. A su vez se manejan controles que permiten al usuario interactuar con la computadora de una forma directa.

1.1.2 Programación orientada a objetos

Está de moda hablar de la Programación orientada a objetos (POO), y no es nuevo este concepto, pero es ahora cuando se está generalizando su uso. Esta moda ha provocado alguna confusión, muchas herramientas han dicho estar orientadas a objeto sin serlo, otras han realizado

implementaciones muy extrañas, y algunas han sabido mantener un equilibrio y una buena implementación, por lo que son verdaderamente recomendables para desarrollar con este nuevo sistema de programación.

Las personas que llevan años realizando programación recuerdan lo difícil y extraño que fue el paso de la programación lineal (con Go To) a la programación estructurada (con Call), ahora toca el paso de la programación estructurada a la Programación Orientada a Objetos (POO). La POO viene a completar algunas de las lagunas de la programación estructurada, cómo ésta vino a solucionar algunos de los problemas de la programación lineal. No es la solución a todos nuestros problemas, pero sí facilita nuestra labor de programación y mantenimiento.

La POO intenta ser un mejor sistema para el desarrollo de aplicaciones. Como toda técnica de programación si se hace mal puede ser desastrosa.

Una de las primeras mejoras que obtenemos por el uso de la POO es el permitir afrontar programas más complejos y de mayor tamaño con menos esfuerzo. El hecho de trabajar con pequeños elementos bien definidos, como son los objetos, nos permite aislar a cada componente de la aplicación del resto y de esa forma aprovechar en mayor medida nuestro esfuerzo.

Una vez adiestrados en las nuevas técnicas de orientación a objetos obtendremos también una mejora considerable en nuestro rendimiento de desarrollo. Las grandes facilidades para el reaprovechamiento del código que nos ofrece la orientación a objeto harán que desarrollemos con mayor velocidad, pero manteniendo unos buenos niveles de calidad.

La reutilización de código en la POO nos otorga una gran flexibilidad. La existencia de la herencia permite modificar las características que necesitamos de una clase de objeto con seguridad de no alterar las especificaciones del mismo y aprovechando todo el desarrollo realizado en el mismo.

El mantenimiento de aplicaciones se ha visto como uno de los grandes problemas de la programación actual. Con las técnicas de POO es más sencillo realizar este mantenimiento. Los objetos son elementos de tamaño pequeño, bien definidos, y por lo tanto más fáciles de mantener. Además la existencia de la herencia nos va a asegurar que la modificación de algunas características de la clase no van a afectar a los desarrollos ya terminados.

La definición correcta de los objetos permitirá tener una mejor estructuración de nuestros programas. Con la POO se tiene un acercamiento más natural a los problemas y por lo tanto los

análisis de aplicaciones orientadas a objeto tienen un acercamiento a la realidad mucho más completa que con la programación estructurada.

La POO es muy fácilmente comprensible en entornos gráficos, pues el hecho de que estos entornos manejen objetos gráficos hace muy recomendable este tipo de programación. Pero debemos tener claro que es totalmente posible realizar POO en entornos de tipo carácter y es posible programar entornos gráficos sin POO.

Tampoco podemos decir que todo sean facilidades. Por una parte la POO nos va a obligar a cambiar nuestra forma de pensar los programas y por lo tanto es necesario un tiempo para que las nuevas técnicas se nos hagan habituales y automáticamente nos salga programar de esta forma.

No basta con dominar estas técnicas para que alcancemos las mejores cotas de productividad, posiblemente sea necesario elaborar unas buenas bibliotecas de programas. Para ello es conveniente desarrollar un par de aplicaciones bajo esta técnicas para darnos cuenta de que es lo que realmente es factible de ser reutilizado y de esa forma organizar nuestras propias bibliotecas de programas.

La depuración de código orientado a objeto es algo más compleja que la depuración de código estructurado. Esto no quiere decir que nuestro código va a ser peor o va a tener más errores, pero sí es cierto que en el caso de producirse un error deberemos recorrer todo el árbol de herencia para encontrarlo, algo que en programación estructurada no tenemos que hacer.

Pero en general, podemos decir que los inconvenientes son realmente menores que las ventajas, y por lo tanto es realmente recomendable el desarrollo con POO.

1.1.3 Visual Basic.

Visual Basic es uno de los tantos lenguajes de programación que podemos encontrar hoy en día. Dicho lenguaje nace del BASIC (Beginner's All-purpose Symbolic Instruction Code) que fue creado en su versión original en el Dartmouth College, con el propósito de servir a aquellas personas que estaban interesadas en iniciarse en algún lenguaje de programación.

Luego de sufrir varias modificaciones, en el año 1978 se estableció el BASIC estándar. La sencillez del lenguaje ganó el desprecio de los programadores avanzados por considerarlo "un lenguaje para principiantes".

Primero fue GW-BASIC, luego se transformó en QuickBASIC y actualmente se lo conoce como Visual Basic. Este lenguaje combina la sencillez del BASIC con un poderoso lenguaje de

programación Visual que juntos permiten desarrollar robustos programas para Windows. Esta fusión de sencillez y la estética permitió ampliar mucho más el monopolio de Microsoft, ya que el lenguaje sólo es compatible con Windows, un sistema operativo de la misma empresa.

Visual Basic ya no es más "un lenguaje para principiantes" sino que es una perfecta alternativa para los programadores de cualquier nivel que deseen desarrollar aplicaciones compatibles con Windows.

Como vimos anteriormente, Visual Basic es una herramienta sencilla pero potente para desarrollar aplicaciones Windows en Basic. Esto no parece ser suficiente para justificar toda la expectación levantada hasta que se cae en la cuenta de que Microsoft Windows es utilizado por millones de personas y desarrollar una aplicación para Windows solía requerir un programador experto en C provisto con una infinidad de documentación y al menos 20 MB de espacio en disco duro para el compilador de C. En las versiones 3 y más recientes de Visual Basic se añade una forma simple de manejar las más potentes bases de datos disponibles. También se añade soporte para los cuadros de diálogo comunes y simplifica la comunicación con otras utilidades de Windows.

Las interfases gráficas de usuario han revolucionado la industria de las microcomputadoras. En lugar del crítico C:> que los usuarios de DOS utilizan, encuentran un escritorio poblado de iconos. Todo esto dá una imagen gráfica de lo que puede ofrecer la computadora. Las aplicaciones para Windows generalmente tienen una interfaz de usuario consistente. Esto significa que los usuarios disponen de más tiempo para dominar la aplicación sin tener que preocuparse de que teclas deben pulsarse, dentro de los menús y cuadros de diálogo.

Antes de la existencia de Visual Basic, el desarrollo de aplicaciones para Windows era mucho más complicado que desarrollar aplicaciones para DOS. Los programadores tenían que preocuparse más de lo que estaba haciendo el ratón, de dónde estaba el usuario dentro de un menú y de si estaba realizando un click o un doble click en un punto determinado. Desarrollar aplicaciones para Windows requería expertos programadores en C, e incluso estos tenían problemas.

Visual Basic ha cambiado esta situación. Se pueden desarrollar aplicaciones para Windows en fracciones de tiempo que se necesitaban anteriormente. Los errores de programación no se generan frecuentemente, y si los hacen son más sencillos de detectar y solventar. No se pierde mucho tiempo en características: las aplicaciones desarrolladas en Visual Basic se ejecutan rápidamente.

1.1.4 Paquetes comerciales de software.

Mediante una búsqueda en internet se localizaron paquetes de software para análisis de Sistemas Eléctricos de Potencia. Las características de estos paquetes se describen a continuación:

- a) CAPE, realizado por la compañía ELECTROCON, con dirección electrónica www.electrocon.com. Es un programa modular que comprende análisis de corto circuito, cálculo de parámetros de líneas de transmisión, coordinación de protecciones y flujos de potencia. El módulo de flujos de potencia utiliza los métodos de Newton-Rhapson y desacoplado rápido, posee una variedad de dispositivos de control de voltaje como lo son los bancos de capacitores, generación remota de potencia reactiva, manipulación de tap's de transformadores y transformadores de corrimiento de fases. Contiene además, reportes textuales y cálculo de condiciones iniciales para análisis de corto circuito. No se especifica el número máximo de nodos ni hace referencia al precio de las licencias de cada uno de los módulos.
- b) EDSA, elaborado por EDSA Micro Corporation, con dirección electrónica www.edsa.com. EDSA también es un programa modular que contiene análisis de corto circuito, análisis de flujos de potencia. Simulación armónica multifase, simulación de estabilidad transitoria, análisis de armónicas y coordinación de protecciones. Todos los módulos tienen una capacidad hasta de 50,000 nodos. EL módulo de flujos de potencia posee elementos de control de voltaje, representación de líneas de transmisión aéreas con el modelo PI, control de intercambio de potencia en áreas y líneas de corriente directa. No se especifica el método utilizado para cálculo de flujos de potencia, tampoco se especifica precio.
- c) ASPEN, realizado por la compañía ASPEN LEAFLET, con dirección electrónica www.aspeninc.com. ASPEN ofrece una serie de 3 paquetes y 2 complementos. El primer paquete se llama OneLiner y sirve para el análisis de corto circuito y coordinación de protecciones para los sistemas de transmisión. El siguiente paquete es el de flujos de potencia, que abarca aspectos como el control de voltaje por medio de capacitores, transformadores de corrimiento de fase y manipulación de tap's de transformadores. El tercer paquete se llama DistriView y abarca flujos de potencia, análisis de corto circuito, coordinación de protecciones y arranque de motores para sistemas de distribución. Los dos complementos, Relay Database y Line Constants Program, se refieren a las bases de datos para relevadores de protección y

parámetros eléctricos de cables y líneas de transmisión. En la página consultada no se especifica un límite en el número de nodos, sin embargo, se ofrece una versión para colegios y universidades de los paquetes OneLinear, Power Flow y Line Constants Programs con un límite de hasta 10 Nodos y un costo de 200 dólares para la primera computadora y 100 dólares para las computadoras adicionales.

- d) Power World Simulator, cuya dirección es www.powerworld.com. Este simulador consta de tres paquetes. El primer paquete se llama Simulator OPF y contiene un simulador para determinar un despacho de carga óptimo. El segundo paquete se llama Simulator ATC y determina la cantidad máxima de MW que se puede transmitir entre dos zonas de un sistema eléctrico de potencia. El tercer paquete se llama Simulator VAST el cual realiza un análisis de estabilidad. Los tres paquetes anteriores pueden manejar hasta 60,000 nodos. Además de los anteriores se ofrece una versión para uso educativo que puede manejar hasta 12 nodos. Los precios que se manejan para estos simuladores varían de los 3,000 dólares hasta los 27,000 dólares.
- e) CYME. Realizado por la compañía CYME INTERNATIONAL, cuya dirección es www.cyme.com. CYME ofrece una serie de paquetes de análisis de los sistemas eléctricos de potencia integrado por módulos de simulación. Se encuentran disponibles los módulos de flujos de potencia, corto circuito, arranque de motores, estabilidad transitoria y análisis de armónicas. El módulo de flujos de potencia maneja los métodos desacoplado rápido y Gauss-Seidel, ofrece solución simultánea para áreas aisladas del sistema, contiene áreas de intercambio de potencia con generadores de control. No se especifica el número máximo de nodos que se manejan ni precio de los paquetes.
- f) NEPLAN. Realizado por Brusarello + Cott + Partner inc., cuya dirección es www.neplan.ch. NEPLAN ofrece los módulos de flujos de potencia, redes de distribución óptimas, corto circuito, análisis de frecuencia, arranque de motores, protección de sobrecorriente, estabilidad transitoria, análisis de confiabilidad, diámetros de cables y diseño de sistemas de tierras. En la página consultada no se especifica el número máximo de nodos ni tampoco precio.

Todos los paquetes anteriores operan bajo Windows, y ofrecen una variedad de ventanas en donde se realizan los diagramas unifilares de los sistemas en estudio, la lectura de datos puede ser realizada con un click del ratón sobre los elementos del diagrama. Ofrecen la posibilidad de copiar

y pegar elementos en el diagrama unifilar, y accesos a bases de datos de los equipos utilizados para la simulación.

1.2 Planteamiento del problema.

Los estudiantes de ingeniería eléctrica de la UNAM, no tienen acceso al aprendizaje mediante el uso de simuladores, como complemento a su desarrollo profesional.

Los laboratorios de cómputo de la UNAM, a los que tienen acceso los estudiantes de ingeniería eléctrica a nivel licenciatura, cuentan con los equipos de cómputo adecuados para manejar simuladores de gran capacidad desarrollados para operar bajo Windows, pero no se cuenta con dichos simuladores, al menos no en los laboratorios de cómputo a los que tienen acceso los estudiantes de ingeniería eléctrica a nivel licenciatura.

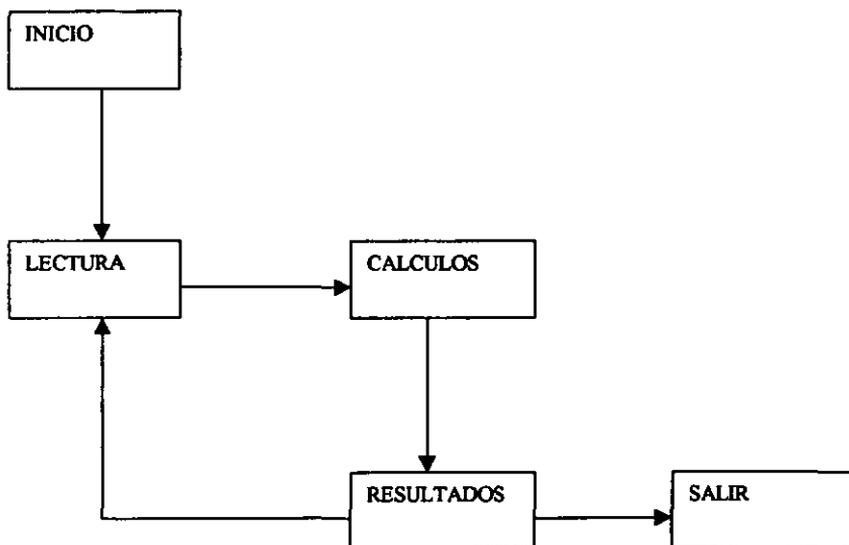
La UNAM a la fecha no ha adquirido simuladores que analicen Sistemas Eléctricos de Potencia que puedan utilizar los estudiantes de licenciatura que estudian esta área de la ingeniería, por otro lado, los estudiantes en su mayoría no pueden comprarlos.

Empresas como Peñoles, Cementos Cruz Azul, Conductores Monterrey o ConduMex, han adquirido paquetes de software para análisis de Sistemas Eléctricos de Potencia (según datos de las empresas que los elaboran). Los cuales son de uso cotidiano en dichas empresas.

1.3 Planteamiento de solución

1.3.1. Planteamiento General

Para diseñar el sistema que nos permitirá obtener las soluciones aquí planteadas, se tendrá en cuenta el planteamiento general de solución descrito a continuación mediante un diagrama de bloques.



Planteamiento General de Solución.

El planteamiento general de solución indica que es necesario diseñar dos pantallas, una para introducir los datos y la otra para leer los resultados. Esto para cada uno de los cuatro módulos de análisis que aquí se plantean.

El diagrama de bloques anterior indica que los datos se deben guardar en archivos desde la plataforma de programación visual. El archivo ejecutable que realiza el cálculo toma los datos de estos archivos para generar los resultados, estos resultados se almacenan en otros archivos. Finalmente los resultados son leídos desde la plataforma de programación visual y desplegados en pantallas diseñadas para tal fin.

Los archivos de datos que se generan desde la plataforma de programación visual, fue necesario adecuarlos a los formatos de lectura de datos de Fortran, dando el formato de los archivos desde Visual Basic.

El conjunto de programas y aplicaciones que se desarrollan en esta tesis toman el nombre de REDELEC, y esta compuesto por dos partes importantes:

- a) Algoritmos de solución.
- b) Programación bajo la plataforma de Windows

Los algoritmos de solución de cada módulo de análisis de REDELEC se programaron en lenguaje de alto nivel Fortran. En donde se realizó una estructura de entrada y salida de datos a través de archivos, con el fin de poder interactuar con el lenguaje visual.

El sistema REDELEC aquí propuesto tiene una presentación similar a las aplicaciones desarrolladas bajo la plataforma Windows, contiene barras de iconos horizontales y barras de iconos verticales, así como barras de menú y todas las características de manipulación de estos que posee Windows.

En esta tesis aprovechamos el potencial que ofrece Visual Basic para realizar programas en ambiente Windows, creando un sistema amigable, que permite generar los archivos de datos que requieren los programas elaborados en Fortran a partir de iconos e imágenes depositadas en las pantallas de diseño, de esta forma, la lectura de datos se realiza de manera dinámica mientras se realiza el dibujo del circuito correspondiente al Sistema Eléctrico de Potencia en estudio.

Los archivos de resultados generados a partir de los algoritmos programados en Fortran son llevados a ventanas de resultados de la aplicación REDELEC, en estas ventanas, se pueden observar de manera simultánea el total de resultados generados desde Fortran, permitiendo realizar comparaciones y análisis de los mismos.

1.3.2 Alcances de REDELEC.

El programa REDELEC tiene los siguientes alcances y características.

- REDELEC contiene cuatro módulos de análisis independientes uno del otro, cada módulo es completamente compatible con Microsoft Windows. Cada módulo de análisis cuenta con una pantalla de lectura de datos y una pantalla de cálculo y lectura de resultados.
- REDELEC permite realizar el dibujo del Sistema Eléctrico de Potencia que se desea analizar.

- El dibujo se realizará a partir de los iconos de la barra vertical. La lectura de datos se puede realizar desde el dibujo, desde la barra de iconos horizontal o desde la barra de menús. En cada dibujo se pueden manipular los datos y el propio dibujo con un click o doble click del ratón.
- Cada dibujo y sus datos correspondientes pueden guardarse en disco duro, y posteriormente puede cargarse nuevamente. En cada módulo se tiene la opción de imprimir tanto el dibujo como los resultados con la impresora predeterminada de Windows.
- Aunque REDELEC esta dimensionado para 100 nodos, se recomienda realizar análisis solo de redes de hasta 15 nodos (para cada módulo).
- El módulo de corrientes de malla, como su nombre lo dice, realiza el cálculo de las corrientes de malla de una red eléctrica, por el método del mismo nombre. La solución de la ecuación matricial que obtiene las corrientes de malla se resuelve por el método de descomposición LU.
- El módulo de Voltajes de nodo, similarmente, realiza el cálculo de los voltajes de nodo por el método con el mismo nombre. Y similarmente, la solución de la ecuación matricial con que se obtienen los voltajes de nodo se resuelve por el método de descomposición LU.
- El módulo de Líneas de transmisión realiza el cálculo de los parámetros eléctricos de operación de la línea, como lo es el voltaje y la corriente en el extremo receptor o transmisor, así como la eficiencia y la regulación de esta. En este módulo se clasifican las líneas como cortas, medianas y largas. Las consideraciones para cada caso se estudian en el capítulo 2 de esta tesis. No se incluye cálculo de resistencia, reactancia o capacitancia de la línea.
- El módulo de líneas de transmisión comprende el cálculo de los parámetros eléctricos de las líneas de transmisión utilizando como base los modelos PI o T. Al momento de realizar los cálculos de parámetros se le pregunta al usuario que modelo eligió para el cálculo.
- El último módulo es el de flujos de potencia. En este se realiza el cálculo de los flujos de potencia, mediante la aplicación del método iterativo de solución de Gauss-Seidel. La solución contempla límites de potencia reactiva en los nodos de generación.
- En cada módulo podemos pasar de las ventanas de lectura a las ventanas de resultados y viceversa. Esto crea la posibilidad de realizar correcciones en la lectura de datos, sin la necesidad de introducir nuevamente el total de estos, para después regresar a las ventanas de resultados y observar los nuevos valores.

Como se podrá observar, REDELEC es un programa sencillo de análisis de sistema eléctricos de potencia diseñado para los estudiantes de la facultad de ingeniería. Los paquetes de software comerciales y simuladores profesionales poseen gran potencial, se pueden realizar cálculos de redes de más de 10,000 nodos, se pueden realizar un sin fin de consideraciones de operación, algunos poseen los manejadores para plotters que permiten imprimir en hojas tamaño plano, algunos otros manejan bases de datos robustas, etc. Todo esto representa bastante tiempo para la captura de datos. Los estudiantes de ingeniería, para fines meramente académicos, solo requieren realizar cálculos de redes desde 2 hasta 20 nodos como máximo, y aplicar los conocimientos adquiridos durante la carrera. REDELEC ofrece estas características, un sistema pequeño con un acceso rápido de datos, además los módulos de cálculo contienen conceptos a nivel licenciatura que son estudiados durante la carrera de Ingeniero Mecánico Electricista.

En cuanto a la parte gráfica, los paquetes comerciales pueden realizar copiado y pegado de elementos, contienen amplias librerías, algunos poseen la facilidad de editar e integrar nuevos elementos a las librerías, algunos otros tienen elementos de color para fácil identificación, algunos poseen interfases con el programa de dibujo AutoCad, etc. REDELEC ofrece una forma sencilla de gráficos, cada elemento es único y posee características propias tanto de parámetros eléctricos como de datos que identifican dentro del dibujo. El poder realizar interfases gráficas con otros programas de dibujo, lleva a la necesidad de dominar el programa de dibujo.

1.4 Objetivos.

Con la presente tesis se pretende diseñar un sistema en computadora mediante el cual sea posible realizar análisis, de tal manera que se observe el comportamiento de los Sistemas Eléctricos de Potencia, analizados bajo diferentes condiciones de operación.

Se busca que el usuario del sistema, que se desarrolla en esta tesis, tenga la facilidad de obtener diversas soluciones, en tiempos relativamente cortos, lo que le permitirá entender de manera rápida como se comporta el sistema eléctrico al modificar las condiciones de operación.

En esta tesis se plantea la forma de resolver cuatro problemas básicos de los Sistemas Eléctricos de Potencia que son:

- a) Cálculo de corrientes de malla
- b) Cálculo de voltajes de nodo.
- c) Cálculo de líneas de transmisión aéreas.

d) Cálculo de flujos de potencia.

El sistema desarrollado está encaminado a facilitar al usuario el análisis de los sistemas eléctricos de potencia, con un sistema amigable y sencillo que permita un manejo fácil de los datos de las redes eléctricas en estudio.

El sistema esta enfocado principalmente con fines académicos, con lo que se pretende que los alumnos de la Facultad de Ingeniería cuenten con un paquete de software de características apropiadas para tales fines. A su vez, los alumnos tendrán la posibilidad de realizar mejoras y ampliaciones al sistema ya que los programas fuente estarán a disposición de quien esté interesado.

1.5 Organización del trabajo.

La presente tesis consta de 5 capítulos, de los cuales el primero corresponde a la introducción del trabajo, en donde se resalta la importancia que tiene el uso de simuladores para la solución de problemas relacionados con la ingeniería eléctrica, además de las limitaciones que tienen los alumnos de la facultad de ingeniería para accesos a este tipo de paquetes de software. En el capítulo 1 además se plantea el alcance del programa que se desarrolla en esta tesis.

El capítulo 2 contiene la representación y modelado de las redes de los Sistemas Eléctricos de Potencia. En donde se eligió el método de descomposición LU para resolver la ecuación matricial mediante la cual se obtienen las corrientes de malla y los voltajes de nodo de una red eléctrica.

Dentro del mismo capítulo 2 también se plantea la solución de la línea de transmisión, en donde se analiza la línea de transmisión como una red de bipuertos, ya sea que se trate de una línea de longitud corta, de longitud media o de longitud larga.

El capítulo 3 hace referencia a la solución del problema de los flujos de potencia, en donde se hace mención de tres métodos iterativos para la solución, que son :

- a) Método de Gauss-Seidel
- b) Método de Newton-Rhapson
- c) Método Desacoplado Rápido.

De los tres métodos de solución, se emplea el de Gauss-Seidel por la simplicidad con que se programa el algoritmo de solución en cualquier lenguaje de alto nivel.

El capítulo 4 hace mención de los antecedentes de programación visual requeridos para la programación del sistema aquí tratado, de tal manera que resulte amigable al usuario.

El capítulo 5 muestra la forma en que se utiliza el sistema desarrollado en esta tesis, para resolver cada uno de los cuatro problemas anteriormente mencionados. Se resuelve un caso base para cada caso, explicando detalladamente la forma en que se debe utilizar el sistema REDELEC.

Capítulo 2. Representación y modelado de las redes de los Sistemas Eléctricos de Potencia.

Un problema muy importante en el diseño y operación de un sistema de potencia es la conservación del voltaje dentro de los límites especificados en varios puntos del sistema. En este tema se describirán ecuaciones con las que se pueden calcular el voltaje, la corriente y la potencia en cualquier punto de la línea de transmisión, siempre que se conozcan estos valores en un punto; este punto es, por lo general, una de las terminales de la línea.

Sin embargo, el propósito de este tema no es solamente describir las ecuaciones pertinentes, sino también proveer la oportunidad de entender los efectos de los parámetros de la línea sobre los voltajes de barra y el flujo de potencia. De esta forma se puede ver la importancia del diseño de la línea.

En los sistemas de potencia modernos, las computadoras se están alimentando continuamente con datos de todo el sistema para propósitos de control e información. Los estudios de flujos de potencia realizados por una computadora, fácilmente suministran respuestas a preguntas concernientes al efecto de conectar o desconectar líneas del sistema o al de cambios en los parámetros de las líneas. Sin embargo, las ecuaciones que se desarrollan en este capítulo son importantes en el entendimiento global de lo que ocurre en un sistema y en el cálculo de la eficiencia de transmisión, las pérdidas y los límites de flujo de potencia en una línea bajo condiciones de estado estable y transitorio.

2.1. Representación de líneas de transmisión.

En este capítulo analizaremos la operación de líneas de transmisión balanceadas bajo condiciones de operación en estado estable. Se describirán las expresiones necesarias para conocer el voltaje y la corriente en cualquier punto a lo largo de la línea de transmisión. La línea será tratada como una red de dos puertos por lo que los parámetros ABCD se obtendrán a partir del circuito π y del circuito T equivalentes de la línea de transmisión.

Es conveniente para representar una línea de transmisión, considerarla como una red de dos puertos. En la figura 2.1. se muestra la representación de una red de dos puertos donde V_S e I_S son el voltaje y la corriente en el extremo transmisor y V_R e I_R son el voltaje y la corriente en el extremo receptor.

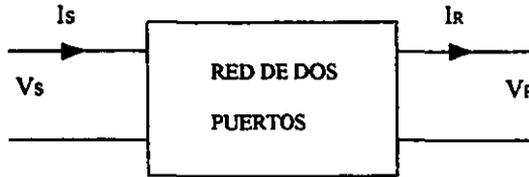


Figura 2.1. Representación de una red de dos puertos.

La relación entre las cantidades del extremo transmisor y el extremo receptor se expresa de la manera siguiente:

$$V_s = AV_R + BI_R \text{ Volts} \quad (2.1)$$

$$I_s = CV_R + DI_R \text{ Volts} \quad (2.2)$$

Expresando las ecuaciones 2.1 y 2.2 en forma matricial:

$$\begin{bmatrix} V_s \\ I_s \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} V_r \\ I_r \end{bmatrix} \quad (2.3)$$

Los parámetros ABCD dependen de las constantes R,L,C y G de la línea de transmisión. La conductancia G usualmente se desprecia para líneas de transmisión aéreas.

Los parámetros ABCD son números complejos, A y D son adimensionales e iguales entre sí, si la línea es la misma cuando se ve desde cada terminal. Las dimensiones de B y C son los ohms y los mhos o siemens, respectivamente. Las constantes se aplican a cualquier red lineal, pasiva y con cuatro terminales en dos lados, y cada lado tiene un par de ellas. A esta representación se le conoce como red de dos puertos

Las ecuaciones generales que relacionan el voltaje y la corriente para las líneas de transmisión establecen el hecho de que los cuatro parámetros de una línea de transmisión, están distribuidos uniformemente a lo largo de la línea. Se describirán más tarde estas ecuaciones generales pero, primeramente, se usarán los parámetros concentrados que dan resultados con una buena exactitud en líneas cortas y medias. Si la línea aérea se clasifica como corta, la capacitancia en derivación es tan pequeña que se puede omitir por completo con una mínima pérdida de exactitud y solo se requiere considerar la resistencia R y la inductancia L en serie para la longitud total de la línea.

Una línea de longitud media se puede representar con suficiente exactitud con R y L como parámetros concentrados, con la mitad de la capacitancia al neutro de la línea concentrada en cada terminal del circuito equivalente.

En lo que se refiere a la capacitancia, se consideran como cortas las líneas de 60 Hz de conductor abierto que tienen menos de 80 km de longitud. Las líneas de longitud media son las que están entre 80 km y 240 km de longitud. Las líneas que tienen más de 240 km requieren de cálculos en términos de constantes distribuidas si se necesita un alto grado de exactitud, aunque para algunos propósitos, se puede usar una representación de parámetros concentrados para líneas hasta de 320 km de largo.

Normalmente las líneas de transmisión se operan con cargas trifásicas balanceadas.

Aunque las líneas no estén espaciadas equilateralmente y no estén transpuestas, la asimetría resultante es ligera y se considera que las fases están balanceadas.

2.1.1. La línea de transmisión corta.

El circuito equivalente para una línea de transmisión corta se muestra en la figura 2.2, en donde I_S e I_R son las corrientes en los extremos generador y receptor, respectivamente, y V_S y V_R son los voltajes línea a neutro en esos mismos extremos. Resolviendo el circuito tenemos que:

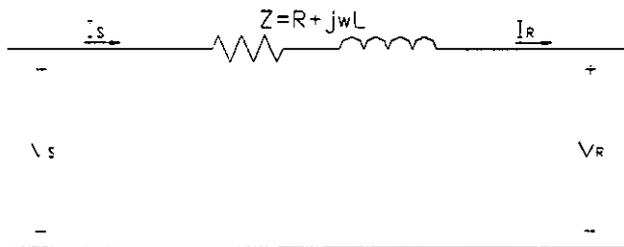


Fig.2.2. Circuito equivalente para una línea de transmisión corta.

$$I_S = I_R \tag{2.4}$$

$$V_S = V_R + I_R \times Z \tag{2.5}$$

Z es la impedancia en serie de secuencia positiva, V_S y V_R son los voltajes de línea a neutro de secuencia positiva, I_S e I_R son las corrientes de línea a neutro de secuencia positiva.

Para evitar que haya confusión entre la impedancia en serie total y la impedancia en serie por unidad de longitud, usaremos la siguiente nomenclatura.

$z=R+j\omega L$ Ω/m , impedancia en serie por unidad de longitud.

$y=G+j\omega C$ S/m, admitancia en derivación por unidad de longitud.

$Z=zl$ Ω , impedancia en serie total.

$Y=y l$ S, admitancia en derivación total.

l =longitud de la línea.

Observando las ecuaciones 2.4 y 2.5 para una línea corta se pueden deducir fácilmente los parámetros ABCD.

$$A=D=1 \text{ P.U}$$

$$B=Z \ \Omega$$

$$C=0 \text{ S}$$

Con base en la ecuación 2.3 se obtiene la representación matricial para una línea corta, a partir de las ecuaciones generales para este tipo de línea.

$$\begin{bmatrix} V_S \\ I_S \end{bmatrix} = \begin{bmatrix} 1 & Z \\ 0 & 1 \end{bmatrix} \begin{bmatrix} V_R \\ I_R \end{bmatrix}$$

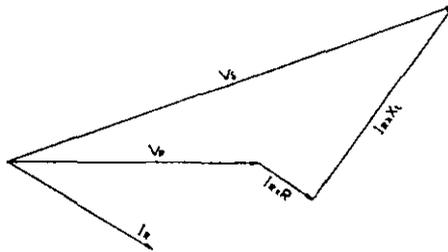
La regulación de voltaje de una línea de transmisión es el cambio en el voltaje en el extremo receptor de la línea cuando la carga varía desde que no hay carga hasta una carga máxima, a un determinado factor de potencia de la carga, mientras el voltaje en el extremo transmisor se mantiene constante.

La expresión para calcular la regulación de voltaje es la siguiente:

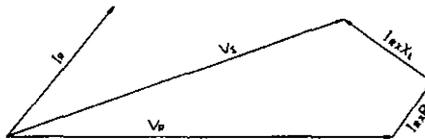
$$\%R = \frac{|V_{Rs/c} - V_{Rp/c}|}{|V_{Rp/c}|} \times 100$$

Donde $V_{Rs/c}$ es la magnitud del voltaje en el extremo receptor sin carga y $V_{Rp/c}$ es la magnitud del voltaje en el extremo receptor a plena carga.

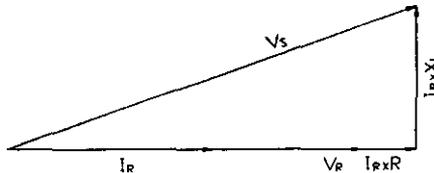
El efecto del factor de potencia de la carga sobre la regulación es ilustrado por los diagramas fasoriales mostrados en la figura 2.3.



a). Factor de potencia de la carga en atraso.



b). Factor de potencia de la carga en adelanto.



c). Factor de potencia de la carga unitario.

Fig.2.3. Diagramas fasoriales para una línea de transmisión.

Sin carga tenemos que, $I_{R\ s/c}=0$ y $V_{S\ p/c}=V_R\ s/c$

La peor regulación de voltaje ocurre para factores de potencia de la carga en atraso, debido a que el voltaje en el receptor sin carga excede con mucho al voltaje en el receptor a plena carga. Si se tienen factores de potencia de la carga en adelanto, la regulación de voltaje es menor e incluso puede ser negativa.

En la práctica el voltaje en las líneas de transmisión decrece cuando están excesivamente cargadas y se incrementa cuando están levemente cargadas.

Ejemplo 2.1. Cálculo de una línea de transmisión de longitud corta considerándola como una red de dos puertos. Tomado de la referencia 8.

Se requieren como datos la resistencia, la inductancia, la longitud de la línea, el voltaje y la corriente en el extremo receptor.

$$r=0.2374[\Omega/\text{km}]$$

$$l=8.6342 \times 10^{-4}[\text{H}/\text{km}]$$

$$Lg=30[\text{km}]$$

$$V_R=11.542[\text{kV}_{LN}]$$

$$I_R=0.0866-j0.052[\text{kA}]$$

$$\omega=2\pi f$$

$$f=60[\text{Hz}]$$

Impedancia y admitancia de la línea.

$$Z=R+j\omega L$$

$$Z=zl=(0.2374)(30)+(8.6342 \times 10^{-4})(30)(2)(3.1416)(60)=7.122+j9.7650[\Omega]$$

Parámetros ABCD de la línea.

$$A=1$$

$$B=Z$$

$$B=7.122+j9.7650[\Omega]$$

$$C=0[\text{S}]$$

$$D=A=1$$

Voltaje y corriente en el extremo transmisor de la línea.

$$V_S = AV_R + BI_R$$

$$I_S = CV_R + DI_R$$

$$V_S = 11.547 + (7.122 + j9.765)(0.0566 - j0.052)$$

$$V_S = 12.6715 + j0.4753[\text{kV}_{LN}]$$

$$I_S = ((-1.12896 \times 10^{-6} + j8.27652 \times 10^{-4})(189.2)) + ((0.9706 + j0.002688)(1.23354 + j0.1758))$$

$$I_S = I_R = 0.0866 - j0.0521[\text{kA}]$$

Potencia aparente en el extremo transmisor de la línea.

$$S_S = 3(V_S(I_S)^*)$$

I^* = Conjugado de la corriente.

$$S_S = 3((12.6715 + j0.4753)(0.0866 + 0.052j))$$

$$S_S = 3.22 + j2.1[\text{MVA}]$$

De la potencia aparente en el extremo transmisor se obtiene tanto la potencia real como la potencia reactiva en el extremo transmisor.

$$P_S = 3.22[\text{MW}]$$

$$Q_S = 2.1[\text{MVAR}]$$

Potencia aparente en el extremo receptor.

$$S_R = 3(V_R(I_R)^*)$$

$$S_R = 3((11.547)(0.0866 + 0.052j))$$

$$S_R = 3 + j1.8[\text{MVA}]$$

De la potencia aparente en el extremo receptor se obtiene tanto la potencia real como la potencia reactiva en el extremo receptor.

$$P_R = 3[\text{MW}]$$

$$Q_R = 1.8[\text{MVAR}]$$

Pérdidas de potencia real y reactiva de la línea.

$$p = P_S - P_R$$

$$p = 3.22 - 3 = 0.22[\text{MW}]$$

$$q = Q_S - Q_R$$

$$q = 2.1 - 1.8 = 0.3[\text{MVAR}]$$

Voltaje entre fases en el extremo transmisor de la línea.

$$|V_{S_{LN}}| = \sqrt{(12.6715)^2 + (0.4753)^2}$$

$$|V_{S_{LN}}| = 12.68[kV_{LN}]$$

$$V_{S_{LL}} = \sqrt{3}V_{S_{LN}}$$

$$V_{S_{LL}} = \sqrt{3}(12.68)$$

$$V_{S_{LL}} = 21.9631[kV_{LL}]$$

Voltaje entre fases en el extremo receptor de la línea.

$$|V_{R_{LN}}| = 11.547[kV_{LN}]$$

$$V_{R_{LL(p/c)}} = \sqrt{3}V_{R_{LN}}$$

$$V_{R_{LL(p/c)}} = \sqrt{3}(11.547)$$

$$V_{R_{LL(p/c)}} = 20[kV_{LL}]$$

Donde p/c significa a plena carga.

Voltaje entre fases en el extremo receptor de la línea sin carga.

$$|A| = 1$$

$$V_{R_{LL(s/c)}} = \frac{V_{S_{LL}}}{|A|}$$

$$V_{R_{LL(s/c)}} = \frac{21.9631}{1}$$

$$V_{R_{LL(s/c)}} = 21.9631[kV_{LL}]$$

Donde s/c significa sin carga.

Factor de potencia en el extremo transmisor de la línea.

$$|S_S| = \sqrt{(3)^2 + (1.8)^2}$$

$$|S_S| = 3.8443[MVA]$$

$$|f_{ps}| = \frac{P_S}{|S_S|}$$

$$|f_{ps}| = \frac{3.22}{3.8443}$$

$$|f_{ps}| = 0.8376$$

Eficiencia de la línea.

$$\%EF = \frac{P_R}{P_S} \times 100$$

$$\%EF = \frac{3}{3.22} \times 100$$

$$\%EF = 93.1677\%$$

Regulación de voltaje de la línea.

$$\%VR = \frac{\left| V_{R_{LL(S/C)}} \right| - \left| V_{R_{LL(P/C)}} \right|}{\left| V_{R_{LL(P/C)}} \right|} \times 100$$

$$\%VR = \frac{21.9631 - 20}{20}$$

$$\%VR = 9.8155\%$$

2.1.2. La línea de longitud media.

Los parámetros ABCD se pueden obtener a partir del modelo T o del modelo π de la línea, una vez que se analiza cada modelo.

Las ecuaciones generales para calcular el voltaje y la corriente en el extremo transmisor de la línea se obtienen a partir de ambos modelos, y los resultados que se obtienen varían muy poco.

2.1.2.1. Modelo π .

En los cálculos de una línea de longitud media se incluye la admitancia paralelo (generalmente capacitancia pura). La admitancia paralelo se divide en dos partes iguales, cada una de estas partes se coloca en los extremos generador y receptor de la línea, es así como se obtiene el llamado **circuito nominal π** .

En este capítulo obtendremos la solución de las líneas de longitud corta media y largas considerando la línea como una red de dos puertos, donde V_S e I_S son el voltaje y la corriente en el extremo generador y V_R e I_R son el voltaje y corriente en el extremo receptor.

La siguiente figura representa al circuito nominal pi de la línea de transmisión.

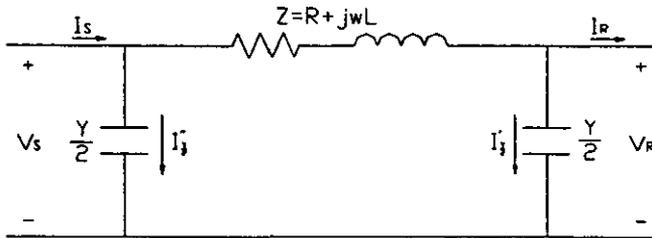


Fig.2.4 Circuito pi equivalente para una línea de transmisión de longitud media.

A continuación se obtienen del circuito mostrado en la fig. 2.4 las ecuaciones para calcular el voltaje y la corriente en el extremo transmisor de la línea de transmisión.

$$Z=R+jwL$$

$$Y_C=jwC$$

$$V_S=V_Z+V_R \quad (2.6)$$

$$(I')_{\frac{Y}{2}} = V_R \times \frac{Y}{2} \quad (2.7)$$

$$V_Z = Z \times I_Z \quad (2.8)$$

$$I_Z = I_R + (I')_{\frac{Y}{2}} \quad (2.9)$$

Sustituyendo la ecuación 2.9 en la ecuación 2.8

$$V_Z = Z \times (I_R + (I')_{\frac{Y}{2}}) \quad (2.10)$$

Sustituyendo la ecuación 2.7 en la ecuación 2.10, y la ecuación 2.10 en la ecuación 2.6.

$$V_S = Z \times I_R + \left(\frac{ZY}{2} + 1\right) \times V_R \quad (2.11)$$

$$I_S = I_Z + I''_{\frac{Y}{2}} \quad (2.12)$$

$$I_Z = I_R + (I')_{\frac{Y}{2}} \quad (2.13)$$

$$(I')_{\frac{Y}{2}} = V_R \times \frac{Y}{2} \quad (2.14)$$

$$I''_{\frac{Y}{2}} = V_S \times \frac{Y}{2} \quad (2.15)$$

$$I_S = I_R + I''_{\frac{Y}{2}} + (I')_{\frac{Y}{2}} \quad (2.16)$$

$$I_S = I_R \left(\frac{ZY}{2} + 1 \right) + \left(\frac{ZY}{4} + 1 \right) Y V_R \quad (2.17)$$

De las ecuaciones 2.11 y 2.17 se obtienen las ecuaciones generales para línea media circuito π .

$$V_S = AV_R + BI_R \text{ [Volts]}$$

$$I_S = CV_R + DI_R \text{ [A]}$$

$$B=Z[\Omega]$$

$$A=D=\left(\frac{ZY}{2} + 1\right) \text{ [Adimensional]}$$

$$C=\left(\frac{ZY}{4} + 1\right) Y \left[\frac{1}{\Omega} \right]$$

Ejemplo 2.2. Cálculo de una línea de transmisión de longitud media tomando como base el modelo π de la línea. Tomado de la referencia 9.

El cálculo de la línea se hará considerándola como una red de dos puertos.

Para el cálculo de la línea se requieren como datos la resistencia, la inductancia, la capacitancia, la longitud de la línea, el voltaje y la corriente en el extremo receptor.

$$r=0.032[\Omega/\text{km}]$$

$$l=9.2841 \times 10^{-4}[\text{H}/\text{km}]$$

$$c=0.01114088 \times 10^{-6}[\text{F}/\text{km}]$$

$$Lg=200[\text{km}]$$

$$V_R=189.2[\text{kV}_{LN}]$$

$$I_R=1.2335+j0.1758[\text{kA}]$$

$$\omega = 2\pi f$$

$$f = 60 \text{ [Hz]}$$

Impedancia y admitancia de la línea.

$$Z = R + j\omega L$$

$$Y = j\omega C$$

$$Z = zl = (0.032)(200) + (j9.2840658E-4)(200)(2)(3.1416)(60) = 6.4 + j70[\Omega]$$

$$Y = yl = (j0.011140879E-6)(200)(2)(3.1416)(60) = j8.4E-4[S]$$

Parámetros ABCD de la línea.

$$A = \frac{ZY}{2} + 1$$

$$A = \frac{(6.4 + j70)(j8.4 \times 10^{-4})}{2} + 1$$

$$A = 0.9706 + j0.002688$$

$$B = Z = 6.4 + j70[\Omega]$$

$$C = \left[\frac{ZY}{4} + 1 \right] Y$$

$$C = \left[\frac{(6.4 + j70)(j8.4 \times 10^{-4})}{4} + 1 \right] (j8.4 \times 10^{-4})$$

$$C = -1.12896 \times 10^{-6} + j8.27652 \times 10^{-4} \left[\frac{1}{\Omega} \right]$$

$$D = A = 0.9706 + j0.002688$$

Voltaje y corriente en el extremo transmisor de la línea.

$$V_S = AV_R + BI_R$$

$$I_S = CV_R + DI_R$$

$$V_S = ((0.9706 + j0.002688)(189.2)) + ((6.4 + j70)(1.23354 + j0.1758))$$

$$V_S = 179.2277 + j87.9813 [\text{kV}_{LN}]$$

$$I_S = ((-1.12896 \times 10^{-6} + j8.27652 \times 10^{-4})(189.2)) + ((0.9706 + j0.002688)(1.23354 + j0.1758))$$

$$I_S = 1.1966 + j0.33051 [\text{kA}]$$

Potencia aparente en el extremo transmisor de la línea.

$$S_S = 3(V_S(I_S)^*)$$

I^* = Conjugado de la corriente.

$$S_S = 3[(179.2277 + j87.9813)(1.1966 - 0.3305j)]$$

$$S_S = 730.6300 + j138.1255[\text{MVA}]$$

De la potencia aparente en el extremo transmisor se obtiene tanto la potencia real como la potencia reactiva en el extremo transmisor.

$$P_S = 730.63[\text{MW}]$$

$$Q_S = 138.1255[\text{MVAR}]$$

Potencia aparente en el extremo receptor.

$$S_R = 3(V_R(I_R)^*)$$

$$S_R = 3[(189.2)(1.23354 - 0.1758j)]$$

$$S_R = 700.15 - j99.7716[\text{MVA}]$$

De la potencia aparente en el extremo receptor se obtiene tanto la potencia real como la potencia reactiva en el extremo receptor.

$$P_R = 700.15[\text{MW}]$$

$$Q_R = -99.7716[\text{MVAR}]$$

Voltaje entre fases en el extremo transmisor de la línea.

$$|V_{S_{LN}}| = \sqrt{(179.223)^2 + (87.981)^2}$$

$$|V_{S_{LN}}| = 199.65[\text{kV}_{LN}]$$

$$V_{S_{LL}} = \sqrt{3}V_{S_{LN}}$$

$$V_{S_{LL}} = \sqrt{3}(199.65)$$

$$V_{S_{LL}} = 345.817[\text{kV}_{LL}]$$

Voltaje entre fases en el extremo receptor de la línea.

$$|V_{R_{LN}}| = 189.2[\text{kV}_{LN}]$$

$$V_{R_{LL(p/c)}} = \sqrt{3}V_{R_{LN}}$$

$$V_{R_{LL(p/c)}} = \sqrt{3}(189.2)$$

$$V_{R_{LL(p/c)}} = 327.7[\text{kV}_{LL}]$$

Donde p/c significa a plena carga.

Voltaje entre fases en el extremo receptor de la línea sin carga.

$$|A| = \sqrt{(0.9706)^2 + (0.002688)^2}$$

$$|A| = 0.9706$$

$$V_{R_{LL(s/c)}} = \frac{V_{S_{LL}}}{|A|}$$

$$V_{R_{LL(s/c)}} = \frac{345.817}{0.9706}$$

$$V_{R_{LL(s/c)}} = 356.293[\text{kV}_{LL}]$$

Donde s/c significa sin carga.

Factor de potencia en el extremo transmisor de la línea.

$$|S_S| = \sqrt{(730.630)^2 + (138.125)^2}$$

$$|S_S| = 743.570[\text{MVA}]$$

$$|f_{ps}| = \frac{P_S}{|S_S|}$$

$$|f_{ps}| = \frac{730.630}{743.570}$$

$$|f_{ps}| = 0.9826$$

Pérdidas de potencia real y reactiva.

$$p = P_S - P_R$$

$$p = 730.63 - 700.15$$

$$p = 30.48[\text{MW}]$$

$$q = Q_S - Q_R$$

$$q = 138.125 - (-99.771)$$

$$q = 237.897[\text{MVAR}]$$

Eficiencia de la línea.

$$\%EF = \frac{P_R}{P_S} \times 100$$

$$\%EF = \frac{700.15}{730.63} \times 100$$

$$\%EF = 95.83\%$$

Regulación de voltaje de la línea.

$$\%VR = \frac{\left| V_{R_{LL(S/C)}} \right| - \left| V_{R_{LL(P/C)}} \right|}{\left| V_{R_{LL(P/C)}} \right|} \times 100$$

$$\%VR = \frac{356.293 - 327.7}{327.7}$$

$$\%VR = 8.72$$

2.1.2.2. Modelo T.

Basándonos en el circuito T de la línea de transmisión podemos también obtener las ecuaciones generales que involucran a los parámetros ABCD. A continuación se obtienen al analizar el circuito que se muestra en la figura 2.5.

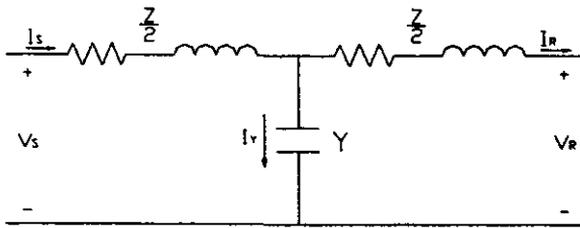


Fig.2.5. Circuito T equivalente para una línea de transmisión de longitud media.

$$V_s = V_z + V_Y \tag{2.18}$$

$$V_z = \left(\frac{Z}{2}\right) I_s \tag{2.19}$$

$$V_Y = \frac{I_Y}{Y} \tag{2.20}$$

$$I_s = I_r + I_Y \tag{2.21}$$

Despejando \$I_Y\$ de 2.20.

$$I_Y = V_Y Y \tag{2.22}$$

$$V_Y = I_r \left(\frac{Z}{2}\right) + V_r \tag{2.23}$$

Sustituyendo 2.19, 2.21 y 2.23 en 2.18.

$$I_Y = \left[\left(\frac{Z}{2}\right) I_r + V_r \right] Y \tag{2.24}$$

Sustituyendo 2.23 en 2.22.

$$V_s = \left(\frac{Z}{2}\right) (I_r + I_Y) + I_r \left(\frac{Z}{2}\right) + V_r \tag{2.25}$$

Sustituyendo 2.24 en 2.25.

$$V_s = \left(Z + \frac{Z^2 Y}{4} \right) I_r + \left(\frac{Z Y}{2} + 1 \right) V_r \tag{2.26}$$

$$I_S = I_R + I_Y \quad (2.27)$$

Sustituyendo 2.24 en 2.27.

$$I_S = \left(1 + \frac{ZY}{2}\right) I_R + YV_R \quad (2.28)$$

Expresando las ecuaciones para el cálculo de V_S e I_S como cuadripolo se tiene lo siguiente:

$$V_S = AV_R + BI_R \text{ [Volts]}$$

$$I_S = CV_R + DI_R \text{ [A]}$$

donde

$$B = Z + \frac{Z^2 Y}{4} \text{ } [\Omega]$$

$$A = D = \frac{ZY}{2} + 1 \text{ [Adimensional]}$$

$$C = Y \left[\frac{1}{\Omega} \right]$$

Donde Z es la impedancia e Y es la admitancia de la línea.

Ejemplo 2.3. Cálculo de una línea de transmisión de longitud media tomando como base el modelo T de la línea. Tomado de la referencia 9.

El cálculo de la línea se hará considerándola como una red de dos puertos, para el cálculo se requieren como datos, la resistencia, la inductancia, la capacitancia, la longitud de la línea, el voltaje y la corriente en el extremo receptor.

$$r=0.032[\Omega/\text{km}]$$

$$l=9.2841 \times 10^{-4}[\text{H}/\text{km}]$$

$$c=0.01114088 \times 10^{-6}[\text{F}/\text{km}]$$

$$Lg=200[\text{km}]$$

$$V_R=189.2[\text{kV}_{LN}]$$

$$I_R=1.2335+j0.1758[\text{kA}]$$

$$\omega=2\pi f$$

$$f= 60 \text{ [Hz]}$$

Impedancia y admitancia de la línea.

$$Z=R+j\omega L$$

$$Y=j\omega C$$

$$Z=zl=(0.032)(200)+(j9.2840658E-4)(200)(2)(3.1416)(60)=6.4+j70[\Omega]$$

$$Y=yl=(j0.011140879E-6)(200)(2)(3.1416)(60)=j8.4E-4[S]$$

Parámetros ABCD de la línea.

$$A = \frac{ZY}{2} + 1$$

$$A = \frac{(6.4 + j70)(j8.4 \times 10^{-4})}{2} + 1$$

$$A = 0.9706 + j0.002688$$

$$B = Z \left[\frac{ZY}{4} + 1 \right]$$

$$B = (6.4 + j70) \left[\frac{(6.4 + j70)(j8.4 \times 10^{-4})}{4} + 1 \right]$$

$$B = 6.212 + j68.98[\Omega]$$

$$C = Y = j8.4 \times 10^{-4}[S]$$

$$D = A = 0.9706 + j0.002688$$

Voltaje y corriente en el extremo transmisor de la línea.

$$V_S = AV_R + BI_R$$

$$I_S = CV_R + DI_R$$

$$V_S = ((0.9706 + j0.002688)(189.2)) + ((6.22 + j68.98)(1.23354 + j0.1758))$$

$$V_S = 179.174 + j86.690[kV_{LN}]$$

$$I_S = (j8.4 \times 10^{-4})(189.2) + ((0.9706 + j0.002688)(1.23354 + j0.1758))$$

$$I_S = 1.1968 + j0.330[kA]$$

Potencia aparente en el extremo transmisor de la línea.

$$S_S = 3(V_S(I_S)^*)$$

I^* = Conjugado de la corriente.

$$S_S = 3((179.174 + j86.69)(1.1968 - 0.330j))$$

$$S_S = 729 + j134.4 \text{ [MVA]}$$

De la potencia aparente en el extremo transmisor se obtiene tanto la potencia real como la potencia reactiva en el extremo transmisor.

$$P_S = 729 \text{ [MW]}$$

$$Q_S = 134.4 \text{ [MVAR]}$$

Potencia aparente en el extremo receptor.

$$S_R = 3(V_R(I_R)^*)$$

$$S_R = 3((189.2)(1.23354 - 0.1758j))$$

$$S_R = 700.155 - j99.784 \text{ [MVA]}$$

De la potencia aparente en el extremo receptor se obtiene tanto la potencia real como la potencia reactiva en el extremo receptor.

$$P_R = 700.155 \text{ [MW]}$$

$$Q_R = -99.784 \text{ [MVAR]}$$

Voltaje entre fases en el extremo transmisor de la línea.

$$|V_{S_{LN}}| = \sqrt{(179.174)^2 + (86.69)^2}$$

$$|V_{S_{LN}}| = 199.044 \text{ [kV}_{LN}\text{]}$$

$$V_{S_{LL}} = \sqrt{3} V_{S_{LN}}$$

$$V_{S_{LL}} = \sqrt{3}(199.65)$$

$$V_{S_{LL}} = 345.75 \text{ [kV}_{LL}\text{]}$$

Voltaje entre fases en el extremo receptor de la línea.

$$|V_{R,LN}| = 189.2 [kV_{LN}]$$

$$V_{R,LL(p/c)} = \sqrt{3} V_{R,LN}$$

$$V_{R,LL(p/c)} = \sqrt{3} (189.2)$$

$$V_{R,LL(p/c)} = 327.7 [kV_{LL}]$$

Donde p/c significa a plena carga.

Voltaje entre fases en el extremo receptor de la línea sin carga.

$$|A| = \sqrt{(0.9706)^2 + (0.002688)^2}$$

$$|A| = 0.9706$$

$$V_{R,LL(s/c)} = \frac{V_{S,LL}}{|A|}$$

$$V_{R,LL(s/c)} = \frac{344.75}{0.9706}$$

$$V_{R,LL(s/c)} = 355.19 [kV_{LL}]$$

Donde s/c significa sin carga.

Factor de potencia en el extremo transmisor de la línea.

$$|S_S| = \sqrt{(729)^2 + (134.4)^2}$$

$$|S_S| = 741.285 [MVA]$$

$$|f_{ps}| = \frac{P_S}{|S_S|}$$

$$|f_{ps}| = \frac{729}{741.285}$$

$$|f_{ps}| = 0.9834$$

Pérdidas de potencia real y reactiva.

$$p = P_S - P_R$$

$$p = 729 - 700.155$$

$$p = 28.845[\text{MW}]$$

$$q = Q_S - Q_R$$

$$q = 134.4 - (-99.784)$$

$$q = 234.184[\text{MVAR}]$$

Eficiencia de la línea.

$$\%EF = \frac{P_R}{P_S} \times 100$$

$$\%EF = \frac{700.155}{729} \times 100$$

$$\%EF = 0.96\%$$

Regulación de voltaje de la línea.

$$\%VR = \frac{\left| V_{R_{LL(S/C)}} \right| - \left| V_{R_{LL(P/C)}} \right|}{\left| V_{R_{LL(P/C)}} \right|} \times 100$$

$$\%VR = \frac{355.195 - 327.7}{327.7}$$

$$\%VR = 8.39$$

Si en la ecuación 2.3, se tienen como datos V_S e I_S , entonces se pueden conocer los valores de V_R e I_R resolviendo el sistema de ecuaciones representado por esta ecuación. Los valores de V_R e I_R estarían dados por:

$$V_R = \frac{D}{AD - CB} V_S - \frac{B}{AD - CB} I_S$$

$$I_R = \frac{-C}{AD - CB} V_S + \frac{A}{AD - CB} I_S$$

Ejemplo 2.4. Cálculo de una línea de transmisión de longitud media, tomando como base el modelo π , el cálculo de la línea se hará considerándola como una red de dos puertos.

Se requieren como datos la resistencia, la inductancia, la capacitancia, la longitud de la línea, el voltaje y la corriente en el extremo transmisor. Este ejemplo fue tomado de la referencia 9, citada en la bibliografía de esta tesis.

$$R=0.032[\Omega/\text{km}]$$

$$L=9.2841 \times 10^{-4}[\text{H}/\text{km}]$$

$$C=0.01114088 \times 10^{-6}[\text{F}/\text{km}]$$

$$L_g=200[\text{km}]$$

$$V_s=179.223+j87.981[\text{kV}_{\text{LN}}]$$

$$I_s=1.1966+j0.3305$$

Impedancia y admitancia de la línea.

$$Z=R+j\omega L$$

$$Y=j\omega C$$

$$Z=Zl=(0.032)(200)+(j9.2840658E-4)(200)(2)(3.1416)(60)=6.4+j70[\Omega]$$

$$Y=Yl=(j0.011140879E-6)(200)(2)(3.1416)(60)=j8.4E-4[\text{S}]$$

Parámetros ABCD de la línea.

$$A = \frac{ZY}{2} + 1$$

$$A = \frac{(6.4 + j70)(j8.4 \times 10^{-4})}{2} + 1$$

$$A = 0.9706 + j0.002688$$

$$B = Z \left[\frac{ZY}{4} + 1 \right]$$

$$B = (6.4 + j70) \left[\frac{(6.4 + j70)(j8.4 \times 10^{-4})}{4} + 1 \right]$$

$$B = 6.212 + j68.98$$

$$C = Y = j8.4 \times 10^{-4}$$

$$D = A = 0.9706 + j0.002688$$

Voltaje y corriente en el extremo transmisor de la línea.

$$V_R = \frac{D}{AD - CB} V_S - \frac{B}{AD - CB} I_S$$
$$I_R = \frac{-C}{AD - CB} V_S + \frac{A}{AD - CB} I_S$$

$$AD = (0.9706 + j0.002688)(0.9706 + j0.002688)$$

$$AD = 0.942 + j0.00522$$

$$CB = (-1.12896 \times 10^{-6} + j8.2752 \times 10^{-4})(6.4 + j70)$$

$$CB = -0.0579 + j0.00522$$

$$AD - CB = (0.942 + j0.00522) - (-0.0579 + j0.00522)$$

$$AD - CB = 1.0$$

$$V_R = (0.9706 + j0.002688)(179.227 + j87.981) - (6.4 + j70)(1.1966 + j0.3305)$$

$$V_R = 189.198 \text{ [kV}_{LN}]$$

$$I_R = - \left[-(1.12896 \times 10^{-6} + j8.27652 \times 10^{-4})(179.227 + j87.981) \right] +$$
$$+ (0.9706 + j0.002688)(1.1966 + j0.3305)$$

$$I_R = 1.2335 + j0.17576 \text{ [kA]}$$

2.1.3 La línea de transmisión larga.

Las líneas de transmisión cortas y medianas, hasta ahora las hemos representado con valores concentrados por unidad de longitud. Para el estudio de las líneas largas, estos parámetros concentrados los debemos tomar distribuidos uniformemente a lo largo de la línea, para de esta manera obtener la solución exacta.

Para tomar en cuenta esta característica, en la figura 2.6 se representa una sección infinitesimal de una línea de transmisión.

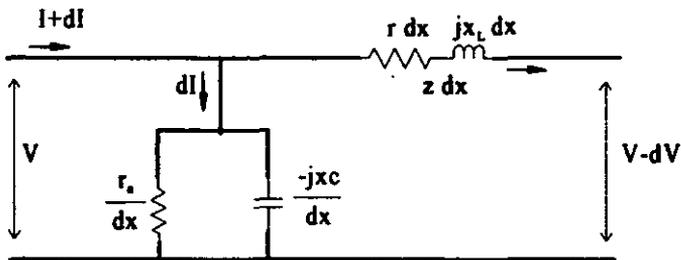


Figura 2.6 Sección infinitesimal de una línea de transmisión.

Donde:

- r = Resistencia efectiva por unidad de longitud.
- x_L = Reactancia inductiva por unidad de longitud.
- $z = r + j x_L$, impedancia en serie por unidad de longitud.
- r_a = Resistencia de aislamineto por unidad de longitud
- x_c = Reactancia capacitiva por unidad de longitud.
- z_1 = Impedancia en paralelo por unidad de longitud.
- y = Admitancia en paralelo por unidad de longitud.
- dx = Longitud de tramo diferencial de línea.
- $z dx$ = Impedancia en serie del tramo de línea de longitud dL
- z_1/dx = Impedancia en paralelo del tramo de línea de longitud dL

Aplicando las ecuaciones básicas de los circuitos eléctricos

$$V = V - dV + Izdx$$

$$dV = Izdx$$

$$\frac{dV}{dx} = Iz \tag{2.29}$$

$$dI = \frac{V}{z_1} dx$$

$$\frac{dI}{dx} = \frac{V}{z_1} \tag{2.30}$$

Las ecuaciones 2.29 y 2.30 son dos ecuaciones diferenciales homogéneas lineales de primer orden. Derivando ambas con respecto a x.

$$\frac{d^2V}{dx^2} = z \frac{dI}{dx} \quad (2.31)$$

$$\frac{d^2I}{dx^2} = \frac{1}{z_1} \left(\frac{dV}{dx} \right) \quad (2.32)$$

Substituyendo 2.30 en 2.31 y 2.29 en 2.32 respectivamente.

$$\frac{d^2V}{dx^2} = \frac{z}{z_1} V \quad (2.33)$$

$$\frac{d^2I}{dx^2} = \frac{z}{z_1} I \quad (2.34)$$

La ecuación 2.33 es una ecuación homogénea lineal de segundo orden con solo la variable V como incógnita cuya solución es del tipo:

$$V = A_1 e^{\gamma x} + A_2 e^{-\gamma x} \quad (2.35)$$

donde A_1 y A_2 son constantes de integración, y:

$$\gamma = \sqrt{zy}$$

La constante γ tienen como unidades $[m^{-1}]$ y se conoce como la constante de propagación, substituyendo 2.35 en 2.29 tenemos:

$$\frac{dV}{dx} = \gamma A_1 e^{\gamma x} - \gamma A_2 e^{-\gamma x} = zI$$

despejando I

$$I = \frac{A_1 e^{\gamma x} - A_2 e^{-\gamma x}}{z/\gamma} \quad (2.36)$$

Analizando el término z/γ tenemos:

$$\frac{z}{\gamma} = \frac{z}{\sqrt{zy}} = \sqrt{\frac{z}{y}} = Z_c$$

El término Z_c se conoce como impedancia característica. La ecuación 2.36 se convierte en:

$$I = \frac{A_1 e^{\gamma x} - A_2 e^{-\gamma x}}{Z_c}$$

Las constantes de integración A_1 y A_2 se evalúan a partir de las condiciones de frontera, si la distancia X se toma a partir del extremo receptor de la línea, para $X = 0$ las ecuaciones 2.35 y 2.36 quedan como:

$$V_R = V(0) = A_1 + A_2$$

$$I_R = I(0) = \frac{A_1 - A_2}{Z_c}$$

resolviendo el sistema de ecuaciones anterior:

$$A_1 = \frac{V_R + Z_c I_R}{2}$$

$$A_2 = \frac{V_R - Z_c I_R}{2}$$

Substituyendo A_1 y A_2 en las ecuaciones (2.35) y (2.36) tenemos:

$$V_s = \left(\frac{V_R + Z_c I_R}{2} \right) e^{\gamma x} + \left(\frac{V_R - Z_c I_R}{2} \right) e^{-\gamma x}$$

$$I_s = \left(\frac{V_R + Z_c I_R}{2Z_c} \right) e^{\gamma x} - \left(\frac{V_R - Z_c I_R}{2Z_c} \right) e^{-\gamma x}$$

Reordenando:

$$V_s = \left(\frac{e^{\gamma x} + e^{-\gamma x}}{2} \right) V_R - Z_c \left(\frac{e^{\gamma x} - e^{-\gamma x}}{2} \right) I_R$$

$$I_s = \frac{1}{Z_c} \left(\frac{e^{\gamma x} + e^{-\gamma x}}{2} \right) V_R - \left(\frac{e^{\gamma x} - e^{-\gamma x}}{2} \right) I_R$$

reorganizando las ecuaciones anteriores con las funciones hiperbólicas

$$V_s = \cosh(\gamma x) V_R + Z_c \sinh(\gamma x) I_R \quad (2.37)$$

$$I_s = \frac{1}{Z_c} \sinh(\gamma x) V_R + \cosh(\gamma x) I_R \quad (2.38)$$

Las ecuaciones 2.37 y 2.38 nos proporcionan los parámetros ABCD de la línea, la cual en forma matricial se representa como:

$$\begin{bmatrix} V \\ I \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} V_R \\ I_R \end{bmatrix} \quad (2.39)$$

La ecuación 2.39 proporciona el valor del voltaje y la corriente en un punto de la línea a una distancia x del extremo receptor, en función de las constantes de la línea y del voltaje y la corriente en el extremo receptor. En el extremo emisor donde $L = x$, tenemos que $V(x)=V_s$ y $I(x) = I_s$ Esto es:

$$\begin{bmatrix} V_s \\ I_s \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} V_R \\ I_R \end{bmatrix}$$

Donde:

$$A = D = \cosh(\gamma x)$$

$$B = Z_c \sinh(\gamma x)$$

$$C = \frac{1}{Z_c} \sinh(\gamma x)$$

La constante de propagación γ es un número complejo y podemos representar su parte real y su parte imaginaria como:

$$\gamma = \alpha + j\beta \text{ [m}^{-1}\text{]}$$

La parte real α se llama constante de atenuación y la parte imaginaria β se llama constante de fase. La constante de atenuación solo afecta la magnitud del voltaje y de la corriente a lo largo de la línea, mientras que la parte imaginaria produce una variación en el ángulo de fase.

Ejemplo 2.5. Cálculo de una línea de transmisión de longitud larga. El cálculo de la línea se hará considerándola como una red de dos puertos. Para el cálculo de la línea se requieren como datos la resistencia, la inductancia, la capacitancia, la longitud de la línea, el voltaje y la corriente en el extremo receptor. Tomado de la referencia 8.

$$r=0.0298[\Omega/\text{km}]$$

$$l=8.965736 \times 10^{-4}[\text{H}/\text{km}]$$

$$c=0.00823784 \times 10^{-6}[\text{F}/\text{km}]$$

$$L_g=320[\text{km}]$$

$$V_R=219.4[\text{kV}_{LN}]$$

$$I_R=0.4558-j0.076[\text{kA}]$$

$$\omega=2\pi f$$

$$f=60 \text{ [Hz]}$$

Impedancia y admitancia de la línea.

$$Z=R+j\omega L$$

$$Y=j\omega C$$

$$\omega=(2)(3.1416)(60)=376.992 \text{ [rad/seg]}$$

$$Z=Zl=(0.0298)(320)+(j8.967536E-4)(320)(376.992)=9.536+j108.16 \text{ } [\Omega]$$

$$Y=Yl=(j0.00823784E-6)(320)(376.992)=j9.938E-4 \text{ [S]}$$

Cálculo de la constante de propagación y la impedancia característica.

$$\gamma = \sqrt{ZY}$$

$$\gamma = \sqrt{(9.536 + j108.16)(j9.938E - 6)}$$

$$\gamma = 0.0144416 + j0.32817 \text{ [m}^{-1}\text{]}$$

γ , es la constante de propagación total (γ multiplicada por la longitud de la línea)

$$Z_C = \sqrt{\frac{Z}{Y}}$$

$$Z_C = \sqrt{\frac{9.536 + j108.16}{j9.938E - 4}}$$

$$Z_C = 329.68 - j14.51 \text{ } [\Omega]$$

Cálculo de los parámetros ABCB.

$$e^{\gamma} = \left(e^{0.0144416} \right) \left(e^{j0.3282} \right)$$

$$e^{-\gamma} = \left(e^{-0.0144416} \right) \left(e^{-j0.3282} \right)$$

$$e^{0.0144416} = 1.0145464$$

$$e^{-0.0144416} = 0.98566$$

$e^{j0.3282}$, es el ángulo en radianes

$e^{-j0.3282}$, es el ángulo en radianes

$$e^{\gamma} = 0.9604 + j0.32703$$

$$e^{-\gamma} = 0.93305 - j0.31772$$

$$A = \frac{e^{\gamma} + e^{-\gamma}}{2}$$

$$A = \frac{(0.9604 + j0.32703) + (0.93305 - j0.31772)}{2}$$

$$A = 0.946725 + j0.004655$$

$$B = Z_C \left[\frac{e^{\gamma} - e^{-\gamma}}{2} \right]$$

$$B = (329.68 - j14.51) \left[\frac{(0.9604 + j0.32703) - (0.93305 - j0.31772)}{2} \right]$$

$$B = 9.2012 + j106.2644 [\Omega]$$

$$C = \frac{1}{Z_C} \left[\frac{e^{\gamma} - e^{-\gamma}}{2} \right]$$

$$C = \frac{1}{328.68 - j14.51} \left[\frac{(0.9604 + j0.32703) - (0.93305 - j0.31772)}{2} \right]$$

$$C = -1.53605E-6 + j9.7788E-4 \left[\frac{1}{\Omega} \right]$$

$$D = A = 0.946725 + j0.004655$$

Voltaje y corriente en el extremo transmisor de la línea.

$$V_S = AV_R + BI_R$$

$$I_S = CV_R + DI_R$$

$$V_S = ((0.9467 + j0.004655)(219.4)) + ((9.2012 + j106.2644)(0.4558 - j0.076))$$

$$V_S = 219.9656 + j48.727 [\text{kV}_{LN}]$$

$$I_S = ((-1.53605E-6 + j9.7788E-4)(219.4)) + ((0.946725 + j0.004655)(0.4558 + j0.076))$$

$$I_S = 0.43197 + j0.1448 [\text{kA}]$$

Potencia aparente en el extremo transmisor de la línea.

$$S_S = 3(V_S(I_S)^*)$$

I^* = Conjugado de la corriente.

$$S_S = 3((219.9656 + j48.727)(0.43197 - 0.1448))$$

$$S_S = 306.23 - j32.4[\text{MVA}]$$

De la potencia aparente en el extremo transmisor se obtiene tanto la potencia real como la potencia reactiva en el extremo transmisor.

$$P_S = 306.23[\text{MW}]$$

$$Q_S = -32.4[\text{MVAR}]$$

Potencia aparente en el extremo receptor.

$$S_R = 3(V_R(I_R)^*)$$

$$S_R = 3((219.4)(0.4558 - 0.076))$$

$$S_R = 299.82 + j49.99[\text{MVA}]$$

De la potencia aparente en el extremo receptor se obtiene tanto la potencia real como la potencia reactiva en el extremo receptor.

$$P_R = 299.82[\text{MW}]$$

$$Q_R = 49.99[\text{MVAR}]$$

Voltaje entre fases en el extremo transmisor de la línea.

$$|V_{S_{LN}}| = \sqrt{(219.9656)^2 + (48.727)^2}$$

$$|V_{S_{LN}}| = 225.298[\text{kV}_{LN}]$$

$$V_{S_{LL}} = \sqrt{3}V_{S_{LN}}$$

$$V_{S_{LL}} = \sqrt{3}(225.298)$$

$$V_{S_{LL}} = 390.23[\text{kV}_{LL}]$$

Voltaje entre fases en el extremo receptor de la línea.

$$|V_{R,LN}| = 219.4[kV_{LN}]$$

$$V_{R,LL(p/c)} = \sqrt{3}V_{R,LN}$$

$$V_{R,LL(p/c)} = \sqrt{3}(219.4)$$

$$V_{R,LL(p/c)} = 380[kV_{LL}]$$

Donde p/c significa a plena carga.

Voltaje entre fases en el extremo receptor de la línea, sin carga.

$$|A| = \sqrt{(0.9467)^2 + (0.004655)^2}$$

$$|A| = 0.9467$$

$$V_{R,LL(s/c)} = \frac{V_{S,LL}}{|A|}$$

$$V_{R,LL(s/c)} = \frac{390.23}{0.9467}$$

$$V_{R,LL(s/c)} = 412.2[kV_{LL}]$$

Donde s/c significa sin carga.

Factor de potencia en el extremo transmisor de la línea.

$$|S_S| = \sqrt{(306.23)^2 + (32.4)^2}$$

$$|S_S| = 307.94[MVA]$$

$$|f_{ps}| = \frac{P_S}{|S_S|}$$

$$|f_{ps}| = \frac{306.23}{307.94}$$

$$|f_{ps}| = 0.9944$$

Pérdidas de potencia real y reactiva.

$$p = P_S - P_R$$

$$p = 306.23 - 299.82$$

$$p = 6.413[MW]$$

$$q = Q_S - Q_R$$

$$q = -32.4 - 49.99$$

$$q = -82.39[\text{MVAR}]$$

Eficiencia de la línea.

$$\%EF = \frac{P_R}{P_S} \times 100$$

$$\%EF = \frac{299.82}{306.23} \times 100$$

$$\%EF = 97.9\%$$

Regulación de voltaje de la línea.

$$\%VR = \frac{\left| V_{R_{LL(S/C)}} \right| - \left| V_{R_{LL(P/C)}} \right|}{\left| V_{R_{LL(P/C)}} \right|} \times 100$$

$$\%VR = \frac{412.1977 - 380}{380} \times 100$$

$$\%VR = 8.47$$

2.2. Diagramas unifilar, de impedancia y de reactancia.

En un diagrama unifilar se omite el neutro del circuito y se indican las partes que lo componen mediante símbolos estándar en lugar de sus circuitos equivalentes. No se muestran los parámetros del circuito y las líneas de transmisión se representan por una sola línea entre dos terminales. El diagrama unifilar indica por medio de una sola línea y por símbolos estándar la manera en que se conectan las líneas de transmisión con los aparatos de un sistema eléctrico.

El propósito de un diagrama unifilar es el de proporcionar información significativa acerca del sistema en forma concisa.

Es importante conocer los puntos en los que se aterriza el sistema, con el fin de calcular la corriente que fluye cuando ocurre una falla asimétrica que involucra a la tierra.

La figura 2.7 muestra el diagrama unifilar de un sistema de potencia sencillo. Dos generadores, uno aterrizado a través de una reactancia y el otro a través de una resistencia están conectados a

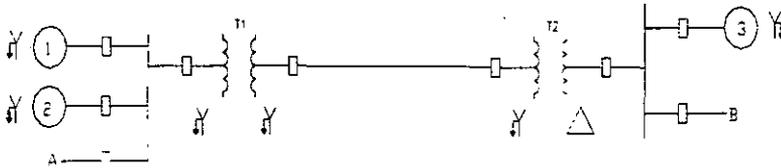


Figura 2.7. Diagrama unifilar de un sistema eléctrico de potencia.

una barra y por medio de un transformador elevador a una línea de transmisión. El otro generador aterrizado a través de una reactancia se conecta a una barra y por medio de un transformador se conecta al extremo opuesto de la línea de transmisión. Una carga está conectada en cada barra. Es común dar información sobre el diagrama, la cual esté relacionada con las cargas, los valores nominales de los generadores y transformadores y con las reactancias de los diferentes componentes del circuito.

2.3. Cantidades en por unidad.

El valor en por unidad de cualquier cantidad se define como la relación de la cantidad a su base y se expresa en forma decimal. La relación en por ciento es cien veces el valor en por unidad. Ambos métodos de cálculo, porcentual y en por unidad son más simples y más informativos que los volts, los amperes y los ohms reales.

El método en por unidad tiene una ventaja sobre el porcentual: el producto de dos cantidades expresadas en por ciento se debe dividir entre cien para obtener el resultado en por ciento.

El voltaje, la corriente, la potencia y la impedancia están relacionadas de tal manera que la selección de los valores base para cualquiera dos de ellos determina la base de los dos restantes. Si se especifican los valores base de corriente y de voltaje, se pueden determinar las bases de impedancia y de potencia.

Por lo general la potencia base en MVA y el voltaje base en kV son las cantidades seleccionadas para especificar las bases. Para sistemas monofásicos, o para los trifásicos, donde el término corriente se refiere a corriente de línea, el de voltaje se refiere a voltaje al neutro y el de potencia se refiere a potencia por fase, las siguientes fórmulas relacionan las diferentes cantidades, [4]:

$$A = \frac{kVA_{1\phi}}{kV_{LN}}$$

$$\Omega = \frac{V_{LN}}{A}$$

$$\Omega = \frac{(kV_{LN})^2}{MVA_{3\phi}}$$

En estas ecuaciones los subíndices 1ϕ y LN significan monofásico y línea a neutro respectivamente.

Debido a que los circuitos trifásicos balanceados se resuelven como si fueran una línea con un neutro de regreso, las bases para las diferentes cantidades en los diagramas de impedancias son los kilovoltamperes por fase y los kilovolts de línea a neutro. Generalmente, los datos que se dan son los kilovoltamperes o megavoltamperes trifásicos totales y los kilovolts línea a línea. Aunque se puede especificar un voltaje de línea como base, el voltaje que se requiere para la solución del circuito monofásico es el voltaje a neutro. El voltaje base a neutro es el voltaje base línea a línea dividido entre $\sqrt{3}$.

La impedancia base y la corriente base se pueden calcular directamente desde valores trifásicos de voltaje y potencia base. Si se interpretan la potencia base y el voltaje base como la potencia total de las tres fases y el voltaje base de línea a línea, se encuentra que:

$$A_{base} = \frac{kVA_{3\phi} \text{ base}}{\sqrt{3} \times kV_{LL} \text{ base}}$$

$$Z_{base} = \frac{(kV_{LL} / \sqrt{3})^2 \times 1000}{kVA_{3\phi} \text{ base} / 3}$$

$$Z_{base} = \frac{(kV_{LL})^2 \times 1000}{kVA_{3\phi} \text{ base}}$$

$$Z_{base} = \frac{(kV_{LL})^2}{MVA_{3\phi} \text{ base}}$$

Para expresar estas relaciones, se han usado subíndices con el fin de hacer énfasis en la distinción de trabajar con cantidades trifásicas y monofásicas. Las ecuaciones para calcular las bases de corriente e impedancia fueron tomadas de la referencia bibliográfica [4].

Se harán evidentes las ventajas de los cálculos en por unidad cuando los problemas por resolver sean muy complejos y, en particular, cuando incluyan transformadores. Generalmente, cuando se dan valores en por unidad de la impedancia y de otros parámetros de un componente del Sistema Eléctrico de Potencia sin especificar las bases, se supone que estos valores están dados en MVA y kV y son los valores nominales de ese componente.

2.3.1. Cambio de base de cantidades en por unidad.

En algunas ocasiones, la impedancia en por unidad de un componente del sistema se expresa sobre una base diferente de la seleccionada en el lugar donde se localiza el componente. Es necesario contar con medios para convertir las impedancias en por unidad de una base a otra, debido a que al hacer cálculos, todas las impedancias de un sistema se deben expresar sobre la misma impedancia base.

Para cambiar la impedancia en por unidad sobre una base dada a impedancia en por unidad sobre una nueva base, se aplica la siguiente ecuación, [4]:

$$Z_{\text{nueva P.U}} = Z_{\text{dada P.U}} \left(\frac{\text{kV}_{\text{dados base}}}{\text{kV}_{\text{nuevos base}}} \right)^2 \left(\frac{\text{kVA}_{\text{nuevos base}}}{\text{kVA}_{\text{dados base}}} \right)$$

La aplicación de la ecuación para el cambio de base consiste en cambiar el valor de la impedancia en por unidad de cualquier componente que se da sobre una base en particular a una nueva base.

En lugar de usar directamente la ecuación para el cambio de base, se puede también hacer el cambio de base convirtiendo primeramente a ohms el valor en por unidad sobre las bases dadas y dividiendo entre la nueva impedancia base.

2.4. Ecuaciones de redes y soluciones. Algoritmos en computadora para la construcción de las matrices de admitancias e impedancias de barra.

En el análisis de los Sistemas Eléctricos de Potencia a gran escala, la corriente que fluye a través de una componente de la red se puede relacionar con la caída de voltaje a través de ella mediante un parámetro de admitancia o de impedancia. La matriz de impedancias de nodo describe el

comportamiento en estado estable de todas las componentes que actúan juntas para formar el sistema, y se basa en el análisis nodal de las ecuaciones de la red.

La matriz de admitancias de nodo de un sistema típico de potencia es grande y puede obtenerse en forma sistemática de construcción de bloques. La aproximación de bloques de construcción es la entrada para el desarrollo de algoritmos que consideran los cambios en la red. Debido a que las matrices de redes son muy grandes, se requieren técnicas de ahorro para incrementar la eficiencia computacional de los programas que se emplean con el fin de resolver los problemas de sistemas de potencia.

En el estudio de flujos de potencia y de análisis de falla del sistema, se hace evidente la importancia del presente capítulo.

La matriz de impedancias de barra es más difícil de construir que la matriz de admitancias de barra, pero el contenido de la información de la matriz de impedancias de barra, es por mucho mayor al de Y_{barra} , debido a que cada elemento de la diagonal de Z_{barra} tiene características importantes de todo el sistema en forma de impedancia de Thévenin, en la barra correspondiente.

La matriz de admitancias de barra se usa ampliamente en los análisis de flujos de potencia. Por otro lado, la matriz de impedancias de barra es igualmente usada en los análisis de fallas de sistemas de potencia. De la misma manera, Y_{barra} y Z_{barra} juegan papeles importantes en el análisis de redes de sistemas de potencia.

2.4.1. Cálculo de los voltajes de nodo.

A continuación se analizará una red a manera de ejemplo, a partir de la cual construiremos la matriz de admitancias de barra, para posteriormente obtener los voltajes de barra(nodo).

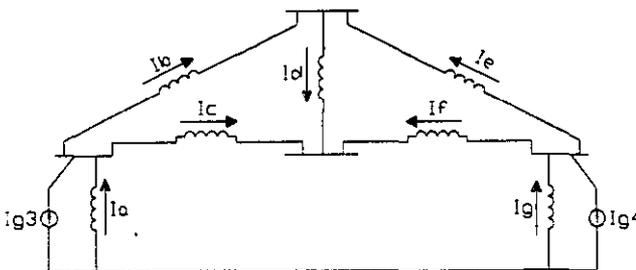


Figura 2.8. Diagrama de admitancias.

La configuración geométrica de las ramas recibe el nombre de topología y está dada por la llamada gráfica dirigida que se muestra en la figura 2.9. En esta gráfica, cada rama de la red de la figura 2.8 está representada entre sus nodos terminales por un segmento de línea recta con la punta de una flecha dirigida en el sentido de la corriente.

Cuando una rama se conecta a un nodo se dice que la rama y el nodo son incidentes. Las ramas de la gráfica que interconectan o alcanzan todos los nodos de la gráfica sin formar una trayectoria cerrada constituyen un árbol. En general una red tiene muchos arboles posibles, por que se tienen diferentes combinaciones de ramas que alcanzan a todos los nodos.

Una gráfica se puede describir en términos de una matriz de incidencia o de conexiones. Se pondrá un especial interés en la matriz de incidencia rama-nodo, que tiene una fila para cada rama y una columna para cada nodo con un elemento a_{ij} en la fila i y en la columna j en concordancia con la siguiente regla:

El elemento a_{ij} es igual a cero si la rama i no está conectada al nodo, es igual a 1 si la corriente en la rama i se aleja del nodo y es igual a -1 si la corriente en la rama i se dirige hacia el nodo.

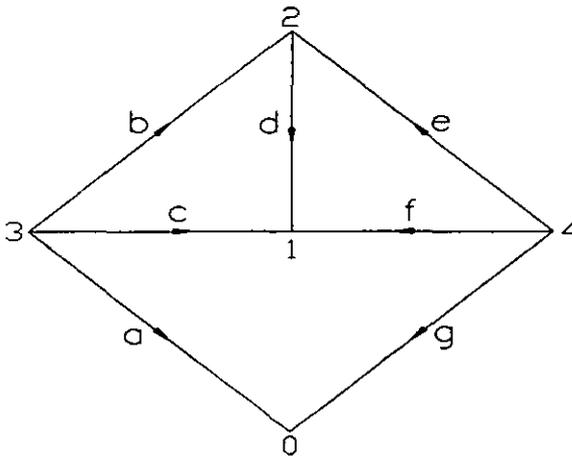


Figura 2.9. Gráfica dirigida de la figura 2.8

Esta regla formaliza el procedimiento general que se usa en la construcción de matrices de coeficientes dadas por las ecuaciones 2.40 y 2.41 para las ramas individuales de una red.

$$V_A = [1 \ -1] \begin{bmatrix} V_M \\ V_N \end{bmatrix} \quad (2.40)$$

$$\begin{bmatrix} 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 \end{bmatrix} \begin{bmatrix} V_M \\ V_N \\ V_P \\ V_Q \end{bmatrix} = \begin{bmatrix} V_A \\ V_B \end{bmatrix} \quad (2.41)$$

Generalmente se selecciona un nodo de referencia para los cálculos de redes. Por ejemplo, se obtiene la siguiente matriz rectangular de rama-nodo, si se selecciona el nodo O como el de referencia en la figura 2.9.

$$A_n = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & -1 & 1 & 0 \\ -1 & 0 & 1 & 0 \\ -1 & 1 & 0 & 0 \\ 0 & -1 & 0 & 1 \\ -1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.42)$$

En la matriz de admitancias elementales que se forma al observar la figura 2.8, cada rama de la red contribuye con un elemento de la diagonal principal que es igual al recíproco de su impedancia de rama.

La matriz de admitancias elementales se puede escribir de forma compacta para el caso más general como se muestra a continuación

$$Y_{pr} = V_{pr} I_{pr} \quad (2.43)$$

A los nodos que no son de referencia en una red se les llama con frecuencia nodos independientes o barras, y cuando se dice que la red tiene N barras, se quiere decir, por lo general, que hay N

nodos independientes sin incluir el de referencia. La matriz A_n tiene una dimensión fila columna de $B \times N$ elementos para cualquier red con B barras y N nodos sin considerar el de referencia.

Se puede observar que cada renglón de la ecuación (2.42) tiene dos elementos que no son cero, pero cuya suma si lo es, excepto para las filas a y g que sólo tienen un elemento que no es cero. Esto se debe a que las ramas a y g de la figura 2.8 tienen una terminal conectada al nodo de referencia cuya columna no se muestra.

El voltaje a través de cada rama se puede expresar como la diferencia de los voltajes en cada terminal de barra, medidos con respecto al nodo de referencia. Por ejemplo, en la figura 2.8 los voltajes en las ramas 1,2,3 y 4 con respecto al nodo de referencia O están designados por V_1 , V_2 , V_3 y V_4 respectivamente y así, las caídas de voltaje a través de las ramas están dadas por:

$$\begin{aligned}
 V_A &= V_3 \\
 V_B &= V_3 - V_2 \\
 V_C &= V_3 - V_1 \\
 V_D &= V_2 - V_1 \\
 V_E &= V_4 - V_2 \\
 V_F &= V_4 - V_1 \\
 V_G &= V_4
 \end{aligned}
 \tag{2.44}$$

Las ecuaciones anteriores expresadas en forma matricial se muestran a continuación:

$$\begin{bmatrix} V_A \\ V_B \\ V_C \\ V_D \\ V_E \\ V_F \\ V_G \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & -1 & 1 & 0 \\ -1 & 0 & 1 & 0 \\ -1 & 1 & 0 & 0 \\ 0 & -1 & 0 & 1 \\ -1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \end{bmatrix}
 \tag{2.45}$$

En forma matricial.

$$V_{pr} = A_n V_n$$

Donde la matriz de coeficientes A_n es la matriz de conexión rama-nodo de la red representada por la figura 2.8, V_{pr} es el vector columna $B \times 1$ de las caídas de voltaje de rama y V_n es el vector columna $N \times 1$ de los voltajes de barra medidos con respecto al nodo de referencia seleccionado. Las ecuaciones 2.40 y 2.41 son aplicaciones particulares de la ecuación 2.45 para ramas individuales. Además se puede observar que la ley de corrientes de Kirchoff en los nodos 1 y 4 de la figura 2.8 nos da:

$$\begin{bmatrix} 0 & 0 & -1 & -1 & 0 & -1 & 0 \\ 0 & -1 & 0 & 1 & -1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \\ i_d \\ i_e \\ i_f \\ i_g \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ i_3 \\ i_4 \end{bmatrix} \quad (2.46)$$

Donde i_3 e i_4 son las corrientes externas que se inyectan en los nodos 3 y 4 de la figura 2.8.

La matriz de coeficientes en este caso es $(A_n)^T$, de nuevo esto es ilustrativo de un resultado general que se aplica a cada red eléctrica, puesto que simplemente establece que la suma de todas las corrientes de rama que inciden en un nodo de una red es igual a la corriente que se inyecta en el nodo, en concordancia con la ley de corrientes de Kirchoff.

Por consiguiente se puede escribir:

$$(A_n)^T \times I_{pr} = I_g \quad (2.47)$$

Donde I_{pr} es el vector columna $B \times 1$ de las corrientes de rama e I_g es el vector columna $N \times 1$ con un elemento que no es cero para cada barra que tiene una fuente externa de corriente.

La matriz A_n describe por completo la topología de la red y es independiente de los valores particulares de los parámetros de la rama. Estos últimos los da la matriz de admitancias elementales. Por lo tanto, dos configuraciones diferentes de red que empleen las mismas ramas tendrán matrices A_n diferentes pero la misma Y_{pr} . Por otro lado, solamente se altera Y_{pr} y no A_n , si ocurren cambios en los parámetros de la rama manteniendo la misma configuración de la red.

Al multiplicar la ecuación 2.43 por $(A_n)^T$ se obtiene:

$$(A_n)^T \times Y_{pr} \times V_{pr} = (A_n)^T \times I_{pr} \quad (2.48)$$

$$(A_n)^T \times Y_{pr} \times V_{pr} = I_g \quad (2.49)$$

El lado derecho de la ecuación 2.48 es igual a I y al sustituir el valor de V_{pr} por la ecuación 2.45 se encuentra:

$$\left\{ (A_n)^T \times Y_{pr} \times A_n \right\} V_n = I_g \quad (2.50)$$

La ecuación 2.50 se puede escribir en una forma más concisa de la siguiente manera:

$$I_g = Y_{barra} \times V_n \quad (2.51)$$

donde la matriz de admitancias de barra Y_{barra} es una matriz de $N \times N$ dada por:

$$Y_{barra} = (A_n)^T \times Y_{pr} \times A_n \quad (2.52)$$

La matriz Y_{barra} tiene una fila y una columna para cada una de las N barras en la red y así, la forma estándar de las cuatro ecuaciones independientes del sistema de ejemplo de la figura 2.8 es:

$$[Y_{barra}] = \begin{bmatrix} Y_{11} & Y_{12} & Y_{13} & Y_{14} \\ Y_{21} & Y_{22} & Y_{23} & Y_{24} \\ Y_{31} & Y_{32} & Y_{33} & Y_{34} \\ Y_{41} & Y_{42} & Y_{43} & Y_{44} \end{bmatrix} \quad (2.53)$$

Cuando se especifican las cuatro corrientes I_1, I_2, I_3 e I_4 , que se inyectan a las barras, las cuatro incógnitas son los voltajes de barra V_1, V_2, V_3 y V_4 .

Con base en la figura 2.10, se obtiene para este caso la matriz de impedancias de rama y resolviendo el sistema de ecuaciones representado por la ecuación 2.55 se obtienen las corrientes de malla.

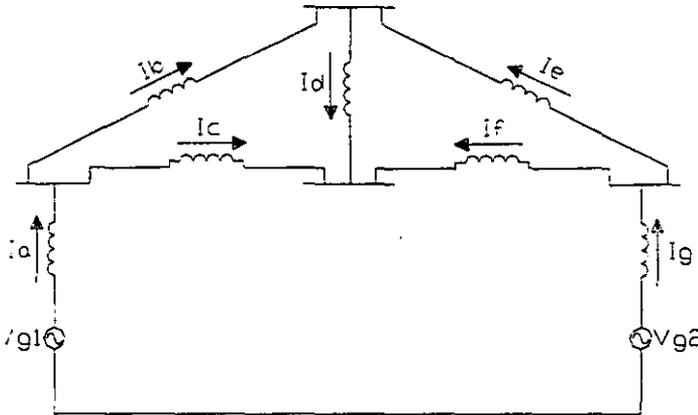


Figura 2.10. Diagrama de impedancias.

Las corrientes de malla se obtienen de manera análoga a los voltajes de nodo, sólo que para obtener en este caso la matriz de impedancias de barra Z_{barra} se requiere obtener a partir de la gráfica orientada de la red la matriz de conexión rama-malla.

La regla para obtener esta matriz es la siguiente:

Para cada rama, si forma parte de la malla considerada y tiene el mismo sentido que el asignado a la malla, el elemento correspondiente de la matriz es igual a 1; si forma parte de la malla y tiene sentido contrario al asignado a la malla, el elemento de la matriz es igual a -1; si no forma parte de la malla, el elemento correspondiente es igual a cero.

Para este caso, se obtiene de la gráfica orientada de la red, la matriz de conexión rama-malla y del diagrama de impedancias se obtiene la matriz de impedancias elementales. Con estos dos datos se obtiene la matriz de admitancias de barra.

La ecuación matricial para obtener Z_{barra} es la siguiente:

$$Z_{barra} = (A_m)^T \times Z_{pr} \times A_m \quad (2.54)$$

Para este caso, se obtiene de la gráfica orientada de la red, la matriz de conexión rama-malla y del diagrama de impedancias se obtiene la matriz de impedancias elementales. Con estos dos datos se obtiene la matriz de admitancias de barra.

La ecuación matricial para obtener Z_{barra} es la siguiente:

$$Z_{\text{barra}} = \left(A_m \right)^T \times Z_{\text{pr}} \times A_m \quad (2.54)$$

donde:

$[Z_{\text{barra}}]$ = Matriz de impedancias de barra

$[Z_{\text{pr}}]$ = Matriz de impedancias elementales

$[A_m]$ = Matriz de conexión rama-malla

Para obtener las corrientes de malla de una red eléctrica se requieren como datos los voltajes de generación y Z_{barra} . La expresión matricial que permite obtener las corrientes de malla es la siguiente:

$$V_g = Z_{\text{barra}} \times I_m \quad (2.55)$$

donde

V_g = Vector de voltajes de generación

I_m = Vector de corrientes de malla

La solución de los sistemas de ecuaciones mediante los cuales se obtienen tanto los voltajes de nodo como las corrientes de malla, se obtendrá aplicando el método de descomposición LU.

A continuación se desarrolla el método de descomposición LU que nos permite resolver un sistema de ecuaciones lineales.

2.4.3. Método de descomposición LU.

El método LU para la solución de sistemas de ecuaciones de $N \times N$ lineales es un método exacto de solución, el cual consiste en la descomposición de una matriz de coeficientes a en el producto de dos matrices triangulares L y U en donde la matriz L es una matriz triangular inferior y la matriz U es una matriz triangular superior con valores de uno en los elementos de su diagonal principal.

El método LU de solución de sistemas de ecuaciones se eligió por la sencillez con que se obtiene la solución de los sistemas de ecuaciones lineales en comparación con otros métodos.

La razón de que este método sea tan recomendado para ser programado, se debe a que se economiza el espacio de almacenamiento. No hay necesidad de almacenar los ceros ya sea en L o en U y los unos de la diagonal principal de U se pueden omitir. Al observar el desarrollo del método, se notará que después de que se usa cualquier elemento de la matriz de coeficientes a , a_{ij} nunca aparece de nuevo en las ecuaciones. Por lo tanto, su lugar en el arreglo original $N \times N$ puede ser utilizado para almacenar un elemento ya sea de la matriz L o de la matriz U.

Debido a que se pueden condensar las matrices L y U en un sólo arreglo, y almacenar sus elementos en el espacio de a , a este método se le ha dado en llamar "esquema compacto".

Una vez que se obtienen los valores de los elementos de las matrices L y U, se procede a obtener primero un vector solución auxiliar, esto se hace al resolver por sustitución hacia adelante el sistema de ecuaciones representado por la siguiente ecuación.

$$Lb=V$$

El vector b se obtiene fácilmente por sustitución hacia adelante. Si ya se conoce el vector b , se procede a resolver el sistema de ecuaciones, el cual se representa mediante la siguiente ecuación:

$$UI=b$$

Donde I es el vector solución del sistema de ecuaciones y U es una matriz triangular superior con valores de uno en su diagonal principal, es por eso que al realizar la sustitución hacia atrás, se obtienen fácilmente los elementos del vector solución del sistema de ecuaciones.

En seguida se obtendrán las ecuaciones generales mediante las cuales se calculan los elementos de las matrices L y U a partir de una matriz de coeficientes de 4×4 .

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} = \begin{bmatrix} 1 & U_{12} & U_{13} & U_{14} \\ 0 & 1 & U_{23} & U_{24} \\ 0 & 0 & 1 & U_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} L_{11} & 0 & 0 & 0 \\ L_{21} & L_{22} & 0 & 0 \\ L_{31} & L_{32} & L_{33} & 0 \\ L_{41} & L_{42} & L_{43} & L_{44} \end{bmatrix} \quad (2.56)$$

Multiplicando las filas de la matriz L por la primera columna de la matriz U en la ecuación 2.56, se obtiene:

$$\begin{aligned}L_{11} &= a_{11} & L_{31} &= a_{31} \\L_{21} &= a_{21} & L_{41} &= a_{41}\end{aligned}$$

La primera columna de la matriz L es la misma que la primera de la matriz a .

Ahora se multiplica la primera fila de la matriz L por las columnas de la matriz U en la ecuación 2.56.

$$\begin{aligned}L_{11} \times U_{12} &= a_{12} \\L_{11} \times U_{13} &= a_{13} \\L_{11} \times U_{14} &= a_{14}\end{aligned} \tag{2.57}$$

Despejando U_{1j} para cada ecuación en 2.57.

$$U_{12} = \frac{a_{12}}{L_{11}} \quad U_{13} = \frac{a_{13}}{L_{11}} \quad U_{14} = \frac{a_{14}}{L_{11}} \tag{2.58}$$

Así queda determinada la primera fila de la matriz U .

Este método se alterna entre obtener una columna de la matriz L y una fila de la matriz U . De manera que, a continuación se obtienen las ecuaciones de la segunda columna de la matriz L , multiplicando las filas de la matriz L por la segunda columna de U .

$$\begin{aligned}L_{11} \times U_{12} + L_{22} &= a_{22} \\L_{31} \times U_{12} + L_{32} &= a_{32} \\L_{41} \times U_{12} + L_{42} &= a_{42}\end{aligned} \tag{2.59}$$

Despejando en las ecuaciones 2.59, se obtienen las ecuaciones para calcular los valores de los elementos de la segunda columna de la matriz L .

$$\begin{aligned}L_{22} &= a_{22} - L_{21} \times U_{12} \\L_{32} &= a_{32} - L_{31} \times U_{12} \\L_{42} &= a_{42} - L_{41} \times U_{12}\end{aligned} \tag{2.60}$$

Procediendo de la misma forma, se obtienen las ecuaciones para calcular todos los valores de los elementos de las matrices L y U.

Elementos de la segunda fila de la matriz U.

$$U_{23} = \frac{a_{23} - L_{21} \times U_{13}}{L_{22}} \quad (2.61)$$

$$U_{24} = \frac{a_{24} - L_{21} \times U_{14}}{L_{22}}$$

Elementos de la tercera columna de la matriz L.

$$L_{33} = a_{33} - L_{31} \times U_{13} - L_{32} \times U_{23} \quad (2.62)$$

$$L_{43} = a_{43} - L_{41} \times U_{13} - L_{42} \times U_{23}$$

Elementos de la tercera fila de la matriz U.

$$U_{34} = \frac{a_{34} - L_{31} \times U_{14} - L_{32} \times U_{24}}{L_{33}} \quad (2.63)$$

Elementos de la cuarta columna de la matriz L.

$$L_{44} = a_{44} - L_{41} \times U_{14} - L_{42} \times U_{24} - L_{43} \times U_{34} \quad (2.64)$$

Las fórmulas generales para obtener los elementos de las matrices L y U correspondientes a la matriz de coeficientes a se muestran a continuación:

$$L_{ij} = a_{ij} - \sum_{k=1}^{j-1} L_{ik} \times U_{kj} \quad j \leq i, i=1,2,\dots,n$$

$$U_{ij} = \frac{a_{ij} - \sum_{k=1}^{i-1} L_{ik} \times U_{kj}}{L_{ii}} \quad i < j, j=2,3,\dots,n$$

Para $j=1$, la regla para L se reduce a.

$$L_{i1} = a_{i1}$$

Para $i=1$, la regla para U se reduce a.

$$U_{1j} = \frac{a_{1j}}{L_{11}}; \text{ donde } L_{11} = a_{11}$$

Ejemplo 2.6. Cálculo de los voltajes de nodo tomando como base la red eléctrica mostrada en la figura 2.8. Tomado de la referencia 4.

De acuerdo a la ecuación 2.51, para poder calcular los voltajes de nodo es necesario conocer la matriz de admitancias de barra y las corrientes que se inyectan a los nodos. Para calcular la matriz de admitancias de barra, es necesario obtener de la gráfica dirigida de la red que se esté analizando la matriz de conexión rama-nodo, y a partir de las ramas de la red obtener la matriz de admitancias de rama.

Matriz de conexión rama-nodo.

$$[A_n] = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & -1 & 1 & 0 \\ -1 & 0 & 1 & 0 \\ -1 & 1 & 0 & 0 \\ 0 & -1 & 0 & 1 \\ -1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Matriz transpuesta de la matriz de conexión rama-nodo.

$$[(A_n)^T] = \begin{bmatrix} 0 & 0 & -1 & -1 & 0 & -1 & 0 \\ 0 & -1 & 0 & 1 & -1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix}$$

Matriz de admitancias de rama.

$$[Y_{pr}] = \begin{bmatrix} -j0.8 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & -j4.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & -j4.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & -j8.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & -j5.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & -j2.5 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & -j0.8 \end{bmatrix}$$

Tomando como base la ecuación 2.52, se procede a calcular la matriz de admitancias de barra.

$$[A_n]^T [Y_{pr}] = \begin{bmatrix} 0.0 & 0.0 & j4.0 & j8.0 & 0.0 & j2.5 & 0.0 \\ 0.0 & j4.0 & 0.0 & -j8.0 & j5.0 & 0.0 & 0.0 \\ -j0.8 & -j4.0 & -j4.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & -j5.0 & -j2.5 & -j0.8 \end{bmatrix}$$

$$[Y_{barra}] = [A_n]^T [Y_{pr}] [A_n] = \begin{bmatrix} -j14.5 & j8.0 & j4.0 & j2.5 \\ j8.0 & -j17.0 & j4.0 & j5.0 \\ j4.0 & j4.0 & -j8.8 & 0.0 \\ j2.5 & j5.0 & 0.0 & -j8.3 \end{bmatrix}$$

Si se conoce la matriz de admitancias de barra y las corrientes que se inyectan a los nodos, el sistema de ecuaciones que se tendría que resolver es el que se representa mediante la ecuación 2.51.

$$\begin{bmatrix} -j14.5 & j8.0 & j4.0 & j2.5 \\ j8.0 & -j17.0 & j4.0 & j5.0 \\ j4.0 & j4.0 & -j8.8 & 0.0 \\ j2.5 & j5.0 & 0.0 & -j8.3 \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \end{bmatrix} = \begin{bmatrix} 0.0 \\ 0.0 \\ -j \\ -0.481 - j0.481 \end{bmatrix}$$

Una vez que se calcula la matriz de admitancias de barra, se puede resolver entonces el sistema de ecuaciones mediante el cual se calculan los voltajes de nodo. La solución de este sistema de ecuaciones se obtiene aplicando el método de descomposición LU.

Aplicando las ecuaciones generales obtenidas para este método se obtendrán los elementos de las matrices L y U. De acuerdo a las ecuaciones generales del método LU, la primera columna de la matriz de admitancias de barra corresponde con la primera columna de la matriz L.

$$L_{11} = -j14.5$$

$$L_{21} = j8.0$$

$$L_{31} = j4.0$$

$$L_{41} = j2.5$$

Elementos de la primera fila de la matriz U.

$$U_{12} = \frac{j8.0}{-j14.5} = -0.5517$$

$$U_{13} = \frac{j4.0}{-j14.5} = -0.27586$$

$$U_{14} = \frac{j2.5}{-j14.5} = -0.1724$$

El método se alterna entre el cálculo de los elementos de una columna de la matriz L y después una fila de la matriz U.

Elementos de la segunda columna de la matriz L.

$$L_{22} = j17.0 - L_{21}U_{12}$$

$$L_{22} = j17.0 - (j8.0)(-0.5517)$$

$$L_{22} = -j12.586$$

$$L_{32} = j4.0 - L_{31}U_{12}$$

$$L_{32} = j4.0 - (j4.0)(-0.5517)$$

$$L_{32} = j6.2068$$

$$L_{42} = j5.0 - L_{41}U_{12}$$

$$L_{42} = j5.0 - (j2.5)(-0.5517)$$

$$L_{42} = j6.37925$$

Elementos de la segunda fila de la matriz U.

$$U_{23} = \frac{j4.0 - L_{21}U_{13}}{L_{22}}$$

$$U_{23} = \frac{j4.0 - (j8.0)(-0.027586)}{-j12.586}$$

$$U_{23} = -0.493157$$

$$U_{24} = \frac{j5.0 - L_{21}U_{14}}{L_{22}}$$
$$U_{24} = \frac{j5.0 - (j8.0)(-0.1724)}{-j12.586}$$
$$U_{24} = -0.50685$$

Elementos de la tercera columna de la matriz L.

$$L_{33} = -j8.8 - L_{31}U_{13} - L_{32}U_{23}$$
$$L_{33} = -j8.8 - (j4.0)(-0.27586) - (j6.2068)(-0.493157)$$
$$L_{33} = -j4.6356$$
$$L_{43} = 0.0 - L_{41}U_{13} - L_{42}U_{23}$$
$$L_{43} = 0.0 - (j2.5)(-0.27586) - (j6.37925)(-0.493157)$$
$$L_{43} = j3.8356$$

Elementos de la tercera fila de la matriz U.

$$U_{34} = \frac{0.0 - L_{31}U_{14} - L_{32}U_{24}}{L_{33}}$$
$$U_{34} = \frac{0.0 - (j4.0)(-0.1724) - (j6.2068)(-0.50685)}{-j4.6356}$$
$$U_{34} = -0.8274$$

Elementos de la cuarta columna de la matriz L.

$$L_{44} = -j8.3 - L_{41}U_{14} - L_{42}U_{24} - L_{43}U_{34}$$
$$L_{44} = 0.0 - (j2.5)(-0.1724) - (j6.37925)(-0.50685) - (j3.8356)(-0.8274)$$
$$L_{44} = -j1.4621$$

El sistema de ecuaciones original se descompone en dos sistemas de ecuaciones muy fáciles de resolver debido a que ambos sistemas de ecuaciones involucran una matriz triangular superior(U) e inferior(L) respectivamente.

Cálculo del vector b por sustitución hacia adelante.

$$\begin{bmatrix} -j14.5 & 0.0 & 0.0 & 0.0 \\ j8.0 & -j12.586 & 0.0 & 0.0 \\ j4.0 & j6.207 & -j4.635 & 0.0 \\ j2.5 & j6.379 & j3.835 & -j1.462 \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix} = \begin{bmatrix} 0.0 \\ 0.0 \\ -j \\ -0.481 - j0.481 \end{bmatrix}$$

$$b_1 = b_2 = 0$$

$$-j4.635b_3 = -j$$

Despejando b3 en la ecuación anterior tenemos que:

$$b_3 = \frac{-j}{-j4.635} = 0.2157$$

Despejando b4 en la ecuación anterior tenemos que:

$$j3.835(0.2157) - j1.462b_4 = -0.481 - j0.481$$

$$b_4 = \frac{-0.481 - j0.481 - j3.835(0.2157)}{-j1.462} = 0.895 - j0.3289$$

Calculo de los voltajes de nodo por sustitución hacia atrás.

$$\begin{bmatrix} 1.0000 & -0.5520 & -0.2760 & -0.1724 \\ 0.0000 & 1.0000 & -0.4931 & -0.5068 \\ 0.0000 & 0.0000 & 1.0000 & -0.8274 \\ 0.0000 & 0.0000 & 0.0000 & 1.0000 \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \end{bmatrix} = \begin{bmatrix} 0.0 \\ 0.0 \\ -0.2157 \\ 0.895 - j0.3289 \end{bmatrix}$$

$$V_4 = 0.895 - j0.3289$$

$$V_3 - 0.8274(0.895 - j0.3289) = 0.2157$$

Despejando V3 en la ecuación anterior tenemos que:

$$V_3 = 0.9561 - j0.2722$$

$$V_2 - 0.4931(0.9561 - j0.2722) - 0.5068(0.8950 - j0.3289) = 0.0000$$

Despejando V_2 en la ecuación anterior tenemos que:

$$V_2 = 0.9250 - j0.3010$$

$$V_1 - 0.5517(0.9250 - j0.3010) - 0.2758(0.9561 - j0.2722) - 0.1724(0.8950 - j0.3289) = 0.0000$$

Despejando V_1 de la ecuación anterior tenemos que:

$$V_1 = 0.9283 - j0.2979$$

Ejemplo 2.7. Cálculo de corrientes de malla, tomando como base la red eléctrica mostrada en la figura 2.10. Tomado de la referencia 4.

Para poder calcular las corrientes de malla, es necesario calcular previamente la matriz de impedancias de barra. La matriz de impedancias de barra, se obtiene teniendo como datos la matriz de conexión rama-malla y la matriz de impedancias de rama.

Matriz de conexión rama-malla.

$$[A_m] = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ -1 & 0 & 1 \\ 1 & -1 & 0 \\ 0 & -1 & 0 \\ 0 & 1 & -1 \\ 0 & 0 & -1 \end{bmatrix}$$

Transpuesta de la matriz de conexión rama-malla.

$$[(A_m)^T] = \begin{bmatrix} 0 & 1 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & -1 & -1 \end{bmatrix}$$

La matriz de impedancias de rama para la red mostrada en la figura 2.9 es la siguiente:

$$[Z_{pr}] = \begin{bmatrix} j1.25 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & j0.25 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & j0.25 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & j0.125 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & j0.2 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & j0.4 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & j1.25 \end{bmatrix}$$

De acuerdo a la ecuación 2.54, la matriz de impedancias de barra se calcula de la siguiente manera:

$$[(A_m)^T][Z_{pr}] = \begin{bmatrix} 0.0 & j0.25 & -j0.25 & j0.125 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & -j0.125 & -j0.2 & j0.4 & 0.0 \\ j1.25 & 0.0 & j0.25 & 0.0 & 0.0 & 0.0 & -j0.4 & -1.25 \end{bmatrix}$$

$$[Z_{barra}] = [(A_m)^T][Z_{pr}][A_m] = \begin{bmatrix} j0.625 & -j0.125 & -j0.25 \\ -j0.125 & j0.725 & -j0.4 \\ -j0.25 & -j0.4 & j3.15 \end{bmatrix}$$

Las corrientes de malla se calculan tomando como base la ecuación 2.55. La solución del sistema de ecuaciones representado por esta ecuación, la obtendremos aplicando el método de descomposición LU.

A partir de las ecuaciones generales del método se obtendrán los elementos de las matrices L y U. De acuerdo a estas ecuaciones, la primera columna de la matriz de impedancias de barra corresponde con la primera columna de la matriz L.

$$L_{11} = j0.625$$

$$L_{21} = j0.125$$

$$L_{31} = -j0.25$$

Elementos de la primera fila de la matriz U.

$$U_{12} = \frac{-j0.125}{j0.625} = -0.2$$

$$U_{13} = \frac{-j0.25}{j0.625} = -0.4$$

El método se alterna entre obtener los elementos de una columna de la matriz L y los elementos de una fila de la matriz U, hasta que se obtienen los elementos de todas las columnas y todas las filas de ambas matrices.

Elementos de la segunda columna de la matriz L.

$$L_{22} = j0.725 - L_{21}U_{12}$$

$$L_{22} = j0.725 - (-j0.125)(-0.2)$$

$$L_{22} = j0.7$$

$$L_{32} = -j0.4 - L_{31}U_{12}$$

$$L_{32} = -j0.4 - (-j0.25)(-0.2)$$

$$L_{32} = -j0.45$$

Elementos de la segunda fila de la matriz U

$$U_{23} = \frac{-j0.4 - L_{21}U_{13}}{L_{22}}$$

$$U_{23} = \frac{-j0.4 - (-j0.125)(-0.4)}{j0.7}$$

$$U_{23} = -0.6428$$

Elementos de la tercera columna de la matriz L.

$$L_{33} = j3.15 - L_{31}U_{13} - L_{32}U_{23}$$

$$L_{33} = j3.15 - (-j0.25)(-0.4) - (-j0.45)(-0.6428)$$

$$L_{33} = j2.7607$$

El sistema de ecuaciones original mediante el cual se calculan las corrientes de malla, se descompone en dos sistemas de ecuaciones muy fáciles de resolver debido a que ambos sistemas de ecuaciones involucran una matriz triangular superior(U) e inferior(L) respectivamente. Estos sistemas se resuelven a continuación.

Resolviendo el siguiente sistema de ecuaciones por sustitución hacia adelante se obtiene el vector b.

$$\begin{bmatrix} j0.625 & 0.000 & 0.000 \\ -j0.125 & j0.700 & 0.000 \\ -j0.250 & -j0.450 & j2.761 \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} 0.000 \\ 0.000 \\ 0.649 - j0.601 \end{bmatrix}$$

$$b_1 = b_2 = 0$$

$$j2.761b_3 = 0.649 - j0.601$$

Despejando b_3 en la ecuación anterior tenemos que:

$$b_3 = \frac{0.649 - j0.601}{j2.761} = 0.2177 - j0.2351$$

Resolviendo el siguiente sistema de ecuaciones por sustitución hacia atrás se obtienen las corrientes de malla.

$$\begin{bmatrix} 1.000 & -0.200 & -0.400 \\ 0.000 & 1.000 & -0.643 \\ 0.000 & 0.000 & 1.000 \end{bmatrix} \begin{bmatrix} I_1 \\ I_2 \\ I_3 \end{bmatrix} = \begin{bmatrix} 0.000 \\ 0.000 \\ 0.218 - j0.235 \end{bmatrix}$$

$$I_3 = 0.218 - j0.235$$

$$I_2 - 0.643I_3 = 0.000$$

Despejando I_2 en la ecuación anterior tenemos que:

$$I_2 = 0.643(0.218 - j0.235)$$

$$I_2 = 0.1399 - j0.1511$$

$$I_1 - 0.2(0.1399 - j0.1511) - 0.4(0.2177 - j0.2351) = 0.0000$$

Despejando I_1 de la ecuación anterior tenemos que:

$$I_1 = 0.1151 - j0.1242$$

Capítulo 3. Flujos de potencia

Durante la década de los cuarentas y principio de los cincuentas, las compañías utilizaban modelos a escala conocidos como analizadores de redes, con el fin de analizar los flujos de potencia, o también conocidos como flujos de carga. Esta manera de analizar las redes eléctricas era demasiado laboriosa y complicada debido a que se necesitaba configurar el sistema para cada condición de carga y generación

Con el desarrollo de las técnicas analíticas y numéricas para la solución de los flujos de carga y la aparición de las computadoras digitales se dio una nueva alternativa para el análisis de los sistemas eléctricos la cual ofrecía una mayor rapidez en la obtención de resultados.

Actualmente se utilizan eficientemente los recursos de la computadora y la capacidad de acceso a archivos de datos extensos, esto hace que el ingeniero analice un gran número de casos, comenzando por las condiciones base del sistema.

Un sistema de potencia puede ser operado en varios estados, pero la mayor parte del tiempo se encuentra en su estado estable de operación. En este estado la frecuencia y el voltaje poseen valores constantes, resultado de un balance cuidadosamente mantenido entre la potencia real y reactiva demandada por la carga y entregada por la fuente de potencia.

3.1 Definición del problema de flujos de potencia

El estudio de flujos de potencia es la solución de una sistema de potencia bajo condiciones de estado estable de operación. Es decir, el problema de los flujos de potencia es el cálculo de la magnitud de los ángulos de fase y la potencia reactiva en los nodos de generación, de los voltajes y los ángulos de fase en los nodos de carga y de los flujos de potencia real y reactiva en las líneas de transmisión.

Los voltajes y las corrientes, en lugares específicos del sistema, se evalúan bajo condiciones predeterminadas, es decir, la generación de potencia real y la magnitud del voltaje en los nodos de generación, las características eléctricas del modelo de la red eléctrica y las potencias reales y reactivas en los nodos de carga son conocidas, siendo estos los datos que requiere el modelo de flujos de potencia para cada caso que es analizado.

3.2 Ecuaciones para el estudio de flujos de carga.

La solución de flujos de potencia para un sistema requiere principalmente de los siguientes pasos:

1. Establecer un circuito equivalente que represente a la red real, mediante un circuito monofásico de fase a neutro con todas las impedancias expresadas en por unidad.
2. Formular un modelo de red matemáticamente adecuado. El modelo deberá describir adecuadamente la relación entre voltajes y potencias en el sistema.
3. Especificar las limitaciones de potencia y voltaje que se deberá aplicar a ciertos nodos de la red.
4. Calcular numéricamente las ecuaciones de flujos de potencia sujeto a las limitaciones anteriores.
5. Cuando todos los voltajes en los nodos se hallan obtenido, debemos finalmente calcular los flujos de potencia actuales en todas las líneas de transmisión.

Para establecer un circuito equivalente los generadores pueden representarse como una fuente de fuerza electromotriz o como una fuente de corriente. Las cargas pueden representarse como impedancias pasivas, como una extracción de corriente, como una extracción de potencia real y reactiva o como una fuente de fuerza electromotriz. La representación más precisa depende del tipo de carga. Las líneas de transmisión se representan mediante su modelo equivalente.

Las ecuaciones de flujos de potencia pueden ser formuladas aplicando el método de mallas o el método de nodos. Pero desde el punto de vista de tiempo y memoria que requiere una computadora, el método de los voltajes de nodo resulta más económico.

Para describir las ecuaciones de flujos nos basaremos en un ejemplo desarrollado en la referencia [12].

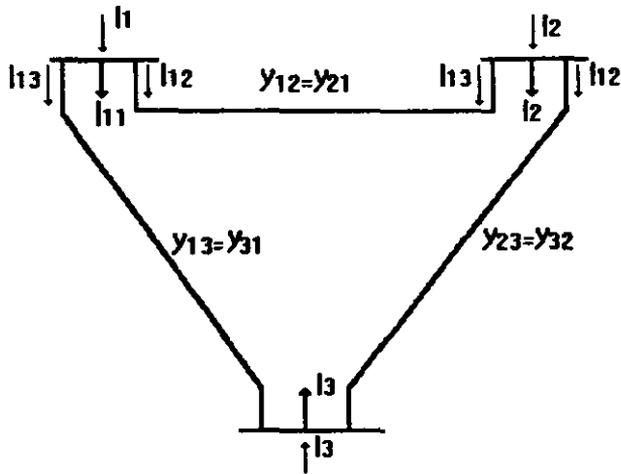


Figura 3.1- Diagrama unifilar

Donde:

y representa las admitancias de la línea de transmisión.

I representa las corrientes que circulan por los nodos de la red.

Para un sistema de tres nodos, como el de la figura 3.1 las ecuaciones de nodo son:

en el nodo 1.

$$I_1 = I_{11} + I_{12} + I_{13}$$

$$I_1 = V_1 y_{11} + (V_1 - V_2) y_{12} + (V_1 - V_3) y_{13}$$

$$I_1 = V_1 y_{11} + V_1 y_{12} - V_2 y_{12} + V_1 y_{13} - V_3 y_{13}$$

$$I_1 = V_1 (y_{11} + y_{12} + y_{13}) - V_2 y_{12} - V_3 y_{13} \quad (3.1)$$

donde y_{11} es la admitancia de carga en el nodo 1. Finalmente la ecuación (3.1) puede ser escrita

como:

$$I_1 = V_1 Y_{11} + V_2 Y_{12} + V_3 Y_{13} \quad (3.2)$$

donde:

$$Y_{11} = y_{11} + y_{12} + y_{13}$$

$$Y_{12} = -y_{12}$$

$$Y_{13} = -y_{13}$$

De una manera similar se obtienen las ecuaciones para los nodos 2 y 3.

$$I_2 = V_1 Y_{21} + V_2 Y_{22} + V_3 Y_{23} \quad (3.3)$$

$$I_3 = V_1 Y_{31} + V_2 Y_{32} + V_3 Y_{33} \quad (3.4)$$

Las ecuaciones (3.2), (3.3) y (3.4) se pueden representar en un arreglo matricial.

$$\begin{bmatrix} I_1 \\ I_2 \\ I_3 \end{bmatrix} = \begin{bmatrix} Y_{11} & Y_{12} & Y_{13} \\ Y_{21} & Y_{22} & Y_{23} \\ Y_{31} & Y_{32} & Y_{33} \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ V_3 \end{bmatrix}$$

donde la matriz cuadrada es conocida como la matriz de admitancias que en forma compacta se escribe como:

$$I_p = \sum_{q=1}^3 Y_{pq} V_q \quad \text{donde } p=1, 2, 3.$$

Y generalizando para un sistema de n nodos

$$I_p = \sum_{q=1}^n Y_{pq} V_q \quad \text{donde } p=1, 2, 3, \dots, n. \quad (3.5)$$

La matriz de admitancias Y_{pq} es simétrica, lo que permite que sólo se almacene una matriz triangular en la computadora, ahorrándose tiempo y memoria.

Si se conoce la interconexión entre varios nodos de un sistema dado, y el valor de la admitancia en cada interconexión, la matriz de admitancias puede formularse como sigue:

1. Los elementos de la diagonal principal Y_{pp} es la suma de las admitancias conectadas en el nodo p.
2. Los elementos fuera de la diagonal principal Y_{pq} son la admitancia entre los nodos p y q con signo negativo.

3.3 Métodos de solución para los flujos de potencia.

Para cada estudio se establecen las condiciones de funcionamiento, esto es, en cada nodo, excepto en uno, se especifica la potencia real. Para un nodo de carga la potencia real es negativa y se determina la potencia reactiva del nodo. Para un nodo de generación la potencia real es positiva,

por lo que se especifica el valor del voltaje de generación y los límites máximo y mínimo de potencia reactiva.

El nodo que se toma como referencia es llamado "oscilante" ó "slack", en este sólo se especifica el módulo y argumento del voltaje.

A partir de la ecuación (3.5) desarrollamos las ecuaciones para resolver el problema de los flujos de potencia.

$$I_p = \sum_{q=1}^n Y_{pq} V_q \quad \text{donde } p=1, 2, 3, \dots, n$$

Escribiendo esta ecuación de otra forma,

$$I_p = Y_{pp} V_p + \sum_{q=1, q \neq p}^n Y_{pq} V_q$$

Despejando V_p ,

$$V_p = \frac{I_p}{Y_{pp}} - \frac{1}{Y_{pp}} \sum_{q=1, q \neq p}^n Y_{pq} V_q \quad (3.6)$$

y recordando que,

$$\begin{aligned} V_p I_p^* &= P_p - jQ_p \\ \therefore I_p &= \frac{P_p - jQ_p}{V_p^*} \end{aligned} \quad (3.7)$$

entonces, sustituyendo (3.7) en (3.6)

$$V_p = \frac{1}{Y_{pp}} \left[\frac{P_p - jQ_p}{V_p^*} - \sum_{q=1, q \neq p}^n Y_{pq} V_q \right] \quad (3.8)$$

para el caso de un nodo de generación, en el cual el voltaje es fijo. De las ecuaciones (3.7) y (3.5) observamos también que:

$$\begin{aligned} P_p - jQ_p &= V_p^* \sum_{q=1}^n Y_{pq} V_q \\ \therefore Q_p &= -\text{Im} \left[V_p^* \sum_{q=1}^n Y_{pq} V_q \right] \end{aligned}$$

El ángulo del voltaje para el nodo de generación se obtiene de la ecuación (3.8), de la que únicamente se toma el ángulo del resultado numérico.

Podemos observar que las ecuaciones anteriores son algebraicas, por lo que representan un modelo de un sistema estático o un sistema operando en estado estable. Además las ecuaciones son no lineales por lo que se requiere utilizar métodos numéricos y la ayuda de una computadora digital para obtener la solución. Usualmente, en el análisis de redes eléctricas, las ecuaciones relacionan voltajes y corrientes, las ecuaciones anteriores relacionan voltajes y potencias por lo que no nos interesa la corriente. La frecuencia está implícita en los valores de las admitancias y se considera constante cuando el sistema opera en estado estable.

De las variables anteriores, las potencias real y reactiva de los nodos de carga están mas allá de nuestro control ya que son determinadas por el consumidor. Las variables que sobran son los voltajes en cada nodo, el ángulo de fase de los voltajes y la potencia real y reactiva en los nodos de generación.

En teoría de control existen variables de control y variables de estado. Las variables de control son utilizadas para manipular o controlar físicamente a las variables de estado. Por lo tanto, podemos decir que las potencias real y reactiva de los nodos de generación son variables de control. Mediante la manipulación de la potencia reactiva de los generadores se afecta la magnitud de los voltajes, similarmente, con la manipulación de la potencia real se afecta el ángulo de fase, por lo tanto, los voltajes y sus respectivos ángulos de fase de cada nodo se denominan variables de estado.

3.3.1 Método de Gauss-Seidel.

La ecuación (3.8) es no lineal y puede ser resuelta por medio de algún método numérico. Un método iterativo para obtener la solución de las ecuaciones de flujos de carga es el de Gauss-Seidel. En este método, el nuevo voltaje calculado V_p^{k+1} , inmediatamente reemplaza a V_p^k y se utiliza en la solución de la ecuación subsecuente.

Inicialmente se especifica un valor de V_p , se sustituye el conjugado de dicho valor obteniéndose un valor corregido de V_p . El valor corregido se utiliza para calcular el voltaje corregido del siguiente nodo. El proceso se repite sucesivamente en todos los nodos, excepto en el nodo oscilante, hasta completar la primera iteración. Después se repite todo el proceso hasta cumplir con un criterio de convergencia.

La ecuación general de flujos de potencia resultante del método de Gauss-Seidel es:

$$V_p^{k+1} = \frac{1}{Y_{pp}} \left[\frac{P_p - jQ_p}{(V_p^k)^*} - \sum_{q=1}^{p-1} Y_{pq} V_q^{k+1} - \sum_{q=p+1}^n Y_{pq} V_q^k \right] \quad (3.9)$$

Debemos tomar en cuenta que en un nodo de generación el voltaje es fijo por lo que se requiere calcular la potencia reactiva con la siguiente ecuación:

$$Q_p^{k+1} = -\text{Im} \left[(V_p^k)^* \left(Y_{pp} V_p^k - \sum_{q=1}^{p-1} Y_{pq} V_q^{k+1} - \sum_{q=p+1}^n Y_{pq} V_q^k \right) \right] \quad (3.10)$$

Donde el subíndice indica el número de nodo y el superíndice indica el número de iteración.

Si la potencia reactiva supera los límites establecidos, el nodo es tratado como un nodo de carga para la iteración en turno, lo que significa que se desconoce el módulo del voltaje y su ángulo y se conoce la potencia real y reactiva. El valor de la potencia reactiva se fija a partir del valor máximo, cuando se rebasa el límite, o del valor mínimo, cuando no se alcanza el límite inferior. En el caso de un nodo de generación, tanto la potencia real como la reactiva son tomadas como positivas, en contraste, en un nodo de carga las potencias real y reactiva se toman como negativas.

En la práctica el proceso de convergencia para el método de Gauss-Seidel es lento y puede ser acelerado si el voltaje de corrección durante iteraciones consecutivas es modificado como:

$$V_p^{k+1}(\text{acc}) = V_p^k + \alpha (V_p^{k+1} - V_p^k)$$

donde α es conocida como el factor de aceleración y es un número real. Según C.L. Wadhwa [12], generalmente se recomienda un valor de 1.6.

En resumen el procedimiento para la solución de las ecuaciones de flujos de carga por el método de Gauss-Seidel es el siguiente:

- Se asumen voltajes de $1 + 0j$ en el nodo de referencia y un valor apropiado de error ϵ para el criterio de convergencia.
- Se calcula la potencia reactiva y el ángulo del voltaje para los nodo de voltaje controlado ó el módulo del voltaje y su ángulo de fase para los nodos de carga utilizando la ecuación (3.9) y (3.10).
- Se calcula el cambio en el voltaje

$$V^k = V_p^{k+1} - V_p^k$$

- Una vez terminada la primera iteración, se toma la mayor desviación de ΔV^k y se compara con la tolerancia especificada, si ΔV^k es menor se finaliza y si no se repite la operación desde el paso 2.

3.3.2 Método de Newton-Raphson.

El Método de Newton-Raphson es otro método numérico utilizado para obtener la solución de las ecuaciones de flujos de potencia. Este método aproxima las ecuaciones simultáneas no lineales a un sistema simultáneo lineal utilizando la expansión en series de Taylor. Las derivadas parciales de orden superior a uno se desprecian en la serie de términos de la expansión de Taylor.

Se puede iniciar con iteraciones por el método de Gauss-Seidel para fijar un buen valor inicial. Estos voltajes iniciales son utilizados para calcular la potencia real P en todos los nodos excepto en el nodo de referencia y la potencia reactiva Q en todos los nodos donde debe ser especificada. Las diferencias entre valores especificados y calculados se emplean para determinar las correcciones en los voltajes de los nodos. El proceso se repite hasta que los valores especificados de P y Q o $|V|^2$ en todos los nodos sean menores que el índice de precisión indicado.

En el nodo p se tiene:

$$P_p - jQ_p = V_p \cdot \sum_{q=1}^n Y_{pq} V_q \quad (3.11)$$

donde

$$V_p = a_p + jb_p \quad (3.12)$$

$$y_{pq} = G_{pq} + jB_{pq} \quad (3.13)$$

Reemplazando (12) y (13) en (3.11)

$$P_p - jQ_p = (a_p - jb_p) \sum_{q=1}^n (G_{pq} - jB_{pq})(a_q + jb_q)$$

Separando la parte real y la parte imaginaria,

$$P_p = \sum_{q=1}^n \{ a_p (a_q G_{pq} - b_q B_{pq}) + b_p (b_q G_{pq} - a_q B_{pq}) \} \quad (3.14)$$

$$Q_p = \sum_{q=1}^n \{ b_p (a_q G_{pq} - b_q B_{pq}) + a_p (b_q G_{pq} - a_q B_{pq}) \} \quad (3.15)$$

donde la parte real es P_p y la parte imaginaria es Q_p .

El voltaje en el nodo oscilante o de referencia es:

$$|V_p|^2 = a_p^2 + b_p^2$$

Retomando el ejemplo de la figura 1, en donde el nodo 1 es de referencia, el nodo 2 es de carga y el nodo 3 es de generación, la aproximación lineal de las ecuaciones simultáneas no lineales es:

$$\begin{bmatrix} \Delta p_2 \\ \Delta p_3 \\ \Delta Q_2 \\ \Delta |V_3|^2 \end{bmatrix} = \begin{bmatrix} \frac{\partial P_2}{\partial a_2} & \frac{\partial P_2}{\partial a_3} & \frac{\partial P_2}{\partial b_2} & \frac{\partial P_2}{\partial b_3} \\ \frac{\partial P_3}{\partial a_2} & \frac{\partial P_3}{\partial a_3} & \frac{\partial P_3}{\partial b_2} & \frac{\partial P_3}{\partial b_3} \\ \frac{\partial Q_2}{\partial a_2} & \frac{\partial Q_2}{\partial a_3} & \frac{\partial Q_2}{\partial b_2} & \frac{\partial Q_2}{\partial b_3} \\ \frac{\partial |V_3|^2}{\partial a_2} & \frac{\partial |V_3|^2}{\partial a_3} & \frac{\partial |V_3|^2}{\partial b_2} & \frac{\partial |V_3|^2}{\partial b_3} \end{bmatrix} \begin{bmatrix} \Delta a_2 \\ \Delta a_3 \\ \Delta b_2 \\ \Delta b_3 \end{bmatrix}$$

El sistema anterior se puede escribir únicamente indicando las matrices siguientes:

$$[B] = [J] [C]$$

La matriz cuadrada de derivadas parciales se llama Jacobiana, esta se obtiene a partir de las derivadas parciales de las ecuaciones (3.14) y (3.15), después se substituyen los valores correspondientes de la iteración k-ésima de a_p^k y b_p^k .

La matriz [B] se obtiene de las ecuaciones (3.16), (3.17) y (3.18) siguientes, los valores calculados se obtienen de la substitución de los valores de a_p^k y b_p^k de la k-ésima iteración en las ecuaciones (3.14) y (3.15), y los valores especificados son los datos del nodo p.

$$\Delta P_p = P_{p,com} - P_{p,cal} \quad (3.16)$$

$$\Delta Q_p = Q_{p,com} - Q_{p,cal} \quad (3.17)$$

$$\Delta |V_p|^2 = |V_{p,com}|^2 - |V_{p,cal}|^2 \quad (3.18)$$

La matriz [C] es la única incógnita, esta puede obtenerse utilizando un método estándar, como lo es el invertir el Jacobiano. La solución del sistema de ecuaciones nos proporciona los valores de Δa_p^k y Δb_p^k por lo tanto la solución final de la iteración es :

$$a_2^{k+1} = a_2^k + \Delta a_2^k$$

$$a_3^{k+1} = a_3^k + \Delta a_3^k$$

**ESTA TESIS NO SALE
DE LA BIBLIOTECA**

$$b_2^{k+1} = b_2^k + \Delta b_2^k$$

$$b_3^{k+1} = b_3^k + \Delta b_3^k$$

Con los valores anteriores de Δa_p^{k+1} y Δb_p^{k+1} el proceso se repite hasta que el elemento de mayor valor de la matriz [B] sea menor que la tolerancia especificada, o el elemento de mayor valor de la matriz [C] sea menor que la tolerancia especificada.

3.3.3 Método desacoplado rápido.

El método desacoplado rápido es una extensión del método de Newton-Raphson formulado en coordenadas polares, el cual resulta en un algoritmo rápido para la solución de flujos de potencia.

Sabemos que:

$$P_p - jQ_p = V_p \sum_{q=1}^n Y_{pq} V_q \quad (3.19)$$

Además, el voltaje y la admitancia en coordenadas polares se expresan como:

$$V_p = |V_p| e^{j\theta_p} \quad (3.20)$$

$$Y_{pq} = |Y_{pq}| e^{-j\theta_{pq}} \quad (3.21)$$

sustituyendo (3.20) y (3.21) en (3.19),

$$P_p + jQ_p = \sum_{q=1}^n |V_p| |V_q| |Y_{pq}| e^{-j(\theta_{pq} + \delta_p + \delta_q)}$$

separando las potencias real y reactiva:

$$P_p = \sum_{q=1}^n |V_p| |V_q| |Y_{pq}| \cos(\theta_{pq} + \delta_p + \delta_q) \quad (3.22)$$

$$Q_p = \sum_{q=1}^n |V_p| |V_q| |Y_{pq}| \sen(\theta_{pq} + \delta_p - \delta_q) \quad (3.23)$$

las ecuaciones (3.22) y (3.23) se pueden escribir en forma matricial de la siguiente manera:

$$\begin{bmatrix} \Delta P \\ \Delta Q \end{bmatrix} = \begin{bmatrix} H & N \\ M & L \end{bmatrix} \begin{bmatrix} \Delta \delta \\ \Delta |V|/|V| \end{bmatrix} \quad (3.24)$$

donde H, N, M y L son elementos de la matriz Jacobiana.

La primera consideración del método desacoplado rápido es que los cambios que suceden en la potencia real (ΔP) son sensitivos principalmente a los cambios de los ángulos de fase, y menos

sensitivos a los cambios en magnitud del voltaje. Similarmente, los cambios en la potencia reactiva (ΔQ) son sensitivos principalmente a los cambios de la magnitud del voltaje y menos sensitivos a los cambios de los ángulos de fase. Con las consideraciones anteriores se pueden despreciar los elementos N y M de la matriz Jacobiana y la ecuación (24) desacoplarse en:

$$[\Delta P] = [H][\Delta \delta] \quad (3.25)$$

$$[\Delta Q] = [L][\Delta |V|/|V|] \quad (3.26)$$

donde los elementos de las matrices H y L son:

$$H_{pq} = \frac{\partial P_p}{\partial \delta_q} = |V_p V_q| [-B_{pq} \cos(\delta_p - \delta_q) + G_{pq} \sin(\delta_p - \delta_q)]$$

$$L_{pq} = \frac{\partial Q_p}{\partial V_q} = |V_p V_q| [-B_{pq} \cos(\delta_p - \delta_q) + G_{pq} \sin(\delta_p - \delta_q)]$$

y los elementos de la diagonal principal de las matrices H y L son:

$$H_{pp} = -Q_p - V^2 B_{pp}$$

$$L_{pp} = Q_p - V^2 B_{pp}$$

En el método desacoplado rápido se utilizan algunas aproximaciones para evaluar los elementos de la matriz Jacobiana.

$$\cos(\delta_p - \delta_q) \cong 1$$

$$G_{pq} \sin(\delta_p - \delta_q) \leq B_{pq}$$

$$Q_p \ll B_{pp} V_p^2$$

por lo tanto, los elementos de la matriz Jacobiana son:

$$L_{pq} = H_{pq} = -|V_p V_q| B_{pq} \quad \text{para } q \neq p$$

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

con lo anterior, las ecuaciones (3.25) y (3.26) pueden escribirse como:

$$[\Delta P_p] = [V_p][V_q][B'_{pq}][\Delta \delta_q] \quad (3.27)$$

$$[\Delta Q_p] = [V_p][V_q][B''_{pq}][\Delta |V_q|/|V_q|] \quad (3.28)$$

donde $[V_p]$ y $[V_q]$ son matrices diagonales; B'_{pq} y B''_{pq} son los elementos de la matriz $[-B_{pq}]$, es decir, la parte imaginaria de los elementos de la matriz de admitancias.

Los dos últimos sistemas, formados por las ecuaciones (3.27) y (3.28) representan al método desacoplado rápido, y se recomienda resolverlos en forma alternada con los valores actuales calculados de magnitud y ángulo de fase de los voltajes complejos nodales.

Hay que tomar en cuenta que en el método desacoplado rápido se omite de B'' el efecto del corrimiento de fase provocado por los cambiadores de fase, y se omite de B' la representación de aquellos elementos de la red que afectan los flujos de potencia reactiva, es decir, los reactores en derivación y transformadores con tap fuera del nominal.

3.4 Cálculo de las pérdidas y flujos de potencia en las líneas de transmisión.

Los flujos de potencia real y reactiva en cada rama de la red se calculan con el valor de los voltajes obtenidos en todos los nodos a partir de un método iterativo. Para cada rama se tiene:

$$P_m + jQ_m = V_p I_m^* \quad (3.29)$$

y sabemos también que:

$$I_m = Y_m (V_p - V_q) \quad (3.30)$$

sustituyendo (3.30) en (3.29) se obtienen los flujos de potencia real y reactiva:

$$P_m + jQ_m = V_p Y_m^* (V_p - V_q) \quad (3.31)$$

la ecuación anterior también se utiliza para el nodo de referencia.

Las pérdidas originadas en las líneas de transmisión se calculan como:

$$p = r_m I_m^2 = r_m [Y_m (V_p - V_q)]^2$$
$$q = -x_m I_m^2 = -x_m [Y_m (V_p - V_q)]^2$$

El análisis anterior nos dice que el total de la producción de potencia real y reactiva debe ser igual al total de la demanda más las pérdidas. Pero las ecuaciones no indican cual debe ser la cantidad de generación de potencia real y reactiva en cada nodo de generación. Si se requiere que el sistema opere de forma óptima, por ejemplo, en el sentido económico, entonces la potencia deberá ser entregada con una relación única. En este caso, entonces, se impone una restricción económica sobre las variables de control para la magnitud de la restricción anterior.

3.5 Aplicaciones del estudio de flujos de carga.

En general el estudio de flujos de potencia se realiza para investigar lo siguiente:

1. Flujos de potencia activa y reactiva a través de las líneas de transmisión.
2. Para determinar el voltaje en los nodos.
3. En la planeación de sistemas de potencia para determinar el efecto del reacomodo de circuitos o adición de nuevos elementos en la red.
4. Efecto de la pérdida temporal de circuitos de generación y transmisión en la red debido a fallas.
5. Para determinar la óptima condición de operación y distribución de carga.
6. Para determinar las pérdidas óptimas del sistema de transmisión.
7. Para evaluar el óptimo rango y cambio de tap de transformadores.
8. Para determinar las condiciones iniciales requeridas para estudios de estabilidad.
9. Para mejorar el voltaje del sistema a partir de cambios en el calibre del conductor.

A partir del estudio de flujos de potencia se obtiene información acerca de la operación de los sistemas de potencia, esta es esencial para el diseño de nuevos sistemas y para planear la extensión y analizar los efectos de cambio en los sistemas ya existentes.

El estudio puede incluir el cálculo de numerosos flujos de potencia bajo condiciones estables de operación así como bajo situaciones de salida de una línea de transmisión o salida de alguna fuente de generación. Se puede examinar el flujo entre áreas y supervisar el cambio de energía para determinar cuando un área esta produciendo la cantidad de energía que se necesita para el intercambio deseado.

3.6 Ejemplos de solución de flujos de potencia.

Para ilustrar un ejemplo de solución de flujos de potencia nos basaremos en un ejemplo desarrollado en la referencia [12]. Considérese un sistema de potencia de cuatro nodos con las características que se enlistan a continuación.

Nodo	Admitancia [P.U]
de 1 a 2	2 - j8.0
de 1 a 3	1 - j4.0
de 2 a 3	0.66 - j2.664
de 2 a 4	1 - j4.0
de 3 a 4	2 - j8.0

El esquema de potencias real y reactiva de cada nodo es el siguiente:

Nodo [P.U]	P [P.U]	Q [P.U]	V_{ini} [P.U]
1	---	---	1.06
2	0.5	0.2	1 - j0.0
3	0.4	0.3	1 - j0.0
4	0.3	0.1	1 - j0.0

Donde el nodo 1, en este caso, es tomado como referencia. Para establecer el nodo de referencia sólo basta con tomar algún nodo de generación, en caso de haber varios, se puede tomar el nodo con la mayor capacidad de potencia real.

Ejemplo 3.1. Método de Gauss-Seidel.

Se pide determinar el voltaje en cada nodo, así como los flujos de potencia en las líneas de transmisión por el método de Gauss-Seidel, con un valor de $\alpha = 1$.

Primeramente, debemos obtener la matriz de admitancias, la cual, después de realizar las operaciones establecidas en el subtema 3.2 queda de la siguiente forma:

$$Y_{pi} = \begin{bmatrix} 3 - j12 & -2 + j8 & -1 + j4.0 & 0.0 \\ -2 + j8 & 3.66 - j14.664 & -0.666 + j2.664 & -1 + j4 \\ -1 + j4.0 & -0.666 + j2.664 & 3.66 - j14.664 & -2 + j8 \\ 0.0 & -1 + j4.0 & -2 + j8 & 3 - j12.0 \end{bmatrix}$$

Después de haber obtenido la matriz de impedancias, procedemos a aplicar las fórmulas de iteración del método elegido, recordando que la potencia aparente para los nodos de carga se toma como negativa, mientras que la potencia aparente para los nodos de generación se toma como positiva.

Aplicando la formula de iteración para el nodo 2 se tiene:

$$V_2^1 = \frac{1}{Y_{22}} \left(\left(\frac{P_2 - jQ_2}{V_2^*} \right) - Y_{21} V_1^0 - Y_{23} V_3^0 - Y_{24} V_4^0 \right)$$

$$V_2^1 = \frac{1}{3.66 - j14.664} \left(\left(\frac{0.5 - j0.2}{1.06^*} \right) - (-2 + j8)(1.06) - (-0.666 + 2.664)(1) - (-1 + j4.0)(1) \right)$$

$$V_2^1 = 1.01183 - j0.28881 \text{ [P.U]}$$

Para el nodo 3:

$$V_3^1 = \frac{1}{Y_{33}} \left(\left(\frac{P_3 - jQ_3}{V_3^*} \right) - Y_{31} V_1^0 - Y_{32} V_2^1 - Y_{34} V_4^0 \right)$$

$$V_3^1 = \frac{1}{3.66 - j14.664} \left(\left(\frac{0.4 - j0.3}{1^*} \right) - (-1 + j4.0)(1.06) - (-0.666 + 2.664)(1.01183 - j0.28881) - (-2 + j8)(1) \right)$$

$$V_3^1 = 0.99284 - j0.0261062 \text{ [P.U]}$$

y para el nodo 4:

$$V_4^1 = \frac{1}{Y_{44}} \left(\left(\frac{P_4 - jQ_4}{V_4^*} \right) - Y_{41} V_1^0 - Y_{42} V_2^1 - Y_{43} V_3^1 \right)$$

$$V_4^1 = \frac{1}{3 - j12} \left(\left(\frac{0.3 - j0.1}{1.0^*} \right) - (-0 + j0.0)(1.06) - (-1 + j4.0)(1.01183 - j0.28881) - (-2 + j8)(0.99284 - j0.0261062) \right)$$

$$V_4^1 = 0.98544 - j0.048599 \text{ [P.U]}$$

Hasta este momento sólo se ha realizado la primera iteración del método, el continuar hasta cubrir un criterio de convergencia nos llevaría mucho tiempo. El problema se resuelve con ayuda de una computadora, para lo cual se realizó un programa en lenguaje Fortran 77, el cual se lista en el anexo.

Utilizando el programa y una computadora personal se obtuvieron los siguientes voltajes en los nodos:

Nodo	Voltaje [P.U]	Angulo [°]	Voltaje [P.U]
1	1.0600000	0.0000000	1.060 +j0.0
2	0.9939784	-4.1066210	0.9914-j.07118
3	0.9668982	-5.2107960	0.9629 - j0.08781
4	0.9613830	-6.1527560	0.9558 -j0.10304

Finalmente con el voltaje en cada uno de los nodos, podemos proceder a calcular los flujos de potencia y las pérdidas en las líneas de transmisión.

Nuevamente, el cálculo de los flujos de potencia y de las pérdidas suele ser largo y tedioso por lo que sólo calcularemos la trayectoria que va del nodo 1 al nodo 2 y viceversa. Una cosa que debemos tomar en cuenta en el resultado de los flujos de potencia es el hecho de tener resultados negativos, lo cual sólo significa que la potencia esta llegando al nodo.

$$P_{12} - jQ_{12} = V_1 * (V_1 - V_2) Y_{12}$$

$$P_{12} - jQ_{12} = 1.06 + j0.0 * (1.06 + j0.0 - 0.9914 - j0.07118)^2 - j8.0$$

$$P_{12} - jQ_{12} = 0.749 - j0.4308 \text{ [P.U]}$$

$$P_{21} - jQ_{21} = V_2 * (V_2 - V_1) Y_{21}$$

$$P_{21} - jQ_{21} = 0.9914 - j0.07118 * (0.9914 + j0.07118 - 1.060 + j0.0)^2 - j8.0$$

$$P_{21} - jQ_{21} = 0.7294 + j0.3526 \text{ [P.U]}$$

Las pérdidas de potencia se obtiene efectuando la resta algebraica del valor de la potencia que sale de un extremo de la línea y del valor de la potencia que llega al otro extremo.

$$p_{21} = P_{12} - P_{21}$$

$$p_{21} = 0.749 - 0.7294$$

$$p_{21} = 0.0196 \text{ [P.U]}$$

$$q_{21} = Q_{12} - Q_{21}$$

$$p_{21} = |0.4308 - 0.3526|$$

$$p_{21} = 0.0782 \text{ [P.U]}$$

En el programa que calcula los voltajes en los nodos, también se realiza el cálculo de los flujos de potencia, que en sistemas con una gran cantidad de nodos es muy útil y reduce considerablemente el tiempo de cálculo.

El programa completo arrojó los resultados siguientes:

de nodo	a nodo	Potencia real [P.U]	Potencia reactiva [P.U]	Pérdidas de potencia real [P.U]	Pérdidas de potencia reactiva [P.U]
1	2	.7489963	-.4305990	.0195383	.0781533
1	3	.4752553	-.3186113	.0171393	.0685571
1	4	.0000000	.0000000	.0000000	.0000000
2	1	-.7294580	.3524457	.0195383	.0781533
2	3	.0673837	-.0598482	.0007261	.0029045
2	4	.1694836	-.0979145	.0022810	.0091241
3	1	-.4581161	.2500542	.0171393	.0685571
3	2	-.0666575	.0569438	.0007261	.0029045
3	4	.1331691	-.0131020	.0005633	.0022533
4	1	.0000000	.0000000	.0000000	.0000000
4	2	-.1672025	.0887903	.0022810	.0091241
4	3	-.1326058	.0108487	.0005633	.0022533

Ejemplo 2.2. Método de Newton-Rhapson.

Considerese el mismo sistema que se planteó en el caso 1 y el mismo esquema de potencias. En este caso también el nodo 1 se toma como referencia.

Al igual que el caso 1, se pide determinar el voltaje en cada nodo, así como los flujos de potencia en las líneas de transmisión por el método de Newton-Raphson.

La matriz de admitancias queda de la misma forma que el caso 1. Siguiendo el método, debemos determinar los valores de G_{pq} y B_{pq} , los cuales son:

$$Y = \begin{bmatrix} G_{11} + B_{11} & G_{12} + B_{12} & G_{13} + B_{13} & G_{14} + B_{14} \\ G_{21} + B_{21} & G_{22} + B_{22} & G_{23} + B_{23} & G_{24} + B_{24} \\ G_{31} + B_{31} & G_{32} + B_{32} & G_{33} + B_{33} & G_{34} + B_{34} \\ G_{41} + B_{41} & G_{42} + B_{42} & G_{43} + B_{43} & G_{44} + B_{44} \end{bmatrix}$$

$$Y = \begin{bmatrix} 3 - j12 & -2 + j8 & -1 + j4.0 & 0.0 \\ -2 + j8 & 3.66 - j14.664 & -0.666 + j2.664 & -1 + j4 \\ -1 + j4.0 & -0.666 + j2.664 & 3.66 - j14.664 & -2 + j8 \\ 0.0 & -1 + j4.0 & -2 + j8 & 3 - j12.0 \end{bmatrix}$$

En el nodo slack:

$$a_1 = 1.06$$

$$b_1 = 0$$

En los restantes nodos, asignamos un valor inicial. En este caso asignamos un valor de $1 + j0.0$ para cada uno, por lo que los valores de a y b restantes son:

$$a_2 = 1 \quad b_2 = 0$$

$$a_3 = 1 \quad b_3 = 0$$

$$a_4 = 1 \quad b_4 = 0$$

Enseguida plantearemos y calcularemos la matriz jacobiana.

$$\begin{bmatrix} \Delta P_2 \\ \Delta P_3 \\ \Delta P_4 \\ \Delta Q_2 \\ \Delta Q_3 \\ \Delta Q_4 \end{bmatrix} = \begin{bmatrix} \frac{\partial P_2}{\partial a_2} & \frac{\partial P_2}{\partial a_3} & \frac{\partial P_2}{\partial a_4} & \frac{\partial P_2}{\partial b_2} & \frac{\partial P_2}{\partial b_3} & \frac{\partial P_2}{\partial b_4} \\ \frac{\partial P_3}{\partial a_2} & \frac{\partial P_3}{\partial a_3} & \frac{\partial P_3}{\partial a_4} & \frac{\partial P_3}{\partial b_2} & \frac{\partial P_3}{\partial b_3} & \frac{\partial P_3}{\partial b_4} \\ \frac{\partial P_4}{\partial a_2} & \frac{\partial P_4}{\partial a_3} & \frac{\partial P_4}{\partial a_4} & \frac{\partial P_4}{\partial b_2} & \frac{\partial P_4}{\partial b_3} & \frac{\partial P_4}{\partial b_4} \\ \frac{\partial Q_2}{\partial a_2} & \frac{\partial Q_2}{\partial a_3} & \frac{\partial Q_2}{\partial a_4} & \frac{\partial Q_2}{\partial b_2} & \frac{\partial Q_2}{\partial b_3} & \frac{\partial Q_2}{\partial b_4} \\ \frac{\partial Q_3}{\partial a_2} & \frac{\partial Q_3}{\partial a_3} & \frac{\partial Q_3}{\partial a_4} & \frac{\partial Q_3}{\partial b_2} & \frac{\partial Q_3}{\partial b_3} & \frac{\partial Q_3}{\partial b_4} \\ \frac{\partial Q_4}{\partial a_2} & \frac{\partial Q_4}{\partial a_3} & \frac{\partial Q_4}{\partial a_4} & \frac{\partial Q_4}{\partial b_2} & \frac{\partial Q_4}{\partial b_3} & \frac{\partial Q_4}{\partial b_4} \end{bmatrix} \begin{bmatrix} \Delta a_2 \\ \Delta a_3 \\ \Delta a_4 \\ \Delta b_2 \\ \Delta b_3 \\ \Delta b_4 \end{bmatrix} = [B] = [J][C]$$

Efectuando la particion de la matriz anterior, tenemos:

$$[B] = \begin{bmatrix} J_1 & J_2 \\ J_3 & J_4 \end{bmatrix} [C]$$

Primeramente calcularemos la matriz jacobiana J_1 . Fuera de la diagonal principal.

$$\frac{\partial P_1}{\partial a_1} = a_p G_{pp} - b_p B_{pp} \quad q \neq p$$

$$\frac{\partial P_2}{\partial a_3} = a_2 G_{23} - b_2 B_{23}$$

$$\frac{\partial P_2}{\partial a_3} = (1)(-0.666) - (0)(2.664)$$

$$\frac{\partial P_2}{\partial a_3} = -0.666$$

de igual forma

$$\frac{\partial P_2}{\partial a_4} = -1$$

$$\frac{\partial P_3}{\partial a_2} = -0.666$$

$$\frac{\partial P_3}{\partial a_4} = -2$$

$$\frac{\partial P_4}{\partial a_2} = -1$$

$$\frac{\partial P_4}{\partial a_3} = -2$$

Los elementos de la diagonal principal de J_1 son:

$$\frac{\partial P_p}{\partial a_q} = 2a_p G_{pp} + \sum_{\substack{q=1 \\ p \neq q}}^n (a_q G_{pq} + b_q B_{pq})$$

$$\frac{\partial P_2}{\partial a_2} = 2a_2 G_{22} + a_1 G_{21} + b_1 B_{21} + a_3 G_{23} + b_3 B_{23} + a_4 G_{24} + b_4 B_{24}$$

$$\frac{\partial P_2}{\partial a_2} = 2(1)(3.66) + (1.06)(-2) + (0)(8) + (1)(0.666) + (0)(2.664) + (1)(-1) + (0)(4)$$

$$\frac{\partial P_2}{\partial a_2} = 3.534$$

de la misma forma:

$$\frac{\partial P_3}{\partial a_3} = 3.594$$

$$\frac{\partial P_3}{\partial a_4} = -2$$

$$\frac{\partial P_4}{\partial a_2} = -1$$

$$\frac{\partial P_4}{\partial a_4} = 3$$

Ahora, para J_2 realizamos exactamente los mismos cálculos.

$$\frac{\partial P_p}{\partial b_q} = a_p B_{pq} - b_p G_{pq} \quad q \neq p$$

$$\frac{\partial P_2}{\partial b_3} = 2.664$$

de igual forma

$$\frac{\partial P_2}{\partial b_4} = 4$$

$$\frac{\partial P_3}{\partial b_2} = 2.664$$

$$\frac{\partial P_3}{\partial b_4} = 8$$

$$\frac{\partial P_4}{\partial b_2} = 4$$

$$\frac{\partial P_4}{\partial b_3} = 8$$

Los elementos de la diagonal principal de J_2 son:

$$\frac{\partial P_p}{\partial b_q} = 2b_p G_{pp} + \sum_{\substack{q=1 \\ p \neq q}}^n (b_q G_{pq} + a_q B_{pq})$$

$$\frac{\partial P_2}{\partial a_2} = -18.93$$

de la misma forma:

$$\frac{\partial P_3}{\partial a_3} = -18.63$$

$$\frac{\partial P_4}{\partial a_4} = -15$$

Para J_3 realizamos exactamente los mismos cálculos que para J_1 y J_2 .

$$\frac{\partial Q_p}{\partial a_q} = a_p B_{pq} + b_p G_{pq} \quad q \neq p$$

$$\frac{\partial Q_2}{\partial a_3} = 2.664$$

de igual forma

$$\frac{\partial Q_2}{\partial a_4} = 4$$

$$\frac{\partial Q_3}{\partial a_2} = 2.664$$

$$\frac{\partial Q_3}{\partial a_4} = 8$$

$$\frac{\partial Q_4}{\partial a_2} = 4$$

$$\frac{\partial Q_4}{\partial a_3} = 8$$

Los elementos de la diagonal principal de J_3 son:

$$\frac{\partial Q_p}{\partial a_p} = 2a_p B_{pp} - \sum_{\substack{q=1 \\ p \neq q}}^n (b_q G_{pq} - a_q B_{pq})$$

$$\frac{\partial Q_2}{\partial a_2} = -14.184$$

de la misma forma:

$$\frac{\partial Q_3}{\partial a_3} = -14.424$$

$$\frac{\partial Q_4}{\partial a_4} = -12$$

Finalmente, para la última partición de la matriz jacobiana J_4 tenemos:

$$\frac{\partial Q_p}{\partial a_q} = -a_p G_{pq} + b_p B_{pq} \quad q \neq p$$

$$\frac{\partial Q_2}{\partial a_3} = 0.666$$

de igual forma

$$\frac{\partial Q_2}{\partial a_4} = 1$$

$$\frac{\partial Q_3}{\partial a_2} = 0.666$$

$$\frac{\partial Q_3}{\partial a_4} = 2$$

$$\frac{\partial Q_4}{\partial a_2} = 1$$

$$\frac{\partial Q_4}{\partial a_3} = -2$$

Los elementos de la diagonal principal de J_4 son:

$$\frac{\partial Q_p}{\partial a_q} = 2b_p B_{pp} + \sum_{\substack{q=1 \\ p \neq q}}^n (a_q G_{pq} + b_q B_{pq})$$

$$\frac{\partial Q_2}{\partial a_2} = -3.786$$

de la misma forma:

$$\frac{\partial Q_3}{\partial a_3} = -3.726$$

$$\frac{\partial Q_4}{\partial a_4} = -3$$

Ahora, los valores calculados de P son:

$$P_p = \sum_{q=1}^n (a_p (a_q G_{pq} + b_q B_{pq}) + b_p (b_q G_{pq} + a_q B_{pq}))$$

$$P_2 = -1.26$$

$$P_3 = -0.06$$

$$P_4 = 0.0$$

y los valores calculados para Q son:

$$Q_p = \sum_{q=1}^n (b_p (a_q G_{pq} + b_q B_{pq}) + a_p (b_q G_{pq} + a_q B_{pq}))$$

$$Q_2 = 0.48$$

$$Q_3 = 0.24$$

$$Q_4 = 0.0$$

Los valores especificados son los datos de P y Q que se dieron al principio del ejemplo. Junto con los valores calculados podemos obtener los incrementos de P y Q.

$$\Delta P_p = P_{\text{especificado}} - P_{\text{calculado}}$$

$$\Delta P_2 = P_{2\text{esp}} - P_{2\text{cal}}$$

$$\Delta P_2 = 0.5 - (-0.126)$$

$$\Delta P_2 = 0.626$$

$$\Delta P_3 = 0.46$$

$$\Delta P_4 = 0.3$$

$$\Delta Q_2 = -0.28$$

$$\Delta P_3 = 0.059$$

$$\Delta P_4 = 0.1$$

Finalmente el sistema de ecuaciones lineales de primer orden para la primera iteración es el siguiente:

$$\begin{bmatrix} 0.626 \\ 0.46 \\ 0.3 \\ -0.028 \\ 0.059 \\ 0.1 \end{bmatrix} = \begin{bmatrix} 3.534 & -0.666 & -1 & -15.144 & 2.664 & 4 \\ -0.666 & 3.606 & -2 & 2.664 & -14.904 & 8 \\ -1 & -2 & 3 & 4 & 8 & -12 \\ 44.472 & 2.664 & 4 & -3.786 & 0.666 & 1 \\ 2.664 & 44.172 & 8 & 0.666 & -3.726 & 2 \\ 4 & 8 & 42.36 & 1 & 2 & -3 \end{bmatrix} \begin{bmatrix} \Delta a_2 \\ \Delta a_3 \\ \Delta a_4 \\ \Delta b_2 \\ \Delta b_3 \\ \Delta b_4 \end{bmatrix}$$

Resolviendo el sistema de ecuaciones:

$$\Delta a_2 = 0.0203$$

$$\Delta a_3 = -0.00405$$

$$\Delta a_4 = -0.00785$$

$$\Delta b_2 = -0.0332$$

$$\Delta b_3 = -0.0247$$

$$\Delta b_4 = -0.02061$$

Finalmente el voltaje después de la primera iteración es:

$$a_2 = a_2^0 + \Delta a_2$$

$$a_2 = 1.0203$$

$$b_2 = b_2^0 + \Delta b_2$$

$$b_2 = -0.0332$$

$$V_2 = 1.0203 - j0.0332$$

de la misma forma:

$$V_3 = 0.9959 - j0.02477$$

$$V_4 = .9921 - j0.02061$$

Al igual que en el caso 1, estos resultados son producto de la primera iteración al aplicar el método. Continuar con las iteraciones resultaría demasiado laborioso, es por eso que, para las demás iteraciones, fue necesario auxiliarse de algoritmos programados específicamente para obtener algunos de los resultados parciales necesarios para obtener la solución hasta la última iteración.

Al final de la última iteración los resultados fueron los siguientes:

Nodo	Voltaje [P.U]	Ángulo [°]	Voltaje [P.U]
1	1.0600000	0.0000000	1.060 +j0.0
2	1.003	-4.93	0.9993-j.0825
3	0.9835	-6.2209	0.977 7- j0.1065
4	0.97928	-6.7844	0.9724 -j0.1156

Capítulo 4. Programación y elaboración de interfaces gráficas. Plataforma *Windows*.

Windows es el entorno gráfico que añade al sistema operativo DOS la característica de sistema gráfico, cambiando la manera en que el usuario interactúa con la computadora. En un entorno gráfico, un usuario simplemente selecciona la herramienta correspondiente a la tarea que desea ejecutar; el resto del trabajo lo hace el sistema.

Windows introduce en el entorno de DOS el Intercambio Dinámico de Datos, conocido como DDE, recurso que posibilita la transferencia de información entre las diversas aplicaciones y la actualización de todas ellas simultáneamente.

Otro recurso de *Windows* no menos importante, es la capacidad de permitir que un usuario ejecute más de una tarea al mismo tiempo(multitarea).

Trabajar bajo el ambiente *Windows* representa una gran ventaja debido a que todas las ventanas se comportan de la misma forma y todas las aplicaciones utilizan los mismos métodos básicos(menús desplegables, botones, etc.) para introducir órdenes.

Otra característica importante de los programas de *Windows* es que deben compartir recursos, muchos recursos se solicitan al sistema operativo antes de que se usen, y una vez que se utilizan, deben regresarse al sistema operativo para que puedan emplearlos otros programas. Esta es la forma como *Windows* controla el acceso a recursos, como la pantalla y otros dispositivos físicos.

En un ambiente como *Windows*, se envían mensajes a un programa cuando ocurre un evento que afecta al programa. Cada mensaje que se envía al programa tiene un propósito específico. La respuesta a mensajes de evento es fundamental para los programas de *Windows*.

Los mensajes son la parte medular de todo programa *Windows*. Entender bien la manera como envía mensajes el sistema operativo *Windows* es de gran ayuda mientras se escribe un programa orientado al objeto.

Un programa desarrollado bajo *Windows* también puede enviar mensajes a otras ventanas. Como cada control usado en un programa *Windows* también es una ventana, en ocasiones los mensajes también se usan para la comunicación con los controles. Existen dos tipos diferentes de mensajes que se manejan con un programa *Windows*.

- a) Mensajes enviados desde el sistema operativo.
- b) Mensajes enviados hacia y desde los controles que manejan la entrada del usuario.

Ejemplos de mensajes que se envían desde el sistema operativo incluyen los que se emplean para decirle al programa que debe iniciar o terminar. Los mensajes que se envían a los controles pueden usarse para cambiar el tipo de letra utilizado por una ventana o su título. Los mensajes que se reciben desde un control incluyen la notificación de que se ha oprimido un botón.

4.1. Interfaz Gráfica del Usuario (GUI).

La GUI es el mecanismo de comunicación entre el usuario y la aplicación. Para realizar una aplicación se crean ventanas llamadas formularios, y sobre ellas se dibujan otros objetos llamados controles, tales como cajas de texto, botones de órdenes, listas desplegables, etc., con el fin de introducir datos o visualizar resultados.

Además se puede añadir a las GUI menús desplegables y menús desplegables flotantes, para proporcionar al usuario un amplio rango de órdenes seleccionables.

Este capítulo proporciona una visión general de la funcionalidad orientada a objetos de los entornos del usuario y del diseñador asociados con Interfaces Gráficas de Usuario.

Tanto el entorno del usuario como el entorno del diseñador de las GUI, están en cierta medida orientados a objetos. Desde la perspectiva del usuario, la pantalla contiene varios objetos visuales, tales como ventanas, botones, barras de desplazamiento e iconos. El usuario puede interactuar con estos objetos a través del teclado, o en la mayoría de los casos, por medio de un ratón. Los objetos, los cuales reflejan bastante aproximadamente el entorno del mundo real contienen datos y procedimientos que permiten cierta manipulación directa, sin invocar continuamente un programa de aplicación.

4.1.1. Características de las GUI.

1. Ventanas que representan gráficamente lo que la computadora está haciendo.
2. Íconos que representan archivos, directorios y otras aplicaciones y entidades del sistema.
3. Cuadros de diálogo, botones, barras de desplazamiento y otros objetos gráficos que permiten al programador y al usuario indicar a la computadora lo que debe hacer y como hacerlo.
4. Menús en pantalla que aparecen o desaparecen bajo el control del dispositivo señalador(apuntador del ratón).

4.1.2. Ventajas de las GUI.

Los objetos en pantalla pueden manipularse para controlar el comportamiento de la computadora. Para el diseñador, las GUIs son el panel de control que gestiona la interacción entre el entorno del sistema operativo, la aplicación subyacente y el usuario.

Las GUIs orientadas a objetos hacen que las aplicaciones se parezcan mucho más a las funciones del mundo real y cada vez menos a los procesos de programación. Esto, a su vez, permite a los usuarios manipular imágenes familiares. La ventaja principal consiste en ayudar a los usuarios a navegar en un entorno que es más parecido al mundo real y menos a un programa de computadora.

Esto permite al usuario centrarse en las tareas en lugar de distraerse con las herramientas. En otras palabras, el usuario emplea menos tiempo entrando y saliendo de los programas y más tiempo realizando una tarea inmediata.

Por ejemplo, para crear un objeto folleto u hoja informativa, el usuario podría abrir un objeto llamado folleto y el objeto podría cargar todas las herramientas de autoedición necesarias.

En los entornos gráficos actuales, los iconos son representaciones gráficas de datos o programas. La interacción entre estos objetos se soporta normalmente mediante alguna forma de intercambio rápido de contexto.

En *Windows* de *Microsoft*, el intercambio de programas se realiza mediante DDE, que proporciona un enlace dinámico entre los datos y las aplicaciones.

En una GUI las cosas en la pantalla se programan como objetos con correspondencia directa entre los objetos de presentación y código.

El enfoque orientado a objetos para la construcción de las GUI permite al diseñador abandonar las sentencias inconexas de los lenguajes tradicionales y utilizar en su lugar objetos representando los elementos clave de la interfaz, como botones o cuadros de diálogo. Los programadores pueden disfrutar de todas las ventajas de la Programación Orientada a Objetos (POO), incluyendo modularidad, reutilización y extensibilidad.

Una gran parte del campo de desarrollo de la mayoría de las GUI está ya basado en objetos y se aproxima mucho o incluso soporta realmente los mecanismos básicos orientados a objetos (métodos, clases y herencia). Por ejemplo, la mayor parte de las GUI están basadas en sucesos. Este proceso basado en sucesos está normalmente soportado por una arquitectura basada

en mensajes, uno de los mecanismos fundamentales de los sistemas orientados a objetos. Debido al polimorfismo, el mismo mensaje enviado a diferentes objetos produce resultados diferentes. Cuando ocurre un suceso, el entorno del sistema determina la naturaleza del suceso y hace que se ejecute la función apropiada.

Utilizando el concepto de herencia, una capacidad que no está disponible en la programación procedimental, el programador puede crear ventanas de subclases que ya tienen la funcionalidad de clases de ventana estandar predefinida y programar solamente las diferencias.

Un formulario es una ventana que sirve de fondo para los controles y para los gráficos situados sobre la misma. Se pueden utilizar tantos formularios como se necesiten y dependiendo de la utilidad que se les dé, estos serán de diferentes tipos. Así se puede crear un formulario para que contenga un gráfico, para visualizar información o para aceptar datos.

4.2. Programación Orientado a Objetos (POO).

La POO es una forma de programación que utiliza objetos, los cuales responden a sucesos. Estos sucesos producen un flujo de mensajes entre los objetos, lo que origina cambios en el estado del objeto que recibe el mensaje.

Un programa tradicional se compone de procedimientos y datos. Un programa orientado a objetos está compuesto solamente por objetos.

En la programación dirigida a sucesos, el código sólo es ejecutado en el momento en que el usuario o el sistema llama al suceso correspondiente. Mientras la aplicación espera un suceso, permanece en el ambiente a menos que el usuario la cierre. En este mismo instante el usuario puede ejecutar otras aplicaciones.

Los objetos que se pueden programar incluyen:

- a) Pantallas o ventanas(formas).
- b) Controles: objetos gráficos diseñados en una ventana, incluyendo cuadros de texto, botones de órdenes, cuadros de diálogo, etiquetas, etc.

La programación que realizamos en DOS, difiere bastante de como se programa una aplicación en *Windows*. Un programa escrito en DOS es un conjunto de sentencias que se ejecutan de arriba abajo, más o menos en el orden que el programador ha diseñado. Una aplicación en *Windows* presenta todas las opciones posibles en uno o más formularios, para que el usuario elija una de ellas. Esto da lugar a una nueva forma de pensar y programar.

Cuando se desarrolla una aplicación utilizando la POO, la secuencia en que van a ejecutarse las sentencias no puede ser prevista por el programador. Por ejemplo, si en lugar de un botón hubiera dos o más botones, claramente se ve que el programador no puede escribir el programa pensando que el usuario va a pulsarlos en una determinada secuencia.

Por lo tanto, para programar una aplicación *Windows* hay que escribir código separado para cada objeto en general, quedando la aplicación dividida en pequeños procedimientos, conducido cada uno de ellos por un suceso. Un suceso es una acción reconocida por un objeto, el suceso puede ser causado por el usuario, por el sistema o indirectamente por el código (por ejemplo, cuando el código carga un formulario se da el suceso load).

Cada formulario y cada control tienen predefinidos un conjunto de sucesos. Cuando ocurre un suceso, se invoca el procedimiento asociado con el objeto para ese suceso.

Un objeto no solamente tiene asociados datos, sino también métodos, los datos se denominan propiedades y los métodos procedimientos.

Una aplicación en *Windows* es conducida por sucesos y orientada a objetos, esto significa que nosotros ligamos unidades de código escritas para un determinado objeto a sucesos que pueden ocurrir sobre dicho objeto, de tal forma que cuando ocurra un suceso se ejecute la unidad de código correspondiente. Es por eso que la unidad que agrupa ese código recibe el nombre de procedimiento conducido por suceso.

Para unir un procedimiento a un objeto, hay que escribir el código correspondiente a dicho procedimiento en la ventana de código asociada con ese objeto.

4.2.1. Orígenes de la POO.

Un cierto número de lenguajes de programación han contribuido a la evolución de los actuales lenguajes orientados a objetos, empezando con LISP(*list processing*), en los años 50, LISP(procesamiento de listas), es un lenguaje de inteligencia artificial que introdujo el concepto de ligadura dinámica y la ventaja de un entorno de desarrollo interactivo en la evolución de los lenguajes orientados a objetos. Simula, desarrollado en los años 60 como lenguaje para programar simulaciones, contribuyó con el concepto de clase y los mecanismos de herencia. La abstracción de datos, fue introducida en los años 70, por lenguajes populares como *Ada* y *Modula-2*.

La fuerza principal que impulsó el desarrollo comercial de los lenguajes orientados a objetos fue el proyecto *Smalltalk* en el Centro de Investigación *Xerox* de Palo Alto(PARC)(*Xerox Palo Alto*

Research Center). Alan Kay describió los objetivos originales para *Smalltalk* a principios de los años 70. Estos objetivos incluían el concepto de clase como se utilizaba en *Simula-67*, y una interfaz gráfica que soporta la manipulación directa del usuario. *Smalltalk* evolucionó a través de varios lanzamientos realizados por PARC, culminando en el lanzamiento de *Smalltalk-80* en 1981. A pesar de este movimiento temprano hacia los lenguajes orientados a objetos, solamente se lograron pequeñas incursiones en la comunidad de la programación general. El progreso fuera de los lenguajes procedimentales sucedía de forma lenta debido a que los lenguajes que soportaban la programación procedimental estaban bien atrincherados en la época en que llegaban los lenguajes orientados a objetos. El progreso reciente se ha acelerado, debido principalmente a la disponibilidad de las extensiones orientadas a objetos para los lenguajes populares, C y Pascal, y a las extensiones prometidas para otros lenguajes comerciales populares como BASIC y COBOL. Dos ámbitos principales han surgido a partir de la última década de evolución del lenguaje orientado a objetos.

Un ámbito es el grupo del lenguaje puro orientado a objetos, en el que casi todo es un objeto. Este grupo incluye *Smalltalk*. El otro ámbito es el grupo híbrido, cuyas construcciones orientadas a objetos se añaden a un lenguaje procedimental. Los miembros de este grupo incluyen C++, *Objective C*, *Common Lisp*(CLOS) y los diferentes lenguajes Pascal orientados a objetos.

Los lenguajes orientados a objetos más maduros como *Smalltalk*, ofrecen también robustas bibliotecas de clases y excelentes conjuntos de herramientas de desarrollo. Estas capacidades se están también incorporando gradualmente a los lenguajes híbridos.

Con la aparición de C en los años 80 como lenguaje de programación extremadamente popular en todas las plataformas, las ampliaciones orientadas a objetos para C han sido una razón fundamental para la atención creciente hacia la propia programación orientada a objetos y la justificación del mayor empleo de C++ sobre *Smalltalk*.

El rendimiento ha sido un factor clave en el aumento de popularidad de C++. El código producido por C++ es casi igual en rendimiento al código producido por los compiladores tradicionales de C, debido al cuidadoso diseño del lenguaje que apunta hacia una generación de código eficaz.

Pascal orientado a objetos es otro híbrido popular. *Apple computer* incorporó a Pascal en 1985 las primeras ampliaciones o extensiones orientadas a objetos para la computadora Macintosh. Otros

lenguajes Pascal orientados a objetos han sido recientemente lanzados por *Microsoft Corporation* y *Borlan International Inc*, para su uso en los entornos de DOS y OS/2.

Al igual que C++, el Pascal orientado a objetos ha sido desarrollado con velocidad y eficacia como requisitos indispensables y no exige un rendimiento significativo en los equipos que lo utilizan.

La compatibilidad hacia el pasado es una ventaja extremadamente importante para C++ y Pascal orientado a objetos. Una gran cantidad de código de programas de la industria del software ya está escrito en C y Pascal. Añadir técnicas orientadas a objetos a un programa ya existente en un lenguaje familiar es mucho más fácil que volver a escribir una aplicación en un lenguaje completamente nuevo.

4.2.2. Mecanismos de la POO.

Los mecanismos básicos de la programación orientada a objetos son: objetos, mensajes, métodos y clases.

4.2.2.1. Objetos.

Los objetos son módulos que contienen los datos y las instrucciones que operan sobre esos datos. Así, dentro de los objetos residen los datos de los lenguajes convencionales, así como cualquier función, instrucción o subrutina que opere sobre ellos. Los objetos, por tanto, son entidades que tienen atributos(datos) y formas de comportamiento(procedimientos particulares). Los objetos son módulos de una aplicación que funcionan juntos para proporcionar una funcionalidad general.

4.2.2.2. Mensajes.

Cuando se ejecuta un programa orientado a objetos, los objetos están recibiendo, interpretando y respondiendo a mensajes de otros objetos. Esto marca una clara diferencia con respecto a los elementos de datos pasivos de los sistemas tradicionales. Por ejemplo, en *visual Basic* un mensaje esta asociado con la declaración de una función o de un procedimiento de tal forma que cuando un objeto recibe un mensaje se ejecuta la correspondiente función o procedimiento de la clase a la que pertenece el objeto.

El objeto emisor del mensaje no necesita conocer la forma en que el objeto receptor está llevando a cabo la solicitud. El objeto que envía el mensaje no sabe de que manera lleva a cabo la solicitud el objeto receptor, solo sabe que se está llevando a cabo.

4.2.2.3. Métodos.

Los procedimientos llamados métodos residen en el objeto y determinan como actúa el objeto cuando recibe un mensaje.

Los métodos se ejecutan en respuesta a mensajes y manipulan los valores de los datos. De hecho, los métodos proporcionan el único mecanismo para cambiar los valores de datos. Los métodos pueden enviar también mensajes a otros objetos solicitando una acción o información. La estructura más interna de un objeto esta oculta para otros usuarios y la única conexión que tiene con el exterior son los mensajes. Los datos que están dentro de un objeto, solamente pueden ser manipulados por los métodos asociados al propio objeto.

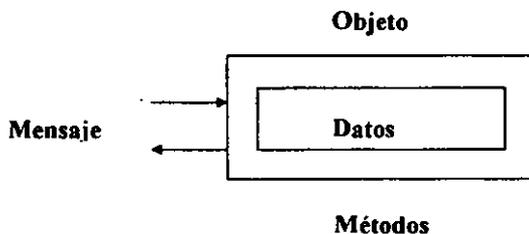


Fig. 4. 1. Representación de un objeto.

4.2.2.4. Clases.

Muchos objetos diferentes pueden actuar de formas muy similares. Un objeto es un modelo o instancia de una clase. Una clase es una descripción de un conjunto de objetos casi idénticos. Una clase consta de métodos y datos que resumen las características comunes de un conjunto de objetos. La posibilidad de abstraer métodos y descripciones de datos comunes de un conjunto de objetos y almacenarlos en una clase es esencial para la potencia de la orientación a objetos. Definir clases significa situar código reutilizable en un depósito común en lugar de volver a expresarlo repetidas veces.

Los objetos se crean cuando se recibe un mensaje solicitando su creación por parte de la clase padre. El nuevo objeto toma sus métodos y datos de su clase padre. Los datos son de dos formas, variables de clase y variables modelo o de instancia. Las variables de clase tienen valores

almacenados en una clase, las variables instancia tienen valores asociados únicamente con cada instancia u objeto creado a partir de una clase.

4.2.3. Características de la POO.

Las características fundamentales de la POO son: abstracción, encapsulamiento, herencia y polimorfismo.

4.2.3.1. Abstracción.

Por medio de la abstracción conseguimos no detenernos en los detalles concretos de las cosas que no interesen en cada momento, sino generalizar y centrarse en los aspectos que permitan tener una visión global del tema. Por ejemplo, el estudio de un ordenador se puede realizar en el ámbito de la transferencia entre registros, centrándose así el estudio en el flujo de información entre las unidades que lo componen, sin importarnos el comportamiento de los circuitos electrónicos que componen estas unidades.

4.2.3.2. Encapsulamiento.

La encapsulación es el término formal que describe el conjunto de métodos y datos dentro de un objeto, de tal manera que el acceso a los datos se permite solamente a través de los propios métodos del objeto. Ninguna otra parte de un programa orientado a objetos puede operar directamente sobre los datos de un objeto. La comunicación entre un conjunto de objetos sucede exclusivamente por medio de mensajes explícitos.

4.2.3.3. Herencia.

La herencia es el mecanismo para compartir automáticamente métodos y datos entre clases y subclases de objetos. La razón principal del uso de la herencia es el reuso, la capacidad de un programa de utilizar código escrito y probado para otro programa. La herencia permite a los programadores crear nuevas clases programando solamente las diferencias con la clase padre.

Los objetos con métodos y datos comunes se reúnen en una clase. Las clases se organizan en jerarquías y los mecanismos de herencia proporcionan automáticamente a cada subclase los métodos y datos de las clases padre. Las subclases se crean programando las diferencias entre las clases disponibles en una biblioteca y los requisitos particulares de la aplicación.

4.2.3.4. Polimorfismo.

Los objetos actúan en respuesta a los mensajes que reciben. El mismo mensaje puede originar acciones completamente diferentes al ser recibido por diferentes objetos. Este fenómeno se conoce

como polimorfismo. Con el polimorfismo, un usuario puede enviar un mensaje genérico y dejar los detalles exactos de la realización para el objeto receptor. El mensaje imprimir, por ejemplo, al ser enviado a una figura o diagrama invocará diferentes métodos de impresión que en el caso de enviar el mismo mensaje imprimir a un documento de texto.

4.3. Programación en *Visual Basic*.

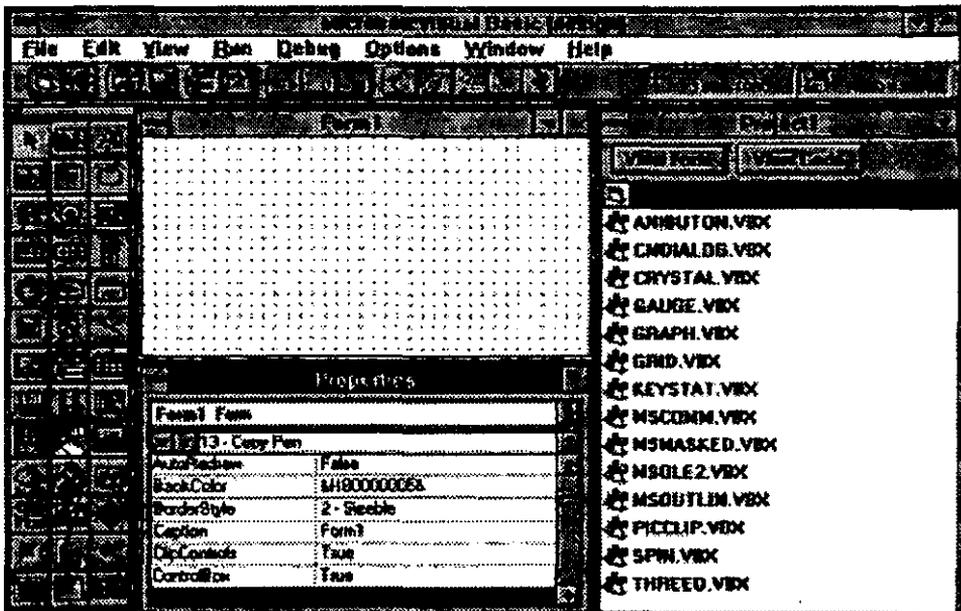


Figura 4.2. Pantalla de presentación de *Microsoft Visual Basic* Versión 3

Microsoft VB es un sistema de desarrollo diseñado especialmente para crear aplicaciones gráficas de una forma rápida y sencilla. Para crear una aplicación, se crean ventanas y sobre ellas se dibujan controles (etiquetas, botones, cajas de texto, etc.) y a continuación se escribe el código fuente relacionado con cada objeto. Esto es, cada objeto está ligado a un código que permanece inactivo hasta que se dé el suceso que lo activa (por ejemplo un clic del ratón).

Visual Basic provee de herramientas que permiten crear ventanas y controles sin escribir código, esto se logra utilizando las características gráficas de *Windows*.

La programación en VB no se inicia con un editor de programas, sino con el diseño de la GUI (Interfaz Gráfica del Usuario). Para ello es necesario estar familiarizado con *Microsoft Windows*.

Visual Basic utiliza fundamentalmente dos herramientas, una que permite realizar los diseños gráficos y un lenguaje de alto nivel.

Visual Basic provee de varias herramientas para facilitar el diseño de cualquier aplicación gráfica, la figura 4. 2 muestra estas herramientas.

Lo que hace que *Visual Basic* sea diferente de cualquier otra herramienta de programación es la facilidad con la que se puede diseñar una pantalla (GUI). Una vez que se ha terminado de dibujar la interfaz, los botones de órdenes, cajas de texto y otros controles que se han colocado en una ventana en blanco, reconocerán automáticamente acciones del usuario, tales como el movimiento del ratón y los *clicks* de los botones.

Tan solo después de diseñar la interfaz de usuario es cuando se empieza a hacer algo que se parezca a programar. Los objetos en *Visual Basic* reconocerán sucesos como los *clicks* del ratón. La respuesta de los objetos dependerá del código que se escriba, se necesitará escribir código para que los controles respondan a los sucesos.

Los programas en los lenguajes de programación convencionales se ejecutan de arriba abajo. En los antiguos lenguajes de programación la ejecución comienza en la primera línea y se desplaza con el flujo del programa a las distintas partes según se necesita. Un programa en *Visual Basic* es un conjunto de diferentes partes de código que son activadas por, y que solamente responden a, los sucesos que se le han indicado que reconozcan. Ahora, en lugar de diseñar un programa que haga lo que el programador piense que debe hacer, el usuario tiene el control.

La mayor parte del código de programación en *Visual Basic* indica al programador el modo de responder a determinados sucesos, como el *click* del ratón, en lo que se denomina procedimientos de suceso. La documentación utiliza el termino proyecto, en lugar de programa, para referirse a la combinación de código de programación e interfaz de usuario que hace que sea posible una aplicación en *Visual Basic*.

A continuación se ve un resumen de los pasos a seguir para diseñar una aplicación en *Visual Basic*:

1. Personalizar la ventana que utiliza el usuario.
2. Decidir los sucesos que deben reconocer los controles en la ventana.
3. Se escriben los procedimientos de suceso para estos.

4. Escribir cualquier procedimiento que necesiten los procedimientos de sucesos para realizar su labor.

Visual Basic permite añadir a las ventanas en blanco menús, cajas de texto, botones de órdenes, botones de opción, cajas de listados, barras de desplazamiento y cajas de archivos y directorios. Se pueden utilizar mallas para manejar datos de tablas, comunicar con otras aplicaciones *Windows* y acceder a bases de datos.

El lenguaje de programación de *Visual Basic* tiene sentencias gráficas de fácil uso, potentes funciones matemáticas incorporadas, funciones de manipulación de cadenas y sofisticadas posibilidades de gestión de archivos.

Visual Basic hace sencilla la creación de grandes programas mediante las modernas técnicas modulares de programación. Esto significa que se puede dividir un programa en módulos, más sencillos de manejar y, por tanto, menos sensibles a los errores. Idealmente, los módulos realizan una sola tarea y tienen una interfaz bien definida con el resto del programa, por lo que puede ser codificado y verificado independientemente.

4.4. Elaboración de interfaces gráficas de usuario para el manejo de los algoritmos generados en computadora.

Nuestra aplicación, elaborada tomando como base el lenguaje de programación orientado a objetos VB, permite obtener la solución de algunos Sistemas Eléctricos de Potencia.

Nuestra aplicación consta de cuatro interfaces gráficas de usuario (ver manual del usuario) que se utilizan para introducir datos a archivos, estos datos son utilizados por archivos ejecutables obtenidos a partir de programas elaborados en FORTRAN, para obtener la solución de cada uno de los Sistemas Eléctricos de Potencia aquí tratados.

La aplicación consta de cuatro interfaces más para la lectura de resultados, desde cada una de estas interfaces se corren los archivos ejecutables, los cuales generan resultados que son enviados a archivos de resultados, estos archivos son leídos desde las interfaces diseñadas para este fin.

Para diseñar la interfaz (medio de comunicación entre el usuario y la aplicación) se siguen los siguientes pasos:

1. Se crea un nuevo formulario
2. Se añaden los controles al formulario.

3. El siguiente paso es especificar las propiedades para los objetos creados. Por ejemplo, título de formulario, títulos de los controles, identificadores de los objetos, valores iniciales, colores, etc.
4. Finalmente se une código al formulario y a los controles para hacer que la aplicación ejecute las tareas deseadas.

Cualquier aplicación que se cree con *Visual Basic* es conducida por sucesos. Esto significa que los formularios y los controles pueden responder a sucesos tales como un *click* del ratón, un mensaje o un tiempo transcurrido. Responder a un suceso significa que cuando este se produzca, tiene que ejecutarse el código asociado con el objeto implicado y para el suceso que se ha producido.

El código que se une a un objeto se denomina procedimiento conducido por un suceso. Para unir un procedimiento a un objeto, hay que escribir el código correspondiente a dicho procedimiento en la ventana de código asociada con ese objeto. Para visualizar la ventana de código de un determinado objeto, hay que hacer un doble clic dentro del área correspondiente a ese objeto.

Cualquier aplicación que se cree con VB es conducida por sucesos, esto significa que los formularios y los controles pueden responder a sucesos tales como un *click* del ratón, un mensaje DDE o un tiempo transcurrido. Responder a un suceso significa que cuando éste se produzca, tiene que ejecutarse el código asociado con el objeto implicado y para el suceso que se ha producido. Todo el código asociado con un formulario y con sus controles se almacena con ese formulario. Cuando una aplicación consta de varios formularios y se quiere compartir código entre ellos, es necesario poner el código en un módulo estándar.

Un módulo estándar es un fichero que contiene los procedimientos que son compartidos por todos los formularios de una aplicación.

Cuando en una aplicación se utilizan muchas órdenes, VB permite agruparlas en uno o más menús, los cuales dan lugar a una barra de menús.

En el manual del usuario desarrollado en el capítulo cinco, se explica en forma detallada la manera en que se introducen los datos que se requieren para correr los archivos ejecutables obtenidos a partir de programas elaborados en FORTRAN, a partir de estos archivos se obtienen resultados que son leídos de igual manera a través de interfaces gráficas de usuario diseñadas para tal fin.

Capítulo 5. Manual del usuario.

El presente manual describe en forma detallada la manera en que se utiliza la aplicación mediante la cual se pueden resolver algunos Sistemas Eléctricos de Potencia. La aplicación fue elaborada usando programación orientada a objetos, se eligió *Visual Basic* como lenguaje de programación.

El problema que se pretende resolver consiste en obtener la solución de redes eléctricas. Al resolver la red es posible obtener las corrientes de malla y los voltajes de nodo para la red que se esté analizando. Para la misma red o para cualquier otra, se pueden obtener además los flujos de potencia y finalmente, si así se requiere, también es posible resolver líneas de transmisión considerándolas como redes de dos puertos.

La forma en que funciona la interfaz gráfica para cada caso se explica a continuación:

1. Desde la interfaz correspondiente, se introducen los datos y que se requieren para resolver la red.
2. Desde la pantalla de presentación de resultados se corren los archivos ejecutables obtenidos a partir de programas elaborados en FORTRAN.
3. Los resultados obtenidos a partir de los archivos ejecutables, se guardan en archivos de resultados.
4. Los archivos de resultados son leídos desde la pantalla de presentación de resultados.

Para poder utilizar el programa en un equipo de cómputo se requiere que como mínimo tenga procesador 386 o similar, contar con *Windows 3* o superior, tener preferentemente 10 MB libres en Disco Duro y un mínimo de 600 KB en memoria RAM exclusivos para nuestro programa, sin tomar en cuenta los recursos de RAM que necesita *Windows*.

5.1. Instalación de la aplicación.

Antes de iniciar la instalación del programa, se debe verificar que exista el siguiente archivo en C:\WINDOWS\SYSTEM\ VBRUN300.DLL, el cual es necesario para que la aplicación pueda ser ejecutada. En caso de no contar con este archivo, el usuario lo puede copiar desde el disco de instalación hasta la trayectoria mencionada.

Después de confirmar que VBRUN300.DLL existe en el disco C, procedemos a realizar la instalación, esta se logra a través del disco flexible que contiene los archivos necesarios para que la aplicación pueda correr correctamente. Para instalar la aplicación se debe insertar el disco flexible

en el manejador de disco A. En la barra de inicio de WINDOWS, en la opción RUN o EJECUTAR, se debe escribir la siguiente instrucción:

A:\INSTALAR

Enseguida se mostrará la ventana de instalación, en la cual es necesario presionar el botón INSTALAR. Después de que se han generado los directorios y grabado la información en el disco duro se debe presionar el botón salir.

Para ejecutar la aplicación bastará con hacer doble *click* en el acceso directo que se generará en el escritorio. Otra forma de correr la aplicación consiste en ejecutar el archivo redelec.exe desde el directorio ARMANDO.

5.2. Entorno visual de REDELEC.

Cuando se inicia REDELEC, se presenta una pantalla principal desde la cual se selecciona el tipo de estudio que se desea realizar. Como se puede observar en la figura 5.0, se tienen cinco opciones, cada una de las cuales dará inicio a la pantalla de lectura de datos correspondiente a cada opción, la opción salir nos permite terminar la aplicación

Salir. La pantalla de inicio se muestra a continuación:

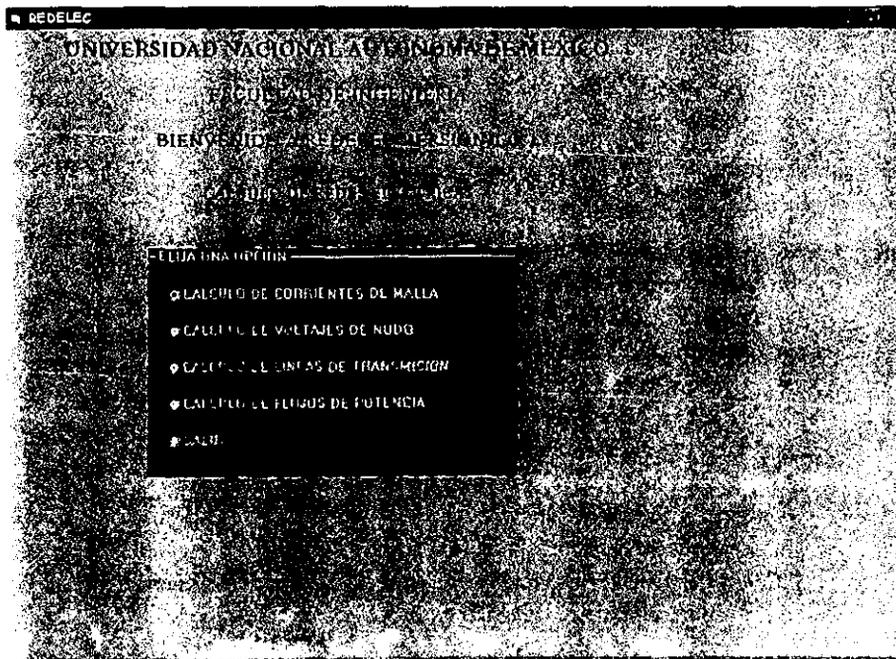


Figura 5.0. Pantalla de inicio.

5.2.1. Barra de menús.

Activar los procedimientos asociados a los menús desplegables situados en la barra de menús es muy común para manejar una aplicación WINDOWS. Esta barra proporciona las herramientas necesarias para desarrollar probar y archivar los estudios realizados.

El menú ARCHIVO contiene las ordenes para trabajar con archivos. Dentro de este se encuentra el submenú GUARDAR, con el cual se despliega el cuadro de diálogo que permite guardar los datos que requiere la aplicación. En este cuadro se deberá escribir el nombre del archivo sin extensión ya que el programa dará la extensión por omisión. Inmediatamente después de este cuadro de diálogo aparece otro con el botón guardar el cual se deberá presionar para que los archivos de datos se guarden correctamente. Cabe mencionar que durante el proceso GUARDAR se generan más de un archivo con el mismo nombre pero diferente extensión. El archivo con la extensión por omisión es el que contiene la información para el manejo adecuado de los archivos relacionados. El submenú GUARDAR COMO permite guardar los datos correspondientes al estudio realizado, con otro nombre. Al pulsar el submenú GUARDAR COMO se visualiza un cuadro de dialogo con el botón guardar, después de que se guarda el archivo de datos con otro nombre. Para completar el guardado de datos es necesario pulsar este botón.

El submenú ABRIR permite recuperar alguno de los circuitos previamente almacenados al guardar los datos correspondientes a cada figura del circuito. Con la opción ABRIR se recupera tanto los datos como el dibujo del circuito correspondiente a los archivos de datos previamente guardados.

De igual manera al seleccionar el submenú GUARDAR se despliega el cuadro de dialogo donde se selecciona el archivo con la extensión por omisión de cada caso analizado. Al pulsar el submenú ABRIR se despliega un cuadro de dialogo con el botón abrir, el cual se deberá presionar para que se abran los archivos de datos que permiten recuperar el circuito previamente guardado. El cuadro de diálogo aparece después de seleccionar el archivo que se desea abrir, en el cuadro de dialogo que aparece al pulsar el submenú ABRIR.

El submenú imprimir permite realizar una impresión de la pantalla actual. El submenú salir permite salir a la pantalla de inicio.

El siguiente menú que se presenta en la barra es DATOS. Como submenús de DATOS se encuentran otros menús que permiten manejar los datos que se introducen de forma directa, es decir que no se requieren dibujar el circuito para generar los datos necesarios para el estudio. Cada

uno de los submenús que contiene DATOS corresponde a un botón de la barra horizontal de herramientas.

El último menú de esta barra es VER, que permite visualizar o no la barra horizontal y la barra vertical. Dentro de este menú también se encuentra la opción para pasar a la pantalla de resultados correspondiente al problema que se esté resolviendo.

5.2.2. Barra de herramientas horizontal.

La barra de herramientas que se encuentra justo debajo de la barra de menús, contiene iconos que permiten introducir los datos de manera directa. Como se mencionó anteriormente, en la barra de menús se encuentran estas mismas opciones. La descripción detallada de cada botón se hace en los párrafos correspondientes a corrientes de malla, voltajes de nodo, líneas de transmisión y flujos de potencia.

5.2.3. Barra de herramienta vertical.

La barra de herramientas vertical se encuentra en la parte izquierda de la pantalla de lectura de datos. Esta barra contiene los iconos necesarios para realizar los dibujos de cada una de las cuatro opciones que tenemos. Similarmente, la descripción detallada se hace en los párrafos correspondientes a corrientes de malla, voltajes de nodo, líneas de transmisión y flujos de potencia.

5.3. Solución de redes eléctricas.

En este caso se obtendrán, para cualquier red eléctrica, tanto las corrientes de malla como los voltajes de nodo. Los datos necesarios para obtener la solución de la red serán introducidos a través de pantallas diseñadas mediante la aplicación del lenguaje de programación orientado a objetos *Visual Basic*. De igual manera, los resultados obtenidos serán leídos desde pantallas diseñadas especialmente para la lectura de los resultados.

5.3.1. Cálculo de corrientes de malla.

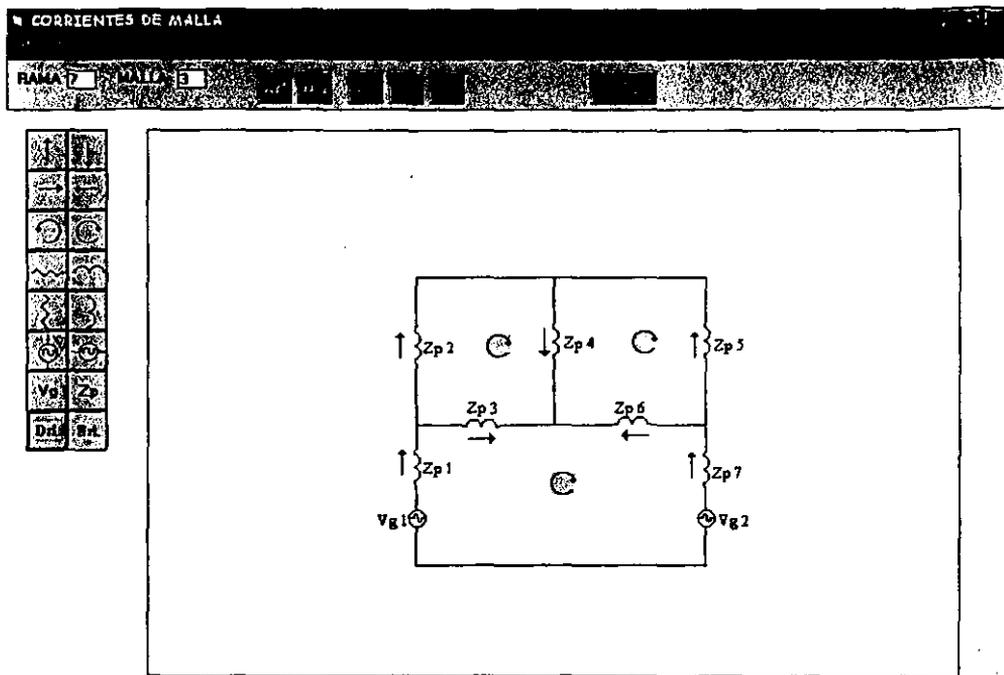


Figura 5.1. Pantalla para el cálculo de las corrientes de malla.

La pantalla que sirve de interfaz para introducir los datos, esta compuesta por una barra de herramientas horizontal y otra vertical. La barra de herramientas vertical contiene los elementos necesarios para dibujar la red eléctrica y la barra horizontal contiene los elementos necesarios para introducir datos directamente, así como para pasar a la pantalla de presentación de resultados. La figura 5.1 muestra la pantalla mediante la cual se introducen los datos.

Desde esta pantalla, también es posible guardar los datos después de que estos han sido leídos, además de que se puede pasar a cualquiera de las otras pantallas que componen la aplicación, por ser esta la pantalla principal de la aplicación.

Inicialmente se recomienda generar el dibujo completo con todas las ramas, mallas, generadores y etiquetas necesarias para entender correctamente el diagrama. Primeramente se deben realizar los trazos de las líneas que servirán de guía y darán tamaño al gráfico.

Sobre la barra de herramientas de cada pantalla que se utiliza para introducir datos, se encuentran los botones DrL y BrL, estos botones se muestran a continuación:



Para dibujar líneas se pulsa previamente el botón DrL, a continuación se comienza a dibujar la línea con un *click* sobre la pantalla, y se termina de dibujar la línea de la misma forma, con un *click* sobre la pantalla.

Si se desea borrar una línea previamente dibujada, entonces hay que pulsar el botón BrL (el puntero del ratón cambia de forma) y a continuación se comienza a borrar cualquier línea con un *click* sobre la pantalla al principio de la línea, y se termina de borrar con otro *click* sobre la pantalla al final de la línea.

Inmediatamente después se deben dibujar cada uno de los elementos que conformaran el dibujo de la red eléctrica.

5.3.1.1. Matriz de conexión rama-malla.

Para introducir los elementos de la matriz de conexión rama-malla en forma directa se requiere inicialmente leer el número de rama y el número de malla. La lectura de estos datos se realiza al momento de pulsar sobre los iconos correspondientes en la barra de herramientas horizontal, estos iconos se muestran a continuación:



Esta matriz se puede generar al pulsar el botón Am, cuando esto ocurre, se le piden al usuario los elementos de la matriz de conexión rama-malla.

La otra forma de generar la matriz de conexión rama-malla es mediante el manejo adecuado de los iconos que representan las direcciones de las corrientes de rama y de los iconos que representan las direcciones de las corrientes de malla. Estos iconos son los siguientes:



Para generar los elementos de la matriz de conexión rama-malla, sobre la barra de herramientas vertical, seleccionamos mediante un *click* con el botón izquierdo del ratón, el icono

correspondiente a la dirección de la corriente de rama que se necesite para empezar a dibujar la red eléctrica.

La figura que indica la dirección de la corriente de rama se hace visible en la parte superior izquierda de la pantalla. Esta figura es posible moverla sobre la pantalla, para lograrlo se requiere hacer *click* sobre ésta, enseguida se puede mover la figura haciendo *click* en cualquier lugar de la pantalla.

Posteriormente se selecciona otra figura y se realiza el mismo procedimiento para moverla hasta el lugar requerido. El procedimiento se realiza hasta que se tienen el total de figuras que representan las direcciones de las corrientes de rama y las corrientes de malla.

En caso de que el usuario necesite borrar alguna de las figuras generadas, basta hacer *click* con el botón derecho sobre la figura. El echo de borrar una figura altera el contéo de figuras y en consecuencia se altera el orden y las características de la matriz de conexión rama-malla. Para corregir esta situación se requiere guardar la información del dibujo y cerrar la aplicación, para después abrirla nuevamente y recuperar el dibujo. Los procedimientos para borrar, guardar y recuperar los dibujos se explicarán con detalle más adelante.

Después de que se han colocado sobre la pantalla el total de las figuras que representan las direcciones de las corrientes de rama y corrientes de malla, se puede realizar la lectura de la matriz de conexión rama-malla.

La lectura de la matriz de conexión rama-malla se realiza por renglón. Cada renglón esta representado por las diferentes figuras que representan la dirección de las corrientes de rama.

Las columnas de la matriz están representadas por las dos posibles figuras que representan las direcciones de las corrientes de rama.

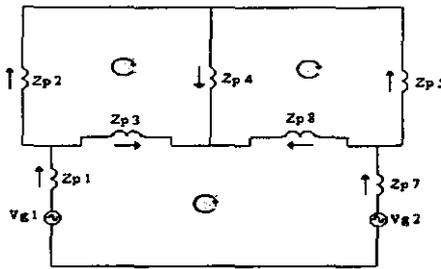
Al estar representados sobre la pantalla el total de renglones y el total de columnas de la matriz de conexión rama-malla, se procede a introducir los elementos de la matriz por renglón, para ello se debe tener identificado el orden en que fueron colocadas sobre la pantalla las figuras que representan la dirección de las corrientes de rama.

Se empieza pulsando dos veces sobre la figura que representa la dirección de la corriente de rama 1. Al hacer esto, se le pregunta al usuario si la corriente de malla 1, pasa por la rama 1 y si tiene el mismo sentido que la corriente en la rama 1. Se le hacen las mismas preguntas al usuario para todas las mallas que contenga la red eléctrica.

Posteriormente, siguiendo el mismo procedimiento, se introducen los elementos correspondientes al renglón 2, renglón 3, etc., dependiendo del número de renglones que contenga la matriz de conexión rama-malla.

En caso de ser necesario introducir nuevamente los elementos de algún renglón de la matriz de conexión rama-malla, esto se logra pulsando dos veces sobre cualquiera de las figuras que representa la dirección de las corrientes de rama. Con esto se entiende que los datos se pueden modificar por renglón, sin que sea necesario volver a leer toda la matriz.

En la siguiente figura:



En el momento en que el usuario selecciona y hace doble *click* sobre la figura que representa la corriente de rama 2, se le pregunta si la corriente de malla 1 pasa por la rama 2 y si tienen el mismo sentido, enseguida se le pregunta si la corriente de malla 2 pasa por la rama 2 y si tienen el mismo sentido, finalmente se le pregunta si la corriente de malla 3 pasa por la rama 2 y si tienen el mismo sentido. Este procedimiento se deberá realizar para las otras 7 corrientes de rama. De la misma forma se realizarán preguntas similares a las anteriores por cada corriente de rama seleccionada.

5.3.1.2. Matriz de impedancias de rama.

Para ingresar los datos de la matriz de impedancias de rama en forma directa, basta hacer *click* en el botón Z_p de la barra horizontal. Para esto, anteriormente se debió ingresar el número de malla y ramas de la red.

Los íconos que se muestran a continuación son útiles para dibujar las impedancias de la red eléctrica.



Para dibujar las impedancias de la red eléctrica, sobre la barra de herramientas vertical, seleccionamos mediante un *click* con el botón izquierdo del ratón, el ícono correspondiente a la figura que se necesite.

La figura en la parte superior izquierda de la pantalla. De manera similar a las direcciones de rama, esta figura es posible moverla sobre la pantalla, para lograrlo se requiere hacer *click* sobre ésta, enseguida sólo basta moverla hasta el lugar requerido y hacer nuevamente *click*.

Para identificar cada una de las impedancias de rama se utiliza el ícono Zp de la barra vertical. Como cualquier otra figura al momento de presionar el ícono Zp aparecerá en la parte superior de la pantalla la etiqueta Zp. Para arrastrarla sólo basta hacer *click* sobre la etiqueta y arrastrar hasta la posición adecuada. Con este procedimiento se realiza de forma automática el conteo y etiquetado de las impedancias.

Para introducir los elementos de la matriz de impedancias de rama, se debe seleccionar la etiqueta de la primera matriz de rama y hacer doble *click*, enseguida se pide el valor de la parte real de la impedancia 1, y después se pide el valor de la parte imaginaria de la impedancia 1. Este procedimiento se deberá realizar para cada una de las impedancias de rama que contenga la red eléctrica.

5.3.1.3. Vector de voltajes de generación.

La forma de ingresar los datos de voltajes de generación en forma directa es mediante el uso del botón Vg de la barra horizontal. Al momento, se le pide al usuario la parte real y la parte imaginaria del voltaje de generación 1. Enseguida se le pide al usuario la parte real y la parte imaginaria del voltaje de generación 2. Así consecutivamente hasta cubrir el total de mallas de la red.

La otra alternativa para ingresar los datos es desde la barra de herraminetas horizontal. Las figuras que representan a los generadores de la red son:



Como cualquier otra figura, al pulsar sobre el botón aparecerá la figura en la parte superior izquierda de la pantalla. Para moverla, se requiere hacer doble *click* sobre ésta y arrastrar sobre la pantalla hasta la posición requerida.

El ícono que representan a la etiqueta Vg, se seleccionan sobre la barra de herramientas vertical mediante un *click* con el botón izquierdo del ratón. Al igual que en los casos anteriores, es

necesario seleccionar con un *click* sobre la etiqueta y así poder moverla sobre la pantalla hasta el lugar donde el usuario desee moverlas.

Para introducir los valores de los voltajes de generación, se requiere hacer *click* en la etiqueta del generador, entonces se pedirá al usuario el valor de la parte real y de la parte imaginaria. Por ejemplo, si se pulsa dos veces con el botón izquierdo del ratón sobre la etiqueta V_{g1} , entonces se le pide al usuario la parte real y la parte imaginaria del voltaje de generación 1. De esta manera, es posible introducir o cambiar el valor de un dato en particular.

5.3.1.4. Lectura de resultados.

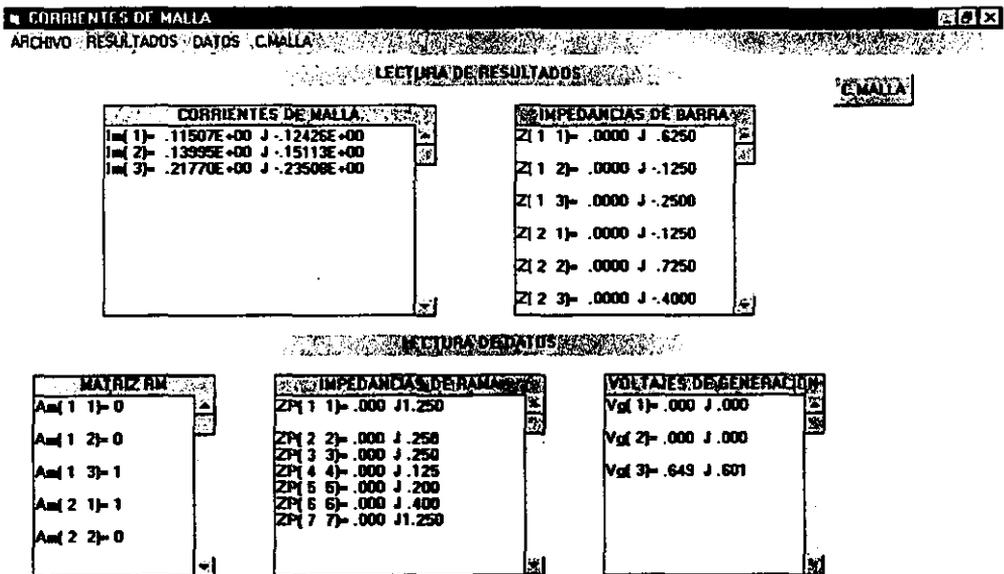


Figura 5.2. Pantalla de lectura de resultados y datos (corrientes de malla).

El icono que permite visualizar la pantalla de presentación de resultados se muestra a continuación:



Recordemos que también se puede visualizar la pantalla desde el menú VER, seleccionando corrientes de malla.

Al pulsar este ícono o seleccionar desde el menú se visualiza la pantalla de presentación de resultados, sobre esta pantalla se encuentra el mismo ícono, el cual al momento de pulsarlo hace posible que corra el archivo ejecutable que calcula las corrientes de malla y la matriz de admitancias de barra. La pantalla de presentación de resultados se muestra en la figura 5. 2.

Para ver los resultados, se pulsa sobre la etiqueta correspondiente en la pantalla de presentación de resultados. Por ejemplo, si queremos ver las corrientes de malla sólo hay que pulsar sobre la etiqueta corrientes de malla. Esta etiqueta es fácilmente identificable sobre la pantalla de presentación de resultados.

También se pueden visualizar los datos sobre esta pantalla, la forma de hacerlo es, al igual que para ver los resultados, pulsando sobre las etiquetas correspondientes. La ventaja de ver los datos, es que se puede confirmar que sean los correctos.

5.3.2. Cálculo de los voltajes de nodo.

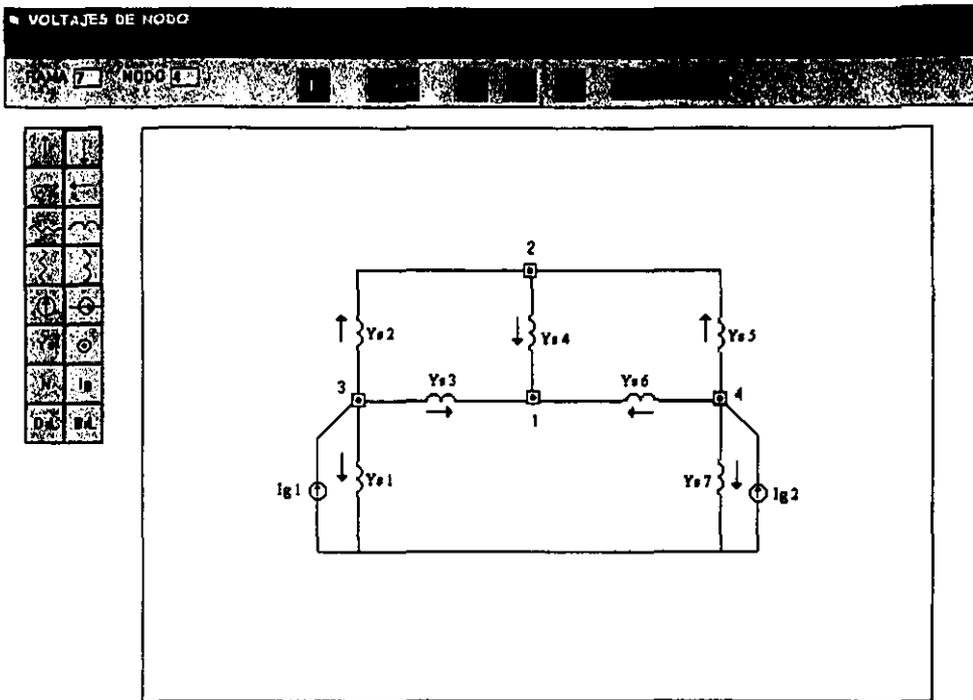


Figura 5.3. Pantalla para el cálculo de voltajes de nodo.

Desde la pantalla de inicio se tiene acceso a la pantalla a través de la cual se introducen los datos necesarios para calcular los voltajes de nodo y la matriz de admitancias de barra, esta pantalla se muestra en la figura 5.3.

Del mismo modo que para el cálculo de las corrientes de malla, los datos se guardan en archivos. Con estos datos, el archivo ejecutable que calcula los voltajes de nodo y la matriz de admitancias de barra, genera los archivos de resultados que son leídos posteriormente desde la pantalla de presentación de resultados. La pantalla de presentación de resultados se muestra en la figura 5.5.

5.3.2.1. Matriz de conexión rama-nodo.

Al igual que en el caso de mallas, los datos se pueden introducir de forma directa o a través del dibujo del circuito que representa la red eléctrica que se quiere resolver. El número de ramas y el

número de nodos se introducen con ayuda de los siguientes botones de la barra de herramientas horizontal.



La matriz se genera al momento de pulsar el botón An sobre la barra de herramientas horizontal, cuando esto ocurre, se le piden al usuario cada uno de los elementos de la matriz de conexión rama-nodo.

Otra forma de generar la matriz de conexión rama-nodo es mediante el uso de los siguientes íconos:



Las flechas indican las diferentes direcciones de las corrientes de rama, y el cuadro con el círculo inscrito representa a los nodos de la red eléctrica.

Cuando se pulsa cualquiera de las figuras que representan las direcciones de las corrientes de rama o la que representan a los nodos de la red, éstas se visualizan sobre la pantalla en la parte superior izquierda. Para arrastrar las imágenes sobre la pantalla, es necesario hacer un *click* con el botón izquierdo y arrastrar la figura sobre la pantalla.

5.3.2.2. Procedimiento para generar la matriz de conexión rama-nodo.

Después de seleccionar los íconos que representan las direcciones de las corrientes de rama, se procede a seleccionar sobre la misma barra de herramientas vertical, el ícono que representa a los nodos de la red.

Cuando ya se tienen el total de ramas y nodos, se procede a iniciar la lectura de datos. Se selecciona la primera corriente de rama con un *click* e inmediatamente después se hacen dos *clicks*, en ese instante se le pregunta al usuario si la rama se conecta al nodo, y si la corriente de rama entra o sale del nodo, si la red tiene más de un nodo, entonces se realizan tantas preguntas como nodos tiene la red.

Tomando como ejemplo la figura 5.4, si seleccionamos y hacemos doble *click* en la corriente de rama 2, se le pregunta al usuario si la rama 2 se conecta al nodo 1 y si la corriente en la rama 2 entra o sale del nodo 1, enseguida se le pregunta al usuario si la rama 2 se conecta al nodo 2 y si la corriente en la rama 2 entra o sale del nodo 2, para el nodo 3 se le pregunta al usuario si la rama 2 se conecta al nodo 3 y si la corriente en la rama 2 entra o sale del nodo 3. Finalmente se le

pregunta al usuario si la rama 2 se conecta al nodo 4 y si la corriente en la rama 2 entra o sale del nodo 4.

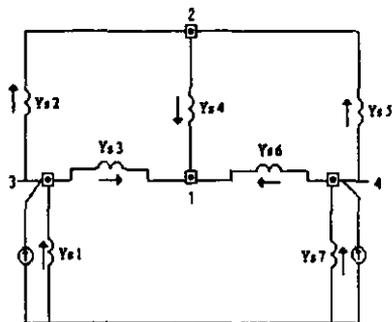


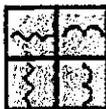
Figura 5.4. Ejemplo para generar la matriz de conexión rama-nodo.

En el circuito de la figura 5.4, se observa que la rama 1 se conecta al nodo 3 y la corriente en la rama 1 llega al nodo 3. Las ramas 2 y 3 también se conectan al nodo 3, pero las corrientes en las ramas 2 y 3 salen del nodo 3.

5.3.2.3. Lectura de impedancias de rama.

Con ayuda del botón Yp de la barra de herramientas horizontal introducimos los datos de las impedancias de rama. Recordemos que previamente se debió introducir los datos del número de ramas y del número de nodos.

Sabemos que la admitancia es el inverso de la impedancia, es por eso que los íconos que se utilizan para generar la matriz de admitancias de rama son los mismos que se utilizan para generar la matriz de impedancias de rama. Estos íconos son los siguientes:



La impedancia se compone de resistencia y reactancia inductiva o capacitiva, si se requiere dibujar la representación de la impedancia, sólo hay que seleccionar los íconos necesarios sobre la barra de herramientas vertical. A continuación se muestra la representación de una impedancia de rama.



El dato correspondiente a la impedancia se puede introducir pulsando dos veces sobre la etiqueta que la identifica, por ejemplo, si se quiere introducir el dato correspondiente a la impedancia de rama 1, sólo hay que pulsar dos veces sobre la etiqueta Zp1.

5.3.2.4. Vector de corrientes de generación.

Para introducir el dato correspondiente a cada corriente de generación, sólo hay que pulsar dos veces sobre la etiqueta que la representa, por ejemplo si se desea introducir el dato correspondiente a la corriente de generación 1, se necesita pulsar dos veces sobre la etiqueta Ig1. La forma de introducir los datos correspondientes al vector de corrientes de generación de forma directa, consiste en pulsar sobre el botón Ig que está contenido en la barra de herramientas horizontal. Este botón se muestra a continuación:



Al pulsar este botón se le pide al usuario la parte real y la parte imaginaria de cada una de las corrientes de generación.

5.3.2.5. Lectura de resultados.

LECTURA DE RESULTADOS

VOLTAJES DE NODO

Vn(1)-	32856E+00	J	-25798E+00
Vn(2)-	32521E+00	J	-30181E+00
Vn(3)-	35826E+00	J	-27223E+00
Vn(4)-	69493E+00	J	-32982E+00

ADMITANCIAS DE BARRA

Y(1 1)-	.000	J	-14.5000
Y(1 2)-	.000	J	8.0000
Y(1 3)-	.000	J	4.0000
Y(1 4)-	.000	J	2.5000
Y(2 1)-	.000	J	8.0000
Y(2 2)-	.000	J	-17.0000
Y(2 3)-	.000	J	4.0000

V.NODO

LECTURA DE DATOS

MATRIZ RHO

An(1 1)-	0
An(1 2)-	0
An(1 3)-	1
An(1 4)-	0
An(2 1)-	0
An(2 2)-	-1

ADMITANCIAS DE BARRA

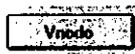
YP(1 1)-	.000	J	-8.000
YP(2 2)-	.000	J	-4.000
YP(3 3)-	.000	J	-4.000
YP(4 4)-	.000	J	-8.000
YP(5 5)-	.000	J	-5.000
YP(6 6)-	.000	J	-2.500

CORRIENTES DE GENERACION

Ig(1)-	.000	J	.000
Ig(2)-	.000	J	.000
Ig(3)-	.000	J	-1.0
Ig(4)-	.481	J	-481

Figura 5.5. Pantalla de lectura de resultados (voltajes de nodo).

Una vez que los datos han sido generados, se procede a correr desde VB el archivo ejecutable que calcula los voltajes de nodo y la matriz de admitancias de barra, esto se hace al pulsar el botón Vnodo contenido en la barra de herramientas horizontal de la figura 5.5. Este botón se muestra a continuación:



Al pulsar este botón se muestra la pantalla de presentación de resultados, sobre esta pantalla se encuentra el mismo botón, el cual al pulsarlo ocasiona que corra el archivo ejecutable que calcula los voltajes de nodo. Los resultados se guardan en archivos y estos archivos son leídos desde la pantalla de presentación de resultados.

Para leer los resultados es suficiente con pulsar sobre la etiqueta correspondiente en la pantalla de presentación de resultados, por ejemplo, si se quiere leer los voltajes de nodo, solo hay que pulsar sobre la etiqueta voltajes de nodo, la cual es fácilmente identificable sobre la pantalla de presentación de resultados.

De igual manera, es posible leer los datos previamente introducidos con tan sólo pulsar sobre la etiqueta correspondiente en la pantalla de presentación de resultados, por ejemplo, si queremos leer las admitancias de barra, sólo hay que pulsar sobre la etiqueta admitancias de barra. Ver fig. 5.5, donde se muestra la pantalla de presentación de resultados.

5.4. Cálculo de líneas de transmisión.

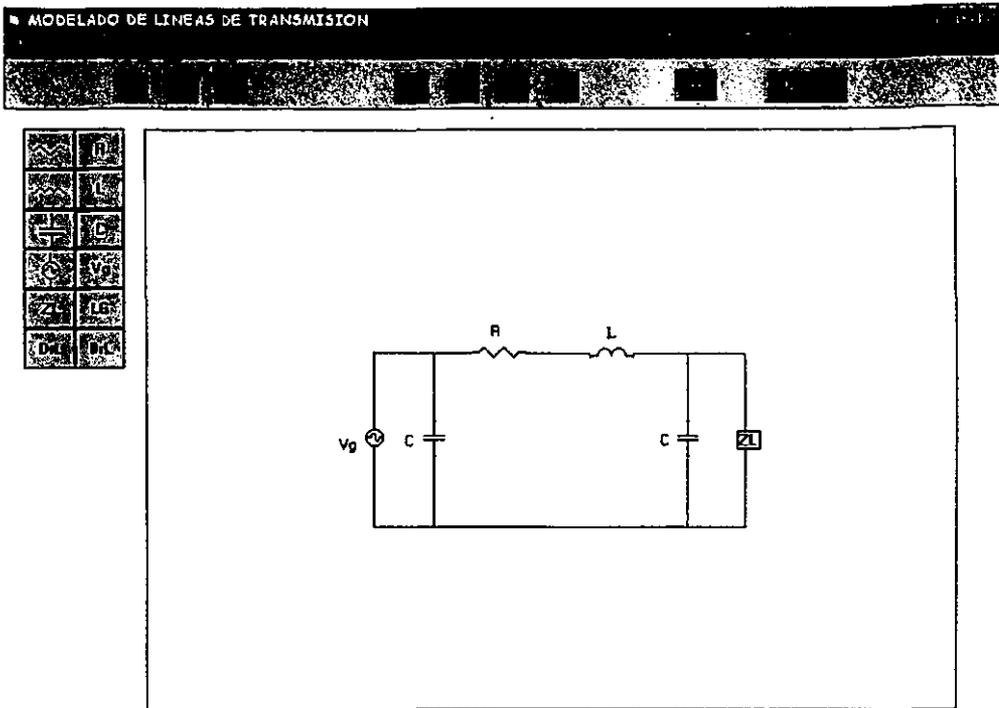


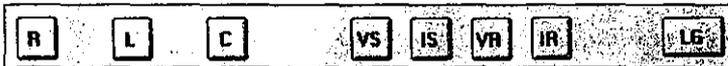
Figura 5.6. Pantalla para el cálculo de líneas de transmisión.

La pantalla de inicio de la aplicación permite pasar a la pantalla mediante la cual se introducen los datos necesarios para el cálculo de las líneas de transmisión, esto se logra al seleccionar Cálculo de líneas de transmisión. Después de que se ha seleccionado la opción, se muestra la pantalla a través de la cual se introducen los datos para el cálculo de las líneas de transmisión. Esta pantalla se muestra en la fig. 5.6. Las líneas de transmisión se clasifican dependiendo de su longitud en líneas cortas, medias y largas. El cálculo de las variables eléctricas requiere como datos los parámetros de la línea y el voltaje y corriente en los extremos receptor o transmisor. Dependiendo de los datos de voltaje y corriente, se obtienen las variables eléctricas ya sea en el extremo receptor o en el extremo transmisor.

Observando la pantalla que se muestra en la fig. 5.6, nos podemos dar cuenta que los valores de los parámetros de la línea y los valores de voltaje y corriente en cualquiera de los dos extremos

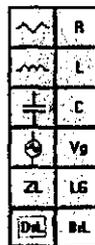
(receptor o transmisor) se pueden introducir a través de la barra de herramientas horizontal, esto se logra con tan sólo pulsar sobre el botón correspondiente.

Por ejemplo, si se pulsa sobre el botón R, se le pregunta al usuario el valor de la resistencia de la línea, si se pulsa sobre el botón Vs, se le pedirá al usuario la parte real y la parte imaginaria del voltaje en el extremo transmisor. Los botones que contiene la barra de herramientas horizontal y que al pulsarlos permiten introducir los datos para el cálculo de líneas, se describen a continuación:



Donde R es la resistencia, L la inductancia, C la capacitancia, Vs es el voltaje en el extremo transmisor, Is es la corriente en el extremo transmisor, Vr es el voltaje en el extremo receptor, Ir es la corriente en el extremo receptor y Lg es la longitud de la línea.

La barra de herramientas vertical contiene los íconos que representan a los parámetros de la línea, la carga, el voltaje en el extremo receptor, el voltaje en el extremo transmisor y sus correspondientes etiquetas. Estas herramientas permiten dibujar el modelo π o T de la línea de transmisión, la figura correspondiente se muestra enseguida.



Otra forma de introducir los datos necesarios para el cálculo de líneas, es pulsando una vez con el botón izquierdo del ratón sobre las etiquetas que representan a los parámetros de la línea, al hacer esto se seleccionan, y mediante un doble *click* se le pide al usuario el dato correspondiente a la etiqueta. Por ejemplo, si se pulsa dos veces sobre la etiqueta Vg, se le pedirá al usuario la parte real y la parte imaginaria del voltaje en el extremo transmisor.

5.4.2 Lectura de resultados.

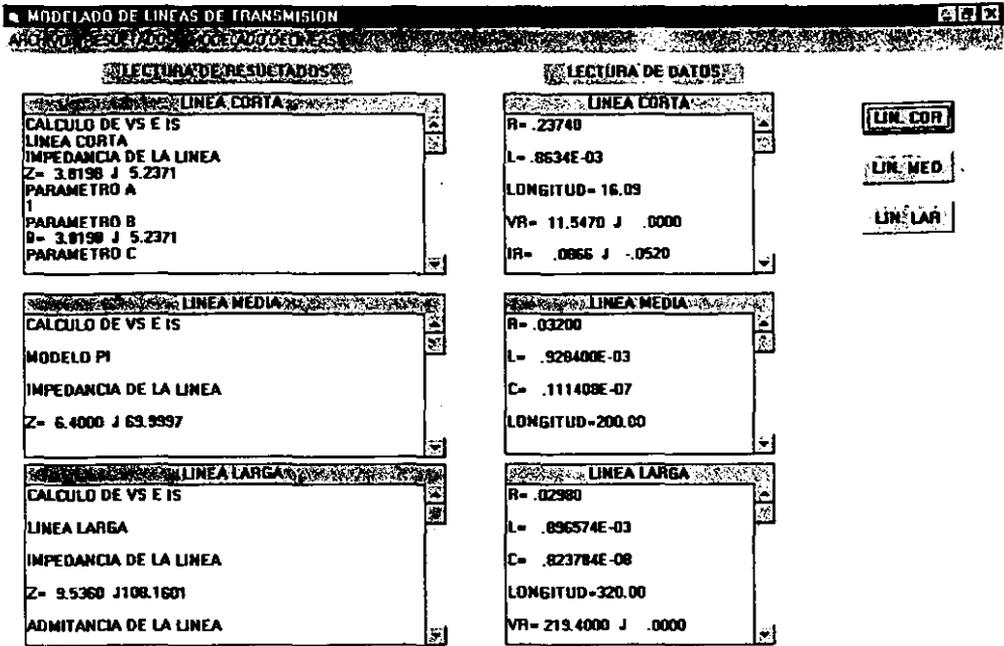


Figura 5.7. Pantalla de lectura de resultados referentes a la línea de transmisión.

Si los datos que se utilizan para el cálculo de la línea ya fueron generados, entonces se pulsa el botón modelado de líneas (figura 5.6), para que se visualice la pantalla de presentación de resultados, o se selecciona la opción modelado de líneas del menú VER. La pantalla de presentación de resultados se muestra en la fig. 5.7.

La pantalla de presentación de resultados contiene tres botones que se utilizan para correr desde *Visual Basic* los archivos ejecutables que calculan las variables eléctricas de la línea de transmisión, estos botones se muestran a continuación:



Al pulsar cualquiera de estos botones se obtienen los resultados correspondientes al tipo de línea, de igual manera, estos resultados se envían a archivos. Los archivos de resultados son leídos desde

la pantalla de presentación de resultados, para ello sólo es necesario pulsar sobre la etiqueta correspondiente al tipo de línea, por ejemplo, si se desea ver los resultados correspondientes a línea corta, sólo hay que pulsar sobre la etiqueta línea corta. Observar la fig. 5.7, donde se muestra la pantalla de presentación de resultados.

Si se desea leer los resultados correspondientes a línea media, entonces se debe pulsar sobre la etiqueta línea media, al hacer esto aparece una etiqueta en la parte inferior izquierda de la pantalla donde se le pide al usuario que indique el tipo de modelo que utilizó para el cálculo.

De igual manera se pueden leer los datos que se introdujeron previamente, con tan solo pulsar sobre la etiqueta correspondiente al tipo de línea (observar la fig. 5.7).

5.5. Cálculo de flujos de potencia (método de Gauss-Seidel).

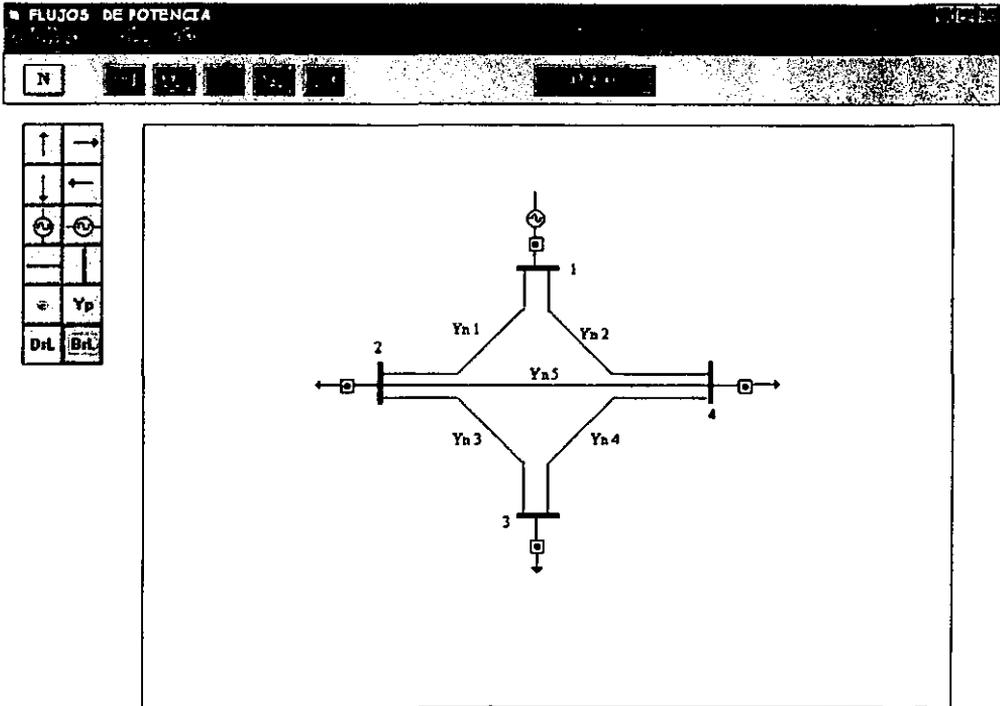


Figura 5.8. Pantalla para el cálculo de flujos de potencia.

Desde la pantalla de inicio de la aplicación se pasa a la pantalla mediante la cual se introducen los datos necesarios para el cálculo de flujos de potencia, esta pantalla se muestra en la fig. 5.8.

Para calcular los flujos de potencia se consideran tres tipos diferentes de nodos, los datos que se introducen para cada tipo de nodo son los siguientes:

Nodo *Slack* (de referencia).

- a) Voltaje.

Nodo de carga.

- a) Potencia real.
- b) Potencia reactiva.

Nodo de generación.

- a) Voltaje.
- b) Ángulo de voltaje.
- c) Límite inferior de potencia reactiva.
- d) Límite superior de potencia reactiva.

Otros datos que se necesitan para el cálculo de flujos son: factor de aceleración, tolerancia, número de nodos y la matriz de admitancias de barra. Estos datos se pueden introducir en forma directa desde la barra de herramientas horizontal, los botones mediante los cuales se introducen estos datos, se muestran a continuación:



Al pulsar sobre estos botones, se le pide al usuario el dato correspondiente al botón pulsado. Los datos referentes a cada tipo de nodo se pueden introducir además al pulsar dos veces sobre el siguiente icono:



Para que este icono se visualice en la parte superior izquierda de la pantalla, sólo hay que seleccionarlo sobre la barra de herramientas vertical. Para mover esta figura sobre la pantalla, se requiere pulsarla una vez para que se seleccione y posteriormente pulsar sobre la pantalla en el lugar al que se desee moverla.

Para que se le pidan al usuario los datos referentes al tipo de nodo, sólo tiene que pulsar una vez sobre la figura que representa al nodo para que ésta se seleccione, y pulsarla dos veces para que se visualice una etiqueta en la parte inferior izquierda de la pantalla, donde se le indica al usuario la manera de elegir el tipo de nodo.

Los iconos restantes que se muestran sobre la barra de herramientas vertical (fig. 5.8), sirven como auxiliares en el dibujo de la red eléctrica.

La etiqueta N se utiliza para numerar los nodos, el procedimiento para mover esta etiqueta sobre la pantalla, es el mismo que se utiliza para mover el icono que representa a los nodos de la red. Para que esta etiqueta se visualice sobre la pantalla, sólo hay que seleccionarla sobre la barra de herramientas vertical, al estar sobre la pantalla y pulsarla, cambia a 1, 2, 3, etc.

5.5.2. Lectura de resultados.

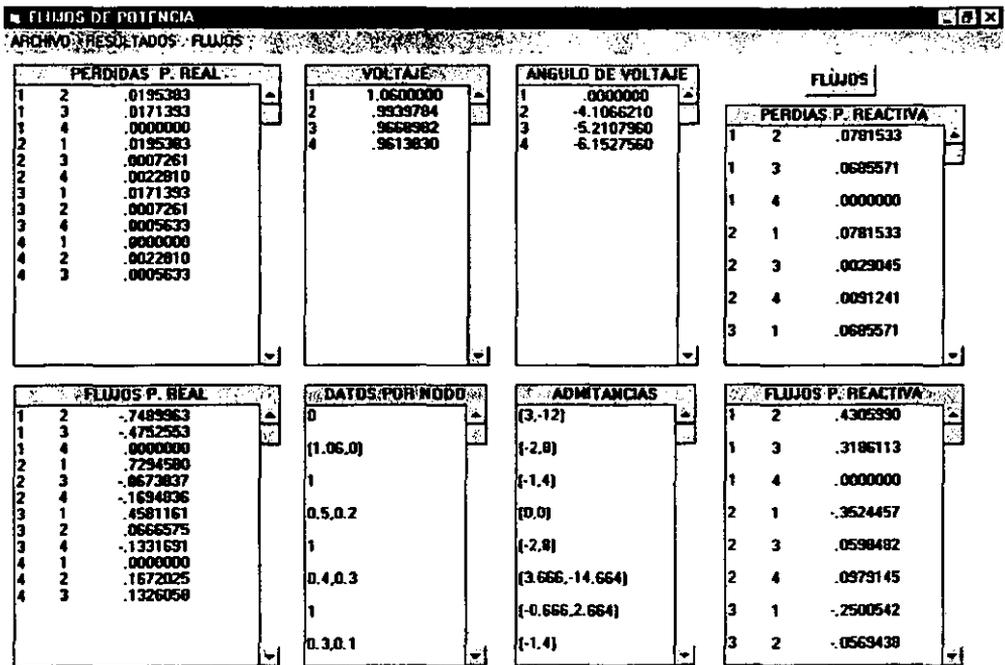


Figura 5.9. Pantalla para lectura de resultados referentes a flujos de potencia.

Una vez que se generaron los datos, se procede a pulsar el botón flujos o a seleccionar la opción flujos del menú VER. El botón flujos se muestra sobre la barra de herramientas horizontal de la fig. 5.8. Al pulsar el botón flujos se muestra la pantalla de presentación de resultados, sobre esta pantalla aparece nuevamente el botón flujos, al momento de pulsar este botón, se ejecuta el archivo que calcula los flujos de potencia, los resultados que genera este archivo, son enviados a

archivos de resultados, estos archivos son leídos desde la pantalla de presentación de resultados, esta pantalla se muestra en la fig. 5.9.

Para leer los resultados, es necesario pulsar sobre la etiqueta correspondiente en la pantalla de presentación de resultados, por ejemplo, si se quiere conocer las pérdidas de potencia real, sólo hay que pulsar sobre la etiqueta pérdidas de potencia real, esta etiqueta es fácilmente identificable sobre la pantalla de presentación de resultados.

5.6. Opciones para editar el dibujo de la red eléctrica.

Cada uno de los dibujos que se generan durante los procesos de estudio de redes eléctricas, se pueden recuperar habiendo guardado previamente las propiedades de cada una de las figuras. Esto se logra debido a que cada que guardamos una figura se guardan datos relacionados con su identificación y posición en la pantalla.

Al almacenar la identificación de las figuras, es posible abrirlas nuevamente y mantener la relación de las figuras con los archivos de datos relacionados a estas. De esta forma, para el caso de corrientes de malla, al recuperar el dibujo de la red eléctrica no es necesario generar nuevamente la matriz de conexión rama-malla, ya que existe la posibilidad de introducir los elementos de la matriz por renglón a partir de las figuras que representan las direcciones de corrientes de rama, como se explica en párrafos anteriores.

Si durante el proceso de generación de los dibujos, el usuario requiere borrar alguna de las figuras, se puede hacer de forma sencilla, sólo basta hacer clic con el botón derecho del mouse sobre la figura que se desee borrar.

El hecho de borrar una figura altera su numeración. Para corregir esta situación y poder pasar a la lectura de los datos, el usuario debe guardar el dibujo y cerrar la aplicación, para después abrir nuevamente la aplicación y recuperar el dibujo del circuito.

Al momento de dibujar algún circuito, REDELEC ofrece la opción de ver las coordenadas horizontal y vertical del punto de la línea que se esté dibujando. Para lograr esto, el usuario debe pulsar la letra C mayúscula o minúscula, con esto se visualiza la ventana que contiene las coordenadas antes mencionadas, esta ventana contiene además la opción salir para hacer que se oculte dicha ventana en el momento que ya no sea requerida.

Conclusiones.

En la actualidad los Sistema Eléctricos de Potencia (SEP) son tan grandes y complejos, que requieren del uso de programas de cómputo especializados para su análisis. Si no se contara con la ayuda de las computadoras y de los simuladores, hacer un análisis de un SEP sería muy tardado y complicado.

En la Facultad de Ingeniería se cuenta con los equipos de cómputo adecuados para que los alumnos puedan recurrir al uso de simuladores para comprender mejor los fenómenos eléctricos. Lamentablemente, los estudiantes de ingeniería eléctrica no tienen acceso a los paquetes de software comerciales y mucho menos a simuladores propios, que les ayuden a entender mejor los fenómenos físicos relacionados con la Ingeniería, en sus diferentes disciplinas.

El uso de este tipo de simuladores reduce la posibilidad de errores humanos. Los operadores de Sistemas Eléctricos de Potencia se enfrentarían a problemas reales generados aleatoriamente por el simulador o, a problemas previamente establecidos y analizados. Los alumnos de la Facultad de Ingeniería tendrían la facilidad de realizar tareas, prácticas e investigación con dichos paquetes de software.

Una de las finalidades de la aplicación que se desarrolla en esta tesis, es que los estudiantes de Ingeniería Eléctrica se familiaricen y acostumbren a utilizar simuladores comerciales de características similares con el fin de que entiendan fácilmente los fenómenos físicos propios de la Ingeniería Eléctrica. La aplicación que se desarrolla en esta tesis no alcanza las posibilidades de un simulador profesional, pero a cambio de esto se pretende que sea didáctico y de fácil uso para los estudiantes de Ingeniería Eléctrica.

REDELEC ofrece la plataforma Windows para interactuar con la computadora, con lo que se logra un sistema amigable y sencillo, permite el fácil manejo de datos. La integración de barras de herramientas facilita el trabajo de dibujo, agregando elementos previamente diseñados y listos para ser editados en la pantalla de lectura de datos. La integración de elementos de dibujo o datos se realiza mediante clics del ratón, lo cual es parte de lo cotidiano cuando se trabaja bajo plataforma Windows.

El hecho de tener la limitante del número de nodos (máximo 15 nodos), permite que los cálculos se realicen de manera rápida, dando como resultado la posibilidad de analizar diferentes condiciones de operación en menos tiempo. Lo que facilitará al alumno el mejor entendimiento del comportamiento de los Sistemas Eléctricos de Potencia.

El método interactivo de solución de utilizado en esta tesis para obtener la solución de flujos de potencia (método de Gauss-Seidel) y el método de descomposición LU, utilizado para obtener las corrientes de malla y los voltajes de nodo, son métodos que facilitan la obtención de resultados, puesto que la programación de estos métodos en lenguaje Fortran resulta sencilla. La programación de estos métodos permite realizar diferentes corridas con diferentes datos, los cuales se manejan desde las Interfases Gráficas de usuario desarrolladas con base en Visual Basic. Con esto se observa el comportamiento de los SEP bajo diferentes condiciones de operación.

La ventaja de esta forma de interactuar con la computadora es la facilidad y flexibilidad del manejo de datos. Cuando se utilizan programas tradicionales, la lectura y escritura se realiza desde un archivo, con formato específico, o desde un listado en la pantalla. La lectura de datos mediante archivos implica conocer siempre el formato, tanto de lectura como de escritura. La lectura de datos mediante un listado en la pantalla se debe realizar sin errores, porque cualquier error en la introducción de datos, implica introducir nuevamente todos los datos.

El hecho de tener a la mano una herramienta como REDELEC, permite realizar comparaciones de métodos, como lo es resolver una red eléctrica aplicando el método de mallas y el método de nodos, además de comprobar las leyes que rigen el comportamiento eléctrico, como lo son las leyes de Kirchoff. También facilita la comprensión de los conceptos que se aprenden en los diferentes cursos relacionados con la ingeniería eléctrica.

Con el módulo de operación de líneas, el usuario de REDELEC entenderá mejor los conceptos de regulación, eficiencia, factor de potencia, etc., y sus efectos sobre la operación de las líneas, además de comprender los conceptos de línea corta, media y larga.

Existen otros conceptos relacionados con las líneas y con otros sistemas, tales como los diagramas fasoriales, los cuales se entienden de mejor forma cuando se cuenta con los resultados que arroja REDELEC en diferentes corridas.

Este sistema sirve de apoyo para resolver los ejercicios propuestos en clase y en los diferentes libros de texto, el alumno puede comparar resultados, para después sacar conclusiones.

El módulo de flujos de potencia de REDELEC permite, como los casos anteriores, entender los conceptos tales como efectos de la potencia real sobre el ángulo de fase, efectos de la potencia reactiva sobre la caída de voltaje, pérdidas de potencia, etc., Este módulo también puede ser utilizado para estudiar una línea de transmisión, sin recurrir a los conceptos de líneas corta, media

o larga, simplemente se estudia un sistema sencillo formado por un generador, una línea de transmisión, y una carga.

El módulo de flujos de potencia de REDELEC nos permitirá observar las consecuencias de los cambios que se sugieran en los nodos del SEP en estudio, tales como agregar una línea, modificar las condiciones de la carga, regular la potencia reactiva en los nodos de generación, etc., con fines de planeación y diseño de los Sistemas Eléctricos de Potencia.

Con el uso de programas como REDELEC, los profesores de la Facultad de Ingeniería podrían tener más elementos para definir una calificación, podrían ir más allá del hecho de resolver los problemas de flujos de potencia o de líneas de transmisión, los profesores podrían hacer que los alumnos resuelvan dos o más ejemplos y tomen sus propias conclusiones, las cuales las presentarían en clase.

Todo sistema puede ser mejorado, al ponerlo en operación se detectan las limitaciones y las posibles deficiencias, por esto surgen nuevas versiones de los sistemas. Es importante que los estudiantes de Ingeniería se involucren con el desarrollo de nuevo paquetes de software y que cada día se den mejoras en ésta y otras aplicaciones que se desarrollen en la Facultad de Ingeniería. La importancia del desarrollo de estos paquetes de software radica en la posibilidad de aplicar los conocimientos adquiridos en las diferentes asignaturas que imparte la Facultad de Ingeniería. Como ya se ha dicho Visual Basic es una herramienta de programación bajo plataforma Windows, que no requiere de grandes conocimientos en materia de programación, los estudiantes del área eléctrica pueden entender fácilmente las estructuras de programación de Visual Basic.

Para lograr el cumplimiento de los objetivos, el archivo fuente se pondrá a disposición de los alumnos. Ellos podrán realizar ampliaciones y modificaciones de acuerdo a lo que más convenga para cubrir las necesidades de las asignaturas del área eléctrica.

El programa REDELEC, para fines de esta tesis, tiene un alcance definido, desafortunadamente, el hecho de ampliar el alcance llevaría más tiempo de programación e investigación. Las ampliaciones o modificaciones que surgieran, podrían ser tema de tesis posteriores, con lo que se llevaría un control de las modificaciones y el paquete de software crecería bajo el control de la coordinación de la carrera.

Bibliografía.

1. - **Prospectiva del Sector Eléctrico 1997-2006**
Secretaría de Energía
México, 1997

2. - **El Sector Eléctrico de México**
Daniel Reséndiz Núñez
Fondo de Cultura Económica
México, 1994

3. - **Hacia el Siglo XXI**
Comisión Federal de Electricidad
Fernandez Cueto
México, 1992

4. - **Análisis de Sistemas de Potencia**
John J. Grainger y William D. Stevenson Jr.
McGraw-Hill
México, 1996

5. - **Análisis de Sistemas de Potencia**
William D. Stevenson Jr.
3ª Edición en español
McGraw-Hill
México, 1988

6. - Análisis de Sistemas de Potencia

William D. Steavenson Jr.

2ª Edición en español

McGraw-Hill

España, 1965

7. - Análisis Numérico

Curtisa F. Gerard; versión de Javier Luis Valls Cabrera.

Alfaomega

México, 1991

8. - Redes Eléctricas

Tercera edición

Jacinto Viqueira Landa

Representaciones y Servicios de Ingeniería.

México 1986.

9. - Power System Analysis and Design.

Glover / Sarma.

PWS-KENT Publishing Company.

Boston 1987.

10. - Introducción al Análisis de los Sistemas Eléctricos de Potencia

Gilberto Enriquez Harper

Limusa

México, 1991

11. -Electrical Power Transmission Systems

J. Robert Eaton, Edwin Cohen

Prentice Hall

New Jersey, 1983

12. - Electrical Power Systems

C.L. Wadwa

Second Edition

John Wiley & Sons

New York, 1991

13. - Electrical Power Systems, Electric Energy Systems Theory – And Introduction

O. I. Elgerd.

McGraw-Hill

USA, 1971

14. -Power System Analysis and Stability

Second Edition

D. R. J. Jha

Dhanpat Rai & Sons

India, 1980

15. -Manual de Visual Basic para Windows

Gary Cornell

McGraw-Hill

España, 1994

16. -C++ Programación Exitosa

Kris Jamsa
Alfaomega
México, 1997

17. - Enciclopedia de Visual Basic 4

Fco. Javier Ceballos Sierra
Alfaomega
México, 1997

18. - Software Orientado a Objetos

Ann L. Winblad, Samuel D. Edwards y David R. King.
Addison-Wesley / Diaz de Santos
EUA, 1994

19. - Visual Basic

José Eduardo Malaf Carvallo
McGraw-Hill
España 1992

20. - Visual Basic

Stefan Dittrich
Marcombo
México, 1993

21. - Programación Orientada a Objetos con C++

Fco. Javier Ceballos Sierra
2ª edición
Alfaomega
México, 1998

Anexo. Algoritmos codificados en lenguaje de programación.

Programa 1. Lincor.for

C ESTE PROGRAMA OBTIENE LAS VARIABLES ELECTRICAS DE UNA LINEA DE
C TRANSMISION CORTA EN EL EXTREMO TRANSMISOR O RECEPTOR, EL PRO
C GRAMA REQUIERE COMO DATOS LOS PARAMETROS DE LA LINEA Y EL VOLTA
C JE Y CORRIENTE EN CUALQUIERA DE LOS DOS EXTREMOS DE LA LINEA.

C
C DECLARACION DE VARIABLES

```
COMPLEX Z,Y,B,Vr,Jr,Va,Is,PSR,PRREAL
REAL L,IRA,IRB,IS2,IS1,ISM,AVS,AIS,VSL,ISM
REAL VS1,VS2,IRM,Isa,Isb,IRa,IRb,IRc,IRd,IRe,IRf,IRg,IRh,IRi,IRj,IRk,IRl,IRm,IRn,IRo,IRp,IRq,IRr,IRs,IRt,IRu,IRv,IRw,IRx,IRy,IRz,IRaa,IRab,IRac,IRad,IRae,IRaf,IRag,IRah,IRai,IRaj,IRak,IRal,IRam,IRan,IRao,IRap,IRaq,IRar,IRas,IRat,IRau,IRav,IRaw,IRax,IRay,IRaz,IRba,IRbb,IRbc,IRbd,IRbe,IRbf,IRbg,IRbh,IRbi,IRbj,IRbk,IRbl,IRbm,IRbn,IRbo,IRbp,IRbq,IRbr,IRbs,IRbt,IRbu,IRbv,IRbw,IRbx,IRby,IRbz,IRca,IRcb,IRcc,IRcd,IRce,IRcf,IRcg,IRch,IRci,IRcj,IRck,IRcl,IRcm,IRcn,IRco,IRcp,IRcq,IRcr,IRcs,IRct,IRcu,IRcv,IRcw,IRcx,IRcy,IRcz,IRda,IRdb,IRdc,IRdd,IRde,IRdf,IRdg,IRdh,IRdi,IRdj,IRdk,IRdl,IRdm,IRdn,IRdo,IRdp,IRdq,IRdr,IRds,IRdt,IRdu,IRdv,IRdw,IRdx,IRdy,IRdz,IRea,IReb,IRec,IRed,IRee,IRef,IReg,IReh,IRei,IRej,IRek,IRel,IRem,IRen,IReo,IRep,IReq,IRer,IRes,IRet,IReu,IRev,IRew,IRex,IRey,IRez,IRfa,IRfb,IRfc,IRfd,IRfe,IRff,IRfg,IRfh,IRfi,IRfj,IRfk,IRfl,IRfm,IRfn,IRfo,IRfp,IRfq,IRfr,IRfs,IRft,IRfu,IRfv,IRfw,IRfx,IRfy,IRfz,IRga,IRgb,IRgc,IRgd,IRge,IRgf,IRgg,IRgh,IRgi,IRgj,IRgk,IRgl,IRgm,IRgn,IRgo,IRgp,IRgq,IRgr,IRgs,IRgt,IRgu,IRgv,IRgw,IRgx,IRgy,IRgz,IRha,IRhb,IRhc,IRhd,IRhe,IRhf,IRhg,IRhh,IRhi,IRhj,IRhk,IRhl,IRhm,IRhn,IRho,IRhp,IRhq,IRhr,IRhs,IRht,IRhu,IRhv,IRhw,IRhx,IRhy,IRhz,IRia,IRib,IRic,IRid,IRie,IRif,IRig,IRih,IRii,IRij,IRik,IRil,IRim,IRin,IRio,IRip,IRiq,IRir,IRis,IRit,IRiu,IRiv,IRiw,IRix,IRiy,IRiz,IRja,IRjb,IRjc,IRjd,IRje,IRjf,IRjg,IRjh,IRji,IRjj,IRjk,IRjl,IRjm,IRjn,IRjo,IRjp,IRjq,IRjr,IRjs,IRjt,IRju,IRjv,IRjw,IRjx,IRjy,IRjz,IRka,IRkb,IRkc,IRkd,IRke,IRkf,IRkg,IRkh,IRki,IRkj,IRkk,IRkl,IRkm,IRkn,IRko,IRkp,IRkq,IRkr,IRks,IRkt,IRku,IRkv,IRkw,IRkx,IRky,IRkz,IRla,IRlb,IRlc,IRld,IRle,IRlf,IRlg,IRlh,IRli,IRlj,IRlk,IRll,IRlm,IRln,IRlo,IRlp,IRlq,IRlr,IRls,IRlt,IRlu,IRlv,IRlw,IRlx,IRly,IRlz,IRma,IRmb,IRmc,IRmd,IRme,IRmf,IRmg,IRmh,IRmi,IRmj,IRmk,IRml,IRmm,IRmn,IRmo,IRmp,IRmq,IRmr,IRms,IRmt,IRmu,IRmv,IRmw,IRmx,IRmy,IRmz,IRna,IRnb,IRnc,IRnd,IRne,IRnf,IRng,IRnh,IRni,IRnj,IRnk,IRnl,IRnm,IRnn,IRno,IRnp,IRnq,IRnr,IRns,IRnt,IRnu,IRnv,IRnw,IRnx,IRny,IRnz,IRoa,IRob,IRoc,IRod,IRoe,IRof,IRog,IRoh,IRoi,IRoj,IRok,IRol,IRom,IRon,IRoo,IRop,IRoq,IRor,IRos,IRot,IRou,IRov,IRow,IRox,IRoy,IRoz,IRpa,IRpb,IRpc,IRpd,IRpe,IRpf,IRpg,IRph,IRpi,IRpj,IRpk,IRpl,IRpm,IRpn,IRpo,IRpp,IRpq,IRpr,IRps,IRpt,IRpu,IRpv,IRpw,IRpx,IRpy,IRpz,IRqa,IRqb,IRqc,IRqd,IRqe,IRqf,IRqg,IRqh,IRqi,IRqj,IRqk,IRql,IRqm,IRqn,IRqo,IRqp,IRqq,IRqr,IRqs,IRqt,IRqu,IRqv,IRqw,IRqx,IRqy,IRqz,IRra,IRrb,IRrc,IRrd,IRre,IRrf,IRrg,IRrh,IRri,IRrj,IRrk,IRrl,IRrm,IRrn,IRro,IRrp,IRrq,IRrr,IRrs,IRrt,IRru,IRrv,IRrw,IRrx,IRry,IRrz,IRsa,IRsb,IRsc,IRsd,IRse,IRsf,IRsg,IRsh,IRsi,IRsj,IRsk,IRsl,IRsm,IRsn,IRso,IRsp,IRsq,IRsr,IRss,IRst,IRsu,IRsv,IRsw,IRsx,IRsy,IRsz,IRta,IRtb,IRtc,IRtd,IRte,IRtf,IRtg,IRth,IRti,IRtj,IRtk,IRtl,IRtm,IRtn,IRto,IRtp,IRtq,IRtr,IRts,IRtt,IRtu,IRtv,IRtw,IRtx,IRty,IRtz,IRua,IRub,IRuc,IRud,IRue,IRuf,IRug,IRuh,IRui,IRuj,IRuk,IRul,IRum,IRun,IRuo,IRup,IRuq,IRur,IRus,IRut,IRuu,IRuv,IRuw,IRux,IRuy,IRuz,IRva,IRvb,IRvc,IRvd,IRve,IRvf,IRvg,IRvh,IRvi,IRvj,IRvk,IRvl,IRvm,IRvn,IRvo,IRvp,IRvq,IRvr,IRvs,IRvt,IRvu,IRvv,IRvw,IRvx,IRvy,IRvz,IRwa,IRwb,IRwc,IRwd,IRwe,IRwf,IRwg,IRwh,IRwi,IRwj,IRwk,IRwl,IRwm,IRwn,IRwo,IRwp,IRwq,IRwr,IRws,IRwt,IRwu,IRwv,IRww,IRwx,IRwy,IRwz,IRxa,IRxb,IRxc,IRxd,IRxe,IRxf,IRxg,IRxh,IRxi,IRxj,IRxk,IRxl,IRxm,IRxn,IRxo,IRxp,IRxq,IRxr,IRxs,IRxt,IRxu,IRxv,IRxw,IRxx,IRxy,IRxz,IRya,IRyb,IRyc,IRyd,IRye,IRyf,IRyg,IRyh,IRyi,IRyj,IRyk,IRyl,IRym,IRyn,IRyo,IRyp,IRyq,IRyr,IRys,IRyt,IRyu,IRyv,IRyw,IRyx,IRyy,IRyz,IRza,IRzb,IRzc,IRzd,IRze,IRzf,IRzg,IRzh,IRzi,IRzj,IRzk,IRzl,IRzm,IRzn,IRzo,IRzp,IRzq,IRzr,IRzs,IRzt,IRzu,IRzv,IRzw,IRzx,IRzy,IRzz
```

C APERTURA DE ARCHIVOS

```
OPEN(130,FILE='LINCOR.RES')
OPEN(140,FILE='LINCOR.DAT')
OPEN(150,FILE='LINCORI.DAT')
OPEN(55,FILE='LINCORF.DAT')
OPEN(60,FILE='TIPOD.DAT')
```

```
WRITE(*,*)'QUE VALORES QUIERES CONOCER: '
WRITE(*,*)
WRITE(*,*)'(1) CONOCER Vs e Is : '
WRITE(*,*)'(2) CONOCER Vr e Ir : '
READ(60,*)N
```

IF (N.EQ.1) THEN

```
WRITE(130,*)'CALCULO DE VS E IS, LINEA CORTA'
WRITE(*,*)'DAME LOS VALORES DE LA RESIST., DE LA INDUCT.'
WRITE(*,*)'Y LA LONGITUD DE LA LINEA'
```

```
READ(140,*)R,L,IR,IS
WRITE(55,100)R,L,IR,IS
```

100 FORMAT(1X,R='F7.5',1X,L='E10.4',1X,IR,IS='F6.2')

```
PI=3.141593
RT=R*LONG
XL=(2*PI*60*L)*LONG
Z=CMPLX(RT,XL)
WRITE(130,*)'IMPEDANCIA DE LA LINEA'
```

150 WRITE(130,150)RT,XL

```
FORMAT(1X,Z='F8.4,2X,J',F8.4)
A=1
WRITE(130,*)'PARAMETRO A'
```

160 WRITE(130,160)A

```
FORMAT(7X,12)
B=Z
B1=REAL(B)
B2=AIMAG(B)
WRITE(130,*)'PARAMETRO B'
```

170 WRITE(130,170)B1,B2

```
FORMAT(1X,B='F8.4,2X,J',F8.4)
C=0
WRITE(130,*)'PARAMETRO C'
```

180 WRITE(130,180)C

```
FORMAT(7X,12)
D=1
WRITE(130,*)'PARAMETRO D'
```

190 WRITE(130,190)D

```
FORMAT(7X,12)
WRITE(*,*)'DAME LOS VALORES DE Vra, Vrb e Ira, Irb : '
READ(140,*)Vra,Vrb,Ira,Irb
Vr=CMPLX(Vra,Vrb)
```

Anexo. Algoritmos codificados en lenguaje de programación.

```
Ir=CMPLX(Ira,Irb)

Vs=A*Vr+B*Ir
VS1=REAL(VS)
VS2=AIMAG(VS)
VSM=SQRT(VS1**2+VS2**2)

Is=C*Vr+D*Ir
IS1=REAL(IS)
IS2=AIMAG(IS)

WRITE(130,*)'VOLTAJE AL NEUTRO EN EL'
WRITE(130,*)'EXTREMO TRANSMISOR'
WRITE(130,200)VS1,VS2
200  FORMAT(1X,'VS=',F9.4,2X,'J',F9.4)

WRITE(130,*)'CORRIENTE EN EL EXTREMO'
WRITE(130,*)'TRANSMISOR'
WRITE(130,210)IS1,IS2
210  FORMAT(1X,'IS=',F7.4,2X,'J',F7.4)
RAIZ=3
VSLL=SQRT(RAIZ)*VSM

C  VOLTAJE LINEA A LINEA EN EL EXTREMO RECEPTOR
VRM=SQRT(VRA**2+VRB**2)
VRL=SQRT(RAIZ)*VRM

C  POTENCIA APARENTE EN EL EXTREMO TRANSMISOR
PSR=3*(VS*CONJG(IS))
PSR1=REAL(PSR)
PSR2=AIMAG(PSR)

C  POTENCIA APARENTE EN EL EXTREMO RECEPTOR
PRREAL=3*(VR*CONJG(IR))
PR1=REAL(PRREAL)
PR2=AIMAG(PRREAL)

WRITE(130,*)'POTENCIA REAL TRANSMITIDA'
WRITE(130,250)PSR1
250  FORMAT(1X,F8.4)
WRITE(130,*)'POTENCIA REACTIVA TRANSMITIDA'
WRITE(130,260)PSR2
260  FORMAT(1X,F8.4)

WRITE(130,*)'POTENCIA REAL RECIBIDA'
WRITE(130,270)PR1
270  FORMAT(1X,F8.4)
WRITE(130,*)'POTENCIA REACTIVA RECIBIDA'
WRITE(130,280)PR2
280  FORMAT(1X,F8.4)

C  PERDIDAS DE POTENCIA
PERD=PSR1-PR1
PERDQ=PSR2-PR2
WRITE(130,*)'PERDIDAS DE POTENCIA REAL'
WRITE(130,290)PERD
290  FORMAT(1X,F8.4)
WRITE(130,*)'PERDIDAS DE POTENCIA REACTIVA'
WRITE(130,300)PERDQ

FPOT=SQRT(PSR1**2+PSR2**2)
FPOT1=PSR1/FPOT
WRITE(130,*)'FACTOR DE POTENCIA EN EL EX'
WRITE(130,*)'TREMOS TRANSMISOR'
WRITE(130,*)FPOT1

300  FORMAT(1X,F8.4)
EF1=PR1/PSR1*100
WRITE(130,*)'EFICIENCIA DE TRANSMISION'
WRITE(130,310)EF1
```

Anexo. Algoritmos codificados en lenguaje de programación.

```
310  FORMAT(1X,F8.4)
      VRSC=VSL/A
      REG=(VRSC-VRL)/VRL*100
      WRITE(130,*)'REGULACION'
      WRITE(130,320)REG
320  FORMAT(1X,F8.4)

      ELSEIF (N.EQ.2) THEN

      WRITE(130,*)'CALCULO DE VR E IR, LINEA CORTA'
      WRITE(*,*)'DAME LOS VALORES DE LA RESIST., DE LA INDUCT.'
      WRITE(*,*)'Y LA LONGITUD DE LA LINEA'
      READ(150,*)R,L,LONG
      WRITE(55,120)R,L,LONG
120  FORMAT(1X,'R=',F7.3/,1X,'L=',E10.4/,1X,'LONGITUD=',F5.2)

      PI=3.141593
      RT=R*LONG
      XL=(2*PI*60*L)*LONG
      Z=CMPLX(RT,XL)
      WRITE(130,*)'IMPEDANCIA DE LA LINEA'
      WRITE(130,153)RT,XL
153  FORMAT(1X,'Z=',F8.4,2X,'J',F8.4)

C    PARAMETROS DE LA LINEA
A=1
  WRITE(130,*)'PARAMETRO A'
  WRITE(130,163)A
163  FORMAT(7X,I2)
  B=Z
  B1=REAL(B)
  B2=AIMAG(B)
  WRITE(130,*)'PARAMETRO B'
  WRITE(130,173)B1,B2
173  FORMAT(1X,'B=',F8.4,2X,'J',F8.4)
  C=0
  WRITE(130,*)'PARAMETRO C'
  WRITE(130,183)C
183  FORMAT(7X,I2)
  D=1
  WRITE(130,*)'PARAMETRO D'
  WRITE(130,193)D
193  FORMAT(7X,I2)

  WRITE(*,*)'DAME LOS VALORES DE Vsa,Vsb e Isa,Isb : '
  WRITE(*,*)
  READ(150,*)Vsa,Vsb,Isa,Isb
  VS=CMPLX(Vsa,Vsb)
  IS=CMPLX(Isa,Isb)

C    MAGNITUD DEL VOLTAJE LINEA A LINEA EN EL EXTREMO TRANSMISOR
  BS1=REAL(VS)
  BS2=AIMAG(VS)
  BMAG=SQRT(BS1**2+BS2**2)
  RAIZ=3
  VSL=SQRT(RAIZ)*BMAG

C    VOLTAJE EN EL EXTREMO RECEPTOR
  VR=(D*VS-(B*IS))/(-(C*B)+(A*D))
  VRA=REAL(VR)
  VRB=AIMAG(VR)

C    CORRIENTE EN EL EXTREMO RECEPTOR
  IR=(A*IS-(C*VS))/(-(C*B)+(A*D))
  IRA=REAL(IR)
  IRB=AIMAG(IR)

  WRITE(130,*)'VOLTAJE AL NEUTRO EN EL'
  WRITE(130,*)'EXTREMO RECEPTOR'
  WRITE(130,300)VRA,VRB
```

Anexo. Algoritmos codificados en lenguaje de programación.

```
500  FORMAT(1X,'VR-',F9.4,2X',F',F9.4)

      WRITE(130,*)'CORRIENTE EN EL EXTREMO'
      WRITE(130,*)'RECEPTOR'
      WRITE(130,510)IRA,IRB
510  FORMAT(1X,1R-',F7.4,2X',F',F7.4)

      PSR=3*(VS*CONJG(IS))
      PSR1=REAL(PSR)
      PSR2=AIMAG(PSR)

      WRITE(130,*)'POTENCIA REAL TRANSMITIDA'
      WRITE(130,*)PSR1
      WRITE(130,*)'POTENCIA REACTIVA TRANSMITIDA'
      WRITE(130,*)PSR2

      PRREAL=3*(VR*CONJG(IR))
      PR1=REAL(PRREAL)
      PR2=AIMAG(PRREAL)

      FAC=SQRT(PR1**2+PR2**2)
      FAC1=PR1/FAC
      WRITE(130,*)'FACTOR DE POTENCIA EN EL'
      WRITE(130,*)'EXTREMO RECEPTOR'
      WRITE(130,*)FAC1

C    MAGNITUD DEL VOLTAJE LINEA A LINEA EN EL EXTREMO RECEPTOR
      VRM=SQRT(VRA**2+VRB**2)
      VRLL=SQRT(RAIZ)*VRM

      WRITE(130,*)'POTENCIA REAL RECIBIDA'
      WRITE(130,*)PR1
      WRITE(130,*)'POTENCIA REACTIVA RECIBIDA'
      WRITE(130,*)PR2
      PERD=PSR1-PR1
      PERDQ=PSR2-PR2
      WRITE(130,*)'PERDIDAS DE POTENCIA REAL'
      WRITE(130,*)PERD
      WRITE(130,*)'PERDIDAS DE POTENCIA REACTIVA'
      WRITE(130,*)PERDQ
      EFI=PR1/PSR1*100
      WRITE(130,*)'EFICIENCIA DE TRANSMISION'
      WRITE(130,*)EFI
      VRSC=VSLL/A
      REG=(VRSC-VRLL)/VRLL*100
      WRITE(130,*)'REGULACION'
      WRITE(130,*)REG
      ENDIF
      STOP
      END
```

Programa 2. Lmedpi.for

```
C    ESTE PROGRAMA OBTIENE LAS VARIABLES ELECTRICAS DE UNA LINEA MEDIA EN
C    EL EXTREMO TRANSMISOR O RECEPTOR. PARA OBTENER LA SOLUCION SE REQUIE
C    REN COMO DATOS LOS PARAMETROS DE LA LINEA Y EL VOLTAJE Y CORRIENTE EN
C    CUALQUIERA DE LOS DOS EXTREMOS. SE UTILIZO PARA EL CALCULO EL MODELO
C    PI DE LA LINEA.
C
```

```
COMPLEX Z,Y,A,B,C,D,Vr,Ir,Vs,Is,PSR,PRREAL
REAL L,IRA,IRB,IS2,ISI,ISM,AVS,AIS,VSLL,VSM
REAL VS1,VS2,IRM,Isa,Isb,LONG
OPEN(130,FILE='LMEDPI.RES')
OPEN(140,FILE='LMEDPI.DAT')
OPEN(150,FILE='LMEDPI1.DAT')
OPEN(*5,FILE='LMEDPIF.DAT')
OPEN(60,FILE='TIPOD.DAT')
WRITE(*,*)'QUE VALORES QUIERES CONOCER: '
```

Anexo. Algoritmos codificados en lenguaje de programación.

```
WRITE(*,*)
WRITE(*,*)(1) CONOCER Vs e Is :
WRITE(*,*)(2) CONOCER Vr e Ir :
READ(60,*)N

IF (N.EQ.1) THEN

WRITE(130,*)'CALCULO DE VS E IS, MODELO PI'
WRITE(*,*)'DAME LOS VALORES DE LA RESIST., DE LA INDUCT.'
WRITE(*,*)'DE LA CAPACITANCIA Y DE LA LONGITUD '
WRITE(*,*)'DE LA LINEA'
READ(140,*)R,L,CAP,LONG
WRITE(55,100)R,L,CAP
WRITE(55,101)LONG
100 FORMAT(1X,'R=',F7.5,/,1X,'L=',E14.6,/,1X,'C=',E14.6)
101 FORMAT(1X,'LONGITUD=',F6.2)

PI=3.141593
RT=R*LONG
XL=(2*PI*60*L)*LONG
XC=(2*PI*60*CAP)*LONG
Z=CMPLX(RT,XL)
WRITE(130,*)'IMPEDANCIA DE LA LINEA'
WRITE(130,150)RT,XL
150 FORMAT(1X,'Z=',F8.4,2X,'I',F8.4)
Y=CMPLX(0,XC)

WRITE(130,*)'ADMITANCIA DE LA LINEA'
WRITE(130,151)XC
151 FORMAT(1X,'Y=',I',F14.10)

A=((Z*Y)2)-1
A1=REAL(A)
A2=AIMAG(A)
AMAO=SQRT(A1**2+A2**2)
WRITE(130,*)'PARAMETRO A'
WRITE(130,*)'PARTE REAL'
WRITE(130,*)A1
WRITE(130,*)'PARTE IMAGINARIA'
WRITE(130,*)A2

B=Z
B1=REAL(B)
B2=AIMAG(B)
WRITE(130,*)'PARAMETRO B'
WRITE(130,*)'PARTE REAL'
WRITE(130,160)B1
160 FORMAT(1X,F11.7)
WRITE(130,*)'PARTE IMAGINARIA'
WRITE(130,170)B2
170 FORMAT(1X,F11.7)
C=Y*(1+((Z*Y)4))
C1=REAL(C)
C2=AIMAG(C)
WRITE(130,*)'PARAMETRO C'
WRITE(130,*)'PARTE REAL'
WRITE(130,*)C1
WRITE(130,*)'PARTE IMAGINARIA'
WRITE(130,*)C2

D=((Z*Y)2)-1
D1=REAL(D)
D2=AIMAG(D)
WRITE(130,*)'PARAMETRO D'
WRITE(130,*)'PARTE REAL'
WRITE(130,*)D1
WRITE(130,*)'PARTE IMAGINARIA'
WRITE(130,*)D2

WRITE(*,*)'DAME LOS VALORES DE Vra, Vrb e Ira, Irb :'
```

Anexo. Algoritmos codificados en lenguaje de programación.

```
READ(140,*)Vra, Vrb,Ira,Irb
WRITE(55,110)Vra, Vrb,Ira,Irb
110  FORMAT(1X,'VR-',F9.4,2X,'J',F9.4,/,1X,'IR-',F9.4,2X,'J',F9.4)
Vr=CMPLX(Vra, Vrb)
Ir=CMPLX(Ira, Irb)
VR1=REAL(Vr)
VR2=AIMAG(Vr)
VRM=SQRT(VR1**2+VR2**2)
Vs=A*Vr+B*Ir
VS1=REAL(VS)
VS2=AIMAG(VS)
VSM=SQRT(VS1**2+VS2**2)
Is=C*Vr+D*Ir
IS1=REAL(IS)
IS2=AIMAG(IS)

WRITE(130,*)'VOLTAJE AL NEUTRO EN EL'
WRITE(130,*)'EXTREMO TRANSMISOR'
WRITE(130,211)VS1, VS2
211  FORMAT(1X,'VS-',F9.4,2X,'J',F9.4)

WRITE(130,*)'CORRIENTE EN EL EXTREMO TRANS'
WRITE(130,*)'MISOR'
WRITE(130,210)IS1, IS2
210  FORMAT(1X,'IS-',F7.4,2X,'J',F7.4)

RAIZ=3
VLL=SQRT(RAIZ)*VSM
VRL=SQRT(RAIZ)*VRM

C  POTENCIA APARENTE EN EL EXTREMO TRANSMISOR
PSR=3*(VS*CONJG(IS))
PSR1=REAL(PSR)
PSR2=AIMAG(PSR)

WRITE(130,*)'POTENCIA REAL TRANSMITIDA'
WRITE(130,*)PSR1
WRITE(130,*)'POTENCIA REACTIVA TRANSMITIDA'
WRITE(130,*)PSR2

C  POTENCIA APARENTE EN EL EXTREMO RECEPTOR
PRREAL=3*(VR*CONJG(IR))
PR1=REAL(PRREAL)
PR2=AIMAG(PRREAL)
WRITE(130,*)'POTENCIA REAL RECIBIDA'
WRITE(130,*)PR1
WRITE(130,*)'POTENCIA REACTIVA RECIBIDA'
WRITE(130,*)PR2

C  PERDIDAS DE POTENCIA
PERD=PSR1-PR1
PERDQ=PSR2-PR2
WRITE(130,*)'PERDIDAS DE POTENCIA REAL'
WRITE(130,*)PERD

WRITE(130,*)'PERDIDAS DE POTENCIA REACTIVA'
WRITE(130,*)PERDQ
EFI=PR1/PSR1*100
WRITE(130,*)'EFICIENCIA DE TRANSMISION'
WRITE(130,*)EFI

FPOT=SQRT(PSR1**2+PSR2**2)
FPOT1=PSR1/FPOT
WRITE(130,*)'FACTOR DE POTENCIA EN EL EX'
WRITE(130,*)'TREMO TRANSMISOR'
WRITE(130,*)FPOT1

VRSC=VLL/AMAG
REG=((VRSC-VRL)/VRL)*100
WRITE(130,*)'REGULACION'
```

Anexo. Algoritmos codificados en lenguaje de programación.

```
WRITE(130,*)REG

ELSEIF (N.EQ.2) THEN

WRITE(130,*)CALCULO DE VR E IR,MODELO PI
WRITE(*,*)DAME LOS VALORES DE LA RESIST., DE LA INDUCT.'
WRITE(*,*)DE LA CAPACITANCIA Y DE LA LONGITUD'
WRITE(*,*)DE LA LINEA'
READ(150,*)R,L,CAP,LONG
WRITE(35,102)R,L,CAP
WRITE(35,103)LONG
102 FORMAT(1X,R='F7.5/,1X,L='E14.6/,1X,C='-E14.6)
103 FORMAT(1X, 'LONGITUD='F6.2)

PI=3.141593
RT=R*LONG
XL=(2*PI*60*L)*LONG
XC=(2*PI*60*CAP)*LONG
Z=CMPLX(RT,XL)
WRITE(130,*)IMPEDANCIA DE LA LINEA'
WRITE(130,152)RT,XL
152 FORMAT(1X,Z='F8.4,2X,'F,F8.4)
Y=CMPLX(0,XC)

WRITE(130,*)ADMITANCIA DE LA LINEA'
WRITE(130,153)XC
153 FORMAT(1X,Y='F,F14.10)

A=((Z*Y)2)+1
A1=REAL(A)
A2=AIMAG(A)
WRITE(130,*)PARAMETRO A'
WRITE(130,*)PARTE REAL'
WRITE(130,*)A1
WRITE(130,*)PARTE IMAGINARIA'
WRITE(130,*)A2

B=Z
B1=REAL(B)
B2=AIMAG(B)
WRITE(130,*)PARAMETRO B'
WRITE(130,*)PARTE REAL'
WRITE(130,180)B1
180 FORMAT(1X,F11.7)
WRITE(130,*)PARTE IMAGINARIA'
WRITE(130,190)B2
190 FORMAT(1X,F11.7)
C=Y*(1+(Z*Y)4))
C1=REAL(C)
C2=AIMAG(C)
WRITE(130,*)PARAMETRO C'
WRITE(130,*)PARTE REAL'
WRITE(130,*)C1
WRITE(130,*)PARTE IMAGINARIA'
WRITE(130,*)C2

D=((Z*Y)2)+1
D1=REAL(D)
D2=AIMAG(D)
WRITE(130,*)PARAMETRO D'
WRITE(130,*)PARTE REAL'
WRITE(130,*)D1
WRITE(130,*)PARTE IMAGINARIA'
WRITE(130,*)D2

WRITE(*,*)DAME LOS VALORES DE Vaa,Vab e Iaa,Iab.'
WRITE(*,*)
READ(150,*)Vaa,Vab,Iaa,Iab
WRITE(55,130)Vaa,Vab,Iaa,Iab
130 FORMAT(1X,'VS='F9.4,2X,'F,F9.4/,1X,'IS='F9.4,2X,'F,F9.4)
```

Anexo. Algoritmos codificados en lenguaje de programación.

```
VS=CMPLX(Vsa, Vsb)
VS1=REAL(VS)
VS2=AIMAG(VS)
VSM=SQRT(VS1**2+VS2**2)

IS=CMPLX(Isa, Isb)
IS1=REAL(IS)
IS2=AIMAG(IS)

VR=(D*VS-(B*IS))/(-C*B)+(A*D)
VRA=REAL(VR)
VRB=AIMAG(VR)

IR=(A*IS-(C*VS))/(-C*B)+(A*D)
IRA=REAL(IR)
IRB=AIMAG(IR)

WRITE(130,*)'VOLTAJE A NEUTRO EN EL'
WRITE(130,*)'EXTREMO TRANSMISOR'
WRITE(130,300)VS1,VS2
300 FORMAT(1X,'VS=',F9.4,2X,'J',F9.4)

WRITE(130,*)'CORRIENTE EN EL EXTREMO'
WRITE(130,*)'TRANSMISOR'
WRITE(130,310)IS1,IS2
310 FORMAT(1X,'IS=',F7.4,2X,'J',F7.4)

RAIZ=3
VSL=SQRT(RAIZ)*VSM

C POTENCIA APARENTE EN EL EXTREMO TRANSMISOR
PSR=3*(VS*CONJG(IS))
PSR1=REAL(PSR)
PSR2=AIMAG(PSR)

WRITE(130,*)'POTENCIA REAL TRANSMITIDA'
WRITE(130,*)PSR1
WRITE(130,*)'POTENCIA REACTIVA TRANSMITIDA'
WRITE(130,*)PSR2

VRM=SQRT(VRA**2+VRB**2)
VRL=SQRT(RAIZ)*VRM

C POTENCIA APARENTE EN EL EXTREMO RECEPTOR
PRREAL=3*(VR*CONJG(IR))
PR1=REAL(PRREAL)
PR2=AIMAG(PRREAL)

WRITE(130,*)'POTENCIA REAL RECIBIDA'
WRITE(130,*)PR1
WRITE(130,*)'POTENCIA REACTIVA RECIBIDA'
WRITE(130,*)PR2

C PERDIDAS DE POTENCIA
PERD=PSR1-PR1
PERDQ=PSR2-PR2
WRITE(130,*)'PERDIDAS DE POTENCIA REAL'
WRITE(130,*)PERD
WRITE(130,*)'PERDIDAS DE POTENCIA REACTIVA'
WRITE(130,*)PERDQ

EFI=PR1/PSR1*100
WRITE(130,*)'EFICIENCIA DE TRANSMISION'
WRITE(130,*)EFI

FAC=SQRT(PR1**2+PR2**2)
FAC1=PR1/FAC
WRITE(130,*)'FACTOR DE POTENCIA EN EL'
WRITE(130,*)'EXTREMO RECEPTOR'
```

```
WRITE(130,*)FAC1

VRSC=VSLL/A
REG=(VRSC-VRLI)/VRLI*100
WRITE(130,*)REGULACION'
WRITE(130,*)REG
ENDIF
STOP
END
```

Programa 3. Lmedt.for

C ESTE PROGRAMA OBTIENE LAS VARIABLES ELECTRICAS DE UNA LINEA MEDIA EN
 C EL EXTREMO TRANSMISOR O RECEPTOR, PARA OBTENER LA SOLUCION SE REQUIE
 C REN COMO DATOS LOS PARAMETROS DE LA LINEA Y EL VOLTAJE Y CORRIENTE EN
 C CUALQUIERA DE LOS DOS EXTREMOS. SE UTILIZO PARA EL CALCULO EL MODELO
 C T DE LA LINEA.
 C

```
COMPLEX Z,Y,A,B,C,D,Vr,lr,Vs,Is,PSR,PRREAL
REAL L,IRA,IRB,IS2,IS1,ISM,AVS,AIS,VSLL,VSM
REAL VS1,VS2,IRM,Isa,Isb,LONG
OPEN(130,FILE='L.MEDT.RES')
OPEN(140,FILE='L.MEDT.DAT')
OPEN(150,FILE='L.MEDT1.DAT')
OPEN(55,FILE='L.MEDTF.DAT')
OPEN(60,FILE='TIPOD.DAT')
WRITE(*,*)YQUE VALORES QUIERES CONOCER: '
WRITE(*,*)
WRITE(*,*) (1) CONOCER Vs e Is : '
WRITE(*,*) (2) CONOCER Vr e lr : '

READ(60,*)N
IF (N.EQ.1) THEN

WRITE(130,*)CALCULO DE VS E IS, MODELO T
WRITE(*,*)DAME LOS VALORES DE LA RESIST., DE LA INDUCT.'
WRITE(*,*)DE LA CAPACITANCIA Y DE LA LONGITUD'
WRITE(*,*)DE LA LINEA'
READ(140,*)R,L,CAP,LONG
WRITE(55,100)R,L,CAP
WRITE(55,101)LONG
100 FORMAT(1X,'R=',F7.5/,1X,'L=',E14.6/,1X,'C=',E14.6)
101 FORMAT(1X,'LONGITUD=',F6.2)

PI=3.141593
RT=R*LONG
XL=(2*PI*60*L)*LONG
XC=(2*PI*60*CAP)*LONG
Z=CMPLX(RT,XL)
WRITE(130,*)IMPEDANCIA DE LA LINEA'
WRITE(130,150)RT,XL
150 FORMAT(1X,'Z=',F8.4,2X,'J',F8.4)
Y=CMPLX(0,XC)

WRITE(130,*)ADMITANCIA DE LA LINEA'
WRITE(130,151)XC
151 FORMAT(1X,'Y=',J,F14.10)

A=(Z*Y)/2+1
A1=REAL(A)
A2=AIMAG(A)
AMAG=SQRT(A1**2+A2**2)
WRITE(130,*)PARAMETRO A'
WRITE(130,*)PARTE REAL'
WRITE(130,*)A1
WRITE(130,*)PARTE IMAGINARIA'
WRITE(130,*)A2
```

Anexo. Algoritmos codificados en lenguaje de programación.

```
B=Z*(1+(Z*Y)/4)
B1=REAL(B)
B2=AIMAG(B)
WRITE(130,*)'PARAMETRO B'
WRITE(130,*)'PARTE REAL'
WRITE(130,160)B1
160 FORMAT(1X,F11.7)
WRITE(130,*)'PARTE IMAGINARIA'
WRITE(130,170)B2
170 FORMAT(1X,F11.7)
C=Y
C1=REAL(C)
C2=AIMAG(C)
WRITE(130,*)'PARAMETRO C'
WRITE(130,*)'PARTE REAL'
WRITE(130,*)C1
WRITE(130,*)'PARTE IMAGINARIA'
WRITE(130,*)C2

D=((Z*Y)/2)+1
D1=REAL(D)
D2=AIMAG(D)
WRITE(130,*)'PARAMETRO D'
WRITE(130,*)'PARTE REAL'
WRITE(130,*)D1
WRITE(130,*)'PARTE IMAGINARIA'
WRITE(130,*)D2

WRITE(*,*)'DAME LOS VALORES DE Vra, Vrb e Ira, Irb .'
READ(140,*)Vra,Vrb,Ira,Irb
WRITE(55,110)Vra,Vrb,Ira,Irb
110 FORMAT(1X,'Vr=',F9.4,2X,'I',F9.4,/,1X,'Ir=',F9.4,2X,'I',F9.4)
Vr=CMPLX(Vra,Vrb)
Ir=CMPLX(Ira,Irb)

Vs=A*Vr+B*Ir
VS1=REAL(VS)
VS2=AIMAG(VS)

Is=C*Vr+D*Ir
IS1=REAL(IS)
IS2=AIMAG(IS)

WRITE(130,*)'VOLTAJE AL NEUTRO EN EL EXTREMO'
WRITE(130,*)'TRANSMISOR'
WRITE(130,200)VS1,VS2
200 FORMAT(1X,'VS=',F9.4,2X,'I',F9.4)

WRITE(130,*)'CORRIENTE EN EL EXTREMO TRANS'
WRITE(130,*)'MISOR'
WRITE(130,210)IS1,IS2
210 FORMAT(1X,'IS=',F7.4,2X,'I',F7.4)

VSM=SQRT(VS1**2+VS2**2)
RAIZ=3
VSL=SQRT(RAIZ)*VSM

C POTENCIA APARENTE EN EL EXTREMO TRANSMISOR
PSR=3*(VS*CONJG(IS))
PSR1=REAL(PSR)
PSR2=AIMAG(PSR)

WRITE(130,*)'POTENCIA REAL TRANSMITIDA'
WRITE(130,*)PSR1
WRITE(130,*)'POTENCIA REACTIVA TRANSMITIDA'
WRITE(130,*)PSR2

VRM=SQRT(VRA**2+VRB**2)
VRL=SQRT(RAIZ)*VRM
```

Anexo. Algoritmos codificados en lenguaje de programación.

```
C POTENCIA APARENTE EN EL EXTREMO RECEPTOR
PRREAL=3*(VR*CONJG(IR))
PR1=REAL(PRREAL)
PR2=AIMAG(PRREAL)
WRITE(130,*)'POTENCIA REAL RECIBIDA'
WRITE(130,*)PR1
WRITE(130,*)'POTENCIA REACTIVA RECIBIDA'
WRITE(130,*)PR2

PERD=PSR1-PR1
PERDQ=PSR2-PR2
WRITE(130,*)'PERDIDAS DE POTENCIA REAL'
WRITE(130,*)PERD
WRITE(130,*)'PERDIDAS DE POTENCIA REACTIVA'
WRITE(130,*)PERDQ
EF1=PR1/PSR1*100

WRITE(130,*)'EFICIENCIA DE TRANSMISION'
WRITE(130,*)EF1

FPOT=SQRT(PSR1**2+PSR2**2)
FPOT1=PSR1/FPOT
WRITE(130,*)'FACTOR DE POTENCIA EN EL EX'
WRITE(130,*)'TREMOS TRANSMISOR'
WRITE(130,*)FPOT1

VRSC=VSSL/AMAG
REG=(VRSC-VRL)/VRL*100
WRITE(130,*)'REGULACION'
WRITE(130,*)REG

ELSEIF (N.EQ.2) THEN

WRITE(130,*)'CALCULO DE VR E IR, MODELO T'
WRITE(*,*)'DAME LOS VALORES DE LA RESIST., DE LA INDUCT.'
WRITE(*,*)'DE LA CAPACITANCIA Y DE LA LONGITUD'
WRITE(*,*)'DE LA LINEA'
READ(150,*)R,L,CAP,LONG
WRITE(55,102)R,L,CAP
WRITE(55,103)LONG
102 FORMAT(1X,'R=',F7.5,/,1X,'L=',E14.6,/,1X,'C=',E14.6)
103 FORMAT(1X,'LONGITUD=',F6.2)

PI=3.141593
RT=R*LONG
XL=(2*PI*60*L)*LONG
XC=(2*PI*60*CAP)*LONG
Z=CMLX(RT,XL)
WRITE(130,*)'IMPEDANCIA DE LA LINEA'
WRITE(130,152)RT,XL
152 FORMAT(1X,'Z=',F8.4,2X,'J',F8.4)
Y=CMLX(0,XC)

WRITE(130,*)'ADMITANCIA DE LA LINEA'
WRITE(130,153)XC
153 FORMAT(1X,'Y=',J,F14.10)

A=(Z*Y)/2+1
A1=REAL(A)
A2=AIMAG(A)
AMAG=SQRT(A1**2+A2**2)
WRITE(130,*)'PARAMETRO A'
WRITE(130,*)'PARTE REAL'
WRITE(130,*)A1
WRITE(130,*)'PARTE IMAGINARIA'
WRITE(130,*)A2

B=Z*(1-(Z*Y)/4)
B1=REAL(B)
B2=AIMAG(B)
```

Anexo. Algoritmos codificados en lenguaje de programación.

```
WRITE(130,*)'PARAMETRO B'  
WRITE(130,*)'PARTE REAL'  
WRITE(130,180)B1  
180 FORMAT(1X,F11.7)  
WRITE(130,*)'PARTE IMAGINARIA'  
WRITE(130,190)B2  
190 FORMAT(1X,F11.7)  
C=Y  
C1=REAL(C)  
C2=AIMAG(C)  
WRITE(130,*)'PARAMETRO C'  
WRITE(130,*)'PARTE REAL'  
WRITE(130,*)C1  
WRITE(130,*)'PARTE IMAGINARIA'  
WRITE(130,*)C2  
  
D=((Z*Y)2)+1  
D1=REAL(D)  
D2=AIMAG(D)  
WRITE(130,*)'PARAMETRO D'  
WRITE(130,*)'PARTE REAL'  
WRITE(130,*)D1  
WRITE(130,*)'PARTE IMAGINARIA'  
WRITE(130,*)D2  
  
WRITE(*,*)'DAME LOS VALORES DE Vsa, Vsb e Isa, Isb :'  
WRITE(*,*)  
READ(150,*)Vsa, Vsb, Isa, Isb  
WRITE(55,130)Vsa, Vsb, Isa, Isb  
130 FORMAT(1X,'VS-',F9.4,2X,'J',F9.4,/,1X,'IS-',F9.4,2X,'J',F9.4)  
  
VS=CMPLX(Vsa,Vsb)  
VS1=REAL(VS)  
VS2=AIMAG(VS)  
  
IS=CMPLX(Isa,Isb)  
IS1=REAL(IS)  
IS2=AIMAG(IS)  
  
VR=(D*VS-(B*IS))/(-(C*B)+(A*D))  
VRA=REAL(VR)  
VRB=AIMAG(VR)  
  
IR=(A*IS-(C*VS))/(-(C*B)+(A*D))  
IRA=REAL(IR)  
IRB=AIMAG(IR)  
  
WRITE(130,*)'VOLTAJE A NEUTRO EN EL'  
WRITE(130,*)'EXTREMO TRANSMISOR'  
WRITE(130,300)VS1,VS2  
300 FORMAT(1X,'VS-',F9.4,2X,'J',F9.4)  
  
IS=CMPLX(Isa,Isb)  
IS1=REAL(IS)  
IS2=AIMAG(IS)  
  
WRITE(130,*)'CORRIENTE EN EL EXTREMO'  
WRITE(130,*)'TRANSMISOR'  
WRITE(130,310)IS1,IS2  
310 FORMAT(1X,'IS-',F7.4,2X,'J',F7.4)  
  
VSM=SQRT(VS1**2+VS2**2)  
RAIZ=3  
VSL=SQRT(RAIZ)*VSM  
C POTENCIA APARENTE EN EL EXTREMO TRANSMISOR  
PSR=3*(VS*CONJG(IS))  
PSR1=REAL(PSR)  
PSR2=AIMAG(PSR)  
WRITE(130,*)'POTENCIA REAL TRANSMITIDA'  
WRITE(130,*)PSR1
```

Anexo. Algoritmos codificados en lenguaje de programación.

```
WRITE(130,*)POTENCIA REACTIVA TRANSMITIDA'
WRITE(130,*)PSR2

VRM=SQRT(VRA**2+VRB**2)
VRLL=SQRT(RAIZ)*VRM

C POTENCIA APARENTE EN EL EXTREMO RECEPTOR
PRREAL=3*(VR*CONJG(IR))
PR1=REAL(PRREAL)
PR2=AIMAG(PRREAL)
WRITE(130,*)POTENCIA REAL RECIBIDA'
WRITE(130,*)PR1
WRITE(130,*)POTENCIA REACTIVA RECIBIDA'
WRITE(130,*)PR2

C PERDIDAS DE POTENCIA
PERD=PSR1-PR1
PERDQ=PSR2-PR2
WRITE(130,*)PERDIDAS DE POTENCIA REAL'
WRITE(130,*)PERD
WRITE(130,*)PERDIDAS DE POTENCIA REACTIVA'
WRITE(130,*)PERDQ

EFI=PR1/PSR1*100
WRITE(130,*)EFICIENCIA DE TRANSMISION'
WRITE(130,*)EFI

FAC=SQRT(PR1**2+PR2**2)
FAC1=PR1/FAC
WRITE(130,*)FACTOR DE POTENCIA EN EL'
WRITE(130,*)EXTREMO RECEPTOR'
WRITE(130,*)FAC1

VRSC=VLL/AMAG
REQ=(VRSC-VRLL)/VRLL*100
WRITE(130,*)REGULACION'
WRITE(130,*)REQ
ENDIF
STOP
END
```

Programa 4. Llarpi.for

C ESTE PROGRAMA OBTIENE LAS VARIABLES ELECTRICAS DE UNA LINEA LARGA EN
C EL EXTREMO TRANSMISOR O RECEPTOR, PARA OBTENER LA SOLUCION SE REQUIE
C REN COMO DATOS LOS PARAMETROS DE LA LINEA Y EL VOLTAJE Y CORRIENTE EN
C CUALQUIERA DE LOS DOS EXTREMOS.
C

```
COMPLEX Z,Y,A,B,C,D,Vr,Ir,Va,Is,ATEN,ZCARAC,AEX,BEX,EXPO
COMPLEX PSR,PRREAL
REAL L,R,CAP,IRA,IRB,ISM,IRM,LONG,IS1,IS2,ISA,ISB
OPEN(130,FILE='LLARPI.RES')
OPEN(140,FILE='LLARPL.DAT')
OPEN(150,FILE='LLARPI1.DAT')
OPEN(35,FILE='LLARPIF.DAT')
OPEN(60,FILE='TIPOD.DAT')
WRITE(*,*)QUE VALORES QUIERES CONOCER: '
WRITE(*,*)
WRITE(*,*)(1) CONOCER Vs e Is: '
WRITE(*,*)(2) CONOCER Vr e Ir: '
READ(60,*)N

IF(N.EQ.1) THEN

WRITE(130,*)CALCULO DE VS E IS, LINEA LARGA'
WRITE(*,*)DAME LOS VALORES DE LA RESIST., DE LA INDUCT.'
```

Anexo. Algoritmos codificados en lenguaje de programación.

```
WRITE(*,*)DE LA CAPACITANCIA Y DE LA LONGITUD
WRITE(*,*)DE LA LINEA
READ(140,*)R,L,CAP,LONG
WRITE(55,100)R,L,CAP
WRITE(55,101)LONG
100 FORMAT(1X,R=,F7.5/,1X,L=,E14.6/,1X,C=,E14.6)
101 FORMAT(1X, LONGITUD=,F6.2)

PI=3.141593
RT=R*LONG
XL=(2*PI*60*L)*LONG
XC=(2*PI*60*CAP)*LONG
Z=CMPLX(RT,XL)
WRITE(130,*)IMPEDANCIA DE LA LINEA
WRITE(130,150)RT,XL
150 FORMAT(1X,Z=,F14.2X,/,F8.4)
Y=CMPLX(0,XC)

WRITE(130,*)Y ADMITANCIA DE LA LINEA
WRITE(130,151)XC
151 FORMAT(1X,Y=,/,F14.10)

ATEN=SQRT((Z/LONG)*(Y/LONG))*LONG
ATEN1=REAL(ATEN)
ATEN2=AIMAG(ATEN)
WRITE(130,*)CONSTANTE DE ATENUACION
WRITE(130,*)PARTE REAL
WRITE(130,*)ATEN1
WRITE(130,*)PARTE IMAGINARIA
WRITE(130,*)ATEN2

ZCARAC=SQRT(Z/Y)
ZC1=REAL(ZCARAC)
ZC2=AIMAG(ZCARAC)
WRITE(130,*)IMPEDANCIA CARACTERISTICA
WRITE(130,*)PARTE REAL
WRITE(130,161)ZC1
161 FORMAT(1X,F12.7)
WRITE(130,*)PARTE IMAGINARIA
WRITE(130,171)ZC2
171 FORMAT(1X,F12.7)

A1=EXP(ATEN1)
ANG1=ATEN2
RA1=A1*COS(ANG1)
RIA1=A1*SIN(ANG1)
AEX=CMPLX(RA1,RIA1)

A2=EXP(-ATEN1)
ANG2=-ATEN2
RA2=A2*COS(ANG2)
RIA2=A2*SIN(ANG2)
BEX=CMPLX(RA2,RIA2)

A=(AEX+BEX)/2
A1=REAL(A)
A2=AIMAG(A)
AMAG=SQRT(A1**2+A2**2)
WRITE(130,*)PARAMETRO A
WRITE(130,*)PARTE REAL
WRITE(130,*)A1
WRITE(130,*)PARTE IMAGINARIA
WRITE(130,*)A2

B=ZCARAC*((AEX-(BEX))/2)
B1=REAL(B)
B2=AIMAG(B)
WRITE(130,*)PARAMETRO B
WRITE(130,*)PARTE REAL
WRITE(130,160)B1
```

Anexo. Algoritmos codificados en lenguaje de programación.

```
160  FORMAT(1X,F12.7)
      WRITE(130,*)'PARTE IMAGINARIA'
      WRITE(130,170)B2
170  FORMAT(1X,F12.7)

      EXPO=(AEX-(BEX))/2
      C=EXPOZCARAC
      C1=REAL(C)
      C2=AIMAG(C)
      WRITE(130,*)'PARAMETRO C'
      WRITE(130,*)'PARTE REAL'
      WRITE(130,*)C1
      WRITE(130,*)'PARTE IMAGINARIA'
      WRITE(130,*)C2

      D=(AEX+BEX)/2
      D1=REAL(D)
      D2=AIMAG(D)
      WRITE(130,*)'PARAMETRO D'
      WRITE(130,*)'PARTE REAL'
      WRITE(130,*)D1
      WRITE(130,*)'PARTE IMAGINARIA'
      WRITE(130,*)D2

      WRITE(*,*)'DAME LOS VALORES DE Vra, Vrb e Ira, Irb :'
      READ(140,*)Vra,Vrb,Ira,Irb
      WRITE(55,110)Vra,Vrb,Ira,Irb
110  FORMAT(1X,VR=-,F9.4,2X,/,F9.4/,1X,IR=-,F9.4,2X,/,F9.4)

      Vr=CMPLX(Vra,Vrb)
      Ir=CMPLX(Ira,Irb)

      Vs=A*Vr+B*Ir
      VS1=REAL(VS)
      VS2=AIMAG(VS)

      Is=C*Vr+D*Ir
      IS1=REAL(IS)
      IS2=AIMAG(IS)

      WRITE(130,*)'VOLTAJE AL NEUTRO EN EL EXTREMO'
      WRITE(130,*)'TRANSMISOR'
      WRITE(130,200)VS1,VS2
200  FORMAT(1X,VS=-,F9.4,2X,/,F9.4)
      WRITE(130,*)'CORRIENTE EN EL EXTREMO TRANS'
      WRITE(130,*)'MISOR'
      WRITE(130,210)IS1,IS2
210  FORMAT(1X,IS=-,F7.4,2X,/,F7.4)
      VSM=SQRT(VS1**2+VS2**2)
      RAIZ=3
      VSLI=SQRT(RAIZ)*VSM

C    POTENCIA APARENTE EN EL EXTREMO TRANSMISOR
      PSR=3*(VS*CONJG(IS))
      PSR1=REAL(PSR)
      PSR2=AIMAG(PSR)
      WRITE(130,*)'POTENCIA REAL TRANSMITIDA'
      WRITE(130,*)PSR1
      WRITE(130,*)'POTENCIA REACTIVA TRANSMITIDA'
      WRITE(130,*)PSR2
      VRM=SQRT(VRA**2+VRB**2)
      VRLI=SQRT(RAIZ)*VRM

C    POTENCIA APARENTE EN EL EXTREMO TRANSMISOR
      PRREAL=3*(VR*CONJG(IR))
      PR1=REAL(PRREAL)
      PR2=AIMAG(PRREAL)
      WRITE(130,*)'POTENCIA REAL RECIBIDA'
      WRITE(130,*)PR1
      WRITE(130,*)'POTENCIA REACTIVA RECIBIDA'
```

Anexo. Algoritmos codificados en lenguaje de programación.

```
WRITE(130,*)PR2
C PERDIDAS DE POTENCIA
PERD=PSR1-PR1
PERDQ=PSR2-PR2
WRITE(130,*)'PERDIDAS DE POTENCIA REAL'
WRITE(130,*)PERD
WRITE(130,*)'PERDIDAS DE POTENCIA REACTIVA'
WRITE(130,*)PERDQ

EFI=PR1/PSR1*100
WRITE(130,*)'EFICIENCIA DE TRANSMISION'
WRITE(130,*)EFI

FPOT=SQRT(PSR1**2+PSR2**2)
FPOT1=PSR1/FPOT
WRITE(130,*)'FACTOR DE POTENCIA EN EL EX'
WRITE(130,*)'TREMPO TRANSMISOR'
WRITE(130,*)FPOT1

VRSC=VSL/AMAG
REG=(VRSC-VRL)/VRL*100
WRITE(130,*)'REGULACION'
WRITE(130,*)REG

ELSEIF (N.EQ.2) THEN

WRITE(130,*)'CALCULO DE VR E IR, LINEA LARGA'
WRITE(*,*)'DAME LOS VALORES DE LA RESIST., DE LA INDUCT.'
WRITE(*,*)'DE LA CAPACITANCIA Y DE LA LONGITUD'
WRITE(*,*)'DE LA LINEA'
READ(150,*)R,L,CAP,LONG
WRITE(55,102)R,L,CAP
WRITE(55,103)LONG
102 FORMAT(1X,R=' ',F7.5/,1X,L=' ',E14.6/,1X,C=' ',E14.6)
103 FORMAT(1X, 'LONGITUD=',F6.2)

PI=3.141593
RT=R*LONG
XL=(2*PI*60*L)*LONG
XC=(2*PI*60*CAP)*LONG
Z=CMPLX(RT,XL)
WRITE(130,*)'IMPEDANCIA DE LA LINEA'
WRITE(130,152)RT,XL
152 FORMAT(1X,Z=' ',F8.4,2X,J,F8.4)
Y=CMPLX(0,XC)

WRITE(130,*)'ADMITANCIA DE LA LINEA'
WRITE(130,153)XC
153 FORMAT(1X,Y=' ',J,F14.10)
ATEN=SQRT(Z*Y)
ATEN1=REAL(ATEN)
ATEN2=AIMAG(ATEN)
WRITE(130,*)'CONSTANTE DE ATENUACION'
WRITE(130,*)'PARTE REAL'
WRITE(130,*)ATEN1
WRITE(130,*)'PARTE IMAGINARIA'
WRITE(130,*)ATEN2

ZCARAC=SQRT(Z/Y)
ZC1=REAL(ZCARAC)
ZC2=AIMAG(ZCARAC)

WRITE(130,*)'IMPEDANCIA CARACTERISTICA'
WRITE(130,*)'PARTE REAL'
WRITE(130,181)ZC1
181 FORMAT(1X,F12.7)

WRITE(130,*)'PARTE IMAGINARIA'
WRITE(130,192)ZC2
```

Anexo. Algoritmos codificados en lenguaje de programación.

192 FORMAT(1X,F12.7)

```
A1=EXP(ATEN1)
ANG1=ATEN2
RA1=A1*COS(ANG1)
RIA1=A1*SIN(ANG1)

AEX=CMPLX(RA1,RIA1)
```

```
A2=EXP(-ATEN1)
ANG2=-ATEN2
RA2=A2*COS(ANG2)
RIA2=A2*SIN(ANG2)

BEX=CMPLX(RA2,RIA2)
```

```
A=(AEX+BEX)/2
A1=REAL(A)
A2=AIMAG(A)
AMAG=SQRT(A1**2+A2**2)
WRITE(130,*)PARAMETRO A'
WRITE(130,*)PARTE REAL'
WRITE(130,*)A1
WRITE(130,*)PARTE IMAGINARIA'
WRITE(130,*)A2
```

EXPO=(AEX-(BEX))/2

```
B=ZCARAC*(((AEX-(BEX))/2)
B1=REAL(B)
B2=AIMAG(B)
WRITE(130,*)PARAMETRO B'
WRITE(130,*)PARTE REAL'
WRITE(130,162)B1
```

162 FORMAT(1X,F12.7)
WRITE(130,*)PARTE IMAGINARIA'
WRITE(130,172)B2
172 FORMAT(1X,F12.7)

```
C=EXPO/ZCARAC
C1=REAL(C)
C2=AIMAG(C)
WRITE(130,*)PARAMETRO C'
WRITE(130,*)PARTE REAL'
WRITE(130,*)C1
WRITE(130,*)PARTE IMAGINARIA'
WRITE(130,*)C2
```

```
D=(AEX+BEX)/2
D1=REAL(D)
D2=AIMAG(D)
WRITE(130,*)PARAMETRO D'
WRITE(130,*)PARTE REAL'
WRITE(130,*)D1
WRITE(130,*)PARTE IMAGINARIA'
WRITE(130,*)D2
```

```
WRITE(*,*)DAME LOS VALORES DE Vsa, Vsb e Isa, Isb :'  
WRITE(*,*)  
READ(150,*)Vsa, Vsb, Isa, Isb  
WRITE(55,130)Vsa, Vsb, Isa, Isb
```

130 FORMAT(1X,'VS=',F9.4,2X,'I',F9.4/,1X,'IS=',F9.4,2X,'I',F9.4)

```
VS=CMPLX(Vsa,Vsb)
VS1=REAL(VS)
VS2=AIMAG(VS)
```

```
IS=CMPLX(Isa,Isb)
IS1=REAL(IS)
IS2=AIMAG(IS)
```

Anexo. Algoritmos codificados en lenguaje de programación.

```
VR=(D*VS-(B*IS))/(-C*B)+(A*D)
VRA=REAL(VR)
VRB=AIMAG(VR)

IR=(A*IS-(C*VS))/(-C*B)+(A*D)
IRA=REAL(IR)
IRB=AIMAG(IR)

WRITE(130,*)'VOLTAJE A NEUTRO EN EL'
WRITE(130,*)'EXTREMO RECEPTOR'
WRITE(130,400)VRA,VRB
400  FORMAT(1X,'VR=',F9.4,2X,'J',F9.4)

WRITE(130,*)'CORRIENTE EN EL EXTREMO'
WRITE(130,*)'RECEPTOR'
WRITE(130,410)IRA,IRB
410  FORMAT(1X,'IR=',F7.4,2X,'J',F7.4)

VSM=SQRT(VS1**2+VS2**2)
RAIZ=3
VSLL=SQRT(RAIZ)*VSM

C  POTENCIA APARENTE EN EL EXTREMO TRANSMISOR
PSR=3*(VS*CONJG(IS))
PSR1=REAL(PSR)
PSR2=AIMAG(PSR)
WRITE(130,*)'POTENCIA REAL TRANSMITIDA'
WRITE(130,*)PSR1
WRITE(130,*)'POTENCIA REACTIVA TRANSMITIDA'
WRITE(130,*)PSR2

VRM=SQRT(VRA**2+VRB**2)
VRL=SQRT(RAIZ)*VRM

C  POTENCIA APARENTE EN EL EXTREMO RECEPTOR
PRREAL=3*(VR*CONJG(IR))
PR1=REAL(PRREAL)
PR2=AIMAG(PRREAL)
WRITE(130,*)'POTENCIA REAL RECIBIDA'
WRITE(130,*)PR1
WRITE(130,*)'POTENCIA REACTIVA RECIBIDA'
WRITE(130,*)PR2

C  PERDIDAS DE POTENCIA
PERD=PSR1-PR1
PERDQ=PSR2-PR2
WRITE(130,*)'PERDIDAS DE POTENCIA REAL'
WRITE(130,*)PERD
WRITE(130,*)'PERDIDAS DE POTENCIA REACTIVA'
WRITE(130,*)PERDQ

EFI=PR1/PSR1*100
WRITE(130,*)'EFICIENCIA DE TRANSMISION'
WRITE(130,*)EFI

FAC=SQRT(PR1**2+PR2**2)
FAC1=PR1/FAC
WRITE(130,*)'FACTOR DE POTENCIA EN EL'
WRITE(130,*)'EXTREMO RECEPTOR'
WRITE(130,*)FAC1

VRSC=VSLL/AMAG
REG=(VRSC-VRL)/VRL*100
WRITE(130,*)'REGULACION'
WRITE(130,*)REG
ENDIF
STOP
END
```

Programa 5. Malla.for

C ESTE PROGRAMA CALCULA LAS CORRIENTES DE MALLA DE UNA RED ELECTRICIA, PARA
C REALIZAR EL CALCULO REQUIERE COMO DATOS EL NUMERO DE RAMAS, EL NUMERO DE
C MALLAS, LA MATRIZ DE CONEXION RAMA-MALLA, EL VECTOR DE VOLTAJES DE GENE
C RACION Y LA MATRIZ DE IMPEDANCIAS DE RAMA.

```
DIMENSION ZB1(20,20),C(20),L(20,20),B(20,20),CP(20),X(20)
DIMENSION A(20,20),ZB(20,20),AT(20,20),ZP(20,20),C1(20,20)
DIMENSION PRE(20,20),PMA(20,20),XREAL(20),XIMA(20),ZP2(20,20)
DIMENSION ZP3(20,20),CR(20),CRI(10),X1(20),X2(20)
COMPLEX ZB,ZB1,C,C1,L,B,SUM,CP,X,ZP
INTEGER P,Q,A
C   CALCULO DE LAS CORRIENTES DE MALLA
OPEN(12,FILE='NR.DAT')
OPEN(13,FILE='NM.DAT')
WRITE(*,*)'CUANTAS RAMAS SON'
READ(12,*)P
WRITE(*,*)'CUANTAS MALLAS SON'
READ(13,*)Q
WRITE(*,*)'DAME LOS ELEMENTOS DE LA MATRIZ DE CONEXION RAMA'
WRITE(*,*)'MALLA'
OPEN(18,FILE='RM.DAT')
OPEN(19,FILE='RM1.DAT')
DO 1 I=1,P
DO 2 J=1,Q
READ(18,57)A(I,J)
WRITE(19,56)J,A(I,J)
57  FORMAT(12)
56  FORMAT(1X,'Am',(I2,1X,I2),',',I2)
2  CONTINUE
1  CONTINUE
c   CALCULO DE LA TRANSPUESTA DE LA MATRIZ DE CONEXION
c   RAMA-MALLA
DO 10 I=1,Q
DO 11 J=1,P
AT(I,J)=A(J,I)
11  CONTINUE
10  CONTINUE
WRITE(*,*)'DAME LOS ELEMENTOS DE ZP'
OPEN(54,FILE='ZP.DAT')
OPEN(53,FILE='ZP1.DAT')
DO 4 I=1,P
DO 3 J=1,P
IF(J.EQ.I) THEN
READ(54,*) ZP(I,J)
ZP2(I,J)=REAL(ZP(I,J))
ZP3(I,J)=ADMAQ(ZP(I,J))
WRITE(53,101)I,J,ZP2(I,J),ZP3(I,J)
101  FORMAT(1X,'ZP',(I2,1X,I2),',',F5.3,2X,'J',F5.3)
ENDIF
3  CONTINUE
4  CONTINUE
C   CALCULO DE LA MATRIZ ZBUS
C1(1,1)=0
DO 51 I=1,Q
DO 52 J=1,P
C1(I,J)=0
ZB(I,J)=0
DO 53 K=1,P
C1(I,J)=C1(I,J)+AT(I,K)*ZP(K,J)
53  CONTINUE
52  CONTINUE
51  CONTINUE
ZB(1,1)=0
DO 12 I=1,Q
DO 13 J=1,Q
ZB(I,J)=0
```

Anexo. Algoritmos codificados en lenguaje de programación.

```
DO 14 K=1,P
ZB(L,J)=ZB(L,J)+C1(L,K)*A(K,J)
14 CONTINUE
ZBI(L,J)=ZB(L,J)
PRE(L,J)=REAL(ZBI(L,J))
PIMA(L,J)=AIMAG(ZBI(L,J))
13 CONTINUE
12 CONTINUE
WRITE(*,*)SE IMPRIME LA MATRIZ ZBUS'
OPEN(64,FILE='ZBARRA.RES')
DO 17 I=1,Q
DO 18 J=1,Q
WRITE(64,66)IJ,PRE(I,J),PIMA(I,J)
66 FORMAT(1X,Z',I',I2,1X,I2,')',F7.4,2X,'J',F7.4)
18 CONTINUE
17 CONTINUE

C CALCULO DE LAS CORRIENTES DE MALLA
DO 55 I=1,Q
DO 902 J=1,Q
IF(J.EQ.1) THEN
L(I,J)=ZBI(I,J)
ENDIF
902 CONTINUE
55 CONTINUE
DO 30 J=1,Q
B(I,J)=ZBI(I,J)/L(I,1)
30 CONTINUE
DO 60 J=2,Q
DO 65 I=J,Q
SUM=(0,0)
DO 70 K=1,J-1
SUM=SUM+(L(I,K))*(B(K,J))
70 CONTINUE
L(I,J)=ZBI(I,J)/(SUM)
65 CONTINUE
B(I,J)=1
DO 80 I=J+1,Q
SUM=(0,0)
DO 90 K=1,J-1
SUM=SUM+(L(J,K))*(B(K,I))
90 CONTINUE
B(I,I)=(ZBI(I,I)-SUM)/L(I,I)
80 CONTINUE
60 CONTINUE
WRITE(*,*)DAME LOS VOLTAJES DE GENERACION'
OPEN(299,FILE='VG.DAT')
OPEN(103,FILE='VG1.DAT')
DO 901 I=1,Q
READ(299,*)C(I)
CR(I)=REAL(C(I))
CRI(I)=AIMAG(C(I))
WRITE(103,102),CR(I),CRI(I)
102 FORMAT(1X,'Vg',I,2,')',F5.3,2X,'J',F5.3)
901 CONTINUE
CP(I)=(C(I))/L(I,1)
DO 100 I=2,Q
SUM=(0,0)
DO 110 J=1,I-1
SUM=SUM+L(I,J)*CP(J)
110 CONTINUE
CP(I)=(C(I)-SUM)/L(I,I)
100 CONTINUE
WRITE(*,*)CORRIENTES DE MALLA'
X(Q)=CP(Q)
DO 120 I=2,Q
NVBL=Q-I+1
SUM=(0,0)
NP1=NVBL+1
DO 130 J=NP1,Q
```

Anexo. Algoritmos codificados en lenguaje de programación.

```
SUM=SUM+B(NVBL,J)*X(J)
130 CONTINUE
X(NVBL)=CP(NVBL)-SUM
120 CONTINUE
OPEN(74,FILE='MALLA.RES')
DO 121 I=1,Q
X1(I)=REAL(X(I))
X2(I)=AIMAG(X(I))
WRITE(74,67)X1(I),X2(I)
67 FORMAT(1X,'m',Y,I2,'-',E12.5,2X,'J',E12.5)
121 CONTINUE
STOP
END
```

Programa 6. Nodo.for

C ESTE PROGRAMA CALCULA LOS VOLTAJES DE NODO DE UNA RED ELECTRICA, PARA
C REALIZAR EL CALCULO REQUIERE COMO DATOS EL NUMERO DE RAMAS, EL NUMERO DE
C NODOS, LA MATRIZ DE CONEXION RAMA-NODO, EL VECTOR DE CORRIENTES DE GENE
C RACION Y LA MATRIZ DE ADMITANCIAS DE RAMA.

```
DIMENSION YB1(20,20),CP1(20),L(20,20),B(20,20),CP(20)
DIMENSION X(20),AP(20,20),YB(20,20),ZP(20,20),CIP(20,20)
DIMENSION YP(20,20),ATP(20,20),PRE(20,20),PIMA(20,20)
DIMENSION VXR(20),VXIM(20),YP2(20,20),YP3(20,20),CPR1(20)
DIMENSION CP11(20),VX(20),VX1(20),VX2(20)
COMPLEX YB,YB1,CP1,X,L,B,SUM,CP,VX,YP,ZP,CIP
INTEGER P,N,AP

OPEN(12,FILE='NR.DAT')
OPEN(13,FILE='NNOD.DAT')
WRITE(*,*)'CUANTAS RAMAS SON'
READ(12,*)P
WRITE(*,*)'CUANTOS NODOS SON'
READ(13,*)N
WRITE(*,*)'DAME LOS ELEMENTOS DE LA MATRIZ DE CONEXION RAMA'
WRITE(*,*)'PUNTO DE UNION'
OPEN(18,FILE='RNOD.DAT')
OPEN(19,FILE='RNOD1.DAT')
DO 8 I=1,P
DO 9 J=1,N
READ(18,57) AP(I,J)
WRITE(19,56)I,J,AP(I,J)
57 FORMAT(I2)
56 FORMAT(1X,'An',Y,I2,1X,I2,'-',I2)
9 CONTINUE
8 CONTINUE

c CALCULO DE LA TRANSPUESTA DE LA MATRIZ DE CONEXION
c RAMA-PUNTO DE UNION
DO 19 I=1,N
DO 20 J=1,P
ATR(I,J)=AP(J,I)
20 CONTINUE
19 CONTINUE
C CALCULO DE LOS ELEMENTOS DE YP
WRITE(*,*)'DAME LOS ELEMENTOS DE ZP'
OPEN(54,FILE='ZP.DAT')
OPEN(53,FILE='YP1.DAT')
DO 97 I=1,P
DO 96 J=1,P
IF(I.EQ.J) THEN
READ(54,*)ZP(I,J)
ENDIF
96 CONTINUE
97 CONTINUE
DO 21 I=1,P
```

Anexo. Algoritmos codificados en lenguaje de programación.

```
DO 22 J=1,P
IF(J.EQ.1) THEN
YP(L,J)=1/ZP(L,J)
YP2(L,J)=REAL(YP(L,J))
YP3(L,J)=AIMAG(YP(L,J))
WRITE(53,103)L,J,YP2(L,J),YP3(L,J)
103 FORMAT(1X,YP,'(',I2,1X,I2,')',='F7.3,2X,'J,F7.3)
ENDIF
22 CONTINUE
21 CONTINUE

C   CALCULO DE LA MATRIZ YBUS'
C1P(1,1)=0
DO 5 I=1,N
DO 6 J=1,P
C1P(L,J)=0
YB(L,J)=0
DO 7 K=1,P
C1P(L,J)=C1P(L,J)+ATP(L,K)*YP(K,J)
7 CONTINUE
6 CONTINUE
5 CONTINUE
YB(1,1)=0
DO 31 I=1,N
DO 32 J=1,N
YB(L,J)=0
DO 33 K=1,P
YB(L,J)=YB(L,J)+C1P(L,K)*AP(K,J)
33 CONTINUE
YB1(L,J)=YB(L,J)
PRE(L,J)=REAL(YB1(L,J))
PIMA(L,J)=AIMAG(YB1(L,J))
32 CONTINUE
31 CONTINUE
WRITE(*,*)SE IMPRIME LA MATRIZ YBUS'
OPEN(73,FILE='YBARRA.RES')
DO 61 I=1,N
DO 62 J=1,N
WRITE(73,10)I,J,PRE(L,J),PIMA(L,J)
10 FORMAT(1X,'(',I2,1X,I2,')',='F8.4,2X,'J,F8.4)
62 CONTINUE
61 CONTINUE

C   CALCULO DE LOS VOLTAJES DE NODO
WRITE(*,*)DAME LAS CORRIENTES DE GENERACION'
OPEN(299,FILE='IG.DAT')
OPEN(103,FILE='IG1.DAT')
DO 71 I=1,N
READ(299,*)CP1(I)
CPR1(I)=REAL(CP1(I))
CPI1(I)=AIMAG(CP1(I))
WRITE(103,102)L,CPR1(I),CPI1(I)
102 FORMAT(1X,Ig,'(',I2,')',='F5.3,2X,'J,F5.3)
71 CONTINUE

DO 81 I=1,N
DO 90 J=1,N
IF(J.EQ.1) THEN
L(L,J)=YB1(L,J)
ENDIF
90 CONTINUE
81 CONTINUE
DO 91 J=1,N
B(1,J)=YB1(1,J)/L(1,1)
91 CONTINUE
DO 900 J=2,N
DO 66 I=1,N
SUM=(0,0)
DO 72 K=1,J-1
SUM=SUM+(L(I,K))*B(K,J))
```

Anexo. Algoritmos codificados en lenguaje de programación.

```
72 CONTINUE
L(I,J)=YB1(I,J)-(SUM)
66 CONTINUE
B(J,J)=1
DO 82 I=J+1,N
SUM=(0,0)
DO 92 K=1,J-1
SUM=SUM+(L(I,K))*(B(K,J))
92 CONTINUE
B(I,I)=(YB1(I,I)-SUM)/L(I,I)
82 CONTINUE
900 CONTINUE
CP(I)=CP1(I)/L(I,1)
DO 101 I=2,N
SUM=(0,0)
DO 112 J=1,I-1
SUM=SUM+L(I,J)*CP(J)
112 CONTINUE
CP(I)=(CP1(I)-SUM)/L(I,I)
101 CONTINUE
WRITE(*,*)VOLTAJES DE NODO
VX(N)=CP(N)
DO 104 I=2,N
NVBL=N-I+1
SUM=(0,0)
NP1=NVBL+1
DO 105 J=NP1,N
SUM=SUM+B(NVBL,J)*VX(J)
105 CONTINUE
VX(NVBL)=CP(NVBL)-SUM
VXR(NVBL)=REAL(VX(NVBL))
VXI(NVBL)=AIMAG(VX(NVBL))
104 CONTINUE
OPEN(73,FILE='VNOD.RES')
DO 73 I=1,N
VX1(I)=REAL(VX(I))
VX2(I)=AIMAG(VX(I))
WRITE(73,11)I,VX1(I),VX2(I)
11 FORMAT(1X,'Vn','(',I2,')',',',E12.5,2X,'I',E12.5)
73 CONTINUE
STOP
END
```

Programa 7. Red11.for

```
C PROGRAMA PARA RESOLVER LAS ECUACIONES DE FLUJOS DE POTENCIA DE
C UNA RED DE N NODOS, LOS DATOS SON LEIDOS DESDE LOS ARCHIVOS
C IMPA.DAT MATRIZ DE ADMITANCIAS
C NODOD.DAT NUMERO DE NODOS
C NODO.DAT CARACTERISTICAS DE NODOS
C TOL.DAT TOLERANCIA
C ALFA.DAT FACTOR DE ACELERACION
C LOS NODO DE CARGA SE IDENTIFICARAN COMO NODOS TIPO 1
C Y LOS NODOD DE GENERACION SE IDENTIFICARAN
C COMO NODOS TIPO 2.
C LOS RESULTADOS SE ENVIAN A LOS SIGUIENTES ARCHIVOS:
C VOLT.RES VOLTAJES DE NODO
C ANGULO.RES ANGULO DE LOS VOLTAJES DE NODO
C PREAL.RES FLUJOS DE POTENCIA REAL
C PREAC.RES FLUJOS DE POTENCIA REACTIVA
C PERR.RES PERDIDAS DE POTENCIA REAL
C PERQ.RES PERDIDAS DE POTENCIA REACTIVA
C
C DECLARACION DE VARIABLES
C
DIMENSION NODO(20),VOLT(20),REAC(20),VREAL(20)
DIMENSION REACT(20,20),AVOLT(20)
```

Anexo. Algoritmos codificados en lenguaje de programación.

```
INTEGER N,M,NN,A,NODOS1L
COMPLEX D
DIMENSION VCON(20),Y(20,20),VOLTACT(20),QINF(20)
DIMENSION QSUP(20),FP(20,20),ABSO(20),U(20),ANG(20)
DIMENSION PR(20,20),PQ(20,20),FPR(20,20),FPQ(20,20)
COMPLEX VOLT,VCON,S,SUM,Y,VOLTACT,DELTA V,VUNIT,FP
INTEGER P,Q

C
C INICIO DEL PROGRAMA CON LA CAPTURA DE DATOS
C
C DECLARACION DE LOS ARCHIVOS DE LECTURA
C
    OPEN (1,FILE='MPA.DAT')
    OPEN (2,FILE='NODO.DAT')
    OPEN (9,FILE='TOL.DAT')
    OPEN (10,FILE='ALFA.DAT')
    OPEN (11,FILE='NODOL.DAT')

C
C DECLARACION DE LOS ARCHIVOS DE RESULTADOS
C
    OPEN (3,FILE='VOLT.RES')
    OPEN (4,FILE='ANGULO.RES')
    OPEN (5,FILE='PREAL.RES')
    OPEN (6,FILE='PREAC.RES')
    OPEN (7,FILE='PERR.RES')
    OPEN (8,FILE='PERQ.RES')

C
C LECTURA DEL NUMERO DE NODOS DESDE EL ARCHIVO
C
100 READ (11,*) N
C
C LECTURA DE TIPO DE NODO Y CARACTERISTICA
C
    NN=N
    DO 102 I=1,N
    READ (2,*) NODO(I)
    IF (NODO(I),EQ.0) THEN
    READ (2,*) VOLT(I)
    END IF
    IF (NODO(I),EQ.1) THEN
    READ (2,*) VREAL(I),REAC(I)
    VOLT(I)=(1,0)
    END IF
    IF (NODO(I),EQ.2) THEN
    READ (2,*) VREAL(I),AVOLT(I)
    READ (2,*) QINF(I),QSUP(I)
    VOLT(I)=(1,0)*AVOLT(I)
    END IF
102 CONTINUE
C
C CAPTURA DE LOS DATOS DE LA MATRIZ DE ADMITANCIAS
C
    DO 104 I=1,N
    DO 105 J=1,N
    READ (1,*) Y(I,J)
105 CONTINUE
104 CONTINUE
C
C LECTURA DE LA TOLERANCIA DESDE ARCHIVO
C
    READ(9,*)TOLI
C
C LECTURA DEL FACTOR DE ACELERACION
C
    READ(10,*)ALFAI
C
C FIN DE LA CAPTURA DE DATOS
C
C INICIO DEL PROGRAMA PRINCIPAL
C
```

Anexo. Algoritmos codificados en lenguaje de programación.

```
IT=0
115 IT=IT+1
    IF (IT.GT.50) THEN
        WRITE(*,*)'EL SISTEMA NO CONVERGE'
        WRITE(3,*)'EL SISTEMA NO CONVERGE'
        WRITE(4,*)'EL SISTEMA NO CONVERGE'
        WRITE(5,*)'EL SISTEMA NO CONVERGE'
        WRITE(6,*)'EL SISTEMA NO CONVERGE'
        WRITE(7,*)'EL SISTEMA NO CONVERGE'
        WRITE(8,*)'EL SISTEMA NO CONVERGE'
    ELSE
        WRITE(*,*)'ITERACION',IT
        DO 110 P=2,N
            IF (NODO(P),EQ,1) THEN
C
C  RUTINA PARA NODO DE CARGA
C
                VCON(P)=CONJG(VOLT(P))
                SUM=(0,0)
C
C  CALCULO DE LA SUMATORIA
C
                DO 111 Q=1,N
                    IF (P.NE.Q) THEN
                        SUM=SUM+Y(P,Q)*VOLT(Q)
                    END IF
111 CONTINUE
C
C  APLICACION DE LA FORMULA DE ITERACION
C
                S=(1,0)*(-VREAL(P))+(0,-1)*(-REAC(P))
                VOLTACT(P)=1/Y(P,P)*((S/VCON(P))-SUM)
C
C  APLICACION DEL FACTOR DE ACELERACION
C
                VOLTACT(P)=VOLT(P)+ALFA1*(VOLTACT(P)-VOLT(P))
C
C  RESTA PARA EL CRITERIO DE CONVERGENCIA
C
                T=CABS(VOLTACT(P))
                V=CABS(VOLT(P))
                W=T-V
                ABSO(P)=ABS(W)
C
                VOLT(P)=VOLTACT(P)
                ELSE
C
C  RUTINA PARA NODO DE GENERACION
C
C  CALCULO DE LA POTENCIA REACTIVA DEL NODO
C
                SUM=(0,0)
                DO 112 Q=1,N
                    SUM=SUM+Y(P,Q)*VOLT(Q)
112 CONTINUE
                REAC(P)=-AJMAG(SUM)
C
C  VERIFICAR SI REVASA LOS LIMITES DE POTENCIA REACTIVA
C
                IF (REAC(P).LT.QINF(P)) THEN
                    WRITE(*,*)'EL NODO',P,' NO ALCANZO EL LIMITE INFERIOR DE Q'
                    REAC(P)=QINF(P)
                IF (REAC(P).GT.QSUP(P)) THEN
                    WRITE(*,*)'EL NODO',P,' REVASO EL LIMITE SUPERIOR DE Q'
                    REAC(P)=QSUP(P)
                END IF
C
C  RUTINA CUANDO LA POTENCIA REACTIVA NO ESTA DENTRO DE LOS LIMITES
C
                VCON(P)=CONJG(VOLT(P))
```

Anexo. Algoritmos codificados en lenguaje de programación.

```

                SUM=(0,0)
C
C  CALCULO DE LA SUMATORIA
C
                DO 114 Q=1,N
                IF (P.NE.Q) THEN
                SUM=SUM+Y(P,Q)*VOLT(Q)
                END IF
114  CONTINUE
C
C  APLICACION DE LA FORMULA DE ITERACION
C
                S=(1,0)*(VREAL(P))+(0,-1)*(REAC(P))
                VOLTACT(P)=1/Y(P,P)*((S/VCON(P))-SUM)
C
C  APLICACION DEL FACTOR DE ACELERACION
C
                VOLTACT(P)=VOLT(P)+ALFA1*(VOLTACT(P)-VOLT(P))
C
C  RESTA PARA EL CRITERIO DE CONVERGENCIA
C
                T=CABS(VOLTACT(P))
                V=CABS(VOLT(P))
                W=T-V
                ABSO(P)=ABS(W)
C
                VOLT(P)=VOLTACT(P)
                ELSE
C
C  CALCULO DE LA SUMATORIA
C
                VCON(P)=CONJG(VOLT(P))
                SUM=(0,0)
                DO 113 Q=1,N
                IF (P.NE.Q) THEN
                SUM=SUM+Y(P,Q)*VOLT(Q)
                END IF
113  CONTINUE
C
C  APLICACION DE LA FORMULA DE ITERACION
C
                S=(1,0)*(VREAL(P))+(0,-1)*(REAC(P))
                VOLTACT(P)=1/Y(P,P)*((S/VCON(P))-SUM)
C
C  OBTENCION DEL ANGULO DEL NODO MEDIANTE UN VECTOR UNITARIO
C
                VUNIT=VOLTACT(P)/CABS(VOLTACT(P))
                VOLTACT(P)=AVOLT(P)*VUNIT
C
C  APLICACION DEL FACTOR DE ACELERACION
C
                VOLTACT(P)=VOLT(P)+ALFA1*(VOLTACT(P)-VOLT(P))
C
C  RESTA PARA EL CRITERIO DE CONVERGENCIA
C
                T=CABS(VOLTACT(P))
                V=CABS(VOLT(P))
                W=T-V
                ABSO(P)=ABS(W)
C
                VOLT(P)=VOLTACT(P)
                END IF
                END IF
110  CONTINUE
C
C  APLICACION DEL CRITERIO DE TOLERANCIA
C
                CONT=0
                DO 124 I=2,N
                IF (ABSO(I).GT.TOLI) THEN
```

Anexo. Algoritmos codificados en lenguaje de programación.

```
        CONT=CONT+1
        END IF
124  CONTINUE
        IF (CONT.NE.0) THEN
            GOTO 115
        END IF
        END IF
C
C  CALCULO DE LAS PERDIDAS Y DE LOS FLUJOS DE POTENCIA
C
        DO 117 P=1,N
            DO 118 Q=1,N
                IF (P.NE.Q) THEN
                    FP(P,Q)=CONJG(VOLT(P))*Y(P,Q)*(VOLT(P)-VOLT(Q))
                    FPR(P,Q)=REAL(FP(P,Q))
                    FPQ(P,Q)=AIMAG(FP(P,Q))
                END IF
118  CONTINUE
117  CONTINUE
            DO 122 P=1,N
                DO 123 Q=1,N
                    IF (P.NE.Q) THEN
                        F=ABS(REAL(FP(P,Q)))-ABS(REAL(FP(Q,P)))
                        PR(P,Q)=ABS(F)
                        F=ABS(AIMAG(FP(P,Q)))-ABS(AIMAG(FP(Q,P)))
                        PQ(P,Q)=ABS(F)
                    END IF
123  CONTINUE
122  CONTINUE
                DO 116 I=1,N
C
C  DESGLOSE EN VOLTAJE Y ANGULO
C
                    U(I)=CABS(VOLT(I))
                    V=REAL(VOLT(I))
                    W=AIMAG(VOLT(I))
                    ANG(I)=ATAN(W/V)*180/3.1416
                    WRITE(3,500) I, U(I)
                    WRITE(4,500) I, ANG(I)
116  CONTINUE
                    DO 120 P=1,N
                        DO 121 Q=1,N
                            IF (P.NE.Q) THEN
                                WRITE(5,501)P,Q,FPR(P,Q)
                                WRITE(6,501)P,Q,FPQ(P,Q)
                                WRITE(7,501)P,Q,PR(P,Q)
                                WRITE(8,501)P,Q,PQ(P,Q)
                            END IF
121  CONTINUE
120  CONTINUE
                    500  FORMAT (I4, 4X, F15.7)
                    501  FORMAT (I4, 4X, I4, 4X, F15.7)
                STOP
            END
```