

FLUJO DE POTENCIA ELECTRICA EN TORNO AL PUNTO CRITICO

Christian Schaerer

José Atlasovich

*Trabajo de Investigación presentado como parte de los requisitos necesarios
para la obtención del título de Ingeniero Electromecánico*

FACULTAD DE CIENCIAS FISICAS Y MATEMATICAS

UNIVERSIDAD NACIONAL DE ASUNCION

*San Lorenzo - Paraguay
Mayo de 1995*

Salmo 23

Salmo de David

*El Señor es mi Pastor; nada me falta.
A verdes pastos me lleva a reposar
y a donde brota el agua fresca me conduce.*

*Fortalece mi alma.
Por el camino del bueno me dirige
por amor de su nombre.*

*Aunque pase por quebradas muy oscuras
no temo ningún mal
porque tú estas conmigo,
tu bastón y tu vara me protegen.*

*Me sirves a la mesa
frente a mis adversarios,
con aceites tú perfumas mi cabeza
y rellenas mi copa.*

*Me acompaña tu bondad y tu favor
mientras dura mi vida,
mi mansión será la casa del Señor
por largo, largo tiempo.*

A nuestros Padres y Hermanos...

Agradecimientos

Nuestra más profunda y sincera gratitud al Prof. Benjamín Barán, quien con sus dotes de gran profesor y excelente persona supo guiarnos con su constante ayuda y consejos hacia una feliz culminación de este proyecto.

Nuestro sincero agradecimiento al Ing. Jorge Achón y al Ing. Walter Causarano, por su gran paciencia y comprensión. Al Ing. Jorge Achón en particular por ayudar a organizar nuestro trabajo y al Ing. Walter Causarano por sus orientaciones en Ingeniería Eléctrica.

Una mención muy especial para la Directora del Centro Nacional de Computación, Prof. Blanca de Trevisán, quien fue la persona que posibilitó nuestra presencia en dicha institución y que no escatimó esfuerzos para que el proyecto llegue a buen puerto.

Nuestra más sincera gratitud a los compañeros del Centro Nacional de Computación que siempre nos brindaron su invaluable ayuda para afrontar los problemas que el fascinante mundo de la computación nos presenta.

La tesis es para nosotros la culminación de muchos años de trabajo y sacrificio en nuestro paso por la facultad. No podemos olvidar de agradecer entonces a todas las personas que de una u otra forma colaboraron en nuestra formación. En especial nos gustaría agradecer al Ing. Miguel Angel Volpe, Ing. Carmen Gómez de Kropff y a Tata, los cuales constante e incansablemente nos brindaron sus consejos.

Nos gustaría además agradecer a nuestros compañeros y amigos, y muy especialmente a la familia Bordón Petretich, a la familia Castro Matto y a la familia Fidabel que siempre nos han tratado como parte de ellas.

Finalmente, agradecemos a nuestras familias que nos dieron el ambiente de felicidad y amor para que de este tiempo vivido nos quede el recuerdo de momentos inolvidables.

TABLA DE CONTENIDO

TABLA DE CONTENIDO	1
Lista de Figuras	4
Lista de Tablas	4
Símbolos y Abreviaturas	5
1. PRESENTACION	10
1.1 Introducción.....	10
1.2 Breve Descripción del Problema	11
1.3 Resumen Histórico.....	13
1.4 Procesamiento Paralelo y Distribuido: Su aplicación a Sistemas de Potencia	17
1.4.1 Computación y Comunicación.....	17
1.4.2 Computación Paralela y Distribuida	18
1.4.3 Software para Computación Distribuida	20
1.5 Objetivos y Organización de esta Obra	21
1.5.1 Originalidad	21
1.5.2 Objetivos generales.....	22
1.5.3 Objetivos específicos	22
1.5.4 Metodología.....	23
1.5.5 Organización.....	23
2. EL FLUJO DE POTENCIA ELECTRICA	24
2.1 Introducción.....	24
2.2 Modelo Matemático de la Red	26
2.3 Definición Analítica del Problema.....	28
2.4 Métodos de Solución	30

2.4.1	Métodos de Gauss y Gauss-Seidel	30
2.4.1.1	Algoritmo Iterativo de Gauss.....	31
2.4.1.2	Algoritmo Iterativo de Gauss-Seidel.....	32
2.4.2	Método de Newton-Raphson.....	34
2.4.2.1	Ecuaciones del Flujo de Carga en un Sistema de Potencia	36
2.4.2.2	Resumen del Método (pasos a seguir)	39
2.4.2.3	Técnicas que hacen al Newton-Raphson Competitivo.....	40
2.4.2.4	Características del Newton-Raphson	41
2.4.3	Métodos Desacoplados Derivados del Newton-Raphson.....	42
3.	CALCULO DEL PUNTO DE COLAPSO	43
3.1	Introducción.....	43
3.2	Métodos para encontrar el Punto de Colapso	46
3.2.1	Método de Continuación	47
3.2.1.1	Reformulación de las Ecuaciones del <i>Load Flow</i>	48
3.2.1.2	Aplicación de un Algoritmo de Continuación.....	51
3.2.1.3	Predicción de la Siguiete Solución.....	51
3.2.1.4	Parametrización y Corrector	53
3.2.1.5	Elección del Parámetro de Continuación.....	55
3.2.1.6	Detección del Punto Crítico.....	56
3.2.2	Método del Punto de Colapso	58
3.2.3	Comparación entre el MC y el PoC.....	60
3.3	Métodos para encontrar el Punto de Colapso más Cercano	60
3.3.1	Método Iterativo	61
3.3.2	Método Directo.....	63
4.	PROPOSICION DE NUEVOS METODOS DE SOLUCION	64
4.1	Descomposición LU.....	64
4.2	PoC con Variante Aceleradora	66
4.2.1	Separación del Sistema	67
4.2.2	Cómo hacer sólo una Descomposición LU	69
4.2.3	Una Dificultad a Salvar	70
4.3	<i>Team Algorithms</i> : Concepto.....	72

4.4 <i>Team Algorithms</i> : Formulación Matemática	74
4.5 Sincronismo y Asincronismo	75
4.6 El PoC en un Sistema de Procesamiento Distribuido	76
4.6.1 Primer Esquema	78
4.6.2 Segundo Esquema	78
4.6.3 Tercer Esquema	81
5. RESULTADOS EXPERIMENTALES	84
5.1 Ambiente Computacional	84
5.2 Problemas de Prueba	85
5.3 Curvas obtenidas con el MC	87
5.4 Comparación entre Diferentes Métodos Secuenciales	91
5.5 Ventajas del <i>Overlapping</i>	94
5.6 Sistemas Distribuidos	95
5.6.1 Métodos Implementados	96
5.6.2 Criterios de Comparación	96
5.6.3 Resultados Experimentales	98
6. CONCLUSIONES	101
Lista de Referencias Bibliográficas	105

Lista de Figuras

Figura 1.1	Localización del Punto Crítico	12
Figura 2.1	Red sencilla mostrando variables nodales.....	26
Figura 2.2	Método de Newton-Raphson.....	34
Figura 3.1	Lugar geométrico de los Puntos de Colapso	44
Figura 3.2	Sistema eléctrico de dos barras	45
Figura 3.3	Método de Continuación.....	47
Figura 3.4	Flujograma del Método de Continuación.....	57
Figura 3.5	Método Iterativo	61
Figura 4.1	Implementación del Segundo Esquema de solución.....	79
Figura 4.2	Comportamiento de los autovectores en la implementación síncrona	81
Figura 5.1	Problema ejemplo de 88 barras	86
Figura 5.2	Partición del problema ejemplo de 88 barras	87
Figura 5.3	Problema IEEE 5. Crecimiento de cargas ‘A’	87
Figura 5.4	Problema IEEE 5. Crecimiento de cargas ‘B’	88
Figura 5.5	Problema IEEE 14. Fases de tensión	89
Figura 5.6	Problema IEEE 14. Módulos de Tensión.....	89
Figura 5.7	Problema PAR 16. Fases de tensión	90
Figura 5.8	Problema PAR 16. Módulos de tensión.....	91
Figura 5.9	Comparación entre métodos secuenciales.....	93

Lista de Tablas

Tabla 5.1	Comparación entre métodos secuenciales.....	92
Tabla 5.2	Resultados que justifican el uso del <i>overlapping</i>	95
Tabla 5.3	Comparación entre los métodos paralelos y secuenciales al resolver el problema IEEE 30.....	98
Tabla 5.4	Comparación entre los métodos paralelos y secuenciales al resolver el problema ejemplo de 88 barras.....	99

Símbolos y Abreviaturas

SÍMBOLO o ABREVIATURA	SIGNIFICADO	PRIMERA REFERENCIA
*	subíndice que designa al punto de colapso más cercano, o superíndice que denota ‘complejo conjugado’	3.1
◦	superíndice que indica solución	4.4
,	“prima”, denota solución predicha	3.2.1.3
$ \cdot $	valor absoluto	3.2.1.5
$\ \cdot\ _q$	norma q de una matriz (cuando q no se especifica, se trata de una norma genérica)	3.2.2
A	matriz auxiliar	4.1
α	factor que hace $\ P(\mathbb{I}k)\ = 1$	3.3.1
A_{km}	submatriz de A	4.2.1
A_r	aceleración relativa	5.6.2
B	vector auxiliar	4.1
B_k	subvector de B	4.2.1
C	conjunto de números complejos	2.2
C	matriz triangular inferior auxiliar	4.2.2
c	vector columna constante	3.2.2
C_{km}	submatriz de C	4.2.2
c_m	subvector de c	4.6
D	matriz triangular superior auxiliar	4.2.2
D_{km}	submatriz de D	4.2.2
ΔP	vector de errores de potencia activa	2.4.2.1
ΔP_k	error de potencia activa en la barra k	2.4.2.1
ΔQ	vector de errores de potencia reactiva	2.4.2.1
$\Delta \theta$	vector de correcciones de fase	2.4.2.1
ΔQ_k	error de potencia reactiva en la barra k	2.4.2.1
ΔV	vector de correcciones de módulo	2.4.2.1

SIMBOLO o ABREVIATURA	SIGNIFICADO	PRIMERA REFERENCIA
$\frac{\Delta V}{V}$	vector cuyas componentes provienen de dividir cada ΔV_k por su correspondiente V_k	2.4.2.1
$\Delta \mathbf{x}(i)$	incremento de \mathbf{x} en la iteración i	2.4.2
Δx_n	n-ésimo elemento de $\Delta \mathbf{x}$	4.2.1
$\Delta \mathbf{x}_{n-1}$	subvector de $\Delta \mathbf{x}$ que incluye sus $n-1$ primeros términos	4.2.1
$d_j^m(i)$	iteración correspondiente a un subvector \mathbf{x}_j (o \mathbf{w}_j), utilizado en el procesador m en la iteración i de este último	4.6.3
\mathbf{E}	vector de tensiones de barra	2.2
e	base del logaritmo natural	3.1
E'_k	valor auxiliar para calcular E_k	2.4.1.1
\mathbf{e}_j	vector fila de ceros, excepto $e_j=1$	3.2.1.3
E_k	tensión de la barra k	2.2
Φ	función no lineal	4.2
$F(\mathbf{x}) = \begin{bmatrix} f_1(\mathbf{x}) \\ \vdots \\ f_n(\mathbf{x}) \end{bmatrix}$	función no lineal $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$	2.4.2
$F(\mathbf{x}, \Pi) = \mathbf{0}$	ecuación del flujo de carga $F : \mathbb{R}^n \times \mathbb{R}^u \rightarrow \mathbb{R}^n$	3.1
$F_m(\mathbf{x}) = \mathbf{0}$	subproblema de $F(\mathbf{x}) = \mathbf{0}$, asignado al procesador m	4.6
$\Phi_m(\mathbf{z}) = \mathbf{0}$	subproblema de $\Phi(\mathbf{z}) = \mathbf{0}$, asignado al procesador m	4.4
G_m	algoritmo iterativo en el procesador m	4.4
\mathbf{H}	submatriz del jacobiano \mathbf{J}	2.4.2.1
η	valor predicho para x_j^a	3.2.1.4
\mathbf{I}	vector de corrientes eléctricas	2.2
I_k	corriente inyectada a la barra k	2.2
\mathbf{J}	matriz jacobiana de $F(\mathbf{x})$	2.4.2
\mathbf{J}_m	jacobiano de $\Phi_m(\mathbf{x}, \mathbf{w}, \lambda)$	4.6.2

SIMBOLO o ABREVIATURA	SIGNIFICADO	PRIMERA REFERENCIA
K	factor de proporcionalidad	3.3.2
κ	constante	4.6.3
L	matriz triangular inferior	4.1
λ	parámetro escalar de carga	3.1
λ_m	versión de λ en el procesador m	4.6.2
λ'_m	valor de λ que utiliza el operador G_m	4.6.2
M	submatriz del jacobiano J	2.4.2.1
MC	método de Continuación	1.3
μ_{Gk}	cte. de crecimiento de pot. activa generada en la barra k	3.2.1.1
μ_{Lk}	cte. de crecimiento de carga en la barra k	3.2.1.1
N	número de barras	2.2
n	dimensión del vector x	2.4.2
ν	versor	3.1
N	submatriz del jacobiano J	2.4.2.1
N_1	número de barras PQ	2.4.2.1
N_2	número de barras PV	2.4.2.1
NR	método de Newton-Raphson	1.2
Π	vector parámetro de carga	3.1
p	número de procesadores	4.4
$P(\Pi)$	función que da la normal a Σ en un punto con carga Π	3.3.1
Π_0	carga en el punto de trabajo	3.1
PoC	método del Punto de Colapso	1.3
$P_{Lk} + jQ_{Lk}$	potencia de carga en la barra k	3.2.1.1
$P_{Gk} + jQ_{Gk}$	potencia generada inyectada en k	3.2.1.1
$P_{Lk_0} + jQ_{Lk_0}$	potencia de carga inicial en la barra k	3.2.1.1
$P_{Gk_0} + jQ_{Gk_0}$	potencia generada inicial en la barra k	3.2.1.1
ψ_k	ángulo de factor de potencia del crecimiento de carga en k	3.2.1.1
θ	vector de fases de tensión	2.4.2.1
θ_k	fase en la barra k	2.3

SIMBOLO o ABREVIATURA	SIGNIFICADO	PRIMERA REFERENCIA
$\theta_{km} = \theta_k - \theta_m$	diferencia de fase entre barras	2.4.2.1
\mathbf{R}	conjunto de números reales	2.4.2
\mathbf{R}	submatriz del jacobiano \mathbf{J}	2.4.2.1
Σ	hipersuperficie en el espacio \mathbf{R}^u , lugar geom. de los puntos de colapso	3.1
σ	tamaño del paso	3.2.1.3
S_p	<i>speedup</i> o aceleración	5.6.2
$S_k = P_k + jQ_k$	potencia inyectada a la barra k	2.2
TA	<i>Team Algorithms</i>	1.4.3
T_j	j -ésimo elemento de \mathbf{T}	3.2.1.3
$\mathbf{T} = \begin{bmatrix} d\theta \\ dV \\ d\lambda \end{bmatrix}$	vector tangente	3.2.1.3
u	dimensión del vector \mathbf{U}	3.1
\mathbf{U}	matriz triangular superior	4.1
\mathbf{u}	autovector derecho de F_x	3.2.2
U_{km}	elemento de \mathbf{U}	4.1
\mathbf{U}_n	n -ésima columna de \mathbf{U}	4.2.1
\mathbf{U}_{n-1}	submatriz de \mathbf{U} que abarca sus $n-1$ primeras columnas	4.2.1
\mathbf{V}	vector de módulos de tensión	2.4.2.1
V_k	modulo de la tensión en la barra k	2.3
\mathbf{w}	autovector izquierdo de F_x	3.2.2
\mathbf{w}_m	subvector de \mathbf{w} que actualiza el procesador m	4.6
$\mathbf{w}^m(i)$	vector \mathbf{w} disponible en el procesador m en la iteración i	4.6.3
ω	‘peso’ en el administrador	4.6.2
ω_m	‘peso’ en el administrador	4.6.2
ξ	constante	4.6.3
\mathbf{x}_0	vector de estado en el punto de trabajo	3.1
x_j^a	parámetro de continuación	3.2.1.4

SIMBOLO o ABREVIATURA	SIGNIFICADO	PRIMERA REFERENCIA
\mathbf{x}_m	subvector de \mathbf{x} que actualiza el procesador m	4.6
x_n	n-ésimo elemento de \mathbf{x}	4.2.1
\mathbf{x}_{n-1}	subvector de \mathbf{x} que incluye sus $n-1$ primeros términos	4.2.1
$\mathbf{x} = \begin{bmatrix} \theta \\ \mathbf{V} \end{bmatrix}$	vector de estado del sistema	2.4.2.1
$\mathbf{x}^a = \begin{bmatrix} \theta \\ \mathbf{V} \\ \lambda \end{bmatrix}$	vector de estado ampliado	3.2.1.4
$\mathbf{x}^m(i)$	vector \mathbf{x} disponible en el procesador m en la iteración i	4.6.3
$\mathbf{Y} = [Y_{km}]$	matriz nodal de admitancias ($\mathbf{Y} \in \mathbb{C}^{n \times n}$)	2.2
$Y_{km} = G_{km} + jB_{km}$	elemento de la matriz \mathbf{Y}	2.4.2.1
y_{km}	admitancia entre las barras k y m	2.2
\mathbf{z}	vector incógnita de un sistema no lineal de ecuaciones	4.4
\mathbf{z}_m	subvector de \mathbf{z} actualizado por G_m	4.4

1. PRESENTACION

Este capítulo presenta un análisis introductorio de la importancia que reviste el estudio del flujo de potencia eléctrica en torno al *punto crítico*. También explica en forma breve qué es *punto crítico* y cuáles son las dificultades matemáticas encontradas al tratar de localizarlo. Expone además un resumen histórico de trabajos publicados respecto al tema.

Se incluye asimismo, una sección que abarca conceptos de computación paralela y distribuida, resaltando su utilidad como herramienta para resolver el problema en forma rápida y eficaz.

Finalmente, este capítulo expone los objetivos de la presente obra, así como la forma en que fue estructurada.

1.1 Introducción

Una variedad de fuerzas (financieras, reguladoras, medioambientales por mencionar algunas) están obligando a las concesionarias de energía a operar sus sistemas de manera a conseguir su máxima capacidad de generación y transmisión.

Consecuentemente, dichas concesionarias están experimentando nuevos tipos de problemas de estabilidad. Esto se nota en los colapsos de sistema por causa de la inestabilidad de tensión, experimentados en años recientes alrededor del mundo [1].

El riesgo de colapso de tensión, el cual en algunas circunstancias puede ocurrir cerca de los regímenes normales de funcionamiento de la red, es un factor importante en la determinación del límite de potencia transferida y ocasiona un fuerte impacto en la operación y planeamiento de sistemas eléctricos grandes o extensos.

Con el creciente interés en el problema de la estabilidad de tensión, mucha atención está siendo otorgada a la investigación de este fenómeno, consiguiéndose mayor comprensión del problema, así como el desarrollo de técnicas analíticas y computacionales que permiten estudiar y mitigar los factores que lo envuelven.

La complejidad de los modelos matemáticos de los sistemas eléctricos no siempre permite su aprovechamiento óptimo, pues hasta hoy no existe un método suficientemente general y accesible que permita estudiar el comportamiento de dichos sistemas en la vecindad del *Punto Crítico* de los mismos.

La dificultad para resolver el flujo de potencia radica en que dichos sistemas no son lineales y presentan características muy diferenciadas de acuerdo al sistema considerado. Por lo tanto, un método computacional que funciona bien en ciertas circunstancias, deja de hacerlo en otras, dando como resultado métodos utilizados en forma empírica, según el criterio del ingeniero o investigador.

Para países como Paraguay, potencialmente exportadores de energía eléctrica, cuyo desarrollo actual está muy ligado a la generación, distribución óptima y control de la misma, es sumamente beneficioso poseer una herramienta de cálculo adecuada y el *know how* necesario para el estudio detallado de sus sistemas y el de aquéllos a los que en el futuro se hallare interconectado.

1.2 Breve Descripción del Problema

Bajo condiciones normales de funcionamiento, los sistemas eléctricos de transmisión, operan en estado estable y el cálculo básico que se requiere para determinar las características de este estado es llamado *load flow* (o flujo de potencia).

El objetivo del *load flow* es determinar las características de estado estable de operación del sistema de generación y transmisión de potencia, para un dado conjunto de cargas en las barras [2].

A una determinada distribución de carga, en condiciones normales de trabajo, pueden estar asociados varios estados de operación estable, de los cuales sólo uno de ellos es de interés operativo. Esto es, existen varias soluciones físicamente posibles, pero sólo una de ellas es utilizada operacionalmente.

A medida que las cargas aumentan en el sistema, las tensiones en las barras disminuyen. Además, al llegar a un determinado valor de las cargas, se tiene una única solución del *load flow* llamada *Punto de Colapso*, *Punto Crítico* o *Límite de Estabilidad de Tensión en Estado Estable del Sistema* [3]. Para valores mayores de las cargas no existe solución, como puede observarse en la figura 1.1.

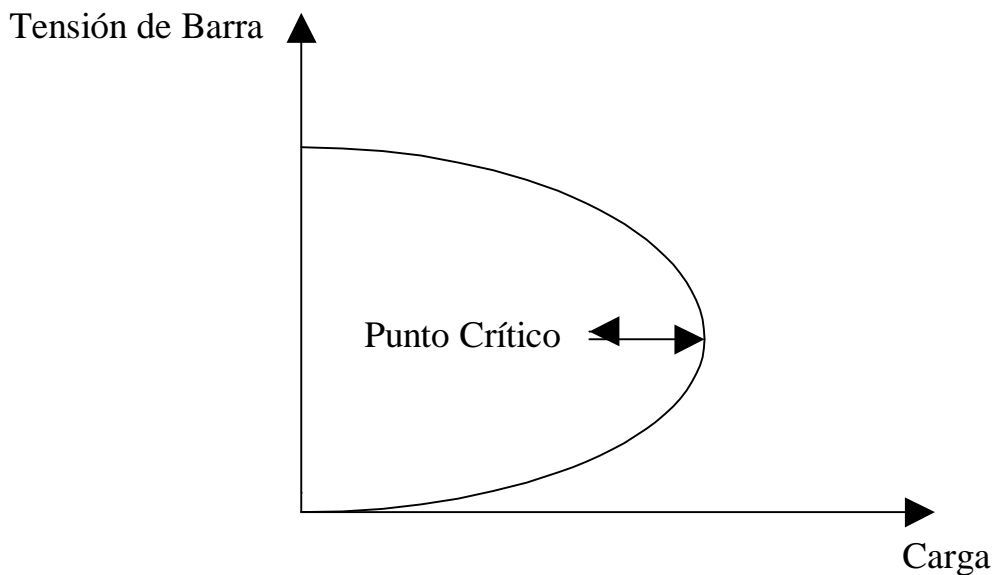


Figura 1.1 Localización del Punto Crítico

Una particular dificultad encontrada en la búsqueda del punto de colapso es que la matriz jacobiana del sistema de ecuaciones correspondiente, se vuelve singular en dicho punto, en consecuencia, intentos de resolver el flujo de carga en las cercanías del punto crítico fallan debido a que los métodos convencionales como el Newton-Raphson (NR) no convergen.

De acuerdo a lo establecido anteriormente, una forma de encontrar el límite de capacidad de cargas del sistema es incrementar las mismas gradualmente hasta que la convergencia no pueda ser obtenida utilizando técnicas ordinarias de *load flow*. Sin embargo, hacer esto requiere intervención

manual. Además, al acercarse al punto crítico, comienzan a surgir problemas de convergencia, y los ingenieros nunca tienen la certeza de donde el límite real se encuentra, convirtiéndose este procedimiento en muy engorroso y arduo [4].

Los métodos presentados en este trabajo son superiores que un *load flow* convencional, puesto que presentan mejores propiedades numéricas en la singularidad del jacobiano, salvando la misma, proporcionando más información y encontrando el *Punto de Colapso* sin necesitar intervención manual.

1.3 Resumen Histórico

La presentación histórica desarrollada a continuación tiene el objetivo de enfatizar el interés que el tema de estabilidad de tensión ha venido recibiendo, así como su evolución en el tiempo y los diferentes puntos de vista desde los cuales ha sido encarado.

En la medida en que los sistemas de potencia se expandían, los problemas de estabilidad de tensión comenzaron a hacerse más frecuentes y de mayor magnitud, un ejemplo de esto último es el *black-out* producido en Francia en el año 1978 [5]. En consecuencia, fueron abiertas un gran número de líneas de investigación en la búsqueda de una comprensión clara del fenómeno.

Es difícil establecer una clasificación temporal de los trabajos publicados, debido a que a medida que los problemas de estabilidad se presentaban, éstos eran abordados con todas las herramientas y conocimientos disponibles para su esclarecimiento, y las líneas de investigación abiertas fueron avanzando en forma paralela.

Estableceremos como punto de partida las décadas del 60 y 70, en las cuales comenzaron a ser publicados con mayor regularidad artículos relativos a la inestabilidad de tensión [6,7,8]. Los mismos incluían descripciones de incidentes ocurridos en sistemas eléctricos.

Posteriores investigaciones [9,10,11] fueron orientadas de tal forma a comprender y explicar el problema del colapso de tensión, a través de un análisis cualitativo.

La mayoría de estos trabajos describían los colapsos de tensión y las circunstancias operativas en que los mismos se habían producido, sin embargo, no presentaban técnicas analíticas para su estudio.

Entretanto, surgen trabajos [12,13,14] relativos a la multiplicidad de las soluciones de un conjunto de ecuaciones no lineales, como es el caso de las ecuaciones que modelan el flujo de carga. Estos trabajos intentan relacionar la existencia de soluciones múltiples con el fenómeno de colapso.

Hay quienes intentan relacionar la existencia de soluciones múltiples con el colapso de tensión utilizando como herramienta la *teoría de la bifurcación* [15,16,17].

Al mismo tiempo, otros investigadores [18,19] intentan analizar el fenómeno, basando su estudio en la teoría de la estabilidad dinámica de sistemas lineales y no lineales.

A fines de la década del 80 e inicio de los 90, son publicados trabajos [20,21,22,23,24,25] que analizan el colapso de tensión y proponen algún tipo de acción. La mayoría de estos trabajos asocian el fenómeno con la máxima carga que el sistema puede soportar.

La idea general expuesta por los trabajos referenciados arriba, es entonces intentar calcular cuál es la máxima carga y a qué distancia se encuentra de las condiciones normales de operación. Para lograrlo, utilizan diferentes tácticas, tales como:

- Aumentar a pasos la carga del sistema e ir resolviendo a cada paso el flujo de potencia. La carga es considerada máxima cuando el algoritmo de flujo de potencia diverge, o las derivadas de tensión respecto a la carga son muy grandes.
- Intentar evitar las dificultades numéricas encontradas en las cercanías del punto de carga máxima y llegar a dicho punto en forma sistemática, sin necesidad de intervención manual, utilizando modelos matemáticos no convencionales, reformulando el problema y otros artificios.
- Intentar resolver el problema analíticamente, lo que requiere el uso de sistemas bastante simplificados, por ejemplo, de dos barras.

A medida que evolucionaron los estudios, las líneas de investigación fueron delimitándose y sin duda una de las de mayor influencia es el análisis de la estabilidad de tensión basado en la determinación del límite de carga.

Dentro de esta línea de investigación, uno de los trabajos que pueden considerarse de mayor importancia es el de Venkataramana Ajjarapu y Colin Christy [3]. En este artículo publicado en el año 1992, se presenta una herramienta llamada *Método de Continuación* (MC) que consiste en un procedimiento para determinar el límite de estabilidad de tensión en estado estable de un sistema, evitando la singularidad en el *punto crítico*.

En el mismo año, un análisis del comportamiento del *load flow* en las cercanías del punto de singularidad, desarrollado por F.D.Gallana y Z.C. Zeng [26], da como principal resultado la formulación analítica de la relación entre la potencia inyectada al sistema y las tensiones resultantes en la vecindad de la singularidad. Este análisis es teóricamente importante puesto que puede predecir y explicar el comportamiento del *load flow* cerca de una singularidad.

El fenómeno del punto de colapso, según Gallana y Zeng, se caracteriza por:

- La no existencia de soluciones para nuevos incrementos de las potencias inyectadas.
- La bifurcación de la tensión en dos soluciones cercanas.
- El súbito colapso de tensión para pequeños incrementos de potencia, debido a que el comportamiento de la solución cerca de la singularidad está dominado por términos de orden cuadrático.
- La naturaleza del colapso, es decir, cuál barra es más susceptible al colapso.

Una observación a ser tomada en cuenta es la hecha por I. Dobson, donde establece que si bien [26] da un marco teórico con muy importantes relevancias acerca del comportamiento del flujo de carga, aún no puede explicar totalmente el fenómeno del colapso de tensión.

Esta crítica, si bien es cierta de alguna manera, no resta importancia a la obra citada, más bien la completa, ya que la intención del *paper*, al decir de

sus autores, es observar el comportamiento del *load-flow* en las cercanías de la singularidad, antes que explicar el fenómeno del colapso de tensión. Además, pretende orientar la investigación y demostrar la necesidad de que el punto de estado estable donde el jacobiano es singular debe ser evitado operativamente.

El problema que presenta la convergencia del *load flow* al acercarse a la singularidad es explicado por René Jean Jumeau y Hsiao Dong Chiang en su trabajo “*Parameterizations of the Load-Flow Equations for Eliminating Ill-conditioning Load Flow Solution*” publicado en 1992 [27]. Los autores presentan discusiones acerca de la teoría de la bifurcación y dependencia entre los parámetros en las ecuaciones.

Jumeau y Chiang muestran que, si las suposiciones iniciales para las variables de estado del sistema no se sitúan dentro de determinadas regiones, la convergencia del *load flow* no es factible. Estas regiones adoptan una forma cónica, de manera tal que a medida que la solución se acerca al punto crítico, la región se va reduciendo y en el *punto crítico* adopta un único valor.

En la misma obra se manifiesta que, dado un sistema no lineal de ecuaciones, un punto de bifurcación es crítico cuando “la naturaleza cualitativa o la multiplicidad del equilibrio cambian repentinamente”. En el caso estudiado, esto significa que para un parámetro en cuestión se tienen múltiples soluciones de un lado del punto crítico mientras que del otro lado ninguna solución es factible. A este punto se dio en llamar *turning point* o *saddle-node bifurcation point*. Algebraicamente, éste es un punto para el cual el jacobiano de la ecuación de *load flow* es singular y es justamente el *punto crítico*.

En el encuentro de invierno de la IEEE en enero de 1992, realizado en Nueva York, Claudio A. Cañizares y Fernando Alvarado [4] describen la implementación de dos métodos para buscar el *Saddle-node bifurcation* y la comparación entre ellos; tales son el *Método de Continuación* ya comentado y el *Método del Punto de Colapso* (PoC).

Ambos presentan características similares en cuanto a la determinación de un límite de carga preciso. Además, el *Método del Punto de Colapso* proporciona información sobre la sensibilidad de la tensión y presenta buena *performance* en cuanto al tiempo de procesamiento, lo cual justifica su uso como herramienta de cálculo. Por otro lado, el *Método de Continuación*.,

aunque más lento, provee información adicional que puede ser usada en el análisis de ciertos aspectos de estabilidad de tensión.

Por tales motivos, los autores creen que ambos métodos se complementan, particularmente en sistemas donde uno de ellos puede fallar.

Los procedimientos tratados anteriormente consideraban una determinada dirección de crecimiento de las cargas. En el encuentro de verano en julio de 1992, en Seattle, Ian Dobson y Liming Lu [28] proponen dos métodos, el *Método Iterativo* y el *Método Directo*, con los cuales puede obtenerse el punto de colapso localmente más cercano a un punto de trabajo dado, es decir, la dirección de crecimiento más peligrosa.

En mayo de 1994, P.Kundur y B.Gao en su obra "*Practical Considerations in Voltage Stability Assessment*" [1], discuten técnicas de análisis de estabilidad de tensión y presentan herramientas analíticas reunidas en el VSTAB, programa comercial que incorpora el análisis modal y el *Método de Continuación*, además de medidas prácticas para prevenir la inestabilidad de tensión.

De lo expuesto en esta sección, es natural percibir el gran interés que está despertando la búsqueda de herramientas y metodologías que ayuden a los ingenieros e investigadores a determinar rápida y eficazmente el punto crítico, de manera a analizar confiablemente el colapso de tensión, ya que éste constituye un fenómeno importante en los estudios de estabilidad de los sistemas.

1.4 Procesamiento Paralelo y Distribuido: Su aplicación a Sistemas de Potencia

1.4.1 Computación y Comunicación

La aplicación de procesamiento paralelo y distribuido a los problemas de sistemas de potencia se encuentra aún en sus primeras etapas de desarrollo. Rápidos progresos en computación y comunicación prometen un revolucionario incremento de la capacidad de procesamiento en sistemas distribuidos [29].

Las tecnologías de computación y comunicación han venido avanzando muy rápidamente en los últimos años. Este suceso es debido a los progresos en la fabricación de chips, con un 40 a 50 por ciento de incremento por año de la velocidad de procesamiento y un aumento de dos a cuatro veces cada dos años de la capacidad de memoria. Mientras tanto, el costo de los equipos ha venido reduciéndose casi en un 50 por ciento cada año [29].

El progreso en las comunicaciones digitales es aún más impresionante. En la actualidad, las redes locales Ethernet o FDDI tienen capacidades de transferir datos a velocidades que varían entre 10 a 100 Mbps [29]. Inclusive existen sistemas que trabajan por encima de 1Gbps, como Sonet.

El gran crecimiento no sólo es debido a los avances tecnológicos, sino también a las exigencias de los usuarios, las cuales pueden enmarcarse en tres grandes grupos.

1. Incremento de los requerimientos computacionales: Los usuarios desean resolver problemas cada vez más complejos con una mayor precisión.
2. Requerimientos estrictos de tiempo: Los usuarios desean resolver sus problemas cada vez con mayor velocidad. Así, tareas que antes se realizaban *off-line*, necesitan ser hoy día computadas *on-line* o en tiempo real.
3. Imagen y Visualización: Los procesos de imagen y visualización son extremadamente intensos y con una demanda creciente, lo que implica la necesidad de procesar y transferir gran cantidad de datos con gran velocidad.

En suma, la tendencia es de un incremento del rendimiento en problemas *standard* y un continuo crecimiento en el tamaño y complejidad de los problemas abordados.

1.4.2 Computación Paralela y Distribuida

Para resolver los complejos problemas computacionales que iban apareciendo a medida que los computadores quedaban disponibles para la industria y la investigación, los fabricantes de computadoras enfocaron su

desarrollo hacia las supercomputadoras con un sólo procesador vectorial cada vez más veloces. Es así que hoy la tecnología permite construir procesadores que están muy cerca del límite teórico de su velocidad, con lo que se puede predecir que el sostenido aumento de velocidad de procesamiento no podrá continuar en los próximos años.

Al mismo tiempo, los microprocesadores fueron haciéndose cada vez mas potentes y baratos por la economía de escala en que se manejan. Como ejemplo, puede citarse el innumerable parque mundial de computadores personales.

Como consecuencia natural de los dos fenómenos arriba citados, la tendencia de la industria y del mercado se volcó hacia la Computación Distribuida/Paralela, por su capacidad de aprovechar el bajo costo de los microprocesadores en el procesamiento de grandes volúmenes de información y para la solución de complejos problemas computacionales.

Actualmente, la tendencia mundial para el procesamiento mediante computadores de los citados problemas, lo constituyen las *Workstations* interconectadas entre sí a través de redes de comunicación. Además, es posible trabajar en ambientes computacionales heterogéneos que aprovechen toda la capacidad computacional existente. En otras palabras, máquinas tan diferentes como computadores personales, *mainframes*, *workstations* y supercomputadores pueden coexistir y realizar un trabajo combinado que redunde en la eficiencia de los problemas computacionales que se estén tratando.

Para formalizar el concepto de *Proceso Distribuido* se transcribe la definición dada en [29]:

Un Sistema de Procesamiento Distribuido es aquél en el cual múltiples procesadores son conectados entre sí de forma tal que puedan compartir información

En esta definición sería interesante resaltar que los computadores comparten información entre sí con el objetivo de resolver un determinado problema, del cual se están ocupando.

1.4.3 Software para Computación Distribuida

Si bien los sistemas distribuidos presentan la mejor relación *performance*/precio del mercado, aún no han sido (totalmente) incorporados a los centros de producción por la escasez de programas (softwares) que puedan aprovechar de forma óptima el potencial de la computación distribuida.

La computación distribuida tiene arquitectura compatible para la utilización de diferentes algoritmos en cada procesador, de forma tal que, resolviendo cada uno de ellos una parte de un problema global, entre todos resuelvan el problema global.

Esta técnica, llamada *Team Algorithm* (TA) [30], permite además trabajar en ambiente asíncrono, es decir, diferentes procesadores trabajando a diferentes velocidades, sin sincronizar necesariamente todas sus tareas.

Como una consecuencia de las dos características citadas, un *Team Algorithm* permite [30]:

- La resolución de sistemas de ecuaciones algebraicas que los algoritmos individuales no resuelven aisladamente
- Permitir la resolución eficiente de sistemas de ecuaciones algebraicas de gran porte, en sistemas computacionales de memoria distribuida, caracterizados por su bajo costo por instrucción.
- Obtener un considerable ahorro en tiempo de procesamiento, cuando problemas de gran porte son fraccionados en subproblemas menores que pueden ser resueltos en paralelo.
- Aprovechar la velocidad de algoritmos rápidos que no siempre convergen en el dominio de interés. Para esto, el TA combina un algoritmo rápido con otro más lento, pero capaz de asegurar la convergencia en el dominio considerado.
- Evitar la divergencia prematura de algunos bloques de la partición, a partir de una apropiada elección de pesos en un administrador de procesos paralelos.

- Acelerar la convergencia, así como hacer posible la resolución de sistemas de ecuaciones ante la presencia de bloques críticos fuertemente acoplados a otros, con la utilización de *overlapping* parcial, una técnica que será discutida en el capítulo 4.

De esta forma, el procesamiento distribuido trae consigo la promesa de su gran capacidad de abordar complejos problemas computacionales con un costo relativamente bajo, lo cual permite a países en vías de desarrollo incursionar en temas antes sólo reservados a aquéllos con capacidad de costearse supercomputadores.

1.5 Objetivos y Organización de esta Obra

1.5.1 Originalidad

El estudio del Problema del Flujo de Potencia Eléctrica en torno al Punto de Colapso es todavía un problema abierto, ya que no existe una solución suficientemente general al problema. Métodos como los utilizados en la actualidad [3,4] consiguen resolver algunos problemas bastante críticos en forma satisfactoria, pero pueden fallar al aplicarlos a sistemas eléctricos con diversas características de alinealidad. Esto lleva a buscar algún método capaz de resolver adecuadamente un universo mayor de problemas.

Una clara alternativa en este sentido, es la utilización de procesadores trabajando en un ambiente computacional distribuido, y la utilización de la técnica de combinación de algoritmos. Dentro de este contexto, los *Team Algorithm* formalizan la combinación de los mismos en un ambiente computacional distribuido y asíncrono.

La contribución que este trabajo pretende realizar, consiste no solo en implementar los algoritmos existentes para la búsqueda del punto de colapso, sino también que los mismos puedan trabajar en un ambiente computacional distribuido, de manera que los estudios de estabilidad de tensión que requieran la resolución de sistemas de ecuaciones de gran porte, puedan ser abordados con los medios computacionales locales.

Mas aún, el alcance de la presente obra, pionera en el país, es desarrollar el *Know How* necesario para aprovechar las prometedoras ventajas de la computación distribuida, especialmente si consideramos las facilidades económicas que los mismos presentan en lugar de los inaccesibles *Mainframes*, hasta hoy utilizados para encarar este tipo de problemas.

De esta forma, intentando formalizar los alcances de esta obra, se establecen los objetivos generales y específicos como sigue.

1.5.2 Objetivos generales

- Resolver el problema de flujo de potencia eléctrica en torno al punto de colapso, considerando las complejas alinealidades que presentan los modelos matemáticos correspondientes.
- Demostrar las ventajas de utilizar *Team Algorithm* en el problema considerado, en comparación con los métodos numéricos publicados hasta la fecha.
- Desarrollar el *Know How* necesario para aprovechar esta técnica de combinación de algoritmos en otros problemas matemáticos que presenten dificultades al intentar ser resueltos con los métodos numéricos utilizados comercialmente.

1.5.3 Objetivos específicos

- Generar una biblioteca de programas con los métodos utilizados para resolver el problema considerado.
- Analizar las características y propiedades de los métodos de solución del problema del Punto de Colapso.
- Comprobar experimentalmente las ventajas de cada método.

1.5.4 Metodología

- Estudio de los últimos trabajos publicados en el área (ver bibliografía).
- Creación de una biblioteca computacional de métodos ya publicados.
- Experimentar con los métodos de solución del problema del punto de colapso, utilizando de preferencia los sistemas de prueba de la IEEE.
- Individualización de los *Team Algorithm* que ofrezcan ventajas adicionales a los métodos computacionales de biblioteca.

1.5.5 Organización

El trabajo fue dividido en seis capítulos. En el capítulo 2 se repasan conceptos de flujo de potencia eléctrica. Los mismos servirán de base para el desarrollo de los siguientes capítulos.

El problema del *Punto de Colapso* se plantea matemáticamente en el capítulo 3. Asimismo, se exponen los métodos convencionales utilizados para su resolución y se presenta una breve comparación entre los mismos.

En el capítulo 4 se desarrollan las bases matemáticas y se exponen los detalles computacionales de dos nuevas propuestas, ambas basadas en el Método del Punto de Colapso presentado en [4]. La primera de ellas consiste en una variante aceleradora de dicho método. La segunda aplica los conceptos de procesamiento distribuido al método en cuestión.

Los resultados experimentales de las distintas implementaciones realizadas son presentados en el capítulo 5. A cada uno de los mismos acompaña una breve discusión, tanto para significar los resultados obtenidos, como para resaltar las virtudes y falencias encontradas en el desempeño de los diferentes métodos.

Finalmente, las conclusiones son presentadas en el capítulo 6, así como los temas que aún necesitan ser investigados.

2. EL FLUJO DE POTENCIA ELECTRICA

El análisis de sistemas operando en régimen de estado estable es ampliamente conocido en ingeniería eléctrica. La intención de este capítulo es repasar conceptos fundamentales que se utilizarán en capítulos posteriores.

A modo de introducción se desarrollarán los objetivos del *load flow*, así como las características principales que debe presentar un método de solución. Posteriormente se desarrollará el modelo matemático de la red y la definición analítica del problema.

Seguidamente, se presentarán métodos de solución del *load flow*, de entre los cuales se hará énfasis en el de Newton Raphson, debido a que los algoritmos de solución del *Punto de Colapso* que serán presentados en los capítulos 3 y 4, se basan en este método.

2.1 Introducción

Bajo condiciones normales de funcionamiento, los sistemas eléctricos de transmisión operan en estado estable, y el cálculo básico que se requiere para determinar las características de este estado es llamado *load flow* [2].

El objetivo del *load flow* es determinar las características de estado estable de operación del sistema de generación y transmisión de potencia para un dado conjunto de cargas en las barras. La potencia activa es normalmente especificada de acuerdo a las prácticas de despacho económico y la magnitud de tensión del generador es normalmente mantenida a un nivel especificado por los reguladores automáticos actuando en la excitatriz de la máquina. Las cargas son normalmente especificadas por sus constantes de potencia activa y reactiva requeridas, asumiéndose invariables para pequeños cambios de tensión y frecuencia durante la operación normal del estado estable.

La solución provee información de módulo y fase de las tensiones, flujo de potencia activa y reactiva en las líneas de transmisión, pérdidas y la potencia reactiva generada o absorbida en las barras controladoras de tensión.

En este capítulo, el problema del *load flow* es formulado analíticamente con la red representada por parámetros lineales y concentrados. Sin embargo, las restricciones de potencia y tensión hacen al problema no lineal. Por consiguiente, el método numérico de solución debe ser de naturaleza iterativa.

Los actuales problemas enfrentados en el desarrollo del *load flow* son: un sostenido aumento de tamaño de los sistemas a ser resueltos, procesamiento *on-line* necesario para controles automáticos, y sistemas de optimización.

Cinco principales propiedades son requeridas de un método de solución de *load flow*:

- Gran velocidad computacional. Esto es especialmente importante cuando se trata con sistemas grandes y procesamiento en tiempo real (*on-line*).
- Bajo consumo de memoria. Esto es importante para sistemas grandes y para el uso de computadores con pequeña capacidad de memoria.
- Confiabilidad de la solución.
- Versatilidad, es decir, capacidad para manejar características convencionales y especiales, y adaptabilidad para incorporar procesos complicados.
- Simplicidad, lo cual implica fácil codificación de los algoritmos en programas de computadoras.

Finalmente, es importante recalcar que no existe un método con el cual siempre pueda llegarse a una solución que reúna los requisitos antes establecidos. Por ejemplo, algunos métodos presentan gran velocidad computacional, pero son en cambio de convergencia sensible, estando ésta supeditada a las características de la red; mientras que otros presentan una fuerte convergencia pero son muy lentos.

2.2 Modelo Matemático de la Red

El primer paso en cualquier análisis de un sistema de energía eléctrica debe ser primero la formulación de un modelo adecuado para la red. Tal modelo debe relacionar las tensiones de la red con las corrientes o potencias de la misma.

Existe, de hecho, abundante bibliografía sobre formulación de redes. Véase en [31] dos formas de representar las mismas: ecuaciones de nudos y ecuaciones de lazos. Se utiliza en esta obra la primera de ellas.

En el método de los nudos [2], las variables son las tensiones complejas (módulo y fase) en los nudos y las corrientes. Para los módulos y las fases de las tensiones deben ser designadas referencias. Generalmente para los módulos la referencia es tierra; mientras que se elige como referencia para las fases, la de uno de los nudos o barras, asignándole el valor cero (usualmente).

Una corriente nodal es aquella que entra a la red por un determinado nudo, proveniente de una fuente y/o de una carga externa a la red. A partir de esta definición, la corriente que entra a la red (desde una fuente) se considera que es positiva, mientras que la que sale de la red (hacia una carga) tiene signo negativo, y la corriente nodal inyectada neta es la suma algebraica de las dos. Asimismo, se puede hablar de una potencia nodal inyectada $S_k = P_k + jQ_k$.

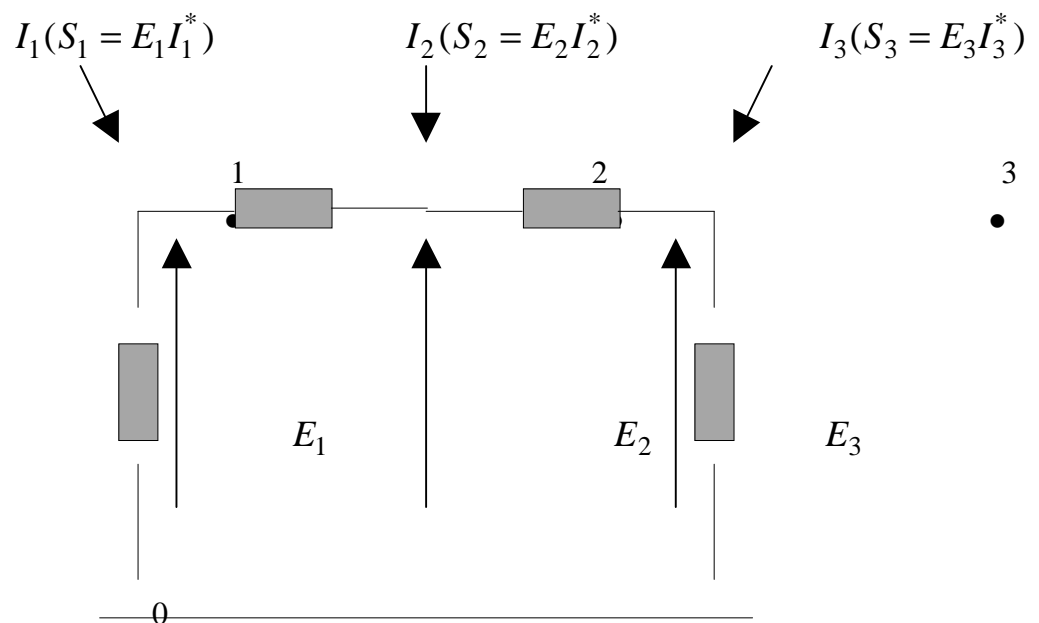


Figura 2.1 Red sencilla mostrando variables nodales

En la figura 2.1 se puede observar una red sencilla mostrando las corrientes nodales, tensiones y potencias.

En el método de los nudos, es conveniente usar admitancias de rama en lugar de impedancias. Designando a las tensiones de los nudos k y m como E_k y E_m respectivamente, y a las admitancias de la rama entre ellos como y_{km} , se tiene que la corriente que fluye en esta rama desde el nudo k hacia el nudo m , está dada por:

$$I_{km} = y_{km}(E_k - E_m) \quad (2.1)$$

Se enumeran $0, 1 \dots N$ los nudos de la red, donde 0 designa el nudo de referencia (tierra). Según la ley de corrientes de Kirchhoff, la corriente I_k inyectada en el nudo k , debe ser igual a la suma de las corrientes que salen del mismo, por consiguiente:

$$I_k = \sum_{m=0 \neq k}^N I_{km} = \sum_{m=0 \neq k}^N y_{km}(E_k - E_m) \quad (2.2)$$

Habiéndose elegido $E_0 = 0$, y si el sistema es lineal:

$$I_k = \sum_{m=0 \neq k}^N y_{km}E_k - \sum_{m=1 \neq k}^N y_{km}E_m \quad (2.3)$$

Si esta ecuación se escribe para todos los nudos, excepto para el de referencia, es decir, para todas las barras en el caso de un sistema de potencia, entonces el conjunto completo de ecuaciones que definen la red se puede expresar en forma matricial como:

$$\begin{bmatrix} I_1 \\ I_2 \\ \vdots \\ I_N \end{bmatrix} = \begin{bmatrix} Y_{11} & Y_{12} & \cdots & Y_{1N} \\ Y_{21} & Y_{22} & \cdots & Y_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ Y_{N1} & Y_{N2} & \cdots & Y_{NN} \end{bmatrix} \cdot \begin{bmatrix} E_1 \\ E_2 \\ \vdots \\ E_N \end{bmatrix} \quad (2.4)$$

donde

$$Y_{kk} = \sum_{m=0 \neq k}^N y_{km} = \text{admitancia propia del nudo } k$$

$$Y_{km} = -y_{km} = \text{admitancia mutua entre los nudos } k \text{ y } m$$

Podemos escribir la ecuación (2.4) en forma vectorial como:

$$\mathbf{I} = \mathbf{Y} \cdot \mathbf{E}, \quad \mathbf{I}, \mathbf{E} \in \mathbb{C}^N, \quad \mathbf{Y} \in \mathbb{C}^{N \times N} \quad (2.5)$$

o bien como una sumatoria:

$$I_k = \sum_{m=1}^N Y_{km} E_m, \quad \forall k \in \{1, \dots, N\} \quad (2.6)$$

La potencia que entra en una barra está dada entonces por:

$$S_k = P_k + jQ_k = E_k I_k^* = E_k \sum_{m=1}^N Y_{km}^* E_m^* \quad (2.7)$$

donde P_k y Q_k son las potencias activas y reactivas netas en las barras o potencias de barra, es decir, la diferencia entre las potencias generada y demandada en cada barra.

2.3 Definición Analítica del Problema

La completa definición del flujo de carga requiere el conocimiento de cuatro variables en cada barra k del sistema [2]:

P_k	potencia activa
Q_k	potencia reactiva
V_k	módulo de la tensión E_k
θ_k	fase de la tensión E_k

Solo son conocidas *a priori* dos variables en cada barra, y la finalidad del estudio del flujo de carga es resolver las ecuaciones para las dos variables restantes en cada barra.

Se define a continuación una clasificación de las barras en base al tipo de especificación de las mismas, es decir, cuales variables son conocidas de antemano:

1. Barra PV o de tensión controlada. La potencia activa total inyectada P_k es especificada, y la magnitud de la tensión es mantenida a un valor determinado mediante la inyección de potencia reactiva. Este tipo de barra generalmente corresponde a un generador, donde P_k es fijada por los reguladores de caudal turbinado, y V_k es fijada por los reguladores automáticos de tensión actuando en la excitación de la máquina; o bien a una barra donde la magnitud de la tensión es fijada por inyección de potencia reactiva mediante capacitores estáticos o compensadores rotativos síncronos, por ejemplo, una sub-estación.
2. Barra PQ o de tensión no controlada. La potencia total inyectada, $P_k + jQ_k$, es especificada en esta barra. En un sistema de potencia corresponde a un centro de consumo de carga, como una ciudad o una industria, donde la potencia demandada está determinada por los requerimientos de los consumidores. Tanto P_k como Q_k se suponen invariables con pequeñas alteraciones de la tensión en la barra.
3. Barra Slack o barra oscilante. Es necesaria para el estudio del flujo de cargas, a raíz de que las pérdidas en el sistema son desconocidas de antemano. Por consiguiente, la potencia inyectada no puede ser especificada en todas las barras. Es usual elegir una de las barras PV disponibles como *slack*, y considerar su potencia activa como desconocida. A la fase de la barra *slack* usualmente se la designa como la de referencia del sistema. En la práctica, corresponde a una estación generadora que tiene la responsabilidad de controlar la frecuencia de la red.

2.4 Métodos de Solución

La complejidad que presenta la obtención de una solución formal del problema de las cargas en una red de energía, radica en las diferencias en el tipo de datos especificados para las distintas barras del sistema. Aunque no es difícil la formulación del número de ecuaciones suficiente, no es práctico obtener una solución analítica. La resolución de los problemas de carga por el método numérico sigue un proceso iterativo, asignando valores estimados en las barras a las tensiones desconocidas, y calculando las tensiones en cada una de las barras, a partir de los valores estimados en las otras y de las potencias real y reactiva especificadas.

De esta manera, se obtiene un nuevo conjunto de tensiones en las barras, que se emplea para calcular otro, y así sucesivamente; cada cálculo de un nuevo conjunto de tensiones se llama *iteración*. El proceso iterativo se repite hasta que los cambios en cada barra sean menores que un valor mínimo especificado [31].

Existen varios métodos iterativos como ser:

- Métodos de Gauss y Gauss-Seidel
- Método de Newton-Raphson
- Métodos desacoplados derivados del Newton-Raphson

2.4.1 Métodos de Gauss y Gauss-Seidel

Para explicar los algoritmos iterativos de Gauss y Gauss-Seidel [31] se emplearán dos ecuaciones deducidas a continuación.

Haciendo algunas transformaciones en (2.7) se tiene:

$$\frac{P_k - jQ_k}{E_k^*} = \sum_{m=1, m \neq k}^N Y_{km} E_m + Y_{kk} E_k \quad (2.8)$$

de donde

$$E_k = \frac{1}{Y_{kk}} \left(\frac{P_k - jQ_k}{E_k^*} - \sum_{m=1 \neq k}^N Y_{km} E_m \right) \quad (2.9)$$

La ecuación (2.9) expresa la tensión de la barra k como función de las potencias activa y reactiva entregadas a la barra por los generadores o suministrada a la carga conectada a la barra, las tensiones estimadas o previamente calculadas en todas las barras y las admitancias propia y mutua de los nudos.

Por otra parte, de (2.7) también puede obtenerse:

$$P_k - jQ_k = E_k^* \sum_{m=1}^N Y_{km} E_m \quad (2.10)$$

de donde

$$Q_k = -\text{Im} \left\{ E_k^* \sum_{m=1}^N Y_{km} E_m \right\} \quad (2.11)$$

en la que el símbolo Im significa “parte imaginaria de”.

2.4.1.1 Algoritmo Iterativo de Gauss

De acuerdo a la ecuación (2.9), el algoritmo iterativo de Gauss que actualiza la tensión en una barra PQ es:

$$E_k(i+1) = \frac{1}{Y_{kk}} \left(\frac{P_k - jQ_k}{E_k^*(i)} - \sum_{m=1 \neq k}^N Y_{km} E_m(i) \right) \quad (2.12)$$

En efecto, para las barras PQ la ecuación (2.12) da un valor de E_k actualizado, utilizando para ello los valores de P_k y Q_k establecidos como dato, y los valores de tensión inicialmente estimados o previamente calculados en la iteración anterior (iteración i).

Como (2.12) se aplica solamente en las barras en las que la potencia activa y reactiva están especificadas, es preciso un paso adicional en las barras en que el valor de la tensión ha de permanecer constante.

Entonces, para actualizar la tensión en una barra PV, se calcula en primer lugar un valor para la potencia reactiva. Con base en la ecuación (2.11), la forma de calcular el valor de $Q_k(i+1)$ en el esquema de Gauss es:

$$Q_k(i+1) = -j \operatorname{Im} \left\{ E_k^*(i) \sum_{m=1}^N Y_{km} E_m(i) \right\} \quad (2.13)$$

Este valor se utiliza para el cálculo indicado en (2.14):

$$E'_k(i+1) = \frac{1}{Y_{kk}} \left(\frac{P_k - jQ_k(i+1)}{E_k^*(i)} - \sum_{m=1, m \neq k}^N Y_{km} E_m(i) \right) \quad (2.14)$$

Este valor $E'_k(i+1)$ debe ser corregido ahora para que coincida con el módulo V_k especificado. Esto es:

$$E_k(i+1) = V_k \frac{E'_k(i+1)}{\|E'_k(i+1)\|} \quad (2.15)$$

El proceso se repite sucesivamente en todas las barras (excepto en la oscilante) a lo largo de la red para completar la primera iteración. Después se vuelve a realizar todo el proceso, una y otra vez, hasta que el valor de la corrección de la tensión en cada barra sea menor que un índice de precisión predeterminado.

En el algoritmo iterativo de Gauss se utiliza el mismo conjunto de valores de tensión para cada iteración.

2.4.1.2 Algoritmo Iterativo de Gauss-Seidel

De acuerdo a la ecuación (2.9), el algoritmo iterativo de Gauss-Seidel que actualiza la tensión en una barra PQ es:

$$E_k(i+1) = \frac{1}{Y_{kk}} \left(\frac{P_k - jQ_k}{E_k^*(i)} - \sum_{m=1}^{k-1} Y_{km} E_m(i+1) - \sum_{m=k+1}^N Y_{km} E_m(i) \right) \quad (2.16)$$

En la ecuación (2.16), los valores de las tensiones en el segundo miembro de la ecuación, son los mejores valores previos para las barras correspondientes.

De una manera análoga a la empleada en el método de Gauss, para las barras PV se requiere un paso previo adicional. La forma de calcular el valor de $Q_k(i+1)$ en el esquema de Gauss-Seidel es:

$$Q_k(i+1) = -j \sum_{m=1}^{k-1} Y_{km} E_m(i+1) - j \sum_{m=k+1}^N Y_{km} E_m(i) \quad (2.17)$$

Este valor se utiliza para el cálculo indicado en (2.18):

$$E_k'(i+1) = \dots = \frac{1}{Y_{kk}} \left(\frac{P_k - jQ_k(i+1)}{E_k^*(i)} - \sum_{m=1}^{k-1} Y_{km} E_m(i+1) - \sum_{m=k+1}^N Y_{km} E_m(i) \right) \quad (2.18)$$

Este valor $E_k'(i+1)$ debe ser corregido ahora para que coincida con el módulo V_k especificado. Esto es:

$$E_k(i+1) = V_k \frac{E_k'(i+1)}{\|E_k'(i+1)\|} \quad (2.19)$$

De forma análoga a la del algoritmo de Gauss, las iteraciones se suceden hasta que el valor de la corrección de la tensión en cada barra sea menor que el índice de precisión predeterminado.

Es posible el desembocamiento en una solución errónea si las tensiones de partida son muy diferentes de los valores correctos. Este desembocamiento puede evitarse si las tensiones de partida tienen valores razonables.

La experiencia con estos métodos de resolución de los problemas de distribución de energía, ha demostrado que se necesita un número excesivo de iteraciones antes de que la tensión corregida esté dentro de un índice aceptable de precisión. Por consiguiente, no son muy utilizados en los actuales sistemas comerciales.

2.4.2 Método de Newton-Raphson

El método de Newton-Raphson es un algoritmo iterativo utilizado para resolver sistemas de n ecuaciones no lineales con igual número de variable desconocidas [2].

$$F(\mathbf{x}) = \begin{bmatrix} f_1(\mathbf{x}) \\ \vdots \\ f_n(\mathbf{x}) \end{bmatrix} = \mathbf{0}, \quad F: \mathbb{R}^n \rightarrow \mathbb{R}^n, \quad \mathbf{x}, \mathbf{0} \in \mathbb{R}^n \quad (2.20)$$

En el problema no lineal, el Newton-Raphson linealiza en cada iteración las ecuaciones dadas, de forma a aproximarse progresivamente a la solución. La linealización puede ser visualizada mejor en el caso de un problema de una sola variable.

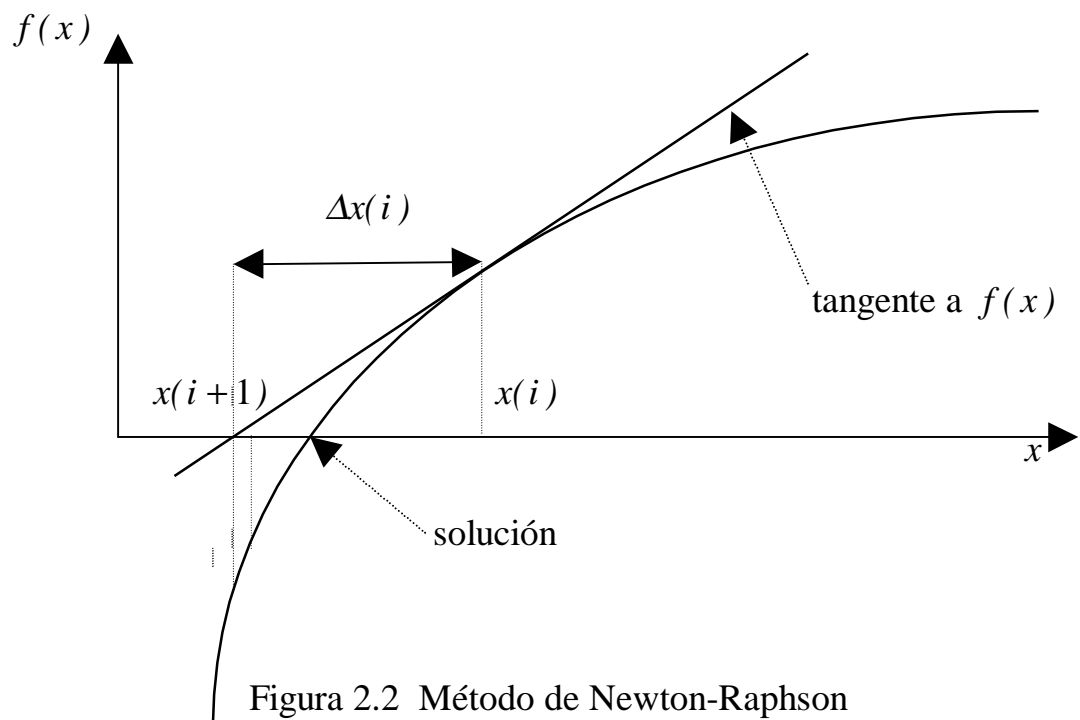


Figura 2.2 Método de Newton-Raphson

En la figura 2.2 se visualiza una función:

$$f(x) = 0, \quad f: \mathbb{R} \rightarrow \mathbb{R}, \quad x \in \mathbb{R} \quad (2.21)$$

En la figura 2.2, $x(i)$ es una aproximación a la solución con error $\Delta x(i)$ en la iteración i . Entonces,

$$f(x(i) + \Delta x(i)) = 0 \quad (2.22)$$

$f(x(i) + \Delta x(i))$ puede ser expandida de acuerdo a la serie de Taylor como:

$$\begin{aligned} f(x(i) + \Delta x(i)) = \\ f(x(i)) + \Delta x(i) \frac{df(x(i))}{dx} + \Delta x(i) \frac{d^2 f(x(i))}{dx^2} + \dots \end{aligned} \quad (2.23)$$

Si la estimación inicial de la variable $x(i)$ es cercana al valor de la solución, el valor de $\Delta x(i)$ será relativamente pequeño y todos los términos de orden superior podrán ser despreciados. Por tanto,

$$f(x(i)) + \Delta x(i) \frac{df(x(i))}{dx} = 0 \quad (2.24)$$

o escrito de otra forma,

$$\Delta x(i) = - \frac{f(x(i))}{\frac{df(x(i))}{dx}} \quad (2.25)$$

El nuevo valor de la variable es entonces obtenido de

$$x(i+1) = x(i) + \Delta x(i) \quad (2.26)$$

La ecuación (2.24) puede ser reescrita como:

$$f(x(i)) = -\mathbf{J}(i)\Delta x(i) \quad (2.27)$$

donde $\mathbf{J}(i) = \frac{df(x(i))}{dx}$

Al extender el método para un conjunto de n ecuaciones con n incógnitas como el indicado en (2.20), \mathbf{J} llega a ser una matriz cuadrada cuyos elementos son las derivadas parciales de primer orden de la función $F(\mathbf{x})$ con respecto a cada uno de los elementos del vector \mathbf{x} . La matriz \mathbf{J} es llamada jacobiano de la función F , definida entonces por:

$$\mathbf{J} = F_{\mathbf{x}} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \cdots & \frac{\partial f_n}{\partial x_n} \end{bmatrix} \quad (2.28)$$

El NR convergerá cuadráticamente si la función tiene primeras derivadas continuas en la vecindad de la solución, la matriz jacobiana no es singular, y la aproximación inicial de \mathbf{x} está cerca de la solución. Sin embargo, el método es sensible al comportamiento de la función. Cuanto más lineal sea la función, la convergencia del método será más rápida y confiable. La existencia de algún pico puede causar retardo, no convergencia o desembocar en una solución que no sea de interés operativo.

2.4.2.1 Ecuaciones del Flujo de Carga en un Sistema de Potencia

Las ecuaciones del flujo de carga de un sistema de potencia están expresadas por (2.7).

Matemáticamente hablando, las ecuaciones complejas del *load flow* no son analíticas, y no pueden ser diferenciadas en forma compleja. De forma a aplicar el método de Newton-Raphson, cada una de las ecuaciones complejas debe ser convertida en dos ecuaciones reales; esto se consigue usando

coordenadas polares o rectangulares en la representación de las tensiones en las barras.

Desarrollando la ecuación (2.7) en coordenadas polares y separando las partes real e imaginaria tendremos:

$$P_k = \sum_{m=1}^N V_k V_m (G_{km} \cos \theta_{km} + B_{km} \sen \theta_{km}) \quad (2.29)$$

$$Q_k = \sum_{m=1}^N V_k V_m (G_{km} \sen \theta_{km} - B_{km} \cos \theta_{km}) \quad (2.30)$$

donde

$$\theta_{km} = \theta_k - \theta_m$$

$$Y_{km} = G_{km} + jB_{km}$$

Relaciones lineales son obtenidas para pequeñas variaciones de las variables θ y V formando el diferencial total de las expresiones (2.29) y (2.30).

Las ecuaciones para una barra PQ son:

$$\Delta P_k = \sum_{m=1}^N \frac{\partial P_k}{\partial \theta_m} \Delta \theta_m + \sum_{m=1}^N \frac{\partial P_k}{\partial V_m} \Delta V_m \quad (2.31)$$

y

$$\Delta Q_k = \sum_{m=1}^N \frac{\partial Q_k}{\partial \theta_m} \Delta \theta_m + \sum_{m=1}^N \frac{\partial Q_k}{\partial V_m} \Delta V_m \quad (2.32)$$

Para una barra PV solamente la ecuación (2.31) es usada, ya que Q_k no es especificada.

Para la barra *slack* no hay ecuaciones.

La magnitud de tensión de las ecuaciones (2.31) y (2.32) para las barras PV y *slack* no son variables, son datos del problema, así como la fase en la barra *slack*.

En resumen, el conjunto completo de ecuaciones (2.31) y (2.32) comprende dos ecuaciones por cada barra PQ y una por cada barra PV. Las incógnitas son V y θ para cada barra PQ y θ para cada barra PV. En consecuencia, el sistema de ecuaciones formado tiene igual cantidad de ecuaciones que incógnitas.

Denominando N_1 al número de barras PQ, y N_2 al número de barras PV, es posible definir el vector de estado del sistema de potencia como:

$$\mathbf{x} = \begin{bmatrix} \theta \\ \mathbf{V} \end{bmatrix} = \begin{bmatrix} \theta_1 \\ \vdots \\ \theta_{N_1+N_2} \\ V_1 \\ \vdots \\ V_{N_1} \end{bmatrix} \quad (2.33)$$

donde la dimensión del vector \mathbf{x} es $n=2N_1+N_2$

El algoritmo (2.27), aplicado al sistema n -dimensional formado con las ecuaciones (2.31) y (2.32), queda establecido entonces como:

$$\begin{bmatrix} \Delta \mathbf{P}(i) \\ \Delta \mathbf{Q}(i) \end{bmatrix} = \begin{bmatrix} \mathbf{H}(i) & \mathbf{N}(i) \\ \mathbf{R}(i) & \mathbf{M}(i) \end{bmatrix} \cdot \begin{bmatrix} \Delta \theta(i) \\ \frac{\Delta \mathbf{V}}{\mathbf{V}}(i) \end{bmatrix} \quad (2.34)$$

donde

- $\Delta \mathbf{P}$ vector de errores de P para todas las barras PQ y PV
- $\Delta \mathbf{Q}$ vector de errores de Q para todas las barras PQ
- $\Delta \theta$ vector de correcciones de θ para todas las barras PQ y PV
- $\Delta \mathbf{V}$ vector de correcciones de V para todas las barras PQ
- $\frac{\Delta \mathbf{V}}{\mathbf{V}}$ vector cuyas componentes provienen de la división de cada $\Delta V_k(i)$ por su correspondiente $V_k(i)$

La división de cada $\Delta V_k (i)$ por su correspondiente $V_k (i)$ sirve para simplificar algunos de los términos del jacobiano sin afectar numéricamente al algoritmo.

Los elementos de las submatrices \mathbf{H} , \mathbf{N} , \mathbf{R} , \mathbf{M} del jacobiano, para las barras k y m son:

para $m \neq k$

$$H_{km} = \frac{\partial P_k}{\partial \theta_m} = V_k V_m (G_{km} \text{sen } \theta_{km} - B_{km} \text{cos } \theta_{km}) \quad (2.35)$$

$$N_{km} = V_m \frac{\partial P_k}{\partial V_m} = V_k V_m (G_{km} \text{cos } \theta_{km} + B_{km} \text{sen } \theta_{km}) \quad (2.36)$$

$$R_{km} = \frac{\partial Q_k}{\partial \theta_m} = -V_k V_m (G_{km} \text{cos } \theta_{km} + B_{km} \text{sen } \theta_{km}) \quad (2.37)$$

$$M_{km} = V_m \frac{\partial Q_k}{\partial V_m} = V_k V_m (G_{km} \text{sen } \theta_{km} - B_{km} \text{cos } \theta_{km}) \quad (2.38)$$

y para $m = k$

$$H_{kk} = \frac{\partial P_k}{\partial \theta_k} = -Q_k - B_{kk} V_k^2 \quad (2.39)$$

$$N_{kk} = V_k \frac{\partial P_k}{\partial V_k} = P_k + G_{kk} V_k^2 \quad (2.40)$$

$$R_{kk} = \frac{\partial Q_k}{\partial \theta_k} = P_k - G_{kk} V_k^2 \quad (2.41)$$

$$M_{kk} = V_k \frac{\partial Q_k}{\partial V_k} = Q_k - B_{kk} V_k^2 \quad (2.42)$$

2.4.2.2 Resumen del Método (pasos a seguir)

El procedimiento que se sigue de acuerdo a [31] es:

1. Se estiman los valores iniciales de las incógnitas (condición inicial).

2. Se calcula P_k para todas las barras PQ y PV, de la ecuación (2.29); y Q_k para todas las barras PQ, de la ecuación (2.30).
3. Se calcula ΔP_k , igual a P_k especificado menos P_k calculado, para todas las barras PQ y PV; y ΔQ_k , igual a Q_k especificado menos Q_k calculado, para todas las barras PQ. Si todos los valores son menores que el índice de precisión predeterminado, paran las iteraciones y se calculan todas las potencias, flujos en las líneas y otros resultados deseados.
4. Si no se ha alcanzado la precisión deseada, se evalúan los elementos del jacobiano, utilizando las ecuaciones (2.35) a (2.42), con las tensiones estimadas para la primera iteración u obtenidas en el paso 6 para las iteraciones siguientes.
5. Se resuelve la ecuación (2.34) para $\Delta \theta(i)$ y $\Delta V(i)$.
6. Se determinan las nuevas tensiones de las barras agregando los incrementos de tensión (módulo y fase) a los valores anteriores.
7. Se regresa al paso 2.

2.4.2.3 Técnicas que hacen al Newton-Raphson Competitivo

La resolución eficiente de (2.34) en cada iteración es crucial para el éxito del NR. Si se usan técnicas matriciales convencionales, los requerimientos de memoria ($\propto n^2$) y tiempo de procesamiento ($\propto n^3$) se volverían prohibitivos para sistemas de gran porte [2].

Las técnicas que hacen al NR competitivo respecto a los otros métodos de *load flow* envuelven la solución de (2.34) mediante técnicas de esparsidad y una ordenada factorización triangular [2], aprovechando que para la mayoría de los problemas la matriz de admitancias es bastante esparsa, y el jacobiano, en consecuencia, también lo es.

Dichas técnicas permiten un ahorro considerable en los requerimientos de memoria y tiempo de procesamiento.

2.4.2.4 Características del Newton-Raphson

Con las técnicas de esparsidad y de factorización triangular, el NR se ha vuelto más rápido que otros métodos para sistemas de gran porte [2]. El número de iteraciones es virtualmente independiente del tamaño del sistema, debido a la convergencia cuadrática del método. Muchos sistemas son resueltos en 2 a 5 iteraciones sin necesidad de factores de aceleración. Como comparación, el número de iteraciones en el método de Gauss-Seidel aumenta casi directamente con el número de barras.

Con buenos programas el tiempo por iteración crece en forma aproximadamente lineal con el número de barras N , así el tiempo total varía con N . Una iteración del Newton-Raphson es equivalente a aproximadamente 7 iteraciones de Gauss-Seidel. Un sistema de 500 barras, el convencional Gauss-Seidel lo resuelve en más o menos 500 iteraciones, siendo la ventaja de velocidad del NR de aproximadamente 15:1, lo que lo hace especialmente atractivo para la resolución de sistemas de gran porte.

Debido a la convergencia cuadrática del método, una gran precisión es obtenida en pocas iteraciones. Esto es importante para el uso de *load flow* en el estudio de corto circuitos y análisis de estabilidad. El método es fácil de extender para incluir transformadores reguladores con *taps* ajustables y variables restringidas. Además, modificaciones en la red son sencillas de realizar.

Los coeficientes de la matriz del jacobiano no son constantes, ellos son funciones de V_k y θ_k , en consecuencia, varían en cada iteración. Sin embargo, luego de unas pocas iteraciones, a medida que los valores de las fases y módulos de las tensiones tienden a sus valores finales, los coeficientes mencionados tienden a permanecer constantes.

Una modificación del NR consiste en calcular el jacobiano para las primeras dos o tres iteraciones solamente, y usar el valor de la última iteración para los siguientes cálculos. Alternativamente, el jacobiano puede ser actualizado cada dos o más iteraciones. Ninguna de estas modificaciones afecta mucho a la convergencia del algoritmo; por el contrario, ayudan a ganar tiempo aunque no ahorran memoria.

2.4.3 Métodos Desacoplados Derivados del Newton-Raphson

Una característica inherente a cualquier sistema eléctrico práctico de transmisión de potencia, operando en estado estable, es la fuerte interdependencia entre potencias activas y fases de tensión, y entre potencias reactivas y módulos de tensión. Correspondientemente, los acoples ‘ $P-V$ ’ y ‘ $Q-\theta$ ’ son relativamente débiles. Existen varios algoritmos propuestos para aprovechar este principio de desacople [2].

Se describe brevemente a continuación el método desacoplado de mayor suceso, el cual está basado en el jacobiano del Newton-Raphson. En la ecuación (2.34), y de acuerdo al principio citado anteriormente, las submatrices $N(i)$ y $R(i)$ pueden ser despreciadas, ya que estas representan el débil acoplamiento entre ‘ $P-V$ ’ y ‘ $Q-\theta$ ’. Luego, las ecuaciones desacopladas resultan:

$$\Delta P(i) = H(i) \cdot \Delta \theta(i) \quad (2.43)$$

$$\Delta Q(i) = M(i) \cdot \frac{\Delta V}{V}(i) \quad (2.44)$$

El método desacoplado se puede comparar favorablemente con el NR convencional. La demanda de memoria es menor que en el NR en un 30 a 40 por ciento. El tiempo empleado por iteración es también menor. Sin embargo, la convergencia del Newton desacoplado es lineal y no cuadrática como en el NR. Por tal motivo, para índices elevados de precisión, más iteraciones son requeridas. Esto es compensado, para resultados de interés práctico con índices menores de precisión, por la rápida convergencia inicial del método. Típicamente, módulos de tensión convergen dentro del 0.3 por ciento de la solución final en la primera iteración. Ángulos de fase convergen más lentamente, pero la solución completa se alcanza en 2 a 5 iteraciones [2].

Simplificaciones adicionales, basadas en las propiedades físicas de los sistemas prácticos, pueden ser efectuadas. Así, los jacobianos de métodos desacoplados pueden suponerse constantes. Los métodos que adoptan este criterio se denominan desacoplados rápidos y son de gran aplicación en la mayoría de los programas comerciales.

3. CALCULO DEL PUNTO DE COLAPSO

En la introducción de este capítulo, se desarrollará desde el punto de vista matemático el concepto del *punto de colapso*, ampliando la breve descripción dada en el capítulo 1.

Seguidamente, se presentarán dos métodos hoy utilizados para el cálculo del *punto de colapso* correspondiente a una dirección de crecimiento de carga especificada, y una comparación entre estos métodos.

Por último, se expondrán dos métodos que permiten calcular el *punto de colapso más cercano* a un punto de trabajo dado.

3.1 Introducción

La planificación de la expansión de un sistema de potencia arranca necesariamente del conocimiento de la dirección de crecimiento en el tiempo de las demandas activas y reactivas, y de las disponibilidades de generación. Para un estado de cargas y generación, se resuelve el modelo del sistema para el caso base y para las contingencias simples, buscando la respuesta de régimen permanente, es decir, los niveles de tensión en las distintas barras.

El siguiente paso en el análisis es identificar las distintas técnicas alternativas que permitan restituir las condiciones anormales a los límites aceptables. De entre estas alternativas, se debe seleccionar aquella que resulte económicamente apropiada [32].

El sistema de ecuaciones que representa el funcionamiento en régimen permanente de un sistema de potencia tiene la forma [28]:

$$F(\mathbf{x}, \Pi) = \mathbf{0} \quad (3.1)$$

donde

$$\mathbf{x} \in \mathbb{R}^n, \quad \underline{\Pi} \in \mathbb{R}^u, \quad F : \mathbb{R}^n \times \mathbb{R}^u \rightarrow \mathbb{R}^n$$

Es conveniente recordar que \mathbf{x} es el vector de estado del sistema de potencia, y que su dimensión es $n=2N_1+N_2$, donde N_1 y N_2 son el número de barras PQ y PV respectivamente. Por otra parte, $\underline{\Pi}$ es un vector parámetro de carga de las potencias activas y reactivas.

A la carga $\underline{\Pi}_0$ en condiciones normales de trabajo, pueden estar asociados varios estados de operación estable [33], de los cuales sólo \mathbf{x}_0 es de interés operativo.

Conforme el vector $\underline{\Pi}$ varía desde la carga de trabajo $\underline{\Pi}_0$, su correspondiente estado estable \mathbf{x} también varía. Cuando el vector $\underline{\Pi}$ alcanza el punto crítico $\underline{\Pi}_1$ en el espacio \mathbb{R}^u (figura 3.1), la solución de régimen permanente alcanza una solución única \mathbf{x}_1 en el espacio \mathbb{R}^n [26,27], denominándose a $(\mathbf{x}_1, \underline{\Pi}_1)$ *punto de colapso* [28]. Este punto está caracterizado por el jacobiano $F_{\mathbf{x}}(\mathbf{x}_1, \underline{\Pi}_1)$ singular, teniendo un único autovalor cero con autovectores izquierdo y derecho distintos de cero [4].

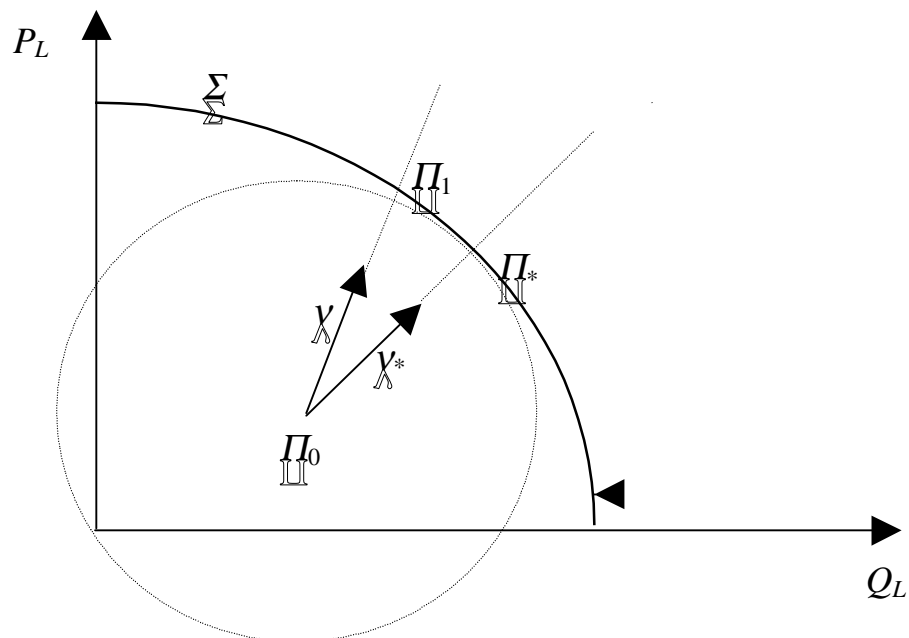


Figura 3.1 Lugar geométrico de los Puntos de Colapso

El punto crítico Π_1 pertenece a una Hipersuperficie Σ en el espacio \mathbb{R}^u , ésta marca el límite de la región en la que es factible la operación del sistema con estabilidad de tensión en régimen permanente, es decir, es el lugar geométrico de los puntos de colapso.

En la figura 3.1 se ilustra el lugar geométrico de los puntos de colapso, correspondiente a un sistema eléctrico de dos barras, con un generador *slack*, una línea de transmisión y una barra PQ con carga variable $P_L + jQ_L$, como se presenta en la figura 3.2. En este caso, tanto el vector de estado del sistema $x = [\theta \ V]^T$, como el vector parámetro de carga $\Pi = [P_L \ Q_L]^T$, son de dimensión 2.

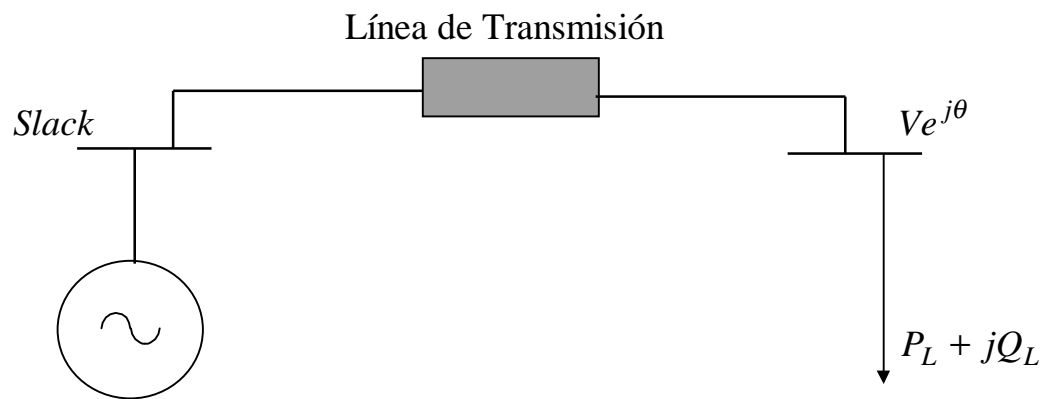


Figura 3.2 Sistema eléctrico de dos barras

A través de una apropiada estimación del crecimiento de las demandas en el tiempo, es posible especificar una dirección según la cual el vector Π se incrementará desde el punto de trabajo Π_0 hasta un punto Π_1 de la superficie Σ (figura 3.1), en el cual colapsaría el sistema si ninguna medida correctiva fuese adoptada.

Si especificamos una dirección particular de incremento de la carga desde el punto de trabajo Π_0 , dada por un vector unitario γ en el espacio \mathbb{R}^u , entonces la evolución de la carga podrá ser representada por

$$\Pi = \Pi_0 + \gamma\lambda \quad (3.2)$$

donde λ es un escalar no negativo llamado *parámetro de carga*.

La ecuación (3.2) representa una Hiper-recta paramétrica en y , en el espacio R^n , siendo el segmento comprendido entre \mathbb{M}^0 y \mathbb{M}^1 , el lugar geométrico, en la dirección del versor $\hat{\mathbf{a}}$, de las cargas que presentan solución de estado estable.

Para un punto \mathbb{M}^1 de la Hipersuperficie Σ se cumple que:

$$\mathbf{r}^1 = \mathbf{r}^0 + \lambda \hat{\mathbf{a}} \quad (3.3)$$

donde λ es entonces un *índice de estabilidad* que mide la proximidad al *punto de colapso* en la dirección $\hat{\mathbf{a}}$ [28].

La figura 3.1 muestra que desde la distribución inicial de carga \mathbb{M}^0 , correspondiente al estado \mathbf{x}_0 , existen diferentes direcciones de crecimiento de carga, y no siempre la dirección $\hat{\mathbf{a}}$ utilizada en la planificación es la más crítica en el cálculo del índice de estabilidad. En la figura, la dirección dada por $\hat{\mathbf{a}}^*$ es la más peligrosa, y a su intersección con Σ se la denomina *punto de colapso más cercano* [28].

En las siguientes secciones se describirán métodos para encontrar el punto de colapso correspondiente a una dirección de crecimiento de cargas especificada, así como también algoritmos que son capaces de localizar el punto de colapso más cercano a un dado punto de trabajo. En ambos casos, las técnicas presentadas salvan las dificultades encontradas en las proximidades del punto de colapso, derivadas de la singularidad del jacobiano, y a raíz de las cuales fallan los programas tradicionales de flujo de carga.

3.2 Métodos para encontrar el Punto de Colapso

Las técnicas más utilizadas para localizar el punto de colapso correspondiente a una dirección de crecimientos dada, salvando la singularidad del jacobiano, son [4]:

- Método de Continuación (MC)
- Método del Punto de Colapso (PoC)

3.2.1 Método de Continuación

El MC permite evitar la singularidad del jacobiano a partir de una ligera reformulación de las ecuaciones de *load flow* [3], aplicando una técnica de continuación basada en una parametrización local.

Para un determinado esquema de crecimiento de cargas, el método encuentra un conjunto de soluciones de *load flow*, a partir de una distribución base de las cargas hasta el punto crítico. Además, el método provee información adicional, como por ejemplo, acerca de la estabilidad de tensión de un sistema y las áreas más propensas al colapso de tensión. También pueden ser determinados diferentes índices de estabilidad de tensión basados en los resultados del algoritmo [1,3].

El principio general del método de continuación es simple. Emplea un esquema de predicción y corrección para encontrar la curva solución del conjunto de las ecuaciones del *load flow*, las cuales previamente han sido reformuladas para incluir el parámetro de carga. Como se muestra en la figura 3.3, el algoritmo parte de una solución conocida y usa una tangente predictora para estimar la siguiente solución, correspondiente a un valor diferente del parámetro de carga. Esta estimación es entonces corregida usando la misma técnica de Newton-Raphson empleada por un *load flow* convencional.

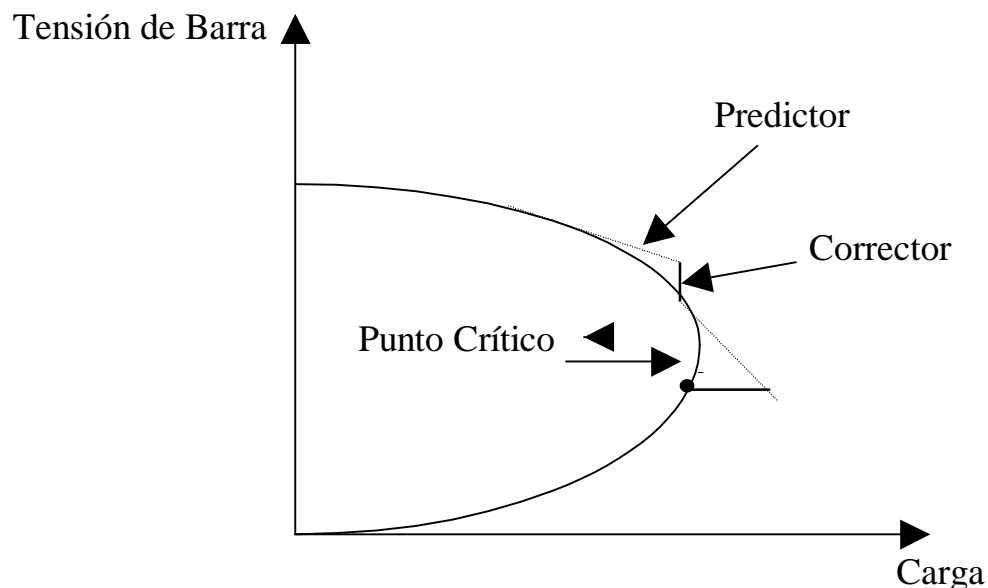


Figura 3.3 Método de Continuación

La parametrización mencionada anteriormente proporciona un medio de identificar cada punto a lo largo de la curva solución y juega un papel importante para evitar la singularidad del jacobiano.

3.2.1.1 Reformulación de las Ecuaciones del *Load Flow*

El *Load Flow* convencional consiste en resolver la ecuación (3.1), esto es:

$$F(\mathbf{x}, \Pi) = \mathbf{0}$$

para un determinado Π . Es decir, dada una distribución de cargas Π , consiste en hallar el estado \mathbf{x} del sistema.

Para aplicar una técnica de continuación parametrizada localmente al problema de flujo de potencia, un parámetro de carga debe ser insertado en las ecuaciones. Introduciendo (3.2) en (3.1), y considerando que Π_0 y λ son datos del problema se tiene:

$$F(\mathbf{x}, \lambda) = \mathbf{0}, \quad F: \mathbb{R}^{n+1} \rightarrow \mathbb{R}^n \quad (3.4)$$

que constituye la ecuación del *Load Flow* convencional reformulada.

Para relacionar estos conceptos con los expuestos en el capítulo 2, esta misma reformulación se desarrollará a continuación con mayor detalle. Así, la ecuación (3.1) puede ser escrita como:

$$P_{Gk} - P_{Lk} - P_k = 0 \quad (3.5)$$

$$Q_{Gk} - Q_{Lk} - Q_k = 0 \quad (3.6)$$

donde,

$P_{Gk} + jQ_{Gk}$ potencia generada, inyectada a la barra k

$P_{Lk} + jQ_{Lk}$ potencia de carga extraída de la barra k

$P_k + jQ_k$ potencia neta inyectada a la barra k

Se recordará que la potencia neta inyectada está determinada por las ecuaciones (2.29) y (2.30), las cuales, para comodidad del lector, son reproducidas a continuación:

$$P_k = \sum_{m=1}^N V_k V_m (G_{km} \cos \theta_{km} + B_{km} \sin \theta_{km})$$

$$Q_k = \sum_{m=1}^N V_k V_m (G_{km} \sin \theta_{km} - B_{km} \cos \theta_{km})$$

Por otro lado, para simular el incremento de potencias, P_{Gk} , P_{Lk} y Q_{Lk} se expresan en función del parámetro λ . Esto puede hacerse desarrollando cada término en dos componentes. Uno de ellos corresponde a la potencia inicial para cada barra k y la otra componente representará el incremento al variar el parámetro λ .

Así, las potencias generadas y de carga serán:

$$P_{Gk} = P_{Gk_0} + \mu_{Gk} \lambda \quad (3.7)$$

$$Q_{Gk} = Q_{Gk_0} \quad (3.8)$$

$$P_{Lk} = P_{Lk_0} + (\mu_{Lk} \cos \psi_k) \lambda \quad (3.9)$$

$$Q_{Lk} = Q_{Lk_0} + (\mu_{Lk} \sin \psi_k) \lambda \quad (3.10)$$

donde

$P_{Gk_0} + jQ_{Gk_0}$ potencia generada inicialmente en la barra k

$P_{Lk_0} + jQ_{Lk_0}$ potencia demandada inicialmente en la barra k

y

- P_{Gk} cte. de crecimiento de potencia activa generada en la barra k
- P_{Lk} cte. de crecimiento de carga en la barra k .
- θ^k ángulo de factor de potencia del crecimiento de carga en k

Restando miembro a miembro (3.7) de (3.9) y (3.8) de (3.10) se obtiene:

$$P_{Lk} - P_{Gk} = (P_{Lk_0} - P_{Gk_0}) + (P_{Lk} \cos \theta^k - P_{Gk})_{\text{vr}} \quad (3.11)$$

$$Q_{Lk} - Q_{Gk} = (Q_{Lk_0} - Q_{Gk_0}) + (Q_{Lk} \sin \theta^k)_{\text{vr}} \quad (3.12)$$

donde las ecuaciones (3.11) y (3.12) escritas en forma vectorial representan (3.2), esto es:

$$\mathbf{r}_k = \mathbf{r}_{k0} + \mathbf{a}_k$$

siempre que se elija una escala adecuada para \mathbf{y} , de manera tal que \mathbf{a}_k sea un versor.

Introduciendo (3.11) y (3.12) en (3.5) y (3.6) se obtienen las ecuaciones reformuladas del *load flow*, que son:

$$(P_{Lk_0} - P_{Gk_0}) + (P_{Lk} \cos \theta^k - P_{Gk})_{\text{vr}} + P_k = 0 \quad (3.13)$$

$$(Q_{Lk_0} - Q_{Gk_0}) + (Q_{Lk} \sin \theta^k)_{\text{vr}} + Q_k = 0 \quad (3.14)$$

las cuales representan (3.4), esto es:

$$F(\mathbf{x}, \mathbf{vr}) = \mathbf{0}$$

3.2.1.2 Aplicación de un Algoritmo de Continuación

En la precedente discusión, las ecuaciones de *load flow* fueron reformuladas para contener un parámetro de carga λ . El siguiente paso es aplicar un algoritmo de continuación al sistema de ecuaciones reformulado.

Sabiendo que $x = \begin{bmatrix} \theta \\ V \end{bmatrix}$,

podemos escribir (3.4) también como:

$$F(\theta, V, \lambda) = 0, \quad 0 = \lambda - \lambda_1 \quad (3.15)$$

donde λ_1 corresponde al punto de colapso en la dirección especificada γ .

Para resolver el problema, el algoritmo de continuación parte de una solución conocida, y usa un esquema predictor-corrector para encontrar soluciones sucesivas a diferentes niveles de carga. Mientras que el corrector es sólo un NR ligeramente modificado, el predictor merece una atención especial.

Es conveniente aclarar que la solución base $(\theta_0, V_0, \lambda_0)$ proviene de un *load flow* convencional.

3.2.1.3 Predicción de la Siguiete Solución

Una vez que una solución base ha sido encontrada ($\lambda=0$), una predicción de la siguiente solución puede ser hecha en la dirección tangente a la curva solución. Por consiguiente, la primera tarea en el proceso predictor es calcular el vector tangente.

Diferenciando (3.15) se obtiene

$$d_{\gamma} F(\theta, V, \lambda)_{\gamma} = F_{\theta} d\theta + F_V dV + F_{\lambda} d\lambda = 0 \quad (3.16)$$

lo que puede ser escrito como:

$$\begin{bmatrix} F_{\wedge} & F_V & F_{\vee} \end{bmatrix} \cdot \begin{bmatrix} d_{\wedge} \\ dV \\ d_{\vee} \end{bmatrix} = \mathbf{0} \quad (3.17)$$

El primer miembro de (3.17) es una matriz de derivadas parciales multiplicada por un vector de diferenciales. La matriz representa el jacobiano del *load flow* convencional aumentado en una columna (F_{\vee}), mientras que el vector es la tangente buscada. Existe, sin embargo, una importante barrera antes de que una única solución para el vector tangente pueda ser obtenida. El problema se presenta en el hecho de que una variable desconocida fue adicionada cuando y fue insertada en las ecuaciones del *load flow*, pero el número de ecuaciones no cambió. Por consiguiente, una ecuación más es necesaria para resolver (3.17).

Este problema puede ser resuelto eligiendo una magnitud no nula (por ejemplo 1) para una de las componentes del vector tangente. En otras palabras, si T es usada para representar al vector tangente;

$$T = \begin{bmatrix} d_{\wedge} \\ dV \\ d_{\vee} \end{bmatrix}, \quad T_j = \pm 1, \quad T \in \mathbb{R}^{n+1}$$

Esto resulta en

$$\begin{bmatrix} F_{\wedge} & F_V & F_{\vee} \\ & e_j & \end{bmatrix} \cdot \begin{bmatrix} T \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \pm 1 \end{bmatrix} \quad (3.18)$$

donde e_j es un vector fila de dimensión $n+1$ con todos los elementos iguales a cero excepto el j -ésimo, el cual es igual a uno. Si el índice j es elegido

correctamente, $T_j = \pm 1$ impone una norma no cero al vector tangente y garantiza que el jacobiano aumentado no será singular en el punto crítico.

Usar +1 o -1 depende de cómo la variable de estado j -ésima esta cambiando cuando la curva solución esta siendo trazada. Si ella está creciendo se debe usar +1, y -1 en caso contrario. Un método para elegir j y el signo de T_j se verá mas adelante.

Una vez que el vector tangente ha sido encontrado resolviendo (3.18), la predicción puede ser hecha como sigue:

$$\begin{pmatrix} \theta \\ V \\ \lambda \end{pmatrix}_{k+1} = \begin{pmatrix} \theta \\ V \\ \lambda \end{pmatrix}_k + \begin{pmatrix} d\theta \\ \sigma dV \\ d\lambda \end{pmatrix} \quad (3.19)$$

donde “prima” denota la solución predicha para el siguiente valor de λ (carga) y σ es un escalar que determina el tamaño del paso. Este último debería ser elegido de manera que la solución predicha este dentro del radio de convergencia del corrector. Si bien un valor constante de σ puede ser usado a través del proceso de continuación, existen métodos más elaborados para elegir el tamaño del paso [4]. Ahora que una predicción ha sido hecha, es necesario un método de corrección de esa aproximación a la solución.

3.2.1.4 Parametrización y Corrector

Toda técnica de continuación tiene un esquema particular de parametrización. Esta última provee un método para identificar cada solución a lo largo de la curva que está siendo trazada. El esquema usado en [3] consiste en una parametrización local.

En la parametrización local, el conjunto original de ecuaciones es aumentado en una ecuación que determina el valor de una de las variables de estado. En el caso de las ecuaciones reformuladas de *load flow*, esto significa especificar ya sea un módulo de tensión, un ángulo de fase o el parámetro de carga λ . En forma de ecuación esto puede ser expresado como sigue:

Sea

$$\mathbf{x}^a = \begin{pmatrix} \theta \\ \mathbf{V} \\ \lambda \end{pmatrix}, \quad \mathbf{x}^a \in \mathbb{R}^{n,1}$$

y sea

$$x_j^a = \eta$$

donde η es un valor apropiado para el j -ésimo elemento x_j^a del vector de estado ampliado \mathbf{x}^a . Entonces, el nuevo conjunto de $n+1$ ecuaciones con $n+1$ incógnitas a ser resuelto, será:

$$\begin{pmatrix} F(\mathbf{x}^a) \\ x_j^a - \eta \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ 0 \end{pmatrix} \quad (3.20)$$

Una vez elegidos un índice j y un valor de η apropiados, una ligera modificación del método de NR (alterado sólo en que una ecuación adicional y una variable de estado fueron agregadas) puede ser usado para resolver el conjunto de ecuaciones (3.20). Se obtiene así el corrector, necesario para corregir los valores predichos en la sección previa.

El índice j usado en el corrector es el mismo que el usado en el predictor, y η será igual a x_j^a , el cual es el valor predicho de x_j^a . De este modo, la variable de estado x_j^a es llamada *parámetro de continuación*. En el predictor, se arbitra que el parámetro de continuación tenga un incremento diferencial no nulo ($dx_j^a = T_j = \bar{\cdot} 1$); y en el corrector, su valor es especificado de manera que las otras variables puedan ser calculadas. ¿Cómo saber entonces que variable de estado debe ser usada como parámetro de continuación?

3.2.1.5 Elección del Parámetro de Continuación

Existen varios caminos para explicar como realizar la elección del parámetro de continuación apropiado. Matemáticamente, el parámetro de continuación es la variable de estado a la que corresponde la mayor componente en el vector tangente. En forma más simple, es la variable de estado que presenta la mayor razón de cambio cerca de la solución dada.

En el caso de un sistema de potencia, el parámetro de carga λ es probablemente la mejor elección cuando se parte de la solución base. Esto es especialmente cierto si el caso base está caracterizado por cargas normales o cargas leves. Bajo tales condiciones, módulos de tensión y ángulos de fase permanecen aproximadamente constantes ante variaciones de la carga.

Por otro lado, una vez que las cargas han sido incrementadas a través de cierto número de pasos del algoritmo de continuación y la curva se aproxima al *punto crítico*, las magnitudes y los ángulos de las tensiones experimentarán cambios significativos para pequeñas variaciones de λ .

En ese momento, λ sería una pobre elección como parámetro de continuación, en vista de que puede cambiar sólo una pequeña cantidad en comparación con las otras variables de estado. Por esta razón, la elección del parámetro de continuación debe ser re-evaluada en cada paso.

Una vez que la elección ha sido realizada para el primer paso, una buena manera de operar en pasos sucesivos es usar

$$x_j : |T_j| = \max \{|T_1|, |T_2|, \dots, |T_{n,1}|\} \quad (3.21)$$

donde el índice j corresponde a la máxima componente del vector tangente.

Seleccionado el parámetro de continuación, se registra el signo de su correspondiente componente de vector tangente, de tal manera que un valor apropiado, +1 o -1, pueda ser asignado a T_j en el próximo cálculo del vector tangente.

3.2.1.6 Detección del Punto Crítico

En el *punto crítico*, la carga (y por consiguiente λ) alcanza un máximo y luego empieza a decrecer. A causa de esto, la componente del vector tangente correspondiente a λ (es decir, $d\lambda$) es cero en el *punto crítico* y negativa debajo del mismo. Por tanto, una vez que la tangente ha sido calculada en el paso predictor, un examen del signo de $d\lambda$ revelará si el *punto crítico* ya ha sido pasado o no.

Si se desea, el método de continuación puede localizar el *punto crítico* con una precisión especificada. Para ello, una vez que se constate que el punto crítico ha sido pasado ($d\lambda$ cambió de signo), se adopta para el siguiente predictor un tamaño de paso modificado. El nuevo σ deberá ser menor en valor absoluto y de signo distinto respecto al último utilizado. Se realiza entonces la predicción-corrección correspondiente. Procediendo de esta forma cada vez que se detecte un cambio del signo de $d\lambda$, las soluciones sucesivas oscilarán alrededor del *punto crítico* acercándose paulatinamente a él.

A medida que esto ocurre, los valores de $d\lambda$ tienden a cero. El criterio de terminación del algoritmo es comparar el valor absoluto de $d\lambda$ con la precisión especificada,

$$|d\lambda| \leq \text{precisión} \quad (3.22)$$

En la figura 3.4 se puede visualizar el algoritmo de continuación en forma de diagrama de flujo.

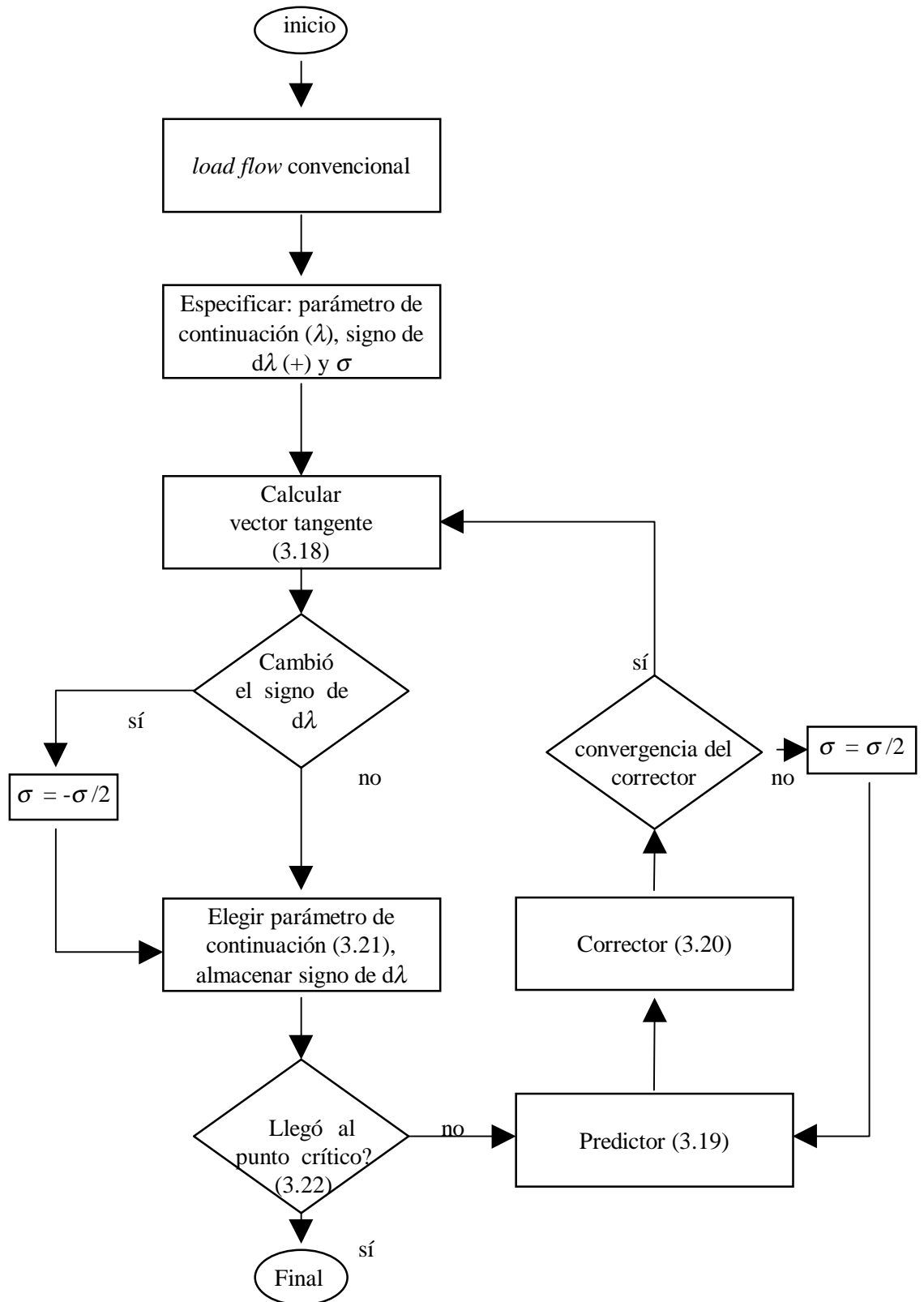


Figura 3.4 Flujoograma del Método de Continuación

3.2.2 Método del Punto de Colapso

El PoC proporciona una manera de calcular directamente el *límite de carga* o *punto crítico* de un sistema de potencia, para una dirección de crecimiento de carga especificada, sin necesidad de levantar la curva tensión-carga. Consiste en resolver uno de los dos siguientes sistemas de ecuaciones [4]:

$$F(\mathbf{x}, \lambda) = \mathbf{0} \quad (3.23 \text{ a})$$

$$F_x(\mathbf{x}, \lambda) \mathbf{u} = \mathbf{0} \quad (3.23 \text{ b})$$

$$\|\mathbf{u}\| = \epsilon \quad (3.23 \text{ c})$$

para el autovector derecho \mathbf{u} del jacobiano F_x , o análogamente

$$F(\mathbf{x}, \lambda) = \mathbf{0} \quad (3.24 \text{ a})$$

$$F_x^T(\mathbf{x}, \lambda) \mathbf{w} = \mathbf{0} \quad (3.24 \text{ b})$$

$$\|\mathbf{w}\| = \epsilon \quad (3.24 \text{ c})$$

para el autovector izquierdo \mathbf{w} de F_x , donde $\mathbf{u}, \mathbf{w} \in \mathbb{R}^n$.

Las ecuaciones (3.23 a) y (3.24 a) representan a la ecuación (3.4), estableciendo que \mathbf{x} es una de las soluciones del régimen permanente. Las ecuaciones (3.23 b) y (3.24 b) garantizan que el estado \mathbf{x} corresponda al punto de colapso, ya que sólo en este punto se tiene un autovalor igual a cero, dado que el jacobiano F_x es singular en dicho punto. Las relaciones (3.23 c) y (3.24 c) garantizan que los autovectores encontrados sean diferentes de cero.

Cualquiera de los sistemas de ecuaciones (3.23) o (3.24) pueden ser resueltos para calcular el punto de colapso. Para resolver este sistema no lineal de $2n+1$ ecuaciones con igual número de incógnitas se puede utilizar, por ejemplo, el método de NR.

Las experiencias con el método del PoC han demostrado que el autovector izquierdo da mejores resultados que su contraparte, el autovector derecho [4].

Cañizares y Alvarado [4] proponen usar una norma infinita como condición para las relaciones (3.23 c) y (3.24 c), es decir:

$$\| \mathbf{u} \|_{\infty} = \max \{ u_j \} = 1 \quad (3.25)$$

para la relación (3.23 c), y

$$\| \mathbf{w} \|_{\infty} = \max \{ w_j \} = 1 \quad (3.26)$$

para la relación (3.24 c). Los autores de esta propuesta encontraron en la misma una manera confiable de hallar el *punto de colapso*.

Desde luego, existen varias alternativas para satisfacer las relaciones (3.23 c) y (3.24 c), de forma a asegurar que el autovector sea distinto de cero en el punto de colapso. Una opción propuesta por Dobson y Lu en [28] es:

$$\mathbf{c}^T \mathbf{u} - 1 = 0 \quad (3.27)$$

para la relación (3.23 c), y

$$\mathbf{c}^T \mathbf{w} - 1 = 0 \quad (3.28)$$

para la relación (3.24 c). En las ecuaciones (3.27) y (3.28) $\mathbf{c} \in \mathbb{R}^n$ es un vector columna constante y distinto de cero.

Cuando el método es aplicado a sistemas de gran porte, existen varios factores que deben ser tomados en consideración, de entre los cuales se puede mencionar una estimación inicial adecuada para las variables de estado y para los autovectores.

El parámetro γ se inicializa a cero. El vector de variables de estado \mathbf{x} se inicializa con los valores obtenidos de aplicar un *load flow* convencional. Suposiciones iniciales para los autovectores \mathbf{u} y \mathbf{w} son obtenidas a partir de 4 o 5 iteraciones del *inverse power method*, mencionado en [4], aplicado al jacobiano inicial del flujo de potencia.

3.2.3 Comparación entre el MC y el PoC

El MC tiene la ventaja de encontrar un conjunto de soluciones a partir del caso base hasta el punto crítico. Estos resultados intermedios proporcionan información adicional acerca de la estabilidad del sistema y de las áreas propensas al colapso de tensión. En efecto, un índice de estabilidad de tensión y un indicador de barras “débiles” pueden ser obtenidos casi sin cálculos adicionales analizando el vector tangente a cada paso. En el contexto de esta obra, la barra “débil” es aquella que está más próxima a experimentar colapso de tensión [3].

El PoC tiene la ventaja de ofrecer información proveniente de los autovectores derecho e izquierdo. El autovector derecho puede ser usado para detectar variables (áreas) en la red propensas al colapso de tensión, mientras que el autovector izquierdo permite desarrollar una estrategia de control óptima usada para evitar puntos de colapso [4].

En resumen, el PoC otorga información de sensibilidad de tensión y sus *performances* de tiempo justifican su uso como herramienta. Este método puede llegar a ser bastante más rápido que el de continuación usando técnicas de esparsidad. Por otro lado, el método de continuación, aunque más lento produce información adicional que puede ser usada en el análisis de otros aspectos de estabilidad. Por estas razones ambos métodos se complementan, particularmente para sistemas donde uno de los métodos puede fallar en la obtención de los resultados deseados [4].

3.3 Métodos para encontrar el Punto de Colapso más Cercano

Dobson y Lu [28] proponen dos métodos para encontrar el punto de colapso más cercano a un punto de trabajo dado, tales son:

- Método Iterativo
- Método Directo

3.3.1 Método Iterativo

El método iterativo de cálculo del punto de colapso más cercano se basa en el uso de dos recursos principales. El primero, es un método para encontrar el punto de colapso correspondiente a una determinada dirección de crecimiento de carga; y el segundo, es una fórmula del vector normal a la Hipersuperficie Σ (ver figura 3.2).

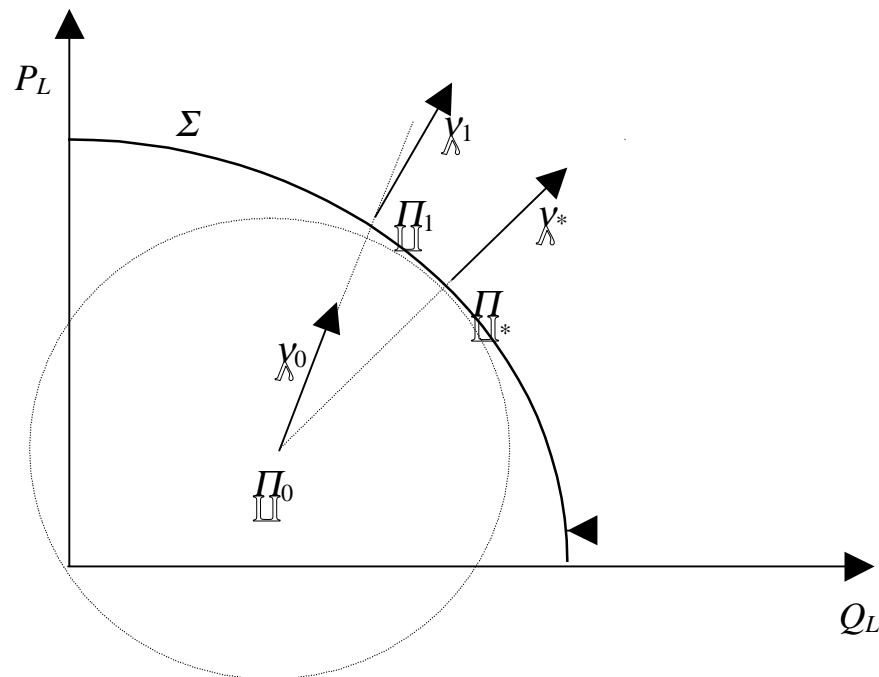


Figura 3.5 Método Iterativo

En lo que se refiere al primer recurso, para encontrar el punto de colapso dada una determinada dirección de crecimiento de carga, se puede usar el método del punto de colapso o el método de continuación.

En cuanto al segundo recurso, la normal χ_1 a la Hipersuperficie Σ en un punto Π_1 (ver figura 3.5) está dada por [28]:

$$\chi_1 = P(\Pi_1) - \alpha F_{\Pi\Pi}^T w_1 \quad (3.29)$$

donde

$$F_{\Pi} = \begin{bmatrix} \frac{\partial f_1}{\partial \Pi_1} & \dots & \frac{\partial f_1}{\partial \Pi_u} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial \Pi_1} & \dots & \frac{\partial f_n}{\partial \Pi_u} \end{bmatrix}, \quad F_{\Pi} \in \mathbb{R}^{n \times u}$$

F_{Π} es una matriz constante, puesto que las potencias de carga aparecen linealmente en las ecuaciones de *load flow*.

Nótese además que w_1 es el autovector izquierdo de $F_x(x_1, \Pi_1)$, correspondiente al autovalor cero. Por otro lado, α es un factor de escala cuya magnitud es elegida de manera que $\|P(\Pi_1)\| = 1$, y cuyo signo se elige tal que un crecimiento de carga en la dirección $P(\Pi_1)$, conduzca a la región donde no existe el equilibrio en régimen permanente.

A continuación se describirá como el cálculo del vector normal y la búsqueda del punto de colapso para una dirección dada pueden ser iterados para calcular la dirección más peligrosa χ^* , con sus correspondientes parámetro escalar λ^* y vector parámetro de carga Π^* . El procedimiento es el siguiente:

1. Se parte de una dirección de cargas inicial χ_0 estimada para χ^* .
2. Dada la dirección χ_{i-1} , se calcula el punto crítico correspondiente a ella, es decir, se calcula x_i, Π_i, λ_i , de manera que $\Pi_i \equiv \Pi_0 + \chi_{i-1} \lambda_i$.
3. Se calcula el autovector izquierdo w_i de $F_x(x_i, \Pi_i)$ correspondiente al autovalor cero.
4. Se adopta como nueva dirección de crecimiento de las cargas la dirección $\chi_i = P(\Pi_i) - \alpha_i F_{\Pi}^T w_i$.
5. Se iteran los pasos 1, 2, 3 hasta que la normal converja al mismo valor entre una iteración y otra. Dicha normal será χ^* , y el valor del vector de carga correspondiente al punto crítico será $\Pi^* = \Pi_0 + \chi^* \lambda^*$.

3.3.2 Método Directo

Las ecuaciones (3.24) fueron formuladas para encontrar el punto de colapso dada una dirección de crecimiento de las cargas. Dobson y Lu [28], con la intención de encontrar el punto crítico más cercano, generalizan las ecuaciones (3.24) de la siguiente forma:

$$F(\mathbf{x}, \Pi) = 0 \quad (3.30 \text{ a})$$

$$F_x^T(\mathbf{x}, \Pi) \mathbf{w} = 0 \quad (3.30 \text{ b})$$

$$K(\Pi - \Pi_0) - F_{\Pi}^T \mathbf{w} = 0 \quad (3.30 \text{ c})$$

$$\mathbf{c}^T F_{\Pi}^T \mathbf{w} = 1 \quad (3.30 \text{ d})$$

donde $K \in \mathbb{R}$ y $\mathbf{c} \in \mathbb{R}^u$ son constantes definidas *a priori*.

Las ecuaciones (3.30 a) y (3.30 b) corresponden a (3.24 a) y (3.24 b) respectivamente. La ecuación (3.30 c) condiciona que $(\Pi - \Pi_0)$ y el vector normal sean paralelos, lo cual asegura que el punto encontrado sea el más cercano.

Finalmente, la ecuación (3.30 d) asegura un autovector \mathbf{w} diferente de cero, de manera análoga a lo establecido con la ecuación (3.24 c).

Las ecuaciones (3.30) representan un sistema no lineal de $2n+u+1$ ecuaciones con igual número de incógnitas, y pueden ser resueltas, por ejemplo, empleando el método de NR.

4. PROPOSICIÓN DE NUEVOS MÉTODOS DE SOLUCIÓN

Este capítulo presenta las bases matemáticas y los detalles computacionales de dos nuevas implementaciones realizadas, ambas basadas en el PoC.

La sección 4.1 aborda la descomposición LU, necesaria para explicar uno de los nuevos métodos propuestos, el *PoC con Variante Aceleradora*, que se desarrolla en la sección 4.2.

En las secciones 4.3 y 4.4 se exponen el concepto y la formulación matemática de los *Team Algorithms*, herramienta que se utiliza en el desarrollo de otra nueva propuesta, el *PoC en un Sistema de Procesamiento Distribuido*.

En la sección 4.5 se explica brevemente el sincronismo y asincronismo como formas de comunicación entre los procesadores de un sistema distribuido. Finalmente, en la sección 4.6 se discuten los detalles de la implementación del *PoC en un Sistema de Procesamiento Distribuido*.

4.1 Descomposición LU

Sea el sistema de n ecuaciones lineales con n incógnitas dado por:

$$A x = B \quad A \in \mathbb{R}^{n \times n} \quad x, B \in \mathbb{R}^n \quad (4.1)$$

La solución de este sistema es:

$$x = A^{-1} B \quad (4.2)$$

En la práctica es raro que la inversa A^{-1} sea realmente útil. La mayor parte de lo que se puede lograr con A^{-1} puede lograrse de manera más eficiente a través de otros medios tales como la descomposición LU [34].

Para resolver (4.1) utilizando descomposición LU se deben obtener a partir de A dos matrices cuadradas: una triangular inferior L (*lower*) y una triangular superior U (*upper*) [34], de modo que

$$A = LU \quad (4.3)$$

quedando entonces (4.1) como

$$LUx = B \quad (4.4)$$

Para resolver (4.4) se realizan dos operaciones con matrices triangulares, siendo la primera:

$$Ux = L^{-1}B \quad (4.5)$$

y la segunda:

$$x = U^{-1}(L^{-1}B) \quad (4.6)$$

Lo que interesa en esta sección, es resaltar dos aspectos de la resolución de un sistema de ecuaciones lineales utilizando descomposición LU.

El primero de ellos es que el número necesario de multiplicaciones para efectuar la descomposición LU, indicada en (4.3), es del orden de n^3 , mientras que para realizar alguna operación con matrices triangulares, como se indica en (4.5) o (4.6), el número de multiplicaciones es sólo del orden de n^2 [34]. En resumen, si n es grande, como ocurre en los sistemas con que se trabaja habitualmente, el tiempo de procesamiento empleado en operaciones con matrices triangulares, es despreciable frente al que se consume en la descomposición LU.

El segundo aspecto a resaltar es que, si la matriz A tiene una singularidad, es decir, un autovalor igual a cero, la descomposición LU con pivotamientos conduce a una matriz U caracterizada por $U_{nn} = 0$. En este caso, y en general si algún elemento de la diagonal de U es cero, la operación

indicada en (4.6) no se puede realizar, el sistema de ecuaciones no tiene solución.

Los dos aspectos resaltados serán tenidos en cuenta en la siguiente sección.

4.2 PoC con Variante Aceleradora

El PoC, como se vio en el capítulo 3, emplea el sistema de ecuaciones (3.24). Si se utiliza la ecuación (3.28), propuesta por Dobson-Lu en [28], para la relación (3.24 c) se tiene:

$$\Phi(\mathbf{x}, \lambda, \mathbf{w}) = \begin{bmatrix} F(\mathbf{x}, \lambda) \\ F_x^T(\mathbf{x}, \lambda)\mathbf{w} \\ \mathbf{c}^T\mathbf{w} - 1 \end{bmatrix} = \mathbf{0} \quad (4.7)$$

donde $\Phi : \mathbb{R}^{2n+1} \rightarrow \mathbb{R}^{2n+1}$.

Para resolver este sistema de $2n+1$ ecuaciones no lineales con igual número de incógnitas, se puede utilizar el método de NR. En cada iteración del NR debe resolverse el siguiente sistema de ecuaciones lineales:

$$\begin{bmatrix} F_x & F_\lambda & \mathbf{0} \\ (F_x^T\mathbf{w})_x & \mathbf{0} & F_x^T \\ \mathbf{0} & 0 & \mathbf{c}^T \end{bmatrix} \begin{bmatrix} \Delta\mathbf{x} \\ \Delta\lambda \\ \Delta\mathbf{w} \end{bmatrix} = \begin{bmatrix} F \\ F_x^T\mathbf{w} \\ \mathbf{c}^T\mathbf{w} - 1 \end{bmatrix} \quad (4.8)$$

Es decir, en cada iteración del NR, se resuelve un sistema de $2n+1$ ecuaciones lineales, que, recordando el primero de los aspectos resaltados en la sección anterior, requiere un número de multiplicaciones del orden de $(2n+1)^3$.

La *Variante Aceleradora* aprovecha la configuración del sistema representado en (4.8) y disminuye el número de cálculos necesarios para obtener $\nabla\mathbf{x}$, $\nabla\lambda$ y $\nabla\mathbf{w}$. Lo logra mediante una separación de (4.8) en dos partes y la resolución de ambos subsistemas con una sola descomposición LU que requiere un número de multiplicaciones del orden de n^3 .

4.2.1 Separación del Sistema

Se puede escribir (4.8) como:

$$F_x \Delta \mathbf{x} + F_\lambda \Delta \lambda = F \quad (4.9 \text{ a})$$

$$(F_x^T \mathbf{w})_x \Delta \mathbf{x} + F_x^T \Delta \mathbf{w} = F_x^T \mathbf{w} \quad (4.9 \text{ b})$$

$$\mathbf{c}^T \Delta \mathbf{w} = \mathbf{c}^T \mathbf{w} = 1 \quad (4.9 \text{ c})$$

Se quiere despejar $\Delta \mathbf{x}$ de (4.9 a) como $\Delta \mathbf{x} = F_x^{-1}(F - F_\lambda \Delta \lambda)$ y sustituir en (4.9 b). Sin embargo, este despeje no se puede realizar, porque F_x^{-1} no existe en el punto de colapso. Se hace entonces primero la descomposición LU de F_x :

$$F_x = LU = L \begin{bmatrix} U_{n-1} \\ U_n \end{bmatrix} \quad (4.10)$$

donde U_{n-1} y U_n son submatrices de U , de dimensiones $(n-1, n-1)$ y $(n, 1)$ respectivamente, y de igual forma se escribe

$$\Delta \mathbf{x} = \begin{bmatrix} \Delta \mathbf{x}_{n-1} \\ \Delta x_n \end{bmatrix} \quad (4.11)$$

donde $\Delta \mathbf{x}_{n-1}$ contiene los $n-1$ primeros elementos de $\Delta \mathbf{x}$, y Δx_n es el n -ésimo elemento.

Teniendo en cuenta entonces (4.10) y (4.11) se puede escribir (4.9 a) como:

$$L \begin{bmatrix} U_{n-1} \\ U_n \end{bmatrix} \begin{bmatrix} \Delta \mathbf{x}_{n-1} \\ \Delta x_n \end{bmatrix} + F_\lambda \Delta \lambda = F \quad (4.12)$$

y realizando algunas transformaciones:

$$\begin{bmatrix} U_{n-1} \\ U_n \end{bmatrix} \begin{bmatrix} \Delta \mathbf{x}_{n-1} \\ \Delta x_n \end{bmatrix} + L^{-1} F_\lambda \Delta \lambda = L^{-1} F$$

$$\begin{bmatrix} U_{n-1} & L^{-1}F_{\nu} \end{bmatrix} \cdot \begin{bmatrix} \varpi x_{n-1} \\ \varpi x_n \end{bmatrix} + U_n \varpi x_n = L^{-1}F$$

donde $\begin{bmatrix} U_{n-1} & L^{-1}F_{\nu} \end{bmatrix}$ es una matriz triangular superior no singular, finalmente se llega a:

$$\begin{bmatrix} \varpi x_{n-1} \\ \varpi x_n \end{bmatrix} = \begin{bmatrix} U_{n-1} & L^{-1}F_{\nu} \end{bmatrix}^{-1} (L^{-1}F - U_n \varpi x_n) \quad (4.13)$$

que da la solución del primer subsistema (4.9 a), una vez que se conozca ϖx_n .

Por otra parte, considerando el termino $(F_x^T \mathbf{w})_x$ de la ecuación (4.9 b) se puede escribir:

$$(F_x^T \mathbf{w})_x = \begin{bmatrix} (F_x^T \mathbf{w})_{x_{n-1}} & (F_x^T \mathbf{w})_{x_n} \end{bmatrix} \quad (4.14)$$

Introduciendo (4.14) en (4.9 b) se obtiene:

$$\begin{bmatrix} (F_x^T \mathbf{w})_{x_{n-1}} & (F_x^T \mathbf{w})_{x_n} \end{bmatrix} \cdot \begin{bmatrix} \varpi x_{n-1} \\ \varpi x_n \end{bmatrix} + \mathbf{0}_{\varpi x_n} + F_x^T \varpi \mathbf{w} = F_x^T \mathbf{w} \quad (4.15)$$

$$\begin{bmatrix} (F_x^T \mathbf{w})_{x_{n-1}} & \mathbf{0} \end{bmatrix} \cdot \begin{bmatrix} \varpi x_{n-1} \\ \varpi x_n \end{bmatrix} + (F_x^T \mathbf{w})_{x_n} \varpi x_n + F_x^T \varpi \mathbf{w} = F_x^T \mathbf{w} \quad (4.16)$$

Sustituyendo $\begin{bmatrix} \varpi x_{n-1} \\ \varpi x_n \end{bmatrix}$ de (4.13) en (4.16) y ordenando se tiene:

$$\begin{aligned} F_x^T \varpi \mathbf{w} + \left((F_x^T \mathbf{w})_{x_n} - \begin{bmatrix} (F_x^T \mathbf{w})_{x_{n-1}} & \mathbf{0} \end{bmatrix} \cdot \begin{bmatrix} U_{n-1} & L^{-1}F_{\nu} \end{bmatrix}^{-1} U_n \right) \varpi x_n = \\ = F_x^T \mathbf{w} - \begin{bmatrix} (F_x^T \mathbf{w})_{x_{n-1}} & \mathbf{0} \end{bmatrix} \cdot \begin{bmatrix} U_{n-1} & L^{-1}F_{\nu} \end{bmatrix}^{-1} L^{-1}F \end{aligned} \quad (4.17)$$

Haciendo

$$A_{12} = (F_x^T \mathbf{w})_{x_n} - \left[(F_x^T \mathbf{w})_{x_{n-1}} \quad \mathbf{0} \right] \cdot \left[U_{n-1} \quad L^{-1} F_{\cdot n} \right]^{-1} U_n \quad (4.18)$$

$$B_1 = F_x^T \mathbf{w} - \left[(F_x^T \mathbf{w})_{x_{n-1}} \quad \mathbf{0} \right] \cdot \left[U_{n-1} \quad L^{-1} F_{\cdot n} \right]^{-1} L^{-1} F \quad (4.19)$$

se puede escribir el sistema de $n+1$ ecuaciones con $n+1$ incógnitas formado a partir de (4.9 c) y (4.17) de la siguiente forma:

$$\underbrace{\begin{bmatrix} F_x^T & A_{12} \\ \mathbf{c}^T & 0 \end{bmatrix}}_{\mathbf{A}} \cdot \underbrace{\begin{bmatrix} \nabla \mathbf{w} \\ \nabla x_n \end{bmatrix}}_{\mathbf{B}} = \underbrace{\begin{bmatrix} B_1 \\ \mathbf{c}^T \mathbf{w} - 1 \end{bmatrix}}_{\mathbf{B}} \quad (4.20)$$

Este es el segundo subsistema, cuyas incógnitas son $\nabla \mathbf{w}$ y ∇x_n . Una vez resuelto, el valor de ∇x_n hallado se introduce en el primer subsistema, efectuándose la operación indicada en (4.13) para hallar ∇x_{n-1} y $\nabla \mathbf{y}$.

4.2.2 Cómo hacer sólo una Descomposición LU

Para resolver (4.20) se debería realizar la descomposición LU de la matriz \mathbf{A} . Para evitar realizar esta nueva descomposición LU, deben encontrarse dos matrices triangulares \mathbf{C} y \mathbf{D} tales que

$$\mathbf{C} \cdot \mathbf{D} = \begin{bmatrix} F_x^T & A_{12} \\ \mathbf{c}^T & 0 \end{bmatrix} \quad (4.21)$$

y cuyos elementos sean ya conocidos.

Recordando de (4.10) que $F_x = \mathbf{L}\mathbf{U}$, entonces $F_x^T = \mathbf{U}^T \mathbf{L}^T$, por lo que se puede escribir (4.21) como:

$$\underbrace{\begin{bmatrix} \mathbf{U}^T & \mathbf{0} \\ \mathbf{C}_{21} & \mathbf{C}_{22} \end{bmatrix}}_{\mathbf{C}} \cdot \underbrace{\begin{bmatrix} \mathbf{L}^T & \mathbf{D}_{12} \\ \mathbf{0} & 1 \end{bmatrix}}_{\mathbf{D}} = \begin{bmatrix} \mathbf{U}^T \mathbf{L}^T & A_{12} \\ \mathbf{c}^T & 0 \end{bmatrix} \quad (4.22)$$

donde la matriz C es triangular inferior y D es triangular superior con los elementos de su diagonal principal iguales a uno.

De (4.20) y (4.22)

$$\begin{array}{c} \left[\begin{array}{cc|cc} U^T & \mathbf{0} & L^T & D_{12} \\ \hline C_{21} & C_{22} & \mathbf{0} & 1 \end{array} \right] \begin{array}{c} \Delta w \\ \Delta x_n \end{array} = \begin{array}{c} B_1 \\ \underbrace{c^T w}_{B} \end{array} \end{array} \quad (4.23)$$

que, una vez hallados C_{21} , C_{22} y D_{12} fácilmente a partir de (4.22), se puede resolver con sólo dos operaciones de matrices triangulares mediante

$$\begin{array}{c} \left[\begin{array}{cc|cc} \Delta w & L^T & D_{12}^{-1} & U^T \\ \hline \Delta x_n & \mathbf{0} & 1 & C_{21} \end{array} \right] \begin{array}{c} \mathbf{0} \\ c^T w \end{array} = \begin{array}{c} B_1 \\ \mathbf{1} \end{array} \end{array} \quad (4.24)$$

y se tiene la solución de (4.20) sin ninguna descomposición LU adicional.

4.2.3 Una Dificultad a Salvar

Recordando el segundo aspecto resaltado en la sección 4.1, como F_x es singular en el punto de colapso, se tiene en la matriz U que $U_{nn} = 0$ en dicho punto. Por consiguiente, un elemento de la diagonal principal de la matriz C de (4.23) será entonces igual a cero, por lo que (4.24) no se puede efectuar.

Para remediar este inconveniente, se escribe (4.20) como

$$\begin{array}{c} \left[\begin{array}{cc|cc} U_{n-1}^T L^T & A_{12n-1} \\ \hline U_n^T L^T & A_{12n} \\ c^T & 0 \end{array} \right] \begin{array}{c} \Delta w \\ \Delta x_n \end{array} = \begin{array}{c} B_{1n-1} \\ B_{1n} \\ \underbrace{c^T w}_{B} \end{array} \end{array} \quad (4.25)$$

e intercambiando las dos últimas filas se llega a

$$\begin{pmatrix} U_{n-1}^T L^T & A_{12_{n-1}} \\ c^T & 0 \\ U_n^T L^T & A_{12_n} \end{pmatrix} \begin{pmatrix} \Delta w \\ \Delta x_n \end{pmatrix} = \begin{pmatrix} B_{1_{n-1}} \\ c^T w_{-1} \\ B_{1_n} \end{pmatrix} \quad (4.26)$$

A partir de aquí se procede como en la subsección 4.2.2, es decir, en primer lugar se encuentran dos matrices C y D triangulares tales que

$$C \cdot D = \begin{pmatrix} U_{n-1}^T L^T & A_{12_{n-1}} \\ c^T & 0 \\ U_n^T L^T & A_{12_n} \end{pmatrix} \quad (4.27)$$

Entonces, se escribe (4.27) como

$$\underbrace{\begin{pmatrix} U_{n-1}^T & 0 \\ C_{21} & 0 \\ U_n^T & C_{32} \end{pmatrix}}_C \underbrace{\begin{pmatrix} L^T & D_{12} \\ 0 & 1 \end{pmatrix}}_D \begin{pmatrix} U_{n-1}^T L^T & A_{12_{n-1}} \\ c^T & 0 \\ U_n^T L^T & A_{12_n} \end{pmatrix} \quad (4.28)$$

donde la matriz C es triangular inferior y D es triangular superior con los elementos de su diagonal principal iguales a uno.

De (4.26) y (4.28)

$$\underbrace{\begin{pmatrix} U_{n-1}^T & 0 \\ C_{21} & 0 \\ U_n^T & C_{32} \end{pmatrix}}_C \underbrace{\begin{pmatrix} L^T & D_{12} \\ 0 & 1 \end{pmatrix}}_D \begin{pmatrix} \Delta w \\ \Delta x_n \end{pmatrix} = \begin{pmatrix} B_{1_{n-1}} \\ c^T w_{-1} \\ B_{1_n} \end{pmatrix} \quad (4.29)$$

que se resuelve mediante

$$\begin{pmatrix} \Delta w \\ \Delta x_n \end{pmatrix} \begin{pmatrix} L^T & D_{12} \\ 0 & 1 \end{pmatrix} \begin{pmatrix} U_{n-1}^T & 0 \\ C_{21} & 0 \\ U_n^T & C_{32} \end{pmatrix} = \begin{pmatrix} B_{1_{n-1}} \\ c^T w_{-1} \\ B_{1_n} \end{pmatrix} \quad (4.30)$$

donde D_{12} , C_{21} y C_{32} se obtienen fácilmente a partir de (4.28) siendo:

$$D_{12} = (U_{n-1}^T)^{-1} A_{12_{n-1}} \quad (4.31 \text{ a})$$

$$C_{21} = c^T (L^T)^{-1} \quad (4.31 \text{ b})$$

$$C_{32} = A_{12_n} - U_n^T D_{12} \quad (4.31 \text{ c})$$

El valor de Δx_n de (4.30) se introduce posteriormente en (4.13) para hallar Δx_{n-1} y $\Delta \lambda$.

Resta hacer una observación respecto a los pivotamientos que se realizan en la descomposición LU y que fueron mencionados en la sección 4.1. Tales pivotamientos se efectúan para evitar posibles divisiones por cero y minimizar errores de redondeo en el algoritmo de descomposición LU.

La implementación de la *Variante* descrita contempla dichos pivotamientos. Sin embargo, por simplicidad, los mismos fueron omitidos en la presente exposición.

4.3 Team Algorithms: Concepto

Hasta aquí fueron presentados métodos secuenciales de cálculo del *punto de colapso*. Estos métodos tienen la desventaja de que en la medida que los problemas crecen en tamaño se necesitan computadores cada vez más potentes, lo que implica mayores costos.

Como se citó en el capítulo 1, la tendencia para afrontar esta dificultad, sobre todo en los países en vías de desarrollo, es volcarse a la resolución de estos problemas a través de la computación distribuida.

Esta última tiene arquitectura compatible para la utilización de diferentes algoritmos en cada procesador, de forma tal que, resolviendo cada uno de ellos una parte de un problema global, entre todos resuelvan el conjunto.

Esta técnica, llamada *Team Algorithm* [30], permite además trabajar en ambiente asíncrono, es decir, diferentes procesadores trabajando a diferentes velocidades cada uno de ellos.

La intención de las implementaciones a ser estudiadas, es encontrar un vector solución que satisfaga (posiblemente con una dada tolerancia) el sistema de ecuaciones algebraicas dado inicialmente. Para esto, se asigna a cada procesador un subproblema junto con las componentes del vector solución que serán actualizadas por ese procesador.

Como el subproblema a ser resuelto en un determinado procesador puede depender también de las componentes del vector no actualizadas por él, cada procesador incluye en el cálculo los datos más actualizados que posee de las componentes actualizadas por los otros procesadores. Para ello se establecen comunicaciones ya sea síncronas o asíncronas entre los procesadores, de forma que los valores más actualizados puedan ser conocidos e incluidos en los cálculos de todos los procesadores.

El conjunto avanza para la solución global del problema de una forma típicamente iterativa. Esto es porque, una vez calculados nuevos valores para las componentes actualizadas por un dado procesador, los mismos deben ser comunicados a los otros procesadores del sistema, para verificar que la solución global para todo el sistema ya fue alcanzada. De no ser así, el proceso iterativo de búsqueda debe continuar y el procesador considerado (que ya encontró una solución local) puede eventualmente recibir nuevos valores de las otras componentes (calculadas en los otros procesadores) que hagan que el subproblema en él colocado, continúe sin ser resuelto. Son así necesarias nuevas iteraciones en ese procesador.

4.4 *Team Algorithms*: Formulación Matemática

En esta sección se considerará un sistema de n ecuaciones algebraicas no lineales dado por:

$$\Phi(z) = 0, \quad z \in \mathbb{R}^n, \quad \Phi: \mathbb{R}^n \rightarrow \mathbb{R}^n \quad (4.32)$$

La idea básica para resolver (4.32) en un sistema distribuido es usar p procesadores, de manera tal que cada uno de ellos resuelve sólo una parte de todo el sistema de ecuaciones y comunica sus resultados parciales a los demás procesadores, resolviendo todos en forma conjunta el problema global [35].

Asumiendo que los elementos de (4.32) puedan ser particionados de modo que

$$\Phi(z) = [\Phi_1^T(z) \quad \Phi_2^T(z) \quad \dots \quad \Phi_p^T(z)]^T \quad (4.33)$$

$$z = [z_1^T \quad z_2^T \quad \dots \quad z_p^T]^T \quad (4.34)$$

se puede escribir (4.32) como:

$$\Phi_m(z) = \theta, \quad m = 1, \dots, p \quad (4.35)$$

Cada subproblema $\Phi_m(z) = \theta$ puede ser resuelto por un algoritmo iterativo representado por el mapeamiento G_m que actualiza z_m , es decir:

$$z_m(i+1) = G_m(z^{(i)}), \quad m = 1, \dots, p \quad (4.36)$$

El algoritmo iterativo es elegido de modo tal que el punto fijo $z^\circ = G(z^\circ)$ de

$$G(z) = [G_1^T(z) \quad G_2^T(z) \quad \dots \quad G_p^T(z)]^T \quad (4.37)$$

sea una solución de (4.32), es decir, $\Phi(z^\circ) = \theta$.

La expresión matemática para $G_m(z)$ depende del algoritmo específico elegido para resolver el correspondiente subproblema. Cuando se eligen dos o más métodos diferentes se tiene una combinación de algoritmos que se dio en llamar *Team Algorithm* [35].

4.5 Sincronismo y Asincronismo

Como ya fue mencionado en la sección 4.3, el objetivo de las implementaciones es encontrar un vector solución z que satisfaga el sistema de ecuaciones algebraicas $\Phi(z) = 0$ dado inicialmente.

En la resolución de problemas en sistemas distribuidos cada procesador intenta resolver el problema a él asignado, utilizando un algoritmo adecuado que permita actualizar las componentes del vector incógnita correspondientes a dicho procesador.

Esto es, cada procesador m resuelve a cada iteración i su sistema de ecuaciones $\Phi_m(z) = 0$, actualizando z_m , con el operador $G_m(z)$, comunicando eventualmente a los otros procesadores del sistema dicho valor actualizado, e incorporando en sus cálculos los valores actualizados de z_j ($j \neq m$) que recibió de los otros procesadores del sistema.

Ahora bien, la comunicación entre los procesadores, es decir, el envío y recepción de las componentes actualizadas, puede ser síncrona o asíncrona.

En un proceso con comunicación síncrona, en una dada iteración, se debe esperar a que todos los procesadores lleguen a una solución local. En ese momento los procesadores intercambian sus datos actualizados. Seguidamente, se inicia una nueva iteración con cada uno de los procesadores dedicado a resolver su subproblema con el conjunto de datos actualizados y así sucesivamente.

La desventaja de la comunicación síncrona consiste en el tiempo desaprovechado, ya que un procesador que llegue a una solución local en una dada iteración, debe esperar ocioso a que el último de los procesadores termine sus cálculos para comunicarse e iniciar una nueva iteración.

En el proceso asíncrono, en cambio, un procesador recibe los datos de los otros sin interrumpir sus cálculos en una determinada iteración; y viceversa, un procesador envía sus datos actualizados a los otros sin importar que éstos se encuentren realizando cálculos. Los datos recibidos por un procesador serán incorporados cuando sean necesarios, es decir, recién al comenzar su siguiente iteración.

Por consiguiente, los procesadores pueden trabajar a diferentes velocidades, sin sincronizar sus tareas necesariamente, pudiendo realizar entonces números de iteraciones dispares.

La ventaja de las implementaciones asíncronas radica en el máximo aprovechamiento del tiempo de procesamiento, aún en circunstancias donde la comunicación pueda acusar retardos aleatorios.

4.6 El PoC en un Sistema de Procesamiento Distribuido

La propuesta de esta sección consiste en aplicar la técnica de los TA al sistema de ecuaciones (4.7), el cual se vuelve a escribir a continuación:

$$\Phi(\mathbf{x}, \lambda, \mathbf{w}) = \begin{cases} F(\mathbf{x}, \lambda) \\ F_x^T(\mathbf{x}, \lambda)\mathbf{w} \\ \mathbf{c}^T \mathbf{w} - 1 \end{cases} = \mathbf{0} \quad (4.38)$$

En el problema del flujo de carga, la partición del sistema ya ha sido enfocada por diferentes autores [36]. Si se considera el uso de dos procesadores, un criterio de partición requiere encontrar:

$$F(\mathbf{x}) = \begin{cases} F_1(\mathbf{x}_1, \mathbf{x}_2) \\ F_2(\mathbf{x}_1, \mathbf{x}_2) \end{cases} = \mathbf{0} \quad (4.39)$$

donde $\mathbf{x} = \begin{cases} \mathbf{x}_1 \\ \mathbf{x}_2 \end{cases}$, de modo que las funciones F_1 y F_2 dependan principalmente de \mathbf{x}_1 y \mathbf{x}_2 respectivamente. De esta forma, un procesador resuelve $F_1(\mathbf{x}_1, \mathbf{x}_2) = \mathbf{0}$ actualizando \mathbf{x}_1 , y el otro procesador resuelve $F_2(\mathbf{x}_1, \mathbf{x}_2) = \mathbf{0}$ actualizando \mathbf{x}_2 .

En el caso del problema del *Punto de Colapso*, la ecuación $F(\mathbf{x}, \lambda) = \mathbf{0}$ de (4.38), no es otra que la del flujo de carga incluyendo un parámetro escalar de crecimiento de cargas λ [3], de manera que se puede optar por el criterio de

partición expuesto anteriormente. En consecuencia, la ecuación (4.38) puede ser escrita como:

$$\Phi \begin{bmatrix} F_1 \\ F_2 \\ F_{1x_1}^T w_1 + F_{2x_1}^T w_2 \\ F_{1x_2}^T w_1 + F_{2x_2}^T w_2 \\ c_1^T w_1 + c_2^T w_2 - 1 \end{bmatrix} = \mathbf{0} \quad (4.40)$$

donde $w = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}$

Por el criterio de partición, se tiene que los elementos de $F_{1x_2}^T$ y $F_{2x_1}^T$ son relativamente pequeños, por lo que se puede ver que la 3ra y 4ta ecuaciones de (4.40) dependen fundamentalmente de x_1 , w_1 y de x_2 , w_2 respectivamente. Las particiones anteriormente citadas pueden ser observadas y realizadas inclusive en forma intuitiva, puesto que, dadas las diferentes topologías de sistemas de potencia, éstas corresponden a los desacoplamientos naturales del mismo.

Por otro lado, la última ecuación depende tanto de w_1 como de w_2 lo que hace difícil establecer *a priori* que partición es la más adecuada para ella, o dicho de otra forma, que procesador la resolverá. Como consecuencia de esto, se plantean tres esquemas de solución.

4.6.1 Primer Esquema

Consiste en asignar la última ecuación a uno de los dos procesadores, conforme (4.41), donde el procesador 1 actualiza x_1 , w_1 y λ , mientras que el procesador 2 actualiza x_2 y w_2 .

El subproblema asignado al procesador 1 es:

$$\Phi_1 = \begin{bmatrix} F_1 \\ F_{1x_1}^T \mathbf{w}_1 + F_{2x_1}^T \mathbf{w}_2 \\ \mathbf{c}_1^T \mathbf{w}_1 + \mathbf{c}_2^T \mathbf{w}_2 - 1 \end{bmatrix} = \mathbf{0} \quad (4.41 \text{ a})$$

y para el procesador 2,

$$\Phi_2 = \begin{bmatrix} F_2 \\ F_{1x_2}^T \mathbf{w}_1 + F_{2x_2}^T \mathbf{w}_2 \\ \mathbf{c}_1^T \mathbf{w}_1 + \mathbf{c}_2^T \mathbf{w}_2 - 1 \end{bmatrix} = \mathbf{0} \quad (4.41 \text{ b})$$

Con este esquema no se obtiene convergencia, lo que se demuestra experimentalmente en el capítulo 5 (tabla 5.2).

4.6.2 Segundo Esquema

Con el fin de lograr la convergencia del método, se decidió que la última ecuación se resuelva en ambos procesadores. Este procedimiento se conoce con el nombre de *overlapping parcial* [37]. Así, es posible agrupar las ecuaciones (4.40) y de manera que el problema local en el procesador 1 sea:

$$\Phi_1 = \begin{bmatrix} F_1 \\ F_{1x_1}^T \mathbf{w}_1 + F_{2x_1}^T \mathbf{w}_2 \\ \mathbf{c}_1^T \mathbf{w}_1 + \mathbf{c}_2^T \mathbf{w}_2 - 1 \end{bmatrix} = \mathbf{0} \quad (4.42)$$

y en el procesador 2

$$\Phi_2 = \begin{bmatrix} F_2 \\ F_{1x_2}^T \mathbf{w}_1 + F_{2x_2}^T \mathbf{w}_2 \\ \mathbf{c}_1^T \mathbf{w}_1 + \mathbf{c}_2^T \mathbf{w}_2 - 1 \end{bmatrix} = \mathbf{0} \quad (4.43)$$

En la figura 4.1 se puede observar la implementación del método de solución a ser utilizado. Cada procesador m resuelve su ecuación local $\Phi_m = 0$,

usando el operador G_m (por ejemplo el algoritmo NR) que actualiza \mathbf{x}_m , \mathbf{w}_m , y su versión λ_m de la variable λ . Nótese además que el valor λ_m que utiliza el operador G_m es el valor de λ obtenido del administrador en el procesador m .

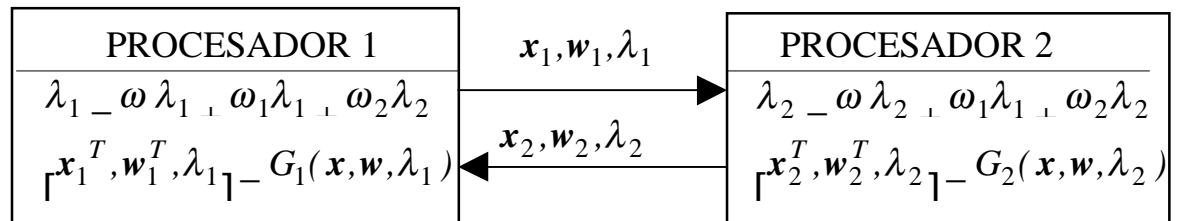


Figura 4.1 Implementación del segundo esquema de solución

Las iteraciones del TA paralelo síncrono, correspondientes a este esquema, pueden ser matemáticamente representadas por las ecuaciones:

$$\begin{aligned}
 & \lambda_1(i+1) - \omega \lambda_1(i) - \omega_1 \lambda_1(i) - \omega_2 \lambda_2(i) \\
 & [\mathbf{x}_1^T(i+1) \quad \mathbf{w}_1^T(i+1) \quad \lambda_1(i+1)] - G_1(\mathbf{x}(i), \mathbf{w}(i), \lambda_1(i+1)) \quad \setminus \\
 & \lambda_2(i+1) - \omega \lambda_2(i) - \omega_1 \lambda_1(i) - \omega_2 \lambda_2(i) \\
 & [\mathbf{x}_2^T(i+1) \quad \mathbf{w}_2^T(i+1) \quad \lambda_2(i+1)] - G_2(\mathbf{x}(i), \mathbf{w}(i), \lambda_2(i+1)) \quad \setminus
 \end{aligned} \tag{4.44}$$

El TA síncrono, ilustrado en la figura 4.1 y expresado por (4.44), funciona de la siguiente forma: el procesador 1 recibe $\mathbf{x}_2(i)$, $\mathbf{w}_2(i)$ y $\lambda_2(i)$. El administrador de este procesador calcula $\lambda_1(i+1)$, valor que el operador G_1 utiliza para actualizar \mathbf{x}_1 , \mathbf{w}_1 y λ_1 . Posteriormente, este procesador transmite los valores actualizados al procesador 2, el cual opera en forma análoga.

Los pesos ω , ω_1 y ω_2 son elegidos de manera a asegurar que para una solución λ° , se cumpla $\lambda_m(i+1) - \lambda_m(i) - \lambda_1(i) - \lambda_2(i) - \lambda^\circ$, lo cual, según (4.44) y su análogo para un proceso asíncrono, implica que

$$\omega_{-1} \quad \omega_1 \quad \omega_2 \quad (4.45)$$

Las experimentaciones demostraron que con este esquema el algoritmo no converge. La causa de esto se explica como sigue. El jacobiano del NR en la resolución del problema local (4.42) en el procesador 1 es:

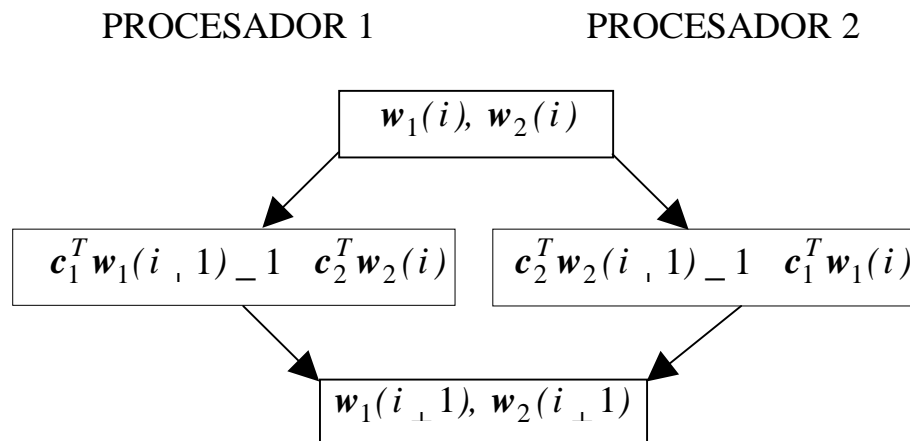
$$J_1 = \begin{pmatrix} F_{1x_1} & F_\lambda & 0 \\ F_{1x_1}^T w_1 & F_{2x_1}^T w_2 & 0 \\ 0 & 0 & c_1^T \end{pmatrix} \quad (4.46)$$

En una determinada iteración i , el algoritmo de NR consiste en resolver el sistema lineal de ecuaciones

$$J_1(i) \begin{pmatrix} \Delta x_1(i) \\ \Delta \lambda_1(i) \\ \Delta w_1(i) \end{pmatrix} = \begin{pmatrix} F_1(i) \\ F_{1x_1}^T w_1(i) + F_{2x_1}^T w_2(i) \\ c_1^T w_1(i) + c_2^T w_2(i) - 1 \end{pmatrix} \quad (4.47)$$

y actualizar las variables correspondientes.

Considerando la última fila de (4.47) y su análoga para el procesador 2, se obtiene el comportamiento de los autovectores en el proceso paralelo síncrono (ver figura 4.2).



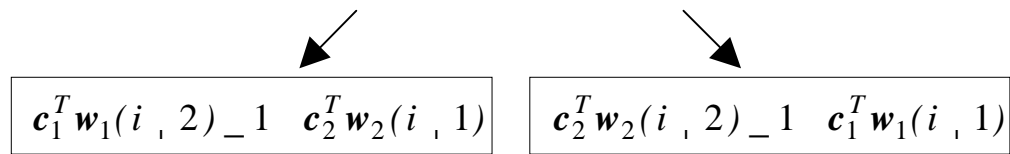


Figura 4.2 Comportamiento de los autovectores en la implementación síncrona

De la figura 4.2 se deduce que:

$$c_1^T w_1(i, 2) - c_1^T w_1(i) - cte \quad c_2^T w_2(i, 2) - c_2^T w_2(i) - cte$$

lo cual impone una restricción adicional sobre las posibles direcciones del autovector w , respecto a la ecuación original (4.38). Como consecuencia de esto, el algoritmo paralelo correspondiente al segundo esquema no converge. A una conclusión similar se puede llegar respecto a la versión asíncrona.

4.6.3 Tercer Esquema

Una modificación que salva esta dificultad y que permite buenos resultados experimentales (ver capítulo 5, tabla 5.2) es reemplazar la última ecuación de (4.40) por :

$$c^T w - \kappa \lambda - \xi = 0 \quad (4.48)$$

con κ y ξ constantes, lo que sigue garantizando una norma distinta de cero para el autovector w , sin la restricción adicional comentada. Introduciendo esta modificación en las ecuaciones (4.40), (4.42) y (4.43) se obtiene:

$$\Phi \begin{bmatrix} F_1 \\ F_2 \\ F_{1x_1}^T w_1 + F_{2x_1}^T w_2 \\ F_{1x_2}^T w_1 + F_{2x_2}^T w_2 \\ c_1^T w_1 + c_2^T w_2 + \kappa \lambda - \xi \end{bmatrix} = \mathbf{0} \quad (4.49)$$

$$\Phi_1 = \begin{bmatrix} F_1 \\ F_{1x_1}^T w_1 + F_{2x_1}^T w_2 \\ c_1^T w_1 + c_2^T w_2 + \kappa \lambda - \xi \end{bmatrix} = \mathbf{0} \quad (4.50)$$

$$\Phi_2 = \begin{bmatrix} F_2 \\ F_{1x_2}^T w_1 + F_{2x_2}^T w_2 \\ c_1^T w_1 + c_2^T w_2 + \kappa \lambda - \xi \end{bmatrix} = \mathbf{0} \quad (4.51)$$

El tercer esquema puede ser representado por la figura 4.1 y su proceso iterativo síncrono por las ecuaciones (4.44), aunque en este caso los operadores G_1 y G_2 (algoritmo NR) se aplican a las ecuaciones (4.50) y (4.51) respectivamente.

Por otro lado, el proceso iterativo asíncrono puede ser representado matemáticamente por las siguientes ecuaciones:

$$\begin{aligned} & \lambda_1(i, 1) = \omega \lambda_1(i) + \omega_1 \lambda_1(i) + \omega_2 \lambda_2(d_2^1(i)) \\ & \begin{bmatrix} x_1^T(i, 1) & w_1^T(i, 1) & \lambda_1(i, 1) \end{bmatrix} = G_1 \begin{bmatrix} x^1(i), w^1(i), \lambda_1(i, 1) \end{bmatrix} \end{aligned} \quad (4.52)$$

$$\begin{aligned} & \lambda_2(i, 1) = \omega \lambda_2(i) + \omega_1 \lambda_1(d_1^2(i)) + \omega_2 \lambda_2(i) \\ & \begin{bmatrix} x_2^T(i, 1) & w_2^T(i, 1) & \lambda_2(i, 1) \end{bmatrix} = G_2 \begin{bmatrix} x^2(i), w^2(i), \lambda_2(i, 1) \end{bmatrix} \end{aligned}$$

donde $d_j^m(i)$ es la iteración correspondiente a un subvector x_j (o w_j), utilizado en el procesador m en la iteración i de este último. G_1 y G_2 se aplican a las ecuaciones (4.50) y (4.51) respectivamente.

El TA asíncrono, ilustrado en la figura 4.1 y expresado por (4.52), funciona de la siguiente forma: el procesador 1 recibe $x_2(d_2^1(i))$, $w_2(d_2^1(i))$ y $\lambda_2(d_2^1(i))$. El administrador de este procesador calcula $\lambda_1(i, 1)$, valor que el operador G_1 utiliza para actualizar x_1 , w_1 y λ_1 . Posteriormente, este procesador transmite los valores actualizados al procesador 2, el cual opera en forma análoga.

5. RESULTADOS EXPERIMENTALES

Este capítulo presenta resultados experimentales obtenidos al implementar los métodos de solución del problema del *Punto de Colapso*. Abarca resultados correspondientes a los métodos expuestos en el capítulo 3 y a las nuevas propuestas presentadas por esta obra en el capítulo 4.

En la sección 5.1 se describe la plataforma computacional empleada para las experimentaciones. Los problemas utilizados para probar los distintos métodos se plantean en la sección 5.2.

La sección 5.3 presenta curvas obtenidas a partir de experimentaciones con el Método de Continuación para diferentes problemas.

La sección 5.4 presenta resultados obtenidos con los métodos convencionales, el MC y el PoC, así como los logros alcanzados al incorporar la Variante Aceleradora al PoC.

Las ventajas que pueden resultar de utilizar *overlapping* parcial al resolver problemas mediante el PoC en un sistema de procesamiento distribuido, quedan demostradas experimentalmente en la sección 5.5, todavía utilizando programas secuenciales.

En la siguiente sección, la 5.6, ya se presentan los resultados obtenidos con el PoC en un sistema de procesamiento distribuido.

5.1 Ambiente Computacional

Los resultados experimentales presentados en este capítulo fueron obtenidos utilizando un sistema distribuido con dos servidores:

DEC 3000 modelo 300
 Procesador: ALPHA 150 Mhz
 Memoria RAM: 32 MB
 SO: OSF/1 Versión 2.0

y

SUN SPARC Station 5
 Procesador: 66 Mhz
 Memoria RAM: 32 MB
 SO: Solaris 5.3

Las implementaciones fueron realizadas en PVM (*Parallel Virtual Machine*) en su versión para lenguaje C. Se utilizó una red Ethernet de 10 Mbps con protocolo de comunicación TCP/IP para interconectar las *workstations*.

Las mediciones correspondientes a los métodos secuenciales fueron realizadas en el procesador mas rápido, en este caso la DEC 3000.

5.2 Problemas de Prueba

Las implementaciones fueron evaluadas por medio de la resolución de los siguientes problemas tipo:

- Problemas tipo de la IEEE (IEEE 5 y IEEE 14)
- Problema de 16 barras creado a partir de parámetros reales de la red eléctrica paraguaya (PAR 16)
- Problema de 26 barras creado a partir de parámetros reales de la red eléctrica paraguaya (PAR 26)
- El *standard* de Monticelli de 30 barras (IEEE 30)
- Un problema ejemplo de 88 barras, creado a partir del *standard* de Monticelli, al que se denominó Prob 88.

En el Prob 88, el original IEEE 30 sin su barra *slack* fue triplicado y los sistemas obtenidos fueron interconectados compartiendo una barra *slack* como se ve en la figura 5.1. Las líneas de interconexión entre los subsistemas 1 y 2 se caracterizan por tener impedancias bajas, mientras que las líneas que unen el subsistema 3 con los otros dos tienen impedancias altas.

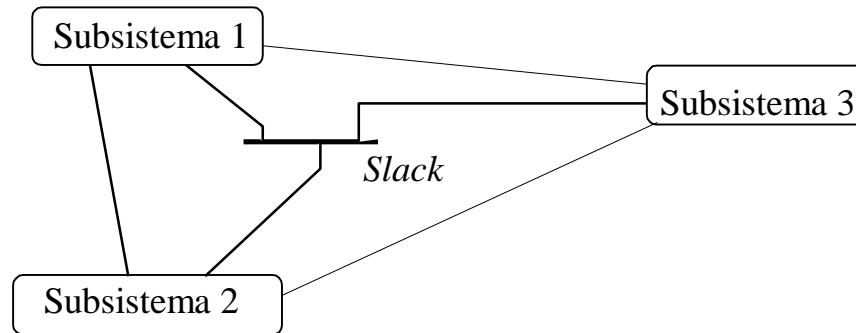


Figura 5.1 Problema Ejemplo de 88 barras

Es importante recalcar que para la resolución eficaz de problemas en un ambiente distribuido heterogéneo como el descrito en la sección 5.1, debe tenerse en cuenta el balanceamiento del trabajo que se le asigna a cada procesador y el desacople que puedan tener los subproblemas entre sí [30].

Un buen balanceamiento del trabajo se consigue asignando a los computadores subproblemas proporcionales a sus capacidades de procesamiento.

Un sistema de ecuaciones presenta un buen desacople cuando las variables correspondientes a los distintos subproblemas dependen muy poco unas de otras. Por ejemplo, una red eléctrica compuesta por dos zonas interconectadas por líneas de transmisión largas, está representada matemáticamente por un sistema de ecuaciones que es posible particionar para obtener dos subproblemas desacoplados.

No siempre los desacoples naturales de las redes eléctricas permiten obtener particiones que puedan además ser bien balanceadas en el sistema de procesadores disponibles. Elegir un criterio óptimo de partición implica un compromiso entre desacople y balanceamiento de carga.

El estudio detallado de los métodos que permiten obtener una buena partición para un determinado problema está fuera del alcance de este trabajo.

En esta obra la partición será considerada como dato del problema. Así, para el problema ejemplo de 88 barras, los subproblemas quedan distribuidos en dos procesadores como se indica en la figura 5.2.

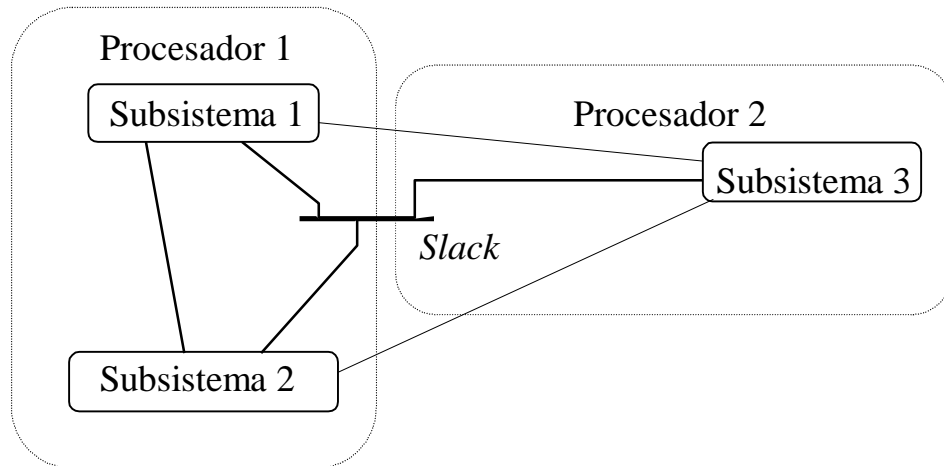


Figura 5.2 Partición del Problema Ejemplo de 88 barras

5.3 Curvas obtenidas con el MC

El objetivo de esta sección es presentar las curvas obtenidas a partir de la implementación secuencial del MC y señalar algunas de sus particularidades.

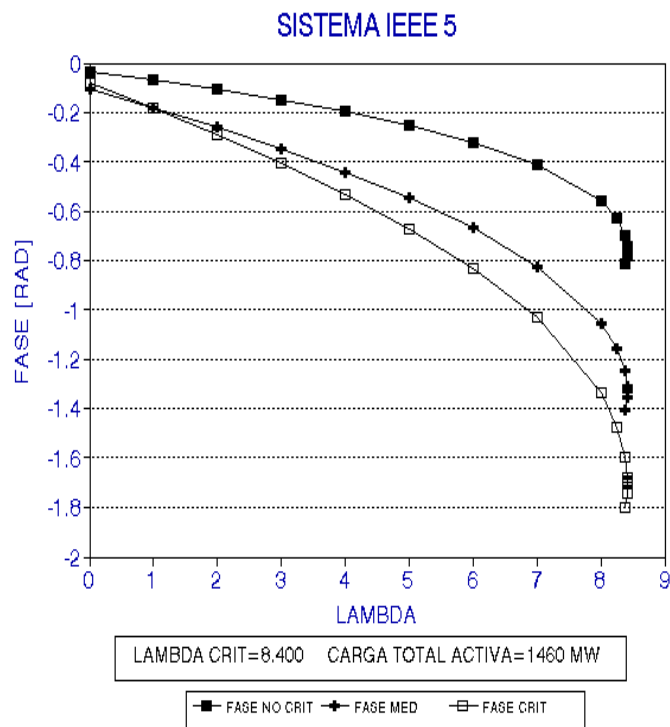


Figura 5.3 Problema IEEE 5. Crecimiento de cargas 'A'

En la figura 5.3, referida al sistema IEEE 5 con una dada distribución de cargas ‘A’, el punto de colapso corresponde a $\lambda = 8.4$ con una Carga Total Activa de 1460 MW. En la figura 5.4, referida también al sistema IEEE 5, pero con otro esquema de crecimiento de cargas ‘B’, se observa que el punto de colapso corresponde a $\lambda = 7.556$ con una Carga Total Activa de 1432 MW.

Las figuras 5.3 y 5.4 confirman entonces lo expuesto en la sección 3.1 e ilustrado en la figura 3.1, es decir, la existencia de diferentes puntos de colapso para direcciones distintas del vector unitario χ .

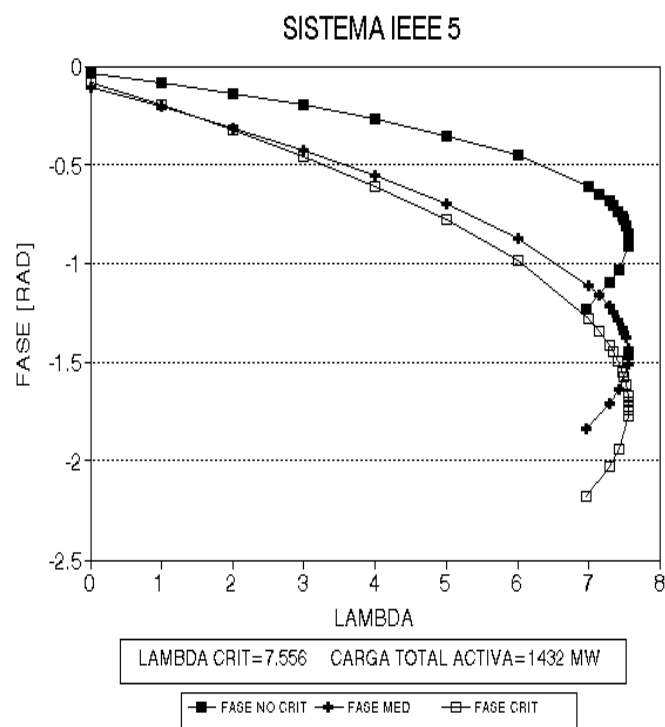


Figura 5.4 Problema IEEE 5. Crecimiento de cargas ‘B’

Dado que el sistema IEEE 5 tiene una barra *slack* y 4 barras PV, el parámetro de continuación, es decir, la magnitud que presenta mayor variación, corresponde necesariamente a la fase de alguna de sus barras PV. En las figuras 5.3 y 5.4, a dicha magnitud se denomina fase crítica.

Las figuras 5.5 y 5.6, correspondientes al sistema IEEE 14, muestran el comportamiento de las fases y módulos de tensión de las barras con el crecimiento de la carga.

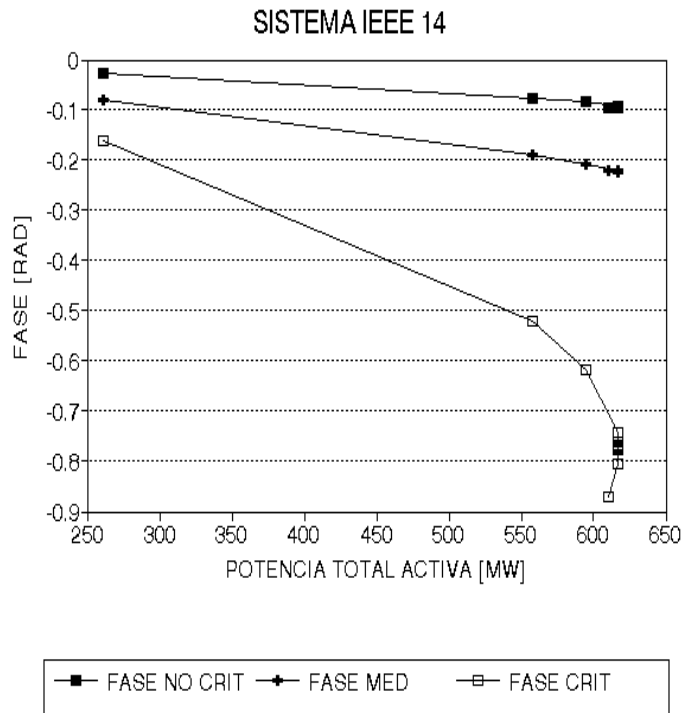


Figura 5.5 Problema IEEE 14. Fases de tensión

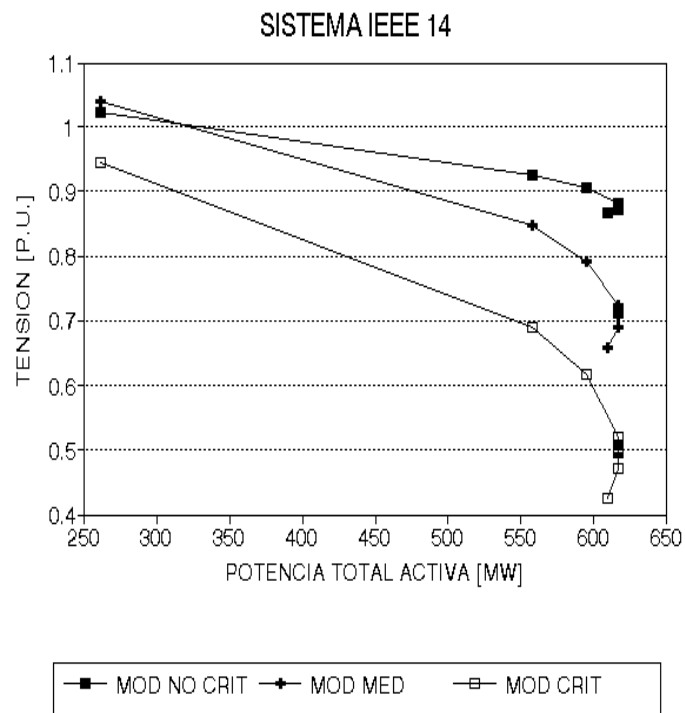


Figura 5.6 Problema IEEE 14. Módulos de tensión

Puede notarse en las figuras 5.5 y 5.6 que, para valores de carga superiores a 623 MW, no existe solución de estado estable para las ecuaciones del flujo de carga. Este valor corresponde al punto de colapso del sistema.

Las curvas obtenidas a partir del sistema PAR 16, construido con parámetros reales de la red eléctrica paraguaya, se presentan en las figuras 5.7 y 5.8.

Se observó de las experimentaciones que, de acuerdo a lo expresado en [3], cuando se parte de una condición inicial base lejana al colapso de tensión, el parámetro de continuación es λ para los primeros puntos encontrados sobre la curva. A medida que la carga se incrementa y el algoritmo se acerca al punto crítico, la magnitud y fase de las tensiones comienzan a experimentar cambios significativos, el parámetro de continuación no permanece fijo y la barra más propensa al colapso puede ir cambiando en cada paso.

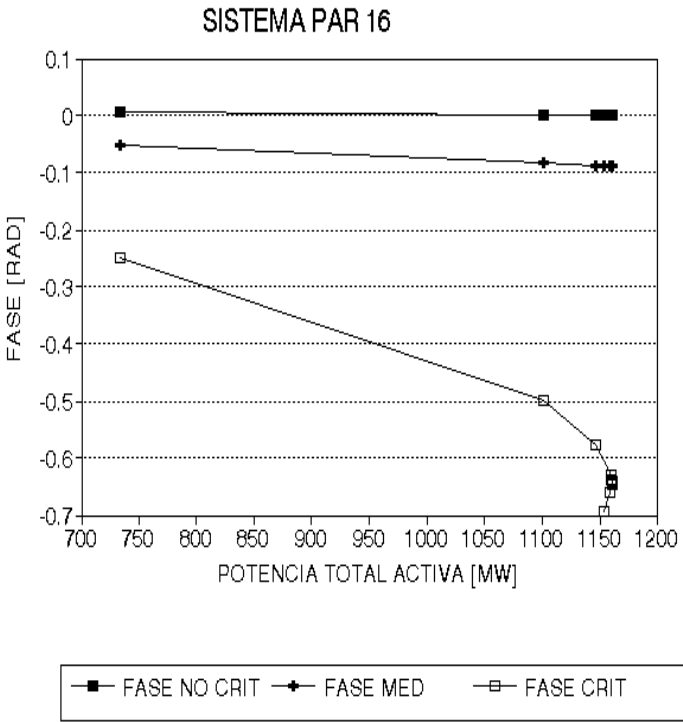


Figura 5.7 Problema PAR 16. Fases de tensión

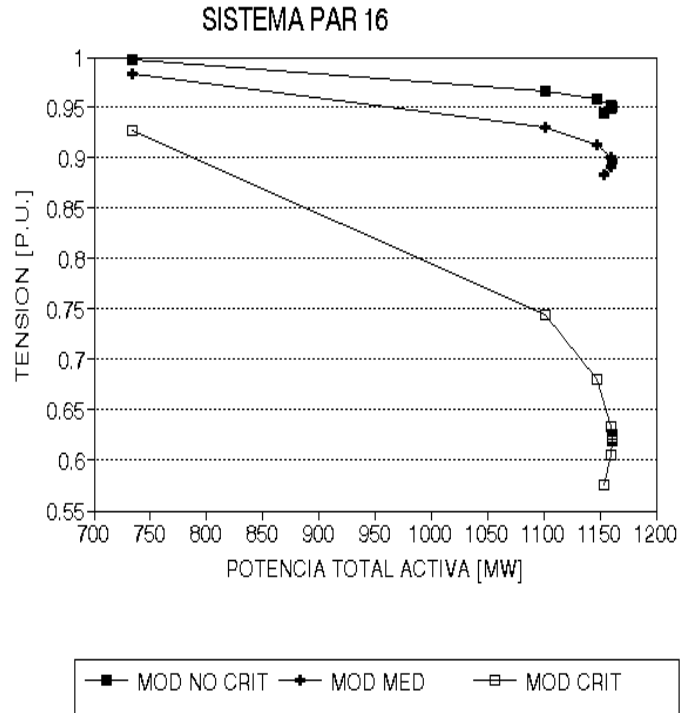


Figura 5.8 Problema PAR 16. Módulos de tensión

Las curvas obtenidas a partir del MC son muy descriptivas acerca del comportamiento del sistema dada una dirección de crecimiento de cargas. Permiten además obtener índices de estabilidad del sistema, así como información relativa a las regiones más propensas a experimentar el colapso de tensión [4].

5.4 Comparación entre Diferentes Métodos Secuenciales

La Tabla 5.1 presenta los resultados experimentales obtenidos con el Método de Continuación, el Método del Punto de Colapso y el Método del Punto de Colapso con Variante Aceleradora, aplicados a los problemas de prueba de la sección 5.2.

Nota 5.1 Es importante aclarar que las mediciones de tiempo en esta obra no incluyen la entrada y salida de datos.

Programa	Converge	Iteraciones *	Tiempo [s]
Sistema IEEE 5			
MC	sí	19	0.05
PoC	sí	6	0.03
PoC con Variante	sí	6	0.12
Sistema IEEE 14			
MC	sí	15	0.55
PoC	sí	6	0.43
PoC con Variante	sí	6	0.27
Sistema PAR 16			
MC	sí	11	0.60
PoC	sí	7	0.68
PoC con Variante	sí	7	0.38
Sistema PAR 26			
MC	sí	14	1.83
PoC	no	-	-
PoC con Variante	no	-	-
Sistema IEEE 30 (Monticelli)			
MC	sí	19	3.30
PoC	sí	8	4.65
PoC con Variante	sí	8	1.65
Prob 88			
MC	sí	19	72.21
PoC	sí	10	166
PoC con Variante	sí	10	35.74

* En el MC se especifica el número de puntos encontrados, mientras que en el PoC se menciona el número de iteraciones del NR.

Tabla 5.1 Comparación entre métodos secuenciales

El PoC con Variante aceleradora resultó ser el más rápido de entre los tres métodos considerados, como se muestra en la Tabla 1. El algoritmo converge en el mismo número de iteraciones que el PoC con un ahorro importante de memoria y tiempo de procesamiento.

Puesto que las ventajas del PoC con Variante Aceleradora fueron demostradas para sistemas de gran porte, era de esperarse que en el caso de sistemas pequeños como el IEEE 5, dichas ventajas no se presenten.

En la figura 5.9, construida a partir de los datos de la tabla 5.1, se ilustra la ventaja potencial de la *Variante* propuesta en la medida que aumenta el número de barras del sistema.

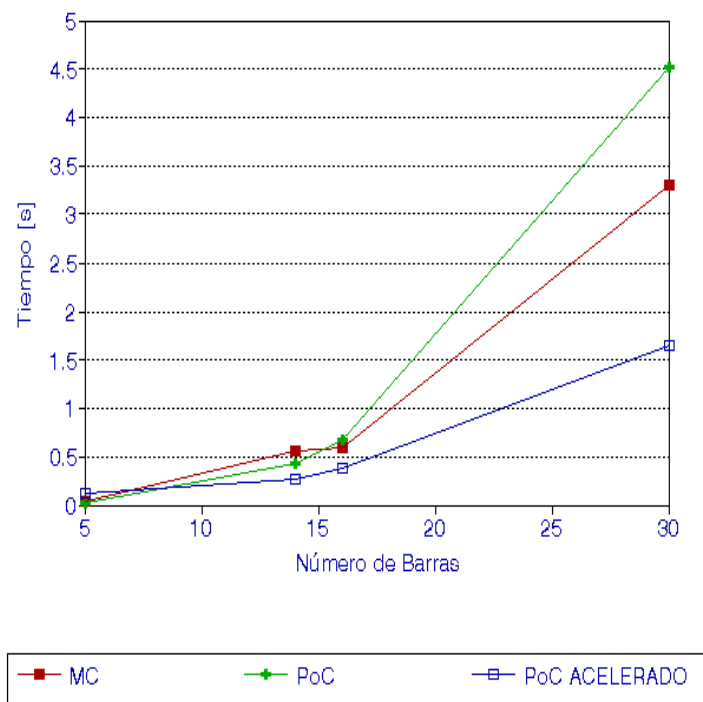


Figura 5.9 Comparación entre métodos secuenciales

Los resultados experimentales muestran que, inclusive para sistemas relativamente pequeños, el PoC presenta tiempos de procesamiento mayores que el MC. Esta característica se va acentuando a medida que aumenta el número de barras.

En cambio, Cañizares y Alvarado en [4] llegan a la conclusión de que, en general, el tiempo de procesamiento empleado por el PoC es la mitad del requerido por el MC.

Esta discrepancia se debe al hecho de que las implementaciones presentadas en esta obra no recurren al uso de técnicas de esparsidad como las utilizadas por Cañizares y Alvarado. Se verifica entonces la necesidad de utilizar dichas técnicas para que el PoC presente las ventajas de tiempo que se le atribuyen en [4].

Se espera que al aplicar estas técnicas a los tres métodos considerados en esta sección, el PoC con Variante siga presentando los menores tiempos de procesamiento.

5.5 Ventajas del *Overlapping*

Para determinar cuál de los tres esquemas presentados en la sección 4.5 sería el más conveniente a desarrollar en las implementaciones paralelas, se elaboraron programas secuenciales conforme al primer y tercer esquema, ya que el segundo se demostró analíticamente que no converge.

Estos programas son:

- **TA secuencial sin *overlapping*:** método iterativo que resuelve cada una de las ecuaciones (4.41 a) y (4.41 b) por el método de NR. Esto es:

Estimar valores iniciales para x_1 , x_2 , w_1 , w_2 y λ

Verificar si converge

DO WHILE (no converge)

 Begin

 Actualiza x_1 , w_1 y λ resolviendo la ecuación (4.41 a) por NR, utilizando x_2 y w_2 estimados inicialmente o calculados en la iteración anterior

 Actualiza x_2 y w_2 resolviendo la ecuación (4.41 b) por NR, utilizando x_1 , w_1 y λ del cálculo anterior

 Verificar si converge

 End

- **TA secuencial con *overlapping***: método iterativo que resuelve cada una de las ecuaciones (4.50) y (4.51) por el método de NR. Esto es:

Estimar valores iniciales para x_1 , x_2 , w_1 , w_2 y λ

Verificar si converge

DO WHILE (no converge)

Begin

Actualiza x_1 , w_1 y λ resolviendo la ecuación (4.50) por NR, utilizando x_2 , w_2 y λ estimados inicialmente o calculados en la iteración anterior

Actualiza x_2 , w_2 y λ resolviendo la ecuación (4.51) por NR, utilizando x_1 , w_1 y λ del cálculo anterior

Verificar si converge

End

En la tabla 5.2 se pueden apreciar claramente las ventajas del *overlapping* parcial. En consecuencia, para las implementaciones en sistemas distribuidos, se recurrió al uso de las ecuaciones propuestas en el tercer esquema.

Método	IEEE 14		IEEE 30		Prob 88	
	Iter	T[s]	Iter	T[s]	Iter	T[s]
TA secuencial <i>s/overlapping</i>	No converge		No converge		No converge	
TA secuencial <i>c/overlapping</i>	20	0.66	13	2.92	23	127

Tabla 5.2 Resultados que justifican el uso del *overlapping*

5.6 Sistemas Distribuidos

En esta sección se presentan los resultados obtenidos al experimentar con el PoC en un sistema de procesamiento distribuido.

5.6.1 Métodos Implementados

Los programas que fueron elaborados para funcionar en sistemas distribuidos son:

- **TA síncrono (con overlapping):** resolución paralela de las ecuaciones (4.50) y (4.51) en forma sincronizada, intercambiando los valores actualizados después de cada iteración. El proceso iterativo síncrono está ilustrado en la figura 4.1 y representado matemáticamente por las ecuaciones (4.44) con G_1 y G_2 aplicados a (4.50) y (4.51) respectivamente.
- **TA asíncrono (con overlapping):** resolución paralela de las ecuaciones (4.50) y (4.51) en forma asíncrona. El proceso iterativo está ilustrado en la figura 4.1 y representado matemáticamente por las ecuaciones (4.52).

5.6.2 Criterios de Comparación

Con la finalidad de facilitar la presentación de los resultados experimentales correspondientes a las implementaciones paralelas realizadas, se definirán dos parámetros de comparación.

Una de las figuras de mérito más utilizadas para comparar implementaciones paralelas y secuenciales es la ‘aceleración’ (o *speedup* S_p) que puede ser definida de diferentes maneras. En este trabajo se utilizará la definición de Stone [38]:

La aceleración o ‘speedup’ S_p está dada por la relación entre el menor tiempo secuencial de entre todos los métodos existentes, y el tiempo de ejecución del algoritmo considerado en un sistema distribuido.

$$S_p = \frac{\text{mejor tiempo secuencial de entre todos los metodos existentes}}{\text{tiempo de la ejecucion paralela considerada}}$$

La definición anterior se muestra ambigua para fines prácticos (Stone, 1987), pues no es posible implementar todos los métodos secuenciales que existen para conocer ‘el mejor tiempo secuencial’. Sin embargo, será utilizada

en el sentido de considerar el mejor tiempo de entre todas las implementaciones realizadas en el contexto secuencial en esta obra. El concepto de ‘speedup’ es utilizado para representar la ganancia como consecuencia de la utilización de varios procesadores.

Considerando la definición de aceleración es posible utilizar un tiempo secuencial bien específico, en lugar del ambiguo ‘mejor tiempo secuencial’. Es así que en muchos casos el *speedup* es definido como la relación de tiempo entre las ejecuciones secuenciales y paralela del mismo algoritmo. Para el caso específico de los TA que combinan métodos diferentes, podemos definir la aceleración relativa de un TA (A_r) considerando como ‘mejor tiempo secuencial’ al menor tiempo de entre todos los algoritmos que son combinados, cuando se utilizan individualmente para resolver el problema global, en un contexto secuencial. Es decir:

La aceleración relativa A_r de un TA que combina diferentes algoritmos es dada por la relación entre el mejor tiempo secuencial de entre todos los métodos combinados, cuando se utilizan individualmente para resolver el problema global, y el tiempo de ejecución del TA considerado, en un sistema distribuido.

$$A_r = \frac{\text{mejor tiempo secuencial de entre los metodos combinados por un TA}}{\text{tiempo de la ejecucion paralela del TA considerado}}$$

La aceleración relativa de un TA, arriba definida, es una figura de mérito apropiada para representar la ganancia al utilizar un TA en un contexto distribuido.

Nota 5.2 Las implementaciones paralelas y secuenciales deben ser comparadas en igualdad de condiciones, por tanto, para los cálculos de *speedup* y aceleración relativa, no se considerarán aquellas implementaciones que incluyan en sus procesos técnicas de esparsidad o la Variante aceleradora propuesta en esta obra.

Dada las definiciones anteriores se establece cuanto sigue:

- Para el cálculo del *speedup* deberá utilizarse como mejor tiempo secuencial el del TA secuencial con *overlapping*.
- Para el cálculo de la aceleración relativa deberá emplearse el tiempo correspondiente al PoC, puesto que las implementaciones paralelas realizadas utilizan dicho método en ambos procesadores.

5.6.3 Resultados Experimentales

En la tabla 5.3 se observan los resultados obtenidos al resolver el problema IEEE 30 de Monticelli. Los mejores tiempos fueron obtenidos con $\kappa=3$. Se puede apreciar que, si bien los métodos paralelos resuelven correctamente el problema, en este caso no presentan ventajas de tiempo en comparación con el mejor método secuencial.

Método	$\kappa = 3$			
	Iter	T[s]	S_p	A_r
PoC *	8	4.65	-	1
TA Secuencial	13	3	1	1.55
TA síncrono	24	7	0.43	0.66
TA asíncrono	49	7	0.43	0.66

* en este método no interviene κ , o se puede considerar $\kappa = 0$

Tabla 5.3 Comparación entre los métodos paralelos y secuenciales al resolver el problema IEEE 30

En la tabla 5.4 se observan los resultados obtenidos al resolver el problema ejemplo de 88 barras. Los mejores tiempos se obtienen con valores de κ comprendidos entre 2 y 3.5.

Método	$\kappa=2$				$\kappa=3$				$\kappa=3.5$			
	Iter	T[s]	S_p	A_r	Iter	T[s]	S_p	A_r	Iter	T[s]	S_p	A_r
PoC *	10	166	-	1								
TA secuencial	23	127	1	1.31	no converge				no converge			
TA síncrono	no converge				21	99	1.30	1.68	14	66	1.92	2.52
TA asíncrono	77	254	0.5	0.65	36	100	1.27	1.66	21	76	1.67	2.18

* en este método no interviene κ , o se puede considerar $\kappa = 0$

Tabla 5.4 Comparación entre los métodos paralelos y secuenciales al resolver el problema ejemplo de 88 barras.

En la tabla 5.4 se puede apreciar que el uso del paralelismo permite disminuir en forma notable el tiempo necesario para la resolución del problema considerado. Así, el mejor *speedup* que se consigue es 1.92 con una aceleración relativa de 2.52.

El valor de *speedup* demuestra que en este caso la implementación síncrona logra ser 1.92 veces más rápida que el mejor algoritmo secuencial, mientras que el valor de aceleración relativa demuestra que la implementación síncrona logra ser 2.52 veces más rápida que el PoC. Se confirma entonces que las implementaciones paralelas presentan, a pesar del mayor número de iteraciones, tiempos menores que el PoC.

Aunque se implementaron versiones síncronas y asíncronas, como se muestra en la tabla 5.4, no fue posible notar las ventajas del asincronismo conforme a lo expuesto en [30]. Esto es debido a que en las experimentaciones se utilizaron sólo dos procesadores, lo que no permite un nivel suficiente de asincronismo.

Como se citó en la sección 5.2, es necesario que los problemas puedan estar bien particionados para su eficaz resolución en un ambiente distribuido. La partición del problema IEEE 30 no es buena, es decir, no presenta un buen desacople ni tampoco un buen balanceamiento. El problema ejemplo de 88 barras, por el contrario, presenta una buena partición.

El ejemplo de Monticelli sirve para demostrar que aunque no se puedan particionar bien ciertos problemas, de todas formas se los puede resolver con sistemas distribuidos existentes en el país. Entonces, si este tipo de problemas a la vez es de gran porte, podrán ser resueltos sin el empleo de supercomputadores no disponibles a nivel local. Esto constituye una de las principales ventajas de la propuesta hasta aquí presentada.

El ejemplo de 88 barras sirve para demostrar que, en la medida en que se consiga un buen balanceamiento y la red eléctrica presente un buen desacople, las implementaciones paralelas, además de resolver los problemas, lo hacen en tiempos menores, lo que constituye otra ventaja de esta propuesta.

Se podría pensar entonces que esta última ventaja sólo se conseguiría con las condiciones establecidas en el párrafo anterior. Esta conclusión no es necesariamente cierta, ya que está basada en resultados de experimentaciones realizadas con sólo dos procesadores. Lastimosamente, hoy no se cuenta con una red de *workstations* con mayor número de procesadores, de forma a realizar una experimentación más exhaustiva.

6. CONCLUSIONES

La presente obra ha intentado relacionar dos áreas de investigación. La primera de ellas es el problema del punto de colapso de un sistema eléctrico de potencia, y la segunda es el procesamiento paralelo y distribuido utilizando TA.

La importancia del estudio del punto de colapso se ha ido incrementando en la medida en que los sistemas eléctricos se expandían y los problemas de estabilidad de tensión se hacían más frecuentes y de mayor magnitud.

La evolución que ha tenido el estudio del punto de colapso se ha presentado en el resumen histórico de esta obra. En la actualidad, una de las líneas de investigación asocia el fenómeno del colapso con la carga máxima que el sistema puede soportar. Esta obra se enmarca dentro de ese contexto.

A su vez, dentro de dicha línea de investigación, se puede encontrar gran variedad de trabajos. Esta obra centró su atención en los trabajos que desarrollaron herramientas de cálculo, y en particular en aquéllos que desarrollaron técnicas cuya metodología sugería su posible implementación en un ambiente computacional distribuido.

En cuanto al procesamiento distribuido, el mismo ha venido realizando progresos importantes en lo que se refiere al desarrollo de tecnologías de bajo costo y alta capacidad de procesamiento, de forma tal que hoy en día la tendencia mundial es manifiesta hacia la utilización de máquinas interconectadas entre sí a través de redes de comunicación. Sin embargo, como se acotó en el capítulo 1, uno de los problemas por los que atraviesa esta tendencia innovadora, es la carencia de *softwares* que exploten sus ventajas potenciales.

La intención de la obra es entonces aprovechar las ventajas que presentan los sistemas de procesamiento distribuido para la resolución eficiente del problema del punto de colapso.

Como las técnicas de cálculo del punto de colapso se basan en conceptos de flujo de potencia eléctrica, este tema fue abordado en el capítulo 2.

Los principales métodos utilizados para la detección del punto de colapso, el MC y el PoC, se describieron en el capítulo 3, y los resultados de compararlos experimentalmente, en el capítulo 5. Con respecto a dichos métodos se pudo notar que:

- El MC presenta mejor convergencia que el PoC, ya que este último es bastante sensible a las estimaciones iniciales de las variables, como pudo observarse en las experimentaciones (problema PAR 26 de la tabla 5.1).
- Aunque las comparaciones de tiempo realizadas por Cañizares y Alvarado muestran al PoC como más rápido que el MC, con una relación de tiempo de 2 a 1 aproximadamente, esto no pudo ser comprobado en las experimentaciones. Es muy posible que la discrepancia presentada sea a causa de que en las implementaciones realizadas no fueron utilizadas técnicas de esparsidad como las empleadas por Cañizares y Alvarado.
- Las curvas obtenidas con el MC son muy descriptivas acerca del comportamiento de las tensiones a medida que la carga aumenta y la precisión de las curvas levantadas puede ser fácilmente controlada por el valor del paso (sección 3.2.1).
- Ambos, el PoC y el MC, son útiles para determinar zonas propensas al colapso de tensión, así como para determinar índices de estabilidad del sistema de potencia. Sin embargo, los índices que proporcionan son diferentes y en cierto sentido se complementan.
- Ambos métodos presentan información complementaria para el estudio del punto de colapso. La complementación es especialmente útil en los casos en que uno de ellos resuelve el problema y el otro no.

Esta obra propone dos nuevas alternativas para el estudio del punto de colapso, las mismas fueron expuestas en el capítulo 4. La primera de ellas es una variante aceleradora del PoC. La misma permite ahorrar tiempo de procesamiento sin perder ninguna de las propiedades del PoC. Como se

observa en el tabla 5.1 , los tiempos obtenidos en comparación con el PoC y el MC justifican su utilización.

La segunda propuesta es la que materializa la principal intención de esta obra, combinando el procedimiento resolutorio del PoC con las características del procesamiento distribuido. Se postularon tres esquemas de solución posibles, pero sólo uno de ellos, el que utiliza el concepto de *overlapping* parcial y reformula las ecuaciones originales, presentó los resultados esperados.

Al comparar los métodos secuenciales y paralelos fue posible llegar a las siguientes conclusiones:

- El uso del paralelismo en varios casos permite disminuir en forma notable el tiempo necesario para la resolución del problema considerado.
- Se demuestra la ventaja de la paralelización utilizando *overlapping* parcial con relación a la paralelización sin *overlapping* cuando existen ecuaciones cuyas variables corresponden a distintos bloques de partición.
- Si bien se esperaba que las versiones asíncronas presenten ventajas sobre las síncronas, esto no pudo ser comprobado, debido a que en las implementaciones realizadas se recurrió al empleo de sólo dos procesadores.
- Se demuestra la necesidad de realizar un buen particionamiento en el caso de implementaciones paralelas, caso contrario no se obtienen aceleraciones adecuadas e inclusive el problema puede no converger [30].

Se consideran los dos puntos siguientes como los principales aportes de esta obra:

1. *Es la primera obra que resuelve en forma eficiente el problema del punto de colapso con más de un procesador.*
2. *Demuestra la posibilidad de resolver en forma eficiente problemas de gran porte en un sistema distribuido existente en nuestro país.*

Para finalizar, se proponen algunos temas de forma a continuar con los trabajos hasta aquí realizados. Estos temas se enmarcan en dos grupos. El

primero de ellos está relacionado con la computación distribuida; y el segundo, con el punto de colapso.

Los temas relacionados con la computación distribuida son:

- La aplicación de los conceptos de *Team Algorithms* a otros problemas de ingeniería.
- Estudio de los métodos de partición óptima y automática de ecuaciones.
- Realizar implementaciones que involucren más de dos procesadores.

Los temas relacionados con el punto de colapso son:

- Desarrollo de *softwares* que permitan el cálculo del punto de colapso más cercano.
- Estudio de índices de estabilidad que puedan ser aplicados a sistemas eléctricos en general.
- Estudio de las medidas más eficaces para evitar el colapso de tensión, y en el caso de producirse un estado de inestabilidad, como mitigar de la mejor manera sus efectos.
- Desarrollo de implementaciones que permitan el análisis en tiempo real de los sistemas eléctricos actuales.

Lista de Referencias Bibliográficas

- [1] P. Kundur, B. Gao, “Practical Considerations in Voltage Stability Assessment”, *IV Symposium of Specialist in Electric Operational and Expansion Planning*. Mayo 1994.
- [2] J. Arrillaga, C.P. Arnold, B. J. Harker, *Computer Modelling of Electrical Power Systems*, John Wiley & Sons, UK. 1983.
- [3] V. Ajjarapu, C. Christy, “The Continuation Power Flow: A Tool for Steady State Voltage Stability Analysis”, *IEEE Transaction on Power Systems*, Vol. 7, N° 1, pg 416 - 423. Febrero 1992.
- [4] C. Cañizares, F. Alvarado, “Point of Collapse and Continuation Methods for Large AC/DC Systems”, *Transaction on Power System*, Vol. 8, N° 1, pg 1-8. Febrero 1993.
- [5] C. Barbier, J. P. Barret, “Analysis of Phenomena of Voltage Collapse on a Transmission Systems”, *Revue Generale de l’Electricité*, Vol. 89, pg 672 - 690. Octubre 1980.
- [6] G. D. Friedlander, “The North East Power Failure”, *Spectrum*, Vol. 3, N° 2, pg. 54-73. Febrero 1966.
- [7] Federal Power Commission Reports, “Prevention of Power failures”, *Studies of the Task Force on the North East Power Interruption*, Vol. 3. 1967
- [8] J. A. Casalla, “Interim Report on the French Blackout of December 19, 1978”, *Report to U.S. Department of Energy*, Office of Utility Systems. Febero 1979.
- [9] W. R. Lachs, “Voltage Collapse in EHV Power Systems”, *IEEE PES Winter Meeting*, New York, paper A-78 057-2. Enero/Febrero 1978.
- [10] W. R. Lachs, “Countering Calamitous System Disturbances”, *International Conference on reliability of Power Supply Systems*, *IEEE Publication N° 225*, pg 79-83, London. 1983.

- [11] W. A. Johnson “EHV Operating Problems Associated with Reactive Control”, *IEEE Trans.*, Vol. Pas-100, N° 3, pg. 1376-1381. Marzo 1981.
- [12] B. K. Johnson, “Extraneous and False Load Flow Solutions”, *IEEE Trans.*, Vol. PAS-96, N° 2, pg. 524-534. Marzo/Abril 1977.
- [13] T. Park, Y. Sekine, “A Method for Analysing Multisolution in Power Flow Analysis”, *Electrical Engineering in Japan*, Vol. 99, N° 2, pg. 95-103. 1979.
- [14] Y. Tamura, Y. Nakanishi, S. Iwamoto, “On the Multiple Solution Structure, Singular Point and Existence Conditions of the Multiple Load-Flow Solutions”, *IEEE PES Winter Meeting*, New York. Febrero 1980.
- [15] S. Chow, J. K. Hale, *Methods of Bifurcation Theory*, New York: Springer-Verlag. 1982.
- [16] A. Vanderbauwhede, *Local Bifurcation and Symmetry*, Boston: Pitman. 1982.
- [17] M. Golubitsky, D. G. Schaeffer, *Singularities and Groups in Bifurcation Theory*, New York: Springer-Verlag. 1985.
- [18] J. Medanic, M. I. Spong, J. Christensen, “Discrete Models of Slow Voltage Dynamics for Under Load Tap-Changing Transformer Coordinations”, *IEEE PES Summer Power Meeting*, Ciudad de Méjico. 1986.
- [19] M. I. Spong, J. Medanic, “Modeling and Control of Slow Voltage Dynamics in Electric Power Systems”, *IFAC Symposium on Large Scale System and Applications*, Switzerland. 1986.
- [20] P. Kundur, “Voltage Instability”, *IEEE Winter Power Meeting*, New York. 1988.
- [21] F. L. Alvarado, T. H. Jung, “Direct Detection of Voltage Collapse Condition”, *Engineering Foundation Conference Proc. on Bulk Power System Voltage Phenomena: Voltage Stability and Security*, Potosí. Setiembre 1988.

- [22] A. Semlyen, B. Gao, W. Janischewskyj, "Calculation of the Extreme Loading Condition of Power System for the Assessment of Voltage Stability", *IEEE PES Winter Meeting*. 1990.
- [23] O. Obadina, G. J. Berg, "Determination of Voltage Stability Limit in Multimachine Power Systems", *IEEE PES Winter Meeting*, New York, paper 88 WM 194-3. Febrero 1988.
- [24] G. J. Berg, "Assesment of Critical Voltage and Load Margins in VAR Compensated Power Transmission Systems", *Electric Power Systems Research*, Vol. 12, pg. 63-69. 1987.
- [25] C. S. Indulkar, B. Viswanathan, S. S. Venkata, "Maximun Power Transfer Limited by Voltage Stability in Series and Shunt Compensated Schemes for A.C. Transmission Systems", *IEEE PES Summer Meeting*, Portland, paper 88 SM 575-3. Julio 1988.
- [26] F.D. Gallana, Z.C. Zeng, "Analisy of the Load Flow Behaviour Near a Jacobian Singularity", *IEEE Transaction on Power Systems*, Vol. 7, N° 3, pg 1362 - 1369. Agosto 1992
- [27] J. Jumeau, Chiang H. "Parameterizations of the Load Flow Equations for Eliminating Ill-conditioning Load Flow Solutions", *IEEE Transaction on Power Systems*, Vol. 8, N° 3, pg 1004-1012. Agosto 1992
- [28] I. Dobson, L. Lu, "New Methods for Computing a Closest Saddle Node Bifurcation and Worst Case Load Power Margin for Voltage Collapse", *IEEE Transaction on Power Systems*, Vol. 8, N° 3, pg 905 - 913. Agosto 1993
- [29] F. Wu, L. Murphy, "Parallel and Distributed Processing: Applications to Power Systems", *IV Simposium of Specialists in Electric Operational and Expansion Planning*, Brasil. Mayo 1994.
- [30] B. Barán, *Estudo de Algoritmos Combinados Paralelos Assíncronos*, Tesis Doctoral COPPE/UFRJ, Rio de Janeiro, Brasil. Octubre 1993.
- [31] W. Stevenson, *Análisis de Sistemas Eléctricos de Potencia*, Segunda Edición, McGraw-Hill. 1975

- [32] B. Barán, W. Causarano, J.J. Achón, J. Atlasovich, C. Schaerer, “Cálculo del Punto de Colapso”, *1er. Seminario Nacional del Sector Eléctrico Paraguayo*, Noviembre 1994.
- [33] M. Weimin, J.S. Thorp, “An Efficient Algorithm to Locate all the Load Flow Solutions”, *IEEE Transaction on Power Systems*, Vol. 8, N° 3, pg 1077 -1083. Agosto 1993.
- [34] B. Noble, J. Daniel, *Algebra Lineal aplicada*, Tercera Edición, Prentice-Hall Hispanoamericana , S.A. 1989.
- [35] B. Barán, E. Kaszkurewicz, A. Bhaya “Distributed Asynchronous Team Algorithms: Application to the Load Flow Problem”, *XIX Conferencia Latinoamericana de Informática*, Bs. Aires, Septiembre 1993.
- [36] A. A. El-Keib, J. Neiplocha, H. Sing, D. J. Maratukulam, “A Decomposed State Estimation Technique Suitable for Parallel Processor Implementation”, *Trans. on Power Systems*, Vol. 7, N° 3. Agosto 1993.
- [37] M. Ikeda, D. Siljak, *Overlapping Decomposition, Expansions and Contractions of Dynamic Systems. Large Scale System 1*, North-Holland Publishing Co., pg 29-38. 1980.
- [38] H. S. Stone, *High-Performance Computer Architecture*, Addison-Wesley Publishing Company. 1987