

ALGORITMO RAPIDO PARA EL CALCULO DE CORTOCIRCUITOS
EN TIEMPO REAL

Eduardo A. Orduña
Francisco F. Garcés
Alberto Vargas

Universidad Nacional de San Juan
Instituto de Energía Eléctrica
República Argentina

RESUMEN

Este trabajo presenta una metodología computacional para el cálculo rápido de potencias de cortocircuito subtransitorias y de apertura; para las primeras, el núcleo fundamental del problema a resolver consiste en calcular la matriz nodal de impedancias $[Z_b]$ a partir de la matriz nodal de admitancias $[Y_b]$ utilizando técnicas de almacenamiento comprimido y un algoritmo para su inversión. El cálculo de las segundas implica en cierto paso del proceso la re-inversión reiterada de $[Y_b]$. Para reducir los tiempos de procesamiento, se ha implementado un procedimiento de cálculo simbólico de $[Z_b]$ basado en evitar el reconocimiento de ceros en la reinversión de $[Y_b]$.

ABSTRACT

This paper presents a computational methodology to the fast calculation of both subtransient and breaking short-circuit powers. For the first ones the central problem to be solved consists in calculating the impedance bus matrix $[Z_b]$ from the admittance bus matrix $[Y_b]$ applying sparse techniques and a suitable algorithm for its inversion. The computation of the second ones involves the repeated re-inversion of $[Y_b]$ in a certain step of the process. To reduce processing-times a symbolic computation procedure has been implemented which is based in avoiding the sparse techniques each time $[Y_b]$ is re-inverted.

1. CALCULO DE CORRIENTES DE CORTOCIRCUITO [I]C2]

Suponiendo una falla trifásica en la barra "k" de una red, las corrientes de cortocircuito en la misma, pueden calcularse con la siguiente ecuación:

$$I_{ccij} = (U_{fi} - U_{fj}) * y_{ij} \quad (1.1)$$

donde:

I_{ccij} : corriente de cortocircuito que fluye de la barra "i" a la barra "j"

y_{ij} : admitancia del componente que vincula la barra "i" con la barra "j"

U_{fi}, U_{fj} : tensiones postfalla de las barras "i" y "j" respectivamente

Las tensiones postfalla en cada barra se calculan aplicando el teorema de superposición según la siguiente ecuación matricial:

$$[U_f] = [U_0] + [U_t] \quad (1.2)$$

donde:

$[U_f]$: vector de tensiones postfalla

$[U_0]$: vector de tensiones prefalla (dato)

$[U_t]$: vector cambio en las tensiones

$[U_t]$ se calcula aplicando el teorema de Thevenin con la siguiente ecuación matricial:

$$[U_t] = [Z_b] * [I_f] \quad (1.3)$$

donde:

$[I_f]$: vector corrientes de falla

$[Z_b]$: matriz de impedancias de barra inversa de la matriz de admitancias de barra $[Y_b]$ (dato)

2. CALCULO DE LA MATRIZ NODAL DE IMPEDANCIAS [Zb]

Como en gral. la matriz $[Y_b]$ es de estructura rala se utilizan técnicas de almacenamiento comprimido. Combinando estas técnicas con el algoritmo computacional propuesto por Takahashi para la formación de $[Z_b]$, se cumple con los siguientes requisitos computacionales para realizar cálculos en tiempo real:

- . BAJOS TIEMPOS DE EJECUCION
- . MODERADO REQUERIMIENTO DE MEMORIA CENTRAL

Este método permite realizar tres tipos diferentes de cálculo de $[Z_b]$:

(1) [Zb] rala

Se calculan únicamente los elementos diagonales y los elementos i_j que se corresponden con los elementos i_j distintos de cero de [Yb]; en otras palabras, se calculan las impedancias propias de cada barra y las impedancias de transferencia de las mismas a sus barras adyacentes.

(2) [Zb] rala, más algunos elementos requeridos especialmente

(3) [Zb] llena

El tipo 1) es utilizado en el cálculo de las corrientes subtransitorias (ver pto. 3).

El tipo 2) es utilizado en el cálculo de corrientes de apertura (ver pto. 5), donde, además de los elementos de [Zb] rala, es necesario conocer algunos elementos adicionales.

2.1 ALGORITMO DE TAKAHASHI [3]

De la definición de [Zb] se tiene que:

$$[Yb] * [Zb] = [I] \quad (2.1)$$

con

[I]: matriz identidad

Estableciendo la hipótesis de que [Yb] es no singular y simétrica, ésta puede factorizarse en las siguientes tres matrices:

$$[Yb] = [L] * [D] * [L]^t \quad (2.2)$$

con

[L]: es una matriz triangular inferior unidad

[D]: es una matriz diagonal

[L]^t: es la transpuesta de [L]

Cada término de la matriz factorizada puede calcularse como sigue:

. términos diagonales

$$d_{ii} = y_{ii} - \sum_{k < i} l_{ik}^2 * d_{kk} \quad \text{para } i = 1, \dots, n \quad (2.3)$$

$k < i$

términos triangulares

$$l_{ij} = (y_{ij} - \sum_{k(j)} l_{ik} * l_{kj} * d_{kk}) / d_{jj} \quad \text{para } i = 2, \dots, n \quad (2.4)$$

Reemplazando la ec.(2.2) en (2.1) queda:

$$[L] * [D] * [L]^t * [Zb] = [I] \quad (2.5)$$

Ahora se define:

$$[W] = ([L] * [D])^{-1} = [D]^{-1} * [L]^{-1} \quad (2.6)$$

Dado que [L] es una matriz triangular inferior unidad, [L]⁻¹ también lo es; como [D] es una matriz diagonal, [D]⁻¹ también lo es, siendo cada término de la inversa, la inversa del término correspondiente de la matriz [D]. Se deduce, por lo tanto, que [W] es una matriz triangular inferior con sus términos diagonales:

$$w_{ii} = 1/d_{ii} \quad \text{para } i = 1, \dots, n$$

Se define otra matriz

$$[T] = [I] - [L]^t \quad (2.7)$$

[T] es estrictamente triangular superior con sus términos diagonales igual a cero.

La introducción de (2.6) y (2.7) en (2.5) da como resultado la siguiente ecuación:

$$[Zb] = [W] + [T] * [Zb] \quad (2.8)$$

Conociendo que [Zb] es simétrica, todos sus elementos son obtenidos sucesivamente como sigue:

$$\begin{aligned} z_{n,n} &= w_{n,n} \\ z_{n-1,n} &= t_{n-1,n} * z_{n,n} \\ z_{n-1,n-1} &= w_{n-1,n-1} + t_{n-1,n} * z_{n-1,n} \\ &\vdots \\ z_{i,j} &= \sum t_{i,k} * z_{k,j} \\ z_{i,i} &= w_{ii} + \sum t_{i,k} * z_{k,i} \quad \text{para } k < j \end{aligned} \quad (2.9)$$

El cálculo de [T] y [W] se denomina " paso de sustitución directa " y el cálculo de los elementos $z_{i,j}$ según (2.9) se denomina " paso sustitución inversa ".

Este método, así presentado, calcula los elementos de [Zb] que están localizados en correspondencia con los elementos no cero de la matriz [T].

Para el cálculo de los elementos de [Zb] correspondientes a barras no adyacentes a una barra determinada, se utiliza la ec.(2.9) como sigue:

$$z_{i,r} = \sum_{i=n, \dots, 1} t_{i,k} * z_{k,r} \quad \text{para } i < r \quad (2.10)$$

3. CALCULO DE LAS CORRIENTES SUBTRANSITORIAS

Incluyendo las admitancias subtransitorias y_{subi} de las fuentes de generación en los elementos diagonales correspondientes a las barras activas en $[Y_b]$, la distribución de corrientes subtransitorias en la red, se obtiene directamente con la aplicación de la ec. (1.1)

En la fig. 3.1 se representa el diagrama de flujo del programa CORSUB para el cálculo de corrientes subtransitorias, donde se ha indicado el nombre de cada subrutina.

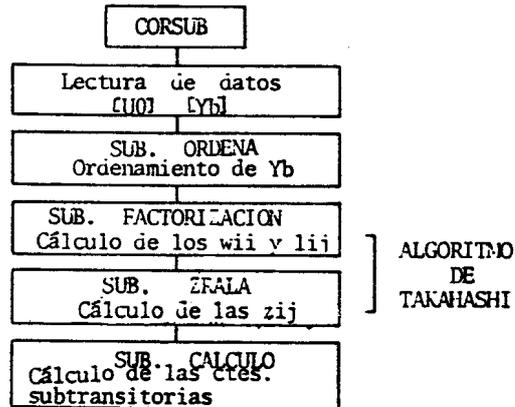


fig. 3.1 DIAGRAMA DE FLUJO DEL PROGRAMA CORSUB

Con el objeto de generar la menor cantidad posible de nuevos elementos innecesarios fuera de la diagonal en el paso de factorización de $[Y_b]$, se realiza el ordenamiento de la misma en la subrutina ORDENA comenzando por las barras que tienen menor número de ramas y finalizando con las que tienen mayor número de ramas conectadas [4].

En las subrutinas FACTORIZACION y ZRALA se encuentra implementado el algoritmo de Takahashi.

En la subrutina FACTORIZACION se calculan los factores $w_{i,i}$ y $l_{i,j}$ correspondientes al paso de sustitución directa.

En la subrutina ZRALA se calculan, con el paso de sustitución inversa, los elementos $z_{i,j}$ de $[Z_b]$.

4. CALCULO DE LAS CORRIENTES DE APERTURA

4.1 RECOMENDACIONES DE LA NORMA VDE 0102 [5][6]

El cálculo de las corrientes de apertura, según indica la norma VDE 0102, se realiza partiendo de los valores de corrientes subtransitorias que las fuentes de generación consideradas aportan a la falla; éstas corrientes son afectadas por un factor que tiene en cuenta el

amortiguamiento que se produce en el período transitorio:

$$I_{api} = \mu_i * I_{subi} \quad (4.1)$$

I_{api} : aporte de corriente de apertura del generador "i" a la falla
 I_{subi} : aporte de corriente subtransitoria del generador "i" a la falla
 μ_i : factor de amortiguamiento del generador "i"

El factor μ_i es función de la distancia eléctrica que existe entre el lugar de la falla y la fuente de generación "i", medida por la relación $I_{subi}/I_{nominali}$, y el tiempo (tap) de despeje de la falla, es decir:

$$\mu_i = f(I_{subi}/I_{nominali} , tap)$$

Si el lugar de la falla es alimentado por varios generadores, la corriente de apertura en el lugar de la falla está compuesta por la suma de las corrientes correspondientes a los diferentes generadores, donde para cada uno habrá que calcular la relación $I_{subi}/I_{nominali}$; en la fig. 4.1 se representan curvas, suministradas por la norma , para la determinación de los factores μ_i .

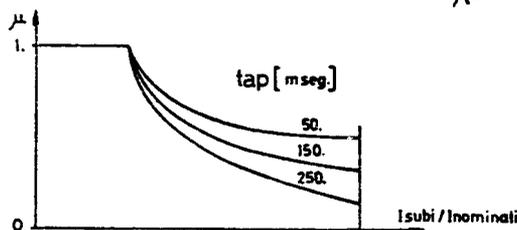


fig. 4.1 $\mu_i = f(I_{subi}/I_{nominali} , tap)$

4.2 CALCULO DE LA DISTRIBUCION DE CORRIENTES DE APERTURA [7]

La desagregación en toda la red o en parte de ella, de la corriente de apertura total aportada a la falla, calculada como se indica en el punto anterior, resulta una tarea de complicada realización.

En este trabajo se encara una solución en base a la determinación del valor de las admitancias de los generadores en el momento de la desconexión de la falla, las cuales incluidas en la $[Yb]$, permiten un cálculo directo de las corrientes de apertura.

Una estimación de las admitancias de los generadores en el momento de la interrupción de la falla, resulta de aplicar las recomendaciones ya mencionadas de la norma VDE 0102, haciendo el siguiente razonamiento:

Si se establece como hipótesis que la tensión U_{fi} en la barra activa "i" se mantiene constante hasta la desconexión de la falla, resultan válidas las siguientes relaciones al aplicar el teorema de Thevenin (fig. 4.2):

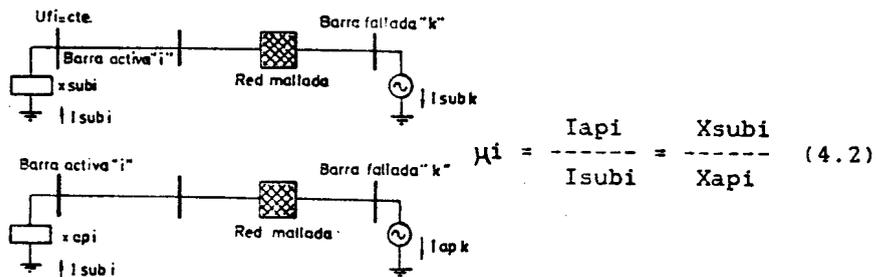


fig. 4.2

luego

$$(4.3)$$

$$x_{api} = x_{subi} / \mu_i$$

$$y_{api} = 1/x_{api} = \mu_i * y_{subi}$$

donde:

- U_{0k} : tensión de Thevenin de la barra "k"
- I_{subi} , I_{api} : corrientes subtransitoria y de apertura respectivamente aportada por el generador "i"
- x_{subi} , x_{api} : reactancias de la fuente de generación durante el período subtransitorio y en el momento de la apertura respectivamente.
- y_{subi} , y_{api} : admitancias de la fuente de generación durante el período subtransitorio y en el momento de la apertura respectivamente.

De esta forma calculando el factor μ_i se puede conocer y_{api} ya que se conoce y_{subi} de los datos de entrada.

El problema que se presenta para implementar un conjunto de programas que calcule en forma rápida según este procedimiento, es considerar que para cada barra del sistema donde se simule falla, la variación de la admitancia de cada generador es diferente debido a que es función de la distancia eléctrica entre la barra activa y la barra fallada, y del tiempo de interrupción de la falla.

De esta forma es necesario realizar una inversión de $[Y_b]$ por cada barra donde se simule falla.

Utilizando para ello el método de Takahashi implementando subrutinas que estructuren su funcionamiento en base a sentencias fortran IF para el reconocimiento de ceros que implica el almacenamiento comprimido, sería imposible obtener tiempos de ejecución que satisfagan los requerimientos que demandan los cálculos en tiempo real.

Con el objeto de solucionar este inconveniente, se implementó un algoritmo computacional denominado "de cálculo simbólico de la matriz [Zb]", con el cual se reducen los tiempos de ejecución.

5. CALCULO SIMBOLICO DE LA MATRIZ [Zb] [7]

Del proceso de cálculo de [Zb] según el método de Takahashi, se observan las siguientes características inherentes al método y al proceso de cálculo:

- 1) los elementos $z_{i,j}$ de [Zb] se calculan, como se observa de la ec. (2.9), a través de la suma de productos entre dos valores, es decir:

$$z_{i,j} = \sum t * z' \quad (5.1)$$

donde

$z_{i,j}$: elemento de [Zb] a calcular
t : factores $t_{i,k}$ correspondientes al cálculo de $z_{i,j}$
z' : elementos de [Zb] ya calculados

y en forma similar, para los elementos diagonales se agrega un término

$$z_{i,i} = w + \sum t * z' \quad (5.2)$$

donde

w : elemento $w_{i,i}$ correspondiente al cálculo de $z_{i,i}$

- 2) las variaciones de las admitancias de los generadores, no alteran la estructura de [Yb] ni de [Zb], por lo que la secuencia de cálculo de los elementos $z_{i,j}$ es siempre la misma.
- 3) La ejecución de sentencias Fortran IF implica tiempos de CPU aproximadamente cien veces mayores que los correspondientes a asignaciones directas a memoria central.

Teniendo en cuenta estas propiedades, se desarrolló un algoritmo computacional para el recálculo de [Zb], sin necesidad de recurrir al reconocimiento de ceros que implica el almacenamiento comprimido, reduciendo, de esta forma, los tiempos de ejecución.

Este algoritmo forma parte del programa CORVDE para el cálculo de corrientes de apertura, cuyo diagrama de bloques se presenta en la fig. 5.1.

En los siguientes apartados se explica la función de cada subrutina.

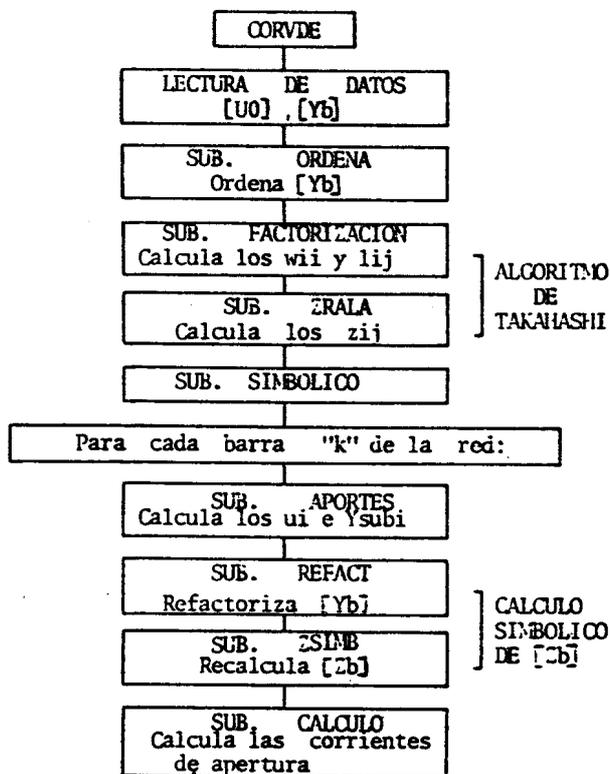


fig. 5.1 DIAGRAMA DE FLUJO DEL PROGRAMA CORVDE

5.1 ORDENAMIENTO DE [Yb] [4]

En la fig. 5.2 se representa el ordenamiento adoptado para [Yb] en el proceso de factorización de la misma.

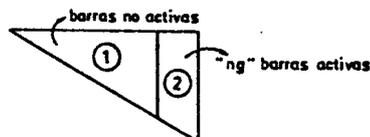


fig. 5.2 ORDENAMIENTO DE [Yb]

Las barras no activas son ordenadas comenzando por las que tienen menor número de ramas y finalizando con las que tienen mayor número de ramas conectadas; este ordenamiento se realiza con el objeto de disminuir la cantidad de elementos nuevos calculados en forma innecesaria en la zona 1; ésto implica una reducción en los requerimientos computacionales.

La zona marcada con 2 tiene un llenado completo al finalizar el cálculo de [Zb], correspondiendo los elementos adicionales a impedancias de transferencia entre barras no activas y activas, y entre estas últimas. Estos elementos son necesarios para calcular los aportes de corriente subtransitoria de c/ generador a la falla.

Esta disposición facilita el cálculo de los elementos de la zona 2, debido a que es conocido a partir de que columna se encuentran.

5.2 CALCULO DE LOS FACTORES μ_i Y REFACTORIZACION DE [Yb]

En primera instancia , antes de comenzar el proceso de simulación de fallas en cada barra, se calcula [Zb] incluyendo en [Yb] las admitancias subtransitorias y_{subi} de los generadores, tal como se explicó para el cálculo de corrientes subtransitorias.

5.2.1 SUBROUTINA APORTES

Para cada barra donde se simule falla, se calculan los aportes de corrientes subtransitorias de cortocircuito I_{subi} de las fuentes de generación utilizando la [Zb] calculada en primera instancia.

Inmediatamente, se calculan los factores de amortiguamiento μ_i para cada generador siguiendo las recomendaciones, ya mencionadas, de la norma VDE 102.

Posteriormente, se calculan las y_{api} de cada generador según la ec. (4.3) y se incluyen en [Yb] en lugar de las y_{subi} originales.

5.2.2 SUBROUTINA REFACT

Debido al reordenamiento original de la estructura de [Yb], las y_{api} reemplazan a las y_{subi} en los últimos "ng" términos diagonales correspondientes a las barras activas , lo que implica recalcular, solamente los factores $t_{i,j}$ y $w_{i,i}$ que se corresponden con los elementos $y_{i,j}$ e $y_{i,i}$ que están ubicados en las últimas "ng" filas. Este paso no compromete los tiempos de ejecución correspondientes a la reinversión de [Yb], aún empleando técnicas de reconocimiento de ceros utilizando sentencias Fortran IF.

5.3 SEGUIMIENTO DE LAS OPERACIONES PARA LA FORMACION DE [Zb]. SUBROUTINA SIMBOLICO.

En este paso se almacena en distintos arreglos la información necesaria para el cálculo de cada elemento de [Zb] rala; esto se realiza luego del paso de la factorización inicial de [Yb] y antes de comenzar con el proceso de cálculo de los aportes para cada barra.

El seguimiento simbólico de las operaciones se realiza de tal manera, que es posible el cálculo de [Zb] columna por columna comenzando por la columna 'n'; para ello se hace un seguimiento lógico de cálculo de los elementos con reconocimientos de ceros.

Para describir cualitativamente el procedimiento se hace referencia a la matriz "8x8" representada en la fig. 5.3, donde se considera el cálculo de solo 3 elementos (11,14 y 15) de la columna 8; a los valores se les ha asignado un número de orden cualquiera en el almacenamiento comprimido.

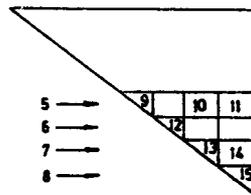


fig. 5.3

Se denomina :

- T : arreglo que contiene los valores $t_{i,j}$ y $w_{i,i}$ calculados en la factorización, almacenados en forma comprimida por fila.
- ZT : arreglo donde se almacenarán los valores de $z_{i,j}$ y $z_{i,i}$ calculados, almacenados en forma comprimida por columna.
- NUPRO : arreglo que contiene para cada elemento, el número necesario de productos $t * z$ para su cálculo.
- NUELE : arreglo que almacena para cada columna, el número de elementos distintos de cero de la misma sin considerar el elemento diagonal.
- LOCAC : arreglo que contiene, para cada elemento, el índice de locación en el arreglo T o ZT.
- INDICE : arreglo bidimensional que almacena los índices de locación de los elementos que intervienen en los productos $t * z$.

Se comienza por el elemento número 14 de ZT ($z_{7,8}$), ya que para el elemento nro. 15 se cumple que:

$$ZT(15)=T(15)= z_{8,8}= w_{8,8}$$

El elemento $z_{7,8}$, según el desarrollo del método :

$$ZT(14) = z_{7,8} = T(14) * ZT(15) = t_{7,8} * z_{8,8}$$

Como es el primer elemento a calcular, luego se hace:

$$\text{LOCAC (1)} = 14$$

es decir, que el primer elemento a calcular ocupará el lugar número 14 en el arreglo ZT; el número necesario de productos $t * z$ para el cálculo del mismo es:

$$\text{NUPRO (1)} = 1$$

posteriormente se almacenan los índices de memoria, en los arreglos respectivos, de los elementos que intervienen en tal producto:

$$\begin{aligned}\text{INDICE (1,1)} &= 14 \\ \text{INDICE (1,2)} &= 15\end{aligned}$$

En forma similar, para el segundo elemento a calcular:

$$\begin{aligned}\text{ZT (11)} &= z_{5,8} = t_{5,7} * z_{7,8} + t_{5,8} * z_{8,8} \quad 6 \\ \text{ZT (11)} &= T_{(10)} * \text{ZT}_{(14)} + T_{(11)} * \text{ZT}_{(15)}\end{aligned}$$

por lo tanto:

$$\begin{aligned}\text{LOCAC (2)} &= 13 \\ \text{NUPRO (2)} &= 2 \\ \text{INDICE(2,1)} &= 10 \\ \text{INDICE(2,2)} &= 14 \\ \text{INDICE(3,1)} &= 11 \\ \text{INDICE(3,2)} &= 15\end{aligned}$$

De la misma forma se puede proceder con los restantes elementos de la columna; luego de ello se almacena el número de elementos distinto de cero de la misma:

$$\text{NUELE(8)} = 4$$

5.4 CALCULO RAPIDO DE LOS ELEMENTOS DE [Zb] SUBROUTINA ZSIMB

En un paso posterior se calcula [Zb] "barriendo" los arreglos formados en el paso anterior, lo cual permite un cálculo rápido de los elementos correspondientes ya que se trabaja solamente con direccionamientos a memoria central.

Para los elementos considerados en el punto anterior se tiene:

para la columna 8

_ elemento diagonal: $\text{ZT (15)} = T (15) = z_{8,8}$

_ elementos fuera de la diagonal:

nro. 1

$$ZT(LOCAC(1)) = T(INDICE(1,1)) * ZT(INDICE(1,2)) \Rightarrow NUPRO(1) = 1$$

$$ZT(14) = T(14) * ZT(15) = z 7,8$$

nro. 2

$$ZT(LOCAC(2)) = T(INDICE(2,1)) * ZT(INDICE(2,2)) +$$

$$T(INDICE(3,1)) * ZT(INDICE(3,2)) \Rightarrow NUPRO(2) = 2$$

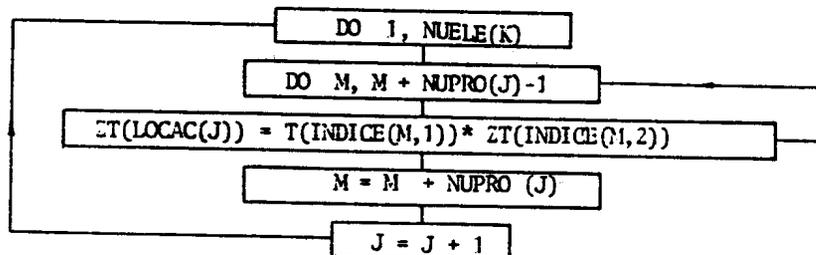
$$ZT(11) = T(10) * ZT(14) +$$

$$T(11) * ZT(15) = z 5,8$$

de la misma forma se procede con los demás elementos de la columna.

La cantidad de elementos calculados por columna se regula con el arreglo NUELE (K).

Genéricamente, para la columna "k", suponiendo que la misma comienza por el elemento "j", y las locaciones de memoria almacenadas en el arreglo INDICE comienzan en la locación "m" para el primer elemento de la columna, se puede aplicar el siguiente diagrama de flujo para el calculo de todos los elementos de la columna :



6. REQUERIMIENTOS COMPUTACIONALES

El conjunto de programas fué desarrollado en lenguaje FORTRAN V e implementado en un calculador de procesos cuyas principales características se resumen en la tabla 1:

TABLA 1

MARCA	DIGITAL
MODELO	MICRO-VAX II
CAPACIDAD	13 MBYTES
VELOCIDAD	1,1 MIPS (*)

(*) MIPS: millones de instrucciones por segundo

En la tabla 2 se detallan las principales características de las redes que se utilizaron como ejemplos para la evaluación de las aptitudes computacionales de los conjuntos de programas presentados. También se muestran los tiempos de ejecución obtenidos para ambos conjuntos, al simular falla trifásica en el total de barras en cada ejemplo.

TABLA 2

REDES EJEMPLOS	NRO. DE BARRAS	NRO. DE RAMAS	NRO. DE GENE.	GRADO DE MALLADO	TI. EJEC. [SEG]-CPU CORVDE	TI. EJEC. [SEG]-CPU CORSUB
	30	41	5	1.4	1.7	0.4
I.E.E.E	57	80	12	1.4	4.1	0.9
	118	185	20	1.6	14.5	2.2
S.N.I	83	93	12	1.13	6.4	1.7

Los requerimientos máximos de memoria central son de 280 kbytes para el conjunto CORVDE y 190 kbytes para el conjunto CORSUB; estos requerimientos se establecen para un sistema de las siguientes características:

TABLA 3

NRO. DE BARRAS	----->	400
NRO. DE RAMAS	----->	800
NRO. DE BARRAS ACTIVAS	----->	70

7. CONCLUSIONES

Teniendo en cuenta los resultados obtenidos y presentados en el punto anterior se llegó a las siguientes conclusiones:

. Los requerimientos máximos de memoria central para ambos conjuntos de programas, permiten sus implementaciones en procesadores de mediana capacidad de memoria real, por ej. en computadores personales. Adicionalmente, el funcionamiento de los conjuntos de programas como parte de un sistema de programas dispuesto para la vigilancia en tiempo real, no comprometerá el funcionamiento del sistema.

. Los tiempos de cálculo de potencias subtransitorias con el conjunto CORSUB, son aceptables inclusive para sistemas con más de 100 barras.

. Los tiempos de cálculo se tornan un tanto elevados para sistemas con más de 100 barras cuando se calculan potencias de apertura con el conjunto CORVDE.

Normalmente, en los estudios en tiempo real, solo se realizan cálculos de cortocircuitos en algún subconjunto de barras o en determinadas barras donde es de especial importancia realizar un seguimiento de los valores de potencia de cortocircuito.

El cálculo rápido de las potencias de apertura según el procedimiento computacional descrito, representa un aporte significativo al tema de cálculo de cortocircuitos en tiempo real.

8. BIBLIOGRAFIA

[1] J. F. Rivera, F. F. Garcés y A. Vargas
" Algoritmos Rápidos para Análisis de Cortocircuitos en Grandes Redes Eléctricas "; Electrotécnica, Vol. 70, n° 5, Sep-Oct 1984

[2] E. Orduña, F. F. Garcés y A. Vargas
" Cálculo de cortocircuitos en tiempo real"; 2da. Reunión Nacional de Sistemas Eléctricos, Noviembre de 1987

[3] K. Takahashi, J. Fagan y M. Chen
" Formation of Sparse Bus Impedance Matrix and its Application to Short Circuit Study "; Paper n° TP11-B, PICA Conference Proceedings 1973

[4] E. C. Ogbuobini, W. F. Tinney and J. W. Walker
" Sparsity-Directed Descomposition for Gaussian Elimination of Matrices ", I.E.E.E. Transactions on Power Apparatus and Systems, Vol. pas 89, n° 1, Jan. 1968, p. p. 141-149

[5] Roeper
" Las Corrientes de Cortocircuito en las Redes Eléctricas "; Ed. Dossat (1970) (Madrid)

[6] Funk
" Cálculo de Corrientes de Cortocircuito "; Paraninfo S.A (1980) (Madrid)

[7] E. Orduña
" Cálculo de cortocircuitos en tiempo real"; Informe final de beca de iniciación; CONICET, 31 de Mayo de 1988.