UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES

"ZARAGOZA"

EVALUACION DE LOS METODOS DE ACELERACION DE LA CONVERGENCIA PARA PROCESOS QUIMICOS CON RECIRCULACION.

T E S I S

QUE PARA OBTENER EL TITULO DE

INGENIERO QUIMICO

P R E S E N T A N

LUIS MANUEL CASTRO LIRA

HECTOR WULFRANO SANCHEZ RIVERA

PEDRO SOLIS RODRIGUEZ

MEXICO, D. F., 1989

UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso

I N D I C E

LISTA DE FIGURAS

LISTA DE TABLAS

INTRODUCCION Y OBJETIVOS

La simulación de procesos químicos ha sido fundamental para el desarrollo de nuevos procesos, el estudio de alternativas de diseño y el mejoramiento de plantas de proceso existentes. Con el empleo de los simuladores de procesos, se evita en la mayoría de los casos, la construcción de plantas piloto, lo cual genera un considerable ahorro de tiempo y recursos económicos, así como el alto riesgo de accidentes.

Para la simulación de un proceso grande, se requiere el empleo de cálculos complejos y tediosos; y si además se consideran procesos con corrientes de recirculación surge la necesidad de generar una estrategia adecuada para atender tales problemas. Por lo cual se han desarrollado una serie de algoritmos para acelerar la convergencia de los métodos de solución empleados y hallar la solución en un tiempo mínimo (Capítulo III).

En este trabajo se tienen como objetivos:

- Analizar los algoritmos para la selección de criterios para la aceleración de la convergencia de los modelos matemáticos.

- Establecer los criterios para la selección de la combinación más adecuada para la formulación y desarrollo de las bases de estos algoritmos.

- Analizar los simuladores de procesos más utilizados tanto a nivel académico como a nivel industrial para resaltar las ventajas de cada uno de ellos así como sus características generales (Capítulo II).

C A P I T U L O I

Generalidades

La simulación de procesos químicos propiamente se refiere al uso de las ecuaciones rigurosas o empíricas que describen matematicamente algún proceso químico o físico mediante un conjunto de programas de computadora, que tienen como finalidad el cálculo de los valores reales de las variables de operación más significativas para dicho proceso, con lo cual se obtiene información acerca del funcionamiento del mismo.

La simulación ha sido de gran trascendencia, ya que puede ser usada como una herramienta para facilitar el diseño, estudio de plantas existentes, para proyectos de nuevas plantas y para el diseño de estrategias óptimas (esto implica el uso innecesario de plantas piloto, que tienen implicito un alto riesgo y un alto costo), y para una multitud de otras aplicaciones, puesto que cada aplicación es característica de cada problema.

La simulación de procesos dentro de la Ingeniería Química se inicio a mediados de los 50's, en este período, el desarrollo de programas de computadora fue exclusivamente para operaciones unitarias individuales, lo que dio las bases para que posteriormente se desarrollaran nuevas técnicas para la simulación de procesos.

A principios de los 60's, los simuladores que eran comunmente aceptados, tenian las siguientes características básicas:

1.- Podian codificarse en un lenguaje de alto nivel (FORTRAN) y eran altamente modulares.

2.- El sistema podia ser fácil de usar, no se requeria ser experto.

3.- Las correlaciones de propiedades físicas deberian ser rigurosas y tan precisas como fuera posible.

Con la creciente aceptación de la simulación de procesos, el período de los 70's, fue caracterizado por el refinamiento de los simuladores de procesos industriales, pero como los simuladores se realizaron para sus respectivas compañias, es decir para resolver los problemas propios de la compañia, se tuvieron problemas serios en lo que respecta a la generalidad y confiabilidad de los simuladores

En el período reciente (1980's), los simuladores se han convertido en una herramienta legítima en la Ingeniería de Procesos, pero es usada principalmente por las compañías que han desarrollado el simulador. Los principales programas desarrollados por Instituciones Académicas, así como los desarrollados por las compañías privadas se muestran en el capítulo II de este trabajo.

Los elementos esenciales de un sistema de cómputo a gran escala, auxiliares en el análisis de procesos químicos e ingenieriles se muestran en la figura --- (1). El sistema puede verse como una estructura construida sobre ciertos blo - ques interactivos, los bloques de un sistema de simulación de procesos son: A) - MODELOS, B) ALGORITMOS, C) "SOFTWARE" Y D) INTERFASE CON EL USUARIO.

Los fundamentos de algunos sistemas consisten de modelos que forman la base para el análisis; los modelos de un proceso químico, usados por sistemas de si - mulación de procesos son todas las relaciones matemáticas, derivadas de las le - yes de la conservación, las ecuaciones de velocidad, relaciones de propiedades - físicas y restricciones de diseño y de control. Los modelos matemáticos toman -- la figura de ecuaciones algebraicas y diferenciales describiendo así el proce - so. Los requerimientos más importantes de los modelos matemáticos son: Que sean apropiados a su uso en términos rigurosos, nivel de detalle, aproximación, validez y generalidad.

Los algoritmos operan sobre los modelos para producir los resultados requeridos, estos resuelven los problemas matemáticos generados por los modelos. El - tipo de problema matemático que deberá resolverse dependera del tipo de análisis de interes, pero, pero incluye solución de ecuaciones diferenciales y algebrai - cas, así como programación no-lineal. Los requerimientos de un buen algoritmo -- son; a) Que sea estable, b) Tan general como sea posible y c) Eficiente en tér - minos de ejecución y almacenamiento claro y funcional. Cuando no se cumplan es - tos requisitos, deberan hacerse los cambios o modificaciones pertinentes.

La parte que es el " software " de la computadora, incluye todo lo requerido para implementar los algoritmos sobre una computadora en particular y su --- sistema de operación, incluido con éste; estan el programa de arquitectura del -

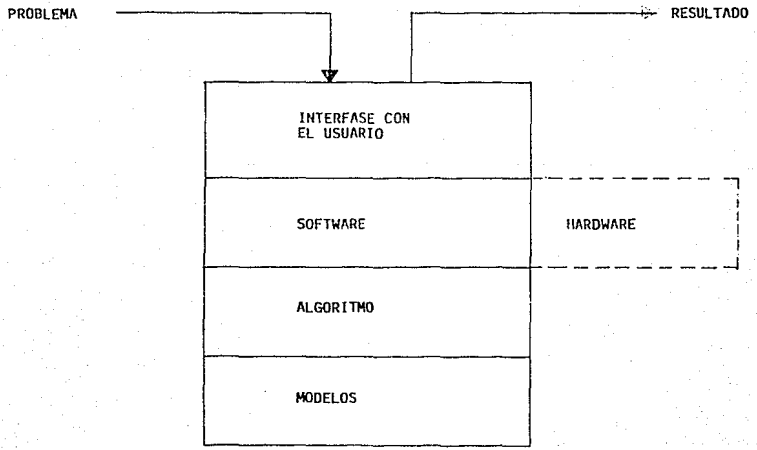


FIGURA No. 1

ESTRUCTURA DE BLOQUES DE UN SISTEMA DE COMPUTO

sistema; la estructura de los datos; la interfase sistema-archivo; el lenguaje de programación; el código de cómputo y la documentación del sistema. Los requerimientos de un software son:

I).- Fácil de entender
II).- Fácil de mantener y modificar
III).- Tan compatible como sea posible

El último bloque de la estructura, es la interfase con el usuario, éste incluye el lenguaje de entrada, por lo cual el usuario describe sus problemas, los reportes que contienen los resultados, la documentación del usuario que explica, el uso del sistema y protocolos para interfases con otros programas o sistemas.

1.1 SIMULACION EN ESTADO ESTACIONARIO

Este tipo de simulación esta basado sobre ciertas premisas, tales como alimentación específica, composición limitada, ésta puede proveer información referida al diseño de la planta y puede predecir un buen rendimiento. Este tipo de simulación esta orientado fundamentalmente a la obtención de los balances de masa y energía, aunque algunas veces esta información no es la que finalmente se requiere, ya que este tipo de datos no produce información suficiente sobre los parámetros y tamaño de equipo específico.

Generalmente para resolver problemas de esta especie, se deben utilizar ecuaciones de equilibrio (P-V-T), desafortunadamente una gran cantidad de plantas químicas es de configuración complicada, involucrando corrientes de recirculación y operaciones de contracorriente lo cual origina un esquema de trabajo para para la aplicación de los métodos de convergencia.

En la modelación en estado estacionario de una planta compleja, aparecen dos tipos de problemas.

1).- CALCULOS DE OPERACION 2).- CALCULOS DE DISEÑO

1).- Una planta es calculada en este modo, si los parámetros de las corrientes de entrada y los datos de especificación de la unidad funcional son conocidos, por lo que los perfiles de las variables dependientes y/o la información de las corrientes de salida son calculados.

2).- Una planta compleja es calculada en modo diseño, si la información incompleta de las corrientes tanto de entrada y salida, así como la serie de incognitas de los parámetros de la unidad, pueden calcularse mediante un método iterativo.

Los calculos de operación pueden calcularse en dos grupos.

A) Simulación abierta
B) Simulación controlada

A).- Los parámetros de la corriente de alimentación y los parámetros de la unidad son dados, reduciendose el problema a determinar las corrientes faltantes.

B).- Una serie de parámetros de ciertas corrientes internas o externas son fijas, por lo que el problema se reduce a ajustar los parámetros desconocidos de la corriente de alimentación para cumplir con las especificaciones.

1.2 SIMULACION DINAMICA

Aunque el campo de la simulación dinámica fue dependiente de máquinas analógicas, la simulación dinámica es ahora manejada por técnicas digitales, relacionada a los sistemas de procesos dependientes del tiempo. El ingeniero químico es quien desarrolla sistemas de ecuaciones diferenciales para describir el proceso y usando circuitos análogos de computadora, él puede resolver grandes redes de ecuaciones diferenciales, con lo cual se obtiene información para examinar la instrumentación de control, condiciones de paro y arranque, respuestas variables para los cambios de alimentación y una gran variedad de problemas dependientes del tiempo.

Los problemas de control de los procesos químicos se incrementan con la complejidad de las nuevas unidades y con la tendencia a integrar el diseño de la -- planta, lo cual implica una investigación a fondo en el desarrollo de la técnica de la simulación dinámica de procesos.

1.3 MODELOS

El corazón de los sistemas de simulación de procesos, son los modelos de - las unidades de operación, la estructura de un modelo se muestra en la figura - (2). Generalmente los modelos producen relaciones algebráicas no lineales de- la forma:

VARIABLES DE SALIDA	=	F (VARIABLES DE) ENTRADA	2.1

Las variables de entrada son las condiciones de las corrientes de entrada y los parámetros del modelo (las variables requeridas para satisfacer el funcionamiento de la unidad de operación). Las variables de salida son las condiciones de las corrientes de salida, resultados del funcionamiento y dimensionamiento (tales como requerimientos de potencia para una bomba o la carga de calor para un cambiador) y variables internas o de retención (tales como temperatura de fase interna, composición y valores de equilibrio "K" en una columna de destilación). Estas variables internas son valores intermedios, no siempre- requeridos en el diagrama de flujo, pero pueden ser almacenados, para usarse - como valores iniciales en la siguiente iteración si la unidad tiene un circuito de recirculación.

Funcionalmente un modelo de una unidad de operación, puede expresarse -- como:

$$F\ (U,X,Y,Z,r\ ) = 0 \qquad 2.2$$

U= Vector de parámetros del modelo.

X= Variables de la corriente de entrada

Y= Variables de la corriente de salida

Z= Variables internas (de retención)

r= Resultados de variables (funcionamiento y dimensionamiento).

El número de ecuaciones es igual a la suma de los números de variables de- corrientes de salida, variables internas y variables de resultados. el número - de grados de libertad es igual al número total de parámetros del modelo y variables de corrientes de entrada.

Hay muchas posibles combinaciones de parámetros de modelo que pueden usarse para especificar una unidad de operación, por ejemplo considerando un in ------

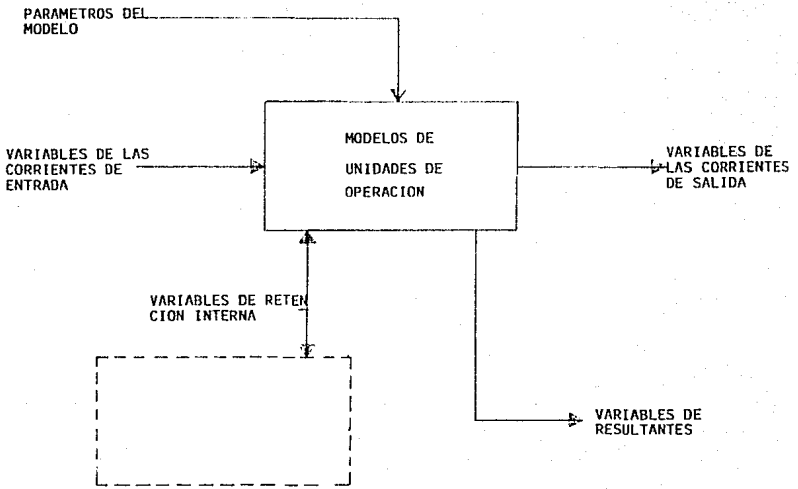
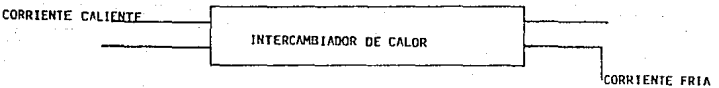


FIGURA No. 2

ESTRUCTURA DE UN MODELO



COEFICIENTE	K	*	*	*
AREA SUPERFICIAL	A	*		
APROX. DE TEMPERATURA	T_{min}		*	
CARGA	Q			*

FIGURA No. 3

ESPECIFICACIONES POSIBLES PARA UN MODELO DE INTERCAMBIADOR

tercambiador de calor a contracorriente como se muestra en la figura (3). se muestran tres conjuntos de especificaciones; 1) Uno puede especificar el coeficiente total de transferencia de calor y el área de superficie, dando-las dos corrientes de entrada, esto completamente determina la condición de las dos corrientes de salida 2) Un segundo conjunto de especificaciones, -- consiste del coeficiente total y de la aproximación de temperaturas 3) Especificar el coeficiente total y la carga de calor. Alguna de estas tres especificaciones llevaría a la convergencia al diagrama de flujo.

El flujo de información (X) en un sistema de simulación se muestra en la figura (4). La fase de cálculo de entrada para el balance de masa y e - nergía en el diagrama de flujo definido, es suficiente detalle para determinar todas las corrientes intermedias y productos y las variables de funcionamientode la unidad y para todas las unidades.

Los resultados del balance de masa y energía, aumentados con datos de dimensionamiento, preveen la entrada al dimensionamiento de equipo. La estima - ción de costos, requiere como entrada, detalles de equipo de la mayoría de estos, más los materiales de construcción y otros datos necesarios para deter - minar el capital invertido.

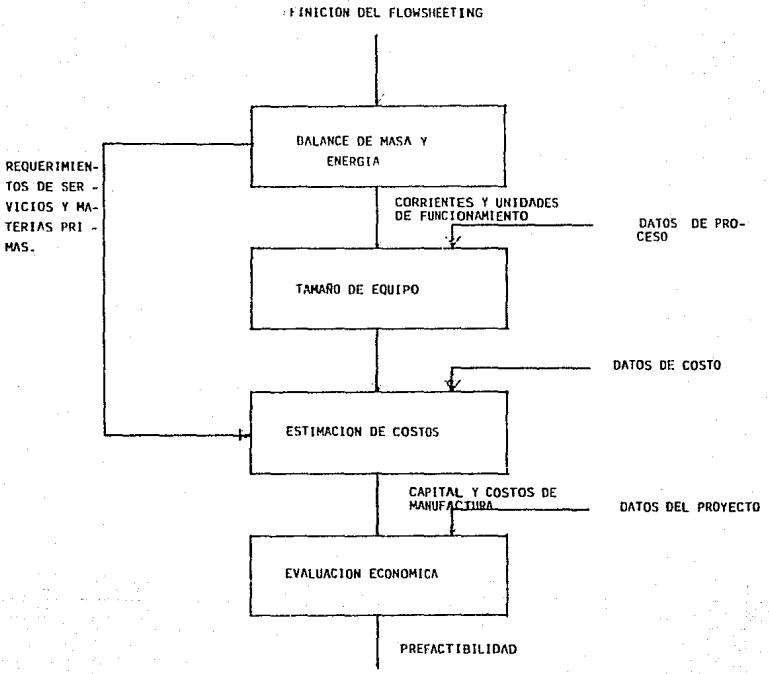


FIGURA No. 4

FLUJO DE INFORMACION EN UN SISTEMA DE "FLOWSHEETING" (SIMULACION DE PROCESOS)

I.4 ALGORITMOS DE CONVERGENCIA

Dentro del diseño y análisis de procesos químicos, petroquímicos y de refinación, los simuladores de procesos constituyen una de las herramientas básicas de cálculo.

La gran mayoría de los simuladores tienen una estructura de tipo modular, de tal forma que para efectuar los cálculos se consideran los módulos que integran el proceso en una forma secuencial, debido a que los módulos únicamente se pueden calcular conociendo sus corrientes de entrada y parámetros del equipo.

En la mayoría de los casos prácticos se presentan recirculaciones entre los módulos de tal forma que no es posible efectuar el cálculo secuencial una sola vez, y es necesario un procedimiento iterativo. Aquí se puede apreciar claramente que el problema de la solución de recirculaciones en un simulador de procesos implica resolver el sistema de ecuaciones :

$$X = G(X) \qquad \text{Forma explícita} \qquad 2.3$$

que desde luego también se puede expresar de la forma :

$$F(X) = X - G(X) = 0 \qquad \text{Forma implícita} \qquad 2.4$$

donde : F es un vector de n-funciones no-lineales.
X un vector de n-variables y
0 es un vector n-dimensional

Es importante señalar que las dos principales dificultades que se encuentran al tratar de resolver el sistema de ecuaciones en (2.4) son las siguientes :

a) En general la única forma de determinar un vector (α) que satisfaga el sistema de ecuaciones, es utilizando un procedimiento numérico de tipo iterativo (de ahí la importancia del estudio que se realiza en este trabajo sobre las técnicas mencionadas).

b) En general no hay garantía de que (2.4) tenga una solución única ; por ejemplo en el sistema de ecuaciones :

$$X_1 + 2X_2 - 3 = 0$$
$$2X_1^2 + X_2^2 - 5 = 0 \qquad 2.5$$

se puede verificar fácilmente que tanto $\alpha = \begin{pmatrix} -.82 \\ 1.91 \end{pmatrix}$ así como $\alpha = \begin{pmatrix} 1.488 \\ 0.756 \end{pmatrix}$, satisfacen estas ecuaciones.

Aunque se pueden formular en la teoría condiciones suficientes para la existencia de una solución única en (2.4) , su aplicación en la práctica es difícil, por lo que normalmente se recurre a argumentos de tipo físico para inferir la unidad o bien se delimita la región en la cual se encuentra la solución de interés. Por ejemplo en (2.5) solo tiene sentido físico, el que X_1 y X_2 tengan valores positivos, la solución de interés será

$$\alpha = \begin{pmatrix} 1.488 \\ 0.756 \end{pmatrix} \qquad 2.6$$

Finalmente, es desde luego posible que no exista una solución en el sistema de ecuaciones, pero en la práctica, esto se debe normalmente a que la formulación de las ecuaciones no es correcta.

1.4. EXISTENCIA DE LA SOLUCION Y CONVERGENCIA DE LOS ALGORITHOS GENERALES DE SOLUCION.

Se tiene el sistema de ecuaciones en (2.5)

$$X = G(X)$$

donde el vector g(X) puede tener diferentes formas, dependiendo del tipo de algoritmo a usar. Para la solución numérica de (2.4), la ecuación anterior puede definir un algoritmo de la forma :

$$X^{k+1} = G(X^k) \qquad 2.7$$

A este algoritmo se le conoce como " Método iterativo de un punto", debido a que a partir del punto X^k se predice el nuevo punto X^{k+1}. Si a --

partir de un valor inicial dado X^0, el algorítmo en (2.7) converge a la solución α ; este generará una secuencia de puntos $\left\{ X^k \right\}_{k=0}^{\infty}$ de tal forma que:

$$\lim_{k \to \infty} X^k = \alpha$$

Es de gran importancia determinar las condiciones bajo las cuales (2.4) tiene una solución única, las condiciones bajo las cuales la ecuación (2.7) convergerá a α y la rapidez de convergencia de las principales clases de algorítmos definidos por la ecuación (2.7). Con objeto de analizar los dos primeros puntos es conveniente introducir las dos definiciones siguientes:

1.- Una región R en un espacio n-dimensional es cerrada sí cualquier sucuencia convergente $\{ X^k \}$, en la que $X^k \in R$, es tal que su límite $\alpha \in R$.
Ejemplo : $a \leq X_1 \leq b$, $c \leq X_2 \leq d$; definen una región cerrada R, mientras que $a \leq X_1 < b$, $c \leq X_2 \leq d$ no lo hacen debido a que en la región definida por estas últimas desigualdades puede haber secuencias cuyo límite converge a (b) y claramente $b \notin R$.

2.- Una función g(X) que mapea el vector real X de dimensión (n), se dice que es un mapeo contráctil con respecto a la norma $/\!/ \cdot /\!/$ en una región cerrada R si :

i) $X \in R \implies g(X) \in R$
ii) $/\!/ g(X)-g(Y) /\!/ \leq L \, /\!/ X-Y /\!/ \; \forall X, Y \in R$, $0 \leq L < 1$

Es decir el mapeo contráctil mantiene la propiedad de la cerradura por (i) y cumple con la condición de Lipschitz por (ii). Con esto último se requiere que el valor de la función así como la derivada de g(X) tengan valores finitos para toda X en R . Notese que la norma $/\!/ \cdot /\!/$ no esta restringida a ninguna en particular.
Como recordatorio conviene señalar que las tres normas más comunes son :

a) $/\!/ X /\!/_1 = \sum_{i=1}^{n} / x_i /$

b) $\| x \|_2 = \left[\sum_{i=1}^{n} | x_i |^2 \right]^{1/2}$ Euclidiana (2.8)

c) $\| x \|_\infty = \max_{1 \leq i \leq n} | x_i |$

Antes de plantear el teorema de existencia de la solución es conveniente considerar lo siguiente :

1.- Si $X^{k+1} = g(X^k)$, donde $g(X)$ es un mapeo contráctil con respecto a la norma $\| \cdot \|$, y $X^0 \in R$ se tiene que para $0 \leq L < 1$

i) $\| X^{k+1} - X^k \| \leq L^k \| X^1 - X^0 \|$ $k = 0, 1, \ldots, n$

ii) $\| X^{m+k} - X^m \| \leq \frac{L^m}{1 - L} \| X^1 - X^0 \|$ $m \geq 0$,

Con lo anterior y las definiciones 1) y 2) se tienen elementos necesarios para demostrar el siguiente teorema, relacionando con la existencia y unicidad de la solución en el sistema de ecuaciones no-lineales (2.4)

TEOREMA 1.-

Si existe una región cerrada R en la que $g(X)$ es un mapeo contráctil con respecto a alguna norma $\| \cdot \|$, se tiene que :

i) El sistema de ecuaciones $X = g(X)$ tiene una solución única en R.

ii) Para cualquier $X \in R$, la secuencia $\left\{ X^k \right\}_{k=0}^{\infty}$ definida por $X^{k+1} = g(X)^k$ converge a α

Hay que notar que el teorema 1, establece condiciones suficientes para la existencia de una solución única en (2.4), por lo que es una condición más bien restrictiva y su aplicación en la práctica no es siempre muy sencilla. Sin embargo es interesante observar que el teorema garantiza que el algoritmo definido por (2.7) converge a la solución para la clase de mapeos considerados y por lo tanto sugiere que el proceso iterativo definido por (2.7) constituye una base racional para resolver numéricamente sistemas de ecuaciones no-lineales. Queda entonces, como punto importante, que es de gran relevancia en la práctica, el analizar la rapidez con la cual las principales clases de algoritmos definidos por (2.7) convergen a la solución.

Para esto es conveniente expander $g(X)$ alrededor de la solución α con la serie de Taylor, hasta los términos de segundo orden.

Para cada componente (i) del vector g(X) se tendrá :

$$g_i(X) = g_i(\alpha) + \sum_{j=1}^{n} \frac{\partial g_i}{\partial x_j}(x_j - \alpha_j) + \frac{1}{2}\sum_{j=1}^{n}\sum_{k=1}^{n}(x_j - \alpha_j)(x_k - \alpha_k)\frac{\partial^2 g_i}{\partial x_j \partial x_k} \qquad (2.9)$$

Considerese ahora la clase de algoritmos de primer orden donde

$$\frac{\partial g_i}{\partial x_j} \qquad (2.10)$$

es continua y acotada en una región R. Como en general (2.10) es diferente de cero se pueden despreciar los términos de segundo orden en (2.9) por lo - que esta expresión se reduce a :

$$g_i(X) - g_i(\alpha) = \sum_{j=1}^{n} \frac{\partial g_i}{\partial x_j}(x_j - x_j) \qquad (2.11)$$

Por notación, definiendo la matriz $S = [s_{ij}]$ como :

$$s_{ij} = x \in R \left/ \frac{\partial g_i}{\partial x_j} \right/ \qquad (2.12)$$

y aplicando la norma $\|\cdot\|$ al vector en (2.11) se tiene :

$$\|g(x) - g(\alpha)\| \leq \|S\| \|x - \alpha\| \qquad (2.13)$$

Para un punto x^k cercano a la solución, (2.13) se puede escribir como :

$$\|g(x^k) - g(\alpha)\| \leq \|S\| \|x^k - \alpha\| \qquad (2.14)$$

y como $x^{k+1} = g(x)^k$, $\alpha = g(x)$, (2.14) se reduce a :

$$\|x^{k+1} - \alpha\| \leq \|S\| \|x^k - \alpha\| \qquad (2.15)$$

Debido a que $\|S\|$ tiene un valor finito, pues las derivadas son acota

das, se puede apreciar de (2.15) que en los algoritmos de primer orden, el error decrecerá cerca de la solución con una relación de tipo lineal. Es - decir los algoritmos de primer orden tendrán una convergencia lineal.

Considerese ahora la clase de algoritmos de segundo orden, en los cuales por definición

$$\frac{\partial g_i}{\partial x_j} = 0 \qquad (2.16)$$

en la región R. En este caso, la ecuación (2.9) se reduce a :

$$g_i(X) - g_i(\alpha) = \frac{1}{2}\sum_{j=1}^{n}\sum_{k=1}^{n}(x_j - \alpha_j)(x_k - \alpha_k)\frac{\partial^2 g_i}{\partial x_j \partial x_k} \qquad (2.17)$$

Ahora si $\frac{\partial^2 g_i}{\partial x_j \partial x_k}$ es continua y acotada en R, se puede definir un escalar finito M tal que :

$$M = \sup_{x \in R}\left|\frac{\partial^2 g_i}{\partial x_j \partial x_k}\right| \qquad 1 \leq i,j,k \leq n \qquad (2.18)$$

aplicando a la ecuación (2.17) el valor absoluto en el miembro izquierdo y la norma $\|\cdot\|_\infty$ al vector $X-\alpha$ del lado derecho se tiene de la ecuación -- (2.18).

$$\left|g_i(X) - g_i(\alpha)\right| \leq \frac{1}{2} n^2 M \|x - \alpha\|_\infty^2 \qquad (2.19)$$

Debido a que $|x_i - \alpha_i| \leq \|X-\alpha\|_\infty$ y como :

$$\left|g_i(X) - g_i(\alpha)\right| \leq \|g(X) - g(\alpha)\|_\infty \qquad (2.20)$$

La ecuación (2.19) se puede escribir como :

$$\|g(X) - g(\alpha)\|_\infty \leq \frac{1}{2}\|X - \alpha\|_\infty^2 n^2 M \qquad (2.21)$$

por lo tanto si $X^{k+1} = g(X^k)$ se tendrá claramente:

$$\|X^{k+1} - \alpha\|_\infty \leq \frac{1}{2} n^2 M \|X^k - \alpha\|_\infty^2 \qquad (2.22)$$

De (2.22) se puede apreciar que para la clase de algoritmos de segundo orden el error cerca de la solución decrecerá siguiendo por lo menos una relación cuadrática. Es decir los algoritmos de segundo orden tendrán una --

convergencia cuadrática. Esto implica que la rapidez de convergencia será bastante mayor que en el caso de convergencia lineal.

Un algoritmo razonable puede al menos, ser linealmente convergente en el sentido de que si X_k es generada por el algoritmo y X_k converge a X^*, entonces para alguna norma $\|\cdot\|$ existe un $\alpha \in (0,1)$ y $k_0 \geq 0$ tal que :

$$\| x_{k+1} - x^* \| \leq \alpha \| x_k - x^* \| \qquad \begin{matrix} k = 0,1, \ldots \\ k \geq k_0 \end{matrix} \qquad (2.23)$$

esto garantiza que el error, eventualmente puede ir decreciendo por el factor $\alpha < 1$. Un algoritmo puede ser superlinealmente convergente en el sentido de que (2.23)tiene alguna secuencia α_k la cual converge a cero

Dennis y More (1974), mostrarón la siguiente propiedad de los métodos de convergencia superlineal :

$$\lim_{k \to +\infty} \| x_{k+1} - x_k \| \Big/ \| x_k - x^* \| = 1 \qquad (2.24)$$

de modo que $x_k \neq x^*$ para $k \geq 0$

La importancia de(2.24) es que de alguna justificación para detener la iteración cuando $\| x_{k+1} - x_k \| \leq \epsilon_1 \| x_k \|$ para alguna tolerancia preestablecida ϵ_1. Este criterio de terminación es frecuentemente usado con uno de la forma $\| F(x_k) \| \leq \epsilon_2$.

El siguiente teorema de Dennis y More (1974) muestra precisamente cuando una iteración es superlinealmente convergente.

TEOREMA .- Suponer que $F: R^n \to R^n$ satisface las siguientes propiedades :

a) El mapeo F es contínuo y diferenciable en una región convexa abierta D.

b) Existe una X^* en D tal que $F(X^*) = 0$ y $F'(X^*)$ es no singular

La notación $F'(X)$ denota la matriz Jacobiana $(\partial_i F_i(X))$ evaluada en X, así que (ayb) garantiza que X^* es la única solución local de las ecuaciones $F(X) = 0$.

Ahora permitase B_k en $L(R^n)$ sea una secuencia de matrices no-singulares. Suponiendo que para alguna X_0 en D la secuencia

$$X_{k+1} = X_k - B_k^{-1} (X_k) \tag{2.25}$$

permanece en D, $X_k \neq X^*$ para $k \geq 0$ y converge a X^* . Entonces X_k converge superlinealmente a X^* si y sólo si :

$$\lim_{k \to \infty} \frac{// \left[B_k - F'(X^*) \right] \left[X_{k+1} - X_k \right] //}{// X_{k+1} - X_k //} = 0 \tag{2.26}$$

La ecuación (2.26) sólo requiere que B_k converga a $F'(X)$ a lo largo de la dirección $S_k = X_{k+1} - X_k$ del método iterativo.

1.4.2 CLASIFICACION DE LOS ALGORITMOS DE CONVERGENCIA

Los sistemas de ecuaciones no-lineales son muy difíciles de resolver por -- cualquiera de las siguientes razones:

1.- Algunas de las funciones no son definidas para ciertos valores de - las variables (las funciones logaritmo y raíz cuadrada, tienen in-- tervalos donde no son definidas).

2.- Algunas soluciones del sistema no son posibles físicamente.

3.- Las funciones son extremadamente no-lineales (debilmente escaladas).

4.- Los sistemas son muy grandes y dispersos.

Shacham (1982), muestra que existen metodos y software que no pueden resolver problemas con las dificultades mencionadas; a menos que el estimado inicial sea muy cercano a la solución.

Los metodos pueden ser clasificados dentro de tres categorías:

I).-Localmente convergentes y que requieren de un buen estimado inicial

II).-Con una región expandida de convergencia

III).-algoritmos de convergencia global para cualquier estimado inicial.

El ejemplo de la figura (5) dará una idea más clara ; es para el - sistema de dos ecuaciones no-lineales mostrada en la misma figura. Existen dos pares de raices reales (1,4) y (4.0715, 0.65027). Cuando se aplica el Método de Newton, si el estimado inicial para X_1 y X_2 está dentro de la región de solución, entonces la convergencia se logra. Por otro lado, si no; el Método de Powell expande la región de convergencia a la frontera deseada. Ahora ; si se utiliza como estimado inicial $X_1^o = 10$ y $X_2^o = 10$, lo cual provoca que el Método de Newton y de Powell fallen, entonces los métodos - de continuación y Homotopia son adecuados; se inicia con la construcción - de una homotopia (h). a relacionar la función F(X) con una familia de parámetros de funciones G(X,T) elegido, tal que G(X,0) = F(X) y la solución de G(X,1) = 0 es conocida o puede obtenerse facilmente por métodos conocidos. Si una homotopia que es lineal se forma en un parámetro de homotopia t_1:

$$h_1 (x_1,x_2,t) = tf_1 (x_1,x_2) + (1-t)\, g_1 (x_1,x_2) = 0 \qquad (2.27)$$

$$h_2 (x_1,x_2,t) = tf_2 (x_1,x_2) + (1-t)\, g_2 (x_1,x_2) = 0 \qquad (2.28)$$

cuando t=0, al inicio de la ruta, h=g=0, donde la solución es X_1^o y X_2^o cuando t=1,h=f=0, donde la rafz X_1 y X_2 será determinada. Si la función de homotopia es continuamente diferenciable y la matriz de derivadas tiene inversa , el teorema de función implicita, garantiza la existencia de un camino continuo que enlace el punto inicial X_1^o y X_2^o a la solución deseada (X_1, X_2). Entonces sólo se necesita seleccionar un apropiado g_1 y g_2 y entonces plancar un esquema para apoyar sobre la ruta mientras se mueve de t= 0 a - t= 1; la eficiencia de este método puede incrementarse, usando métodos de matrices dispersas en trato con el Jacobiano ó dando un algorítmo del tamaño de paso de integración.

Recientemente Shacham desarrolló un algorítmo nuevo. "CONLES" que es capaz de resolver ecuaciones con restricciones físicas y absolutas; las restricciones absolutas se manejan por el algorítmo usando el Método de Newton con longitud de paso restringida ó Broyden. Las restricciones físicas son convertidas a restricciones absolutas o usando funciones penalizadas o manejandolas directamente usando el Método de continuación. Para matrices casi singulares, el algorítmo usa una modificación del Método Levenberg-Marquardt. La eficiencia de la solución puede incrementarse, usando una nueva técnica de descomposición en dos pasos:

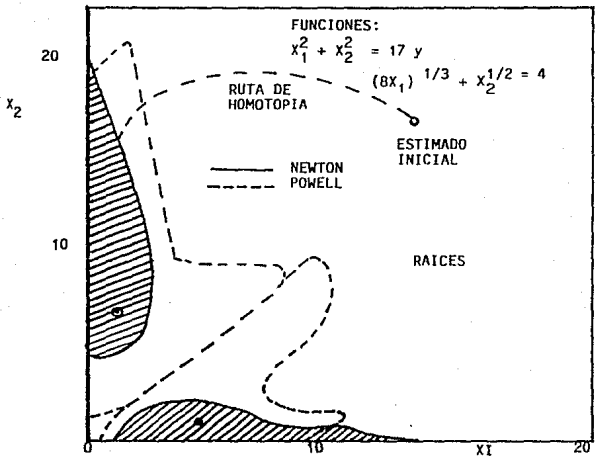


FIGURA No. 5

HOMOTOPIA

REGIONES DE CONVERGENCIA PARA LA SOLUCION DE UN PAR DE ECUACIONES NO-LINEALES

Primero : Algunas de las ecuaciones no-lineales por nuevas variables y adicionando nuevas ecuaciones no-lineales al sistema.

Segundo : La división del subconjunto lineal de ecuaciones se realiza con eliminación Gaussiana.

Esta técnica de descomposición puede usarse para problemas con y- sin restricciones. La implementación numérica del algorítmo es más importante que el algorítmo en si mismo, ya que permite hacer un programa estable y eficiente.

I.5 ENFOQUES DE LA SIMULACION DE PROCESOS

Un proceso químico puede ser visualizado como un diagrama formado por bloques, en el cual cada bloque representa una unidad de proceso (estos diagramas son comunmente llamados diagramas de simulación de procesos), interconectados por arcos, los cuales representan las corrientes de proceso, la figura (6) muestra un ejemplo de un diagrama de flujo de proceso formado por: Un reactor, un separador y dos mezcladores de corriente. Como ya se menciono anteriormente cada unidad de proceso esta modelada por un conjunto distinto de ecuaciones, las variables presentes en estas ecuaciones son las variables de las corrientes de entrada y de salida (x_i, flujo, temperatura y presion), parámetros de los equipos (como son números de etapas, la relación de flujo para una columna de destilación), variables internas (tales como composición y temperatura de cada etapa en una columna de destilación) y propiedades termofísicas.

En aplicaciones típicas, el número de ecuaciones puede variar en un rango de unos cientos, a más de 10,000 . Debido al tamaño de los problemas es necesario el explotar su estructura para hacer los problemas más manejables, de las diferentes formas de la estructura surgen diferentes enfoques para la solución de los problemas de la simulación de procesos a pesar de que los sistemas de ecuaciones resueltos por diferentes enfoques son esencialmente los mismos. Motard (1975), Hlavacek (1977), Rosen (1980) y Evans (1981) han hecho varias revisiones sobre los trabajos realizados en esta área.

Westerberg (1979) clasifica los enfoques de la siguiente forma: Secuencial modular, Simultaneo modular ó doble rompimiento y Orientado a ecuaciones con rompimiento y Orientado a ecuaciones con linearización simultanea.

A continuación se da un análisis sobre cada uno de estos enfoques, en este trabajo se comparan solamente el enfoque modular secuencial y el modular simultaneo.

I.5.1 SECUENCIAL MODULAR

El término secuencial modular fue utilizado por Westerberg (1979) para describir el enfoque utilizado esencialmente en todos los sistemas de simulación.

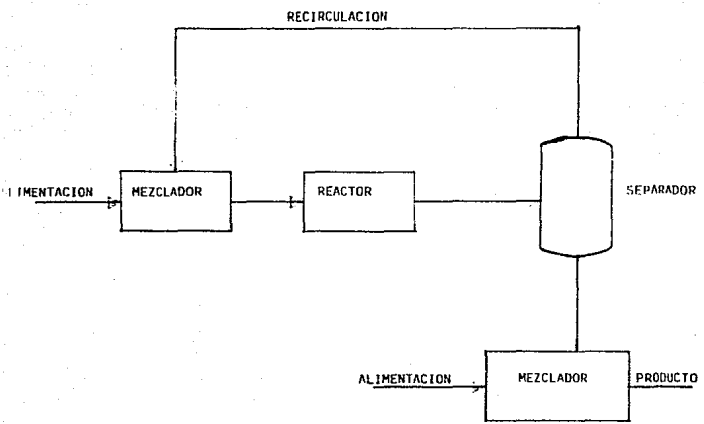


FIGURA No. 6

DIAGRAMA DE FLUJO DE PROCESO

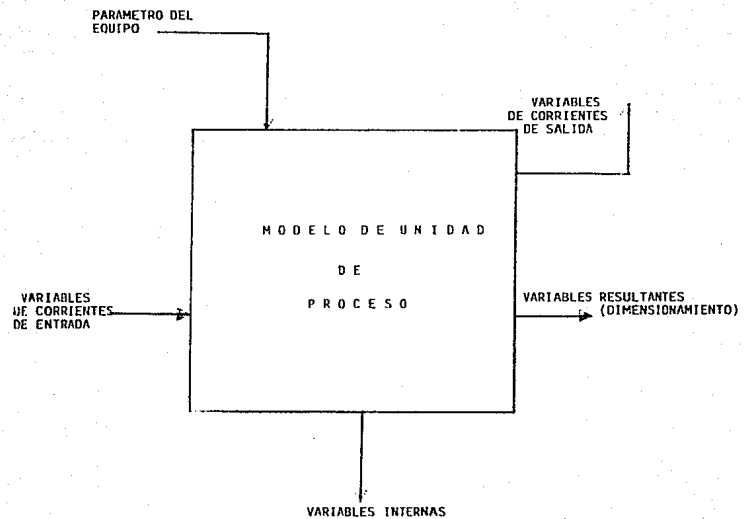


FIGURA No. 7

MODELO DE OPERACION UNITARIA

a nivel industrial. En este enfoque un modelo de cálculo es desarrollado para cada tipo de operación unitaria, el cual calcula las variables de las corrientes de salida como función de las variables de las corrientes de entrada y los parámetros de equipo, en la figura (7) se muestra un modelo.

Estos modelos son entonces llamados, en orden secuencial para simular el proceso. Las corrientes de recirculación son cortadas y llevadas a la convergencia mediante algun algoritmo iterativo, el estimado inicial (valor) de las corrientes de recirculación son proporcionadas por el diseñador. Las especificaciones incognitas deben ser resueltas en forma iterativa hasta que estas son satisfechas, y estas pueden servir para calcular una variable de entrada ó un parámetro de equipo dando una variable de salida.

Para resolver un diagrama de flujo de proceso mediante el método secuencial modular es necesario la partición del diagrama, selección de conjuntos de corte y determinación de la secuencia de cálculo, como ejemplo se considera el ejemplo de la figura (8), el cual contiene mezcladores (M), reactores (R) y separadores (D). Este tiene la característica de que la recuperación de un componente por el separador D_1 será ajustada para cumplir con la concentración deseada en S_6. El flujo de información 1_1 indica que la recuperación en la separación va a ser determinada por la especificación ESPEC.

La idea básica de los procedimientos de partición es la identificación y l condensación de conjuntos de módulos (conocidos como ciclos máximos ó redes i rreducibles) que deben ser resueltos en forma simultánea.

Los métodos para identificación de recirculaciones para propositos de participación pueden ser clasificados en dos tipos: Potencias de matriz adyacente (Kehat y Shacham (1973), la matriz adyacente se forma a partir de la información obtenida de las corrientes de la red de flujo de proceso, un elemento a_{ij} representa el número de corrientes que estan fluyendo de la unidad (i) a la unidad (j). El segundo es el rastreo de rutas, en este método la red de flujo de procesn es analizado tomando como base una unidad y rastreando el flujo de información a las unidades subsecuentes. (Henley y Williams (1973). Estos fueron revisados por Gros, Kaijaluto y Mattsson (1977). En el ejemplo de la figura (8), el subsistema 1 es independiente del subsistema 11, por lo que debe ser resuelto primero el subsistema 1 y posteriormente el subsistema 11.

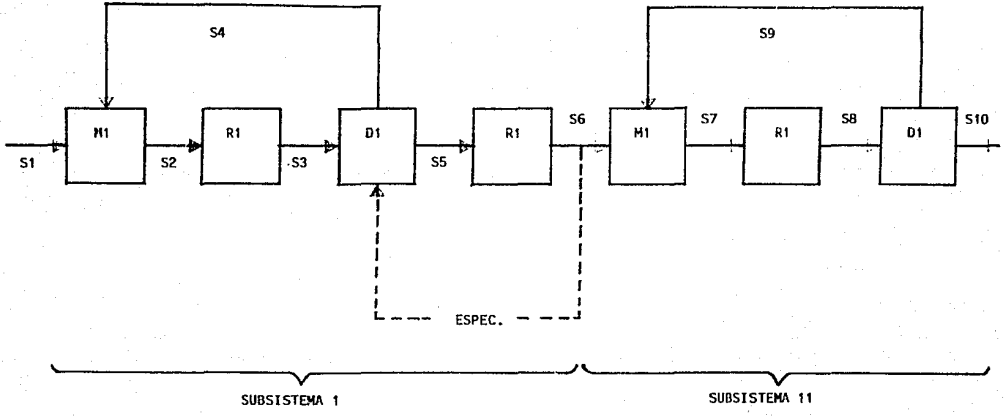


FIGURA No. 8

DIAGRAMA DE FLUJO PARA ILUSTRAR LA ESTRUCTURA COMPUTACIONAL
DEL ENFOQUE SECUENCIAL MODULAR

El rompimiento determina cuales corrientes ó flujos de información pueden ser cortados para convertir el diagrama de flujo en una gráfica acíclica.

En el subsistema I, la corriente S_4 y el flujo de información I_1 son seleccionados.

La agrupación de los cálculos determina que corrientes de corte van a converger simultaneamente y el orden en el cual se agrupan las corrientes de corte para ser llevadas a la convergencia. En el subsistema I, es posible agrupar la convergencia de la corriente de recirculación dentro de la convergencia de la especificación o viceversa, a la corriente de recirculación y la especificación de diseño podrán converger simultaneamente.

Una vez que la agrupación de las corrientes de corte ha sido especificada es necesario determinar la secuencia de cálculo. Finalmente un algorítmo de convergencia apropiado debe utilizarse.

Un ingeniero de proceso puede efectuar los pasos de partición, rompimiento, agrupamiento y determinación de la secuencia de cálculo, por simple inspección del diagrama de flujo, estos no son muy complicados.

Para la partición, el algoritmo de Sargent y Westerberg (1964) en una forma modificada como redes independientes cubierta y presentada por Targan (1972) es muy eficiente y efectiva. Para la selección de las variables de las corrientes de corte el algorítmo desarrollado por Motard y Westerberg (1971) basado sobre los criterios de Upadhye y Grens (1975) y Barchers (1975), se ha probado que son adecuados.

Para la convergencia de las corrientes de recirculación, la mayoría de los algoritmos de convergencia toman la forma;

$$Y^{k+1} = Y^k - t^k J^{k-1} f^k \qquad (2.29)$$

donde :

Y^k = Variable de corte en la iteración k.

$f^k = Y^k - r^k$ (diferencia entre el valor supuesto y el valor calculado para la variable de corte).

t^k = Factor escalar de amortiguamiento (normalmente igual a uno).
$(j)^{-1}$ = Jacobiano (inverso).

El método más conocido es el de sustitución directa (que es lento pero estable) ó métodos de aceleración tales como El método de Wegstein (1971) ó - El método de Broyden (Broyden (1965)), estos métodos se presentan desarrollados en el capítulo III. Cada método tiene su propia representación para la aproximación del Jacobiano (matriz de derivadas parciales). Más adelante se - hablara con más detalle de los métodos numéricos y sus características.

Algunas de las ventajas que presenta este enfoque son:

1).- Debido a que el flujo de información en este enfoque es análogo al flujo de materiales, el metódo es facilmente comprendido.

2).- El enfoque permite la participación del diagrama de flujo en pequeños conjuntos de unidades, para que las secciones dificiles de calcular puedan ser examinadas más profundamente.

3).- Los módulos pueden emplear uno ó más algoritmos especializados para resolver las ecuaciones que describen cada módulo. Como un resultado el cálculo de los módulos puede ser muy eficiente y robusto.

4).- Debido a la forma como son relacionadas las variables y parámetros en los módulos estos son fáciles de construir.

5).- Cuando los cálculos de recirculaciones son bastante lentos para converger, son normalmente poco estables.

Los dos problemas principales que afectan seriamente la eficiencia del - enfoque son:

1) El manejo de especificaciones de diseño.
2) La presencia de multiples circuitos de iteración anidados, como se ve en la figura (10).

Debido a que los circuitos de especificaciones o de control son los extremos en la jerarquía de las iteraciones. Inmediatamente después estan las corrientes de corte, posteriormente los circuitos de iteración de los módulos y finalmente en el nivel más profundo los del cálculo de propiedades físicas requeridas por los módulos.

1.5.2 MODULAR SIMULTANEO

Una alternativa al método secuencial modular es el enfoque modular simultaneo o de doble rompimiento, siendo este un híbrido entre el enfoque secuencial y los orientados a ecuaciones. Como ya se mencionó el secuencial - modular funciona satisfactoriamente en procesos cuyos circuitos de recirculación son independientes. Pero para diagramas de flujo complejos con circuitos de recirculación supuestos y anidados, así como con reestructuraciones de diseño, la técnica secuencial presenta un pobre comportamiento en la convergencia.

Para mejorar el comportamiento en la convergencia con respecto a los - cálculos de circuitos anidados y especificaciones de diseño, manteniendo la estabilidad y la eficiencia de los módulos de cálculo existentes, un concepto de convergencia diferente utilizado, el enfoque modular ha sido estudiado por Rosen (1979). El concepto esta basado en la solución de todas las - corrientes del diagrama de flujo (no únicamente las corrientes de corte) y las especificaciones de diseño (restricciones) simultaneamente, de aquí su nombre " simulación modular simultanea ".

La idea del enfoque es, arrancar con un estimado de la solución del diagrama de flujo, para :

1) Aproximar las soluciones que describen los módulos con un conjunto - de ecuaciones aproximado.
2) Solución de las ecuaciones aproximadas para todas las unidades en el proceso simultaneamente, para un nuevo estimado de la solución del - diagrama de flujo.

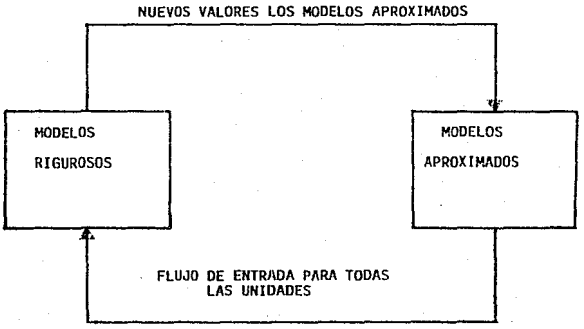


FIGURA No. 9

ALGORITMO DE DOBLE ROMPIMIENTO

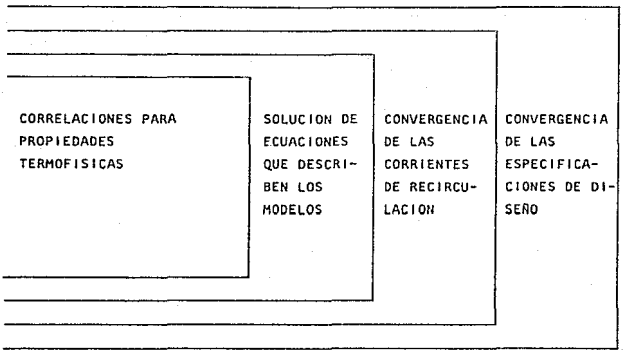


FIGURA 10 .- MULTIPLES CIRCUITOS DE ITERACION ANIDADOS DEL ENFOQUE SECUENCIAL MODULAR

3).- Usar el nuevo estimado para generar un nuevo conjunto de ecuaciones aproximadas.

Este proceso es repetitivo hasta que el número estimado de la solución es igual al estimado anterior dentro de una tolerancia de la convergencia.

Ya que las ecuaciones aproximadas que describen el diagrama de flujo - son resueltas simultaneamente en cada iteración, una condición requerida -- por estas ecuaciones es de que puedan ser resueltas con relativa facilidad. Esencialmente todos los enfoques simultaneos modulares involucran la utili - zación de módulos durante cada iteración para generar una aproximación al Jacobiano, el cual es usado para efectuar una iteración.

El método emplea dos tipos de modelos, Simples y Rigurosos (ver figura 9). Los modelos rigurosos son utilizados en el modular secuencial para determinar parámetros de modelos simples y los modelos simples son resueltos - determinando todas las variables de las corrientes y así permitir que los modelos rigurosos sean llamados nuevamente. En esencia todas las recirculaciones y las especificaciones de control son resueltas simultaneamente.

El primer trabajo utilizó aproximaciones lineales para cada unidad (Nagiey (1964), Rosen (1962), Mahalec (1979).

$$Y = Ax + b \qquad (2.30)$$

Donde:

(X y Y) son las variables de las corrientes de entrada y salida de una unidad respectivamente (figura (11)), y (A) es una matriz diagonal-- de coeficientes lineales y (b) es un vector.

Esta aproximación lineal satisface la condición de fácil solución esta -- blecida anteriormente, ya que ésta es lineal y el número total de variables - de corrientes es mucho menor que el número de variables total del diagrama de flujo. Utilizando módulos de cálculo, la matriz de coeficientes lineales puede ser calculada numéricamente ya sea por el método de fracción de sepa - - -

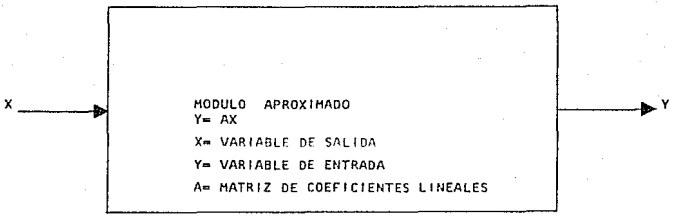


FIGURA 11.- MODELO APROXIMADO UTILIZADO POR ROSEN (1962) PARA EL ENFOQUE SIMULTANEO MODULAR

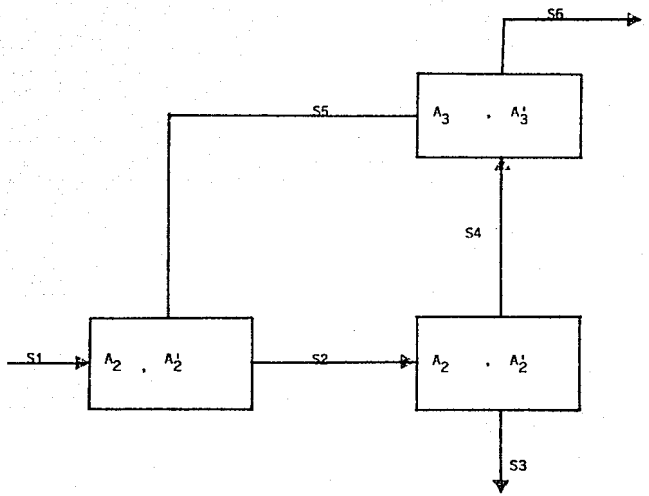


FIGURA 12

PROCESO HIPOTETICO USADO PARA ILUSTRAR LA MATRIZ LINEAL DEL PROCESO (FRACCION DE SEPARACION) - DEL ENFOQUE SIMULTANEO MODULAR.

ración propuesto por Rosen (1962), con :

$$A_{ij} = Y_i / X_i \quad , \quad i = j$$
$$A_{ij} = 0 \quad , \quad i \neq j \text{ y } b = 0 \tag{2.31}$$

tal como se muestra en la figura (12) para un proceso hipotético.

Varios tipos de modelos lineales han sido propuestos siguiendo el desarrollo de Rosen. Raviez y Norman (1964) y Naphtali (1964) proponen el modelo tipo gradiente, similar al requerido en el procedimiento de Newton para la solución de ecuaciones no-lineales.

Mahalec (1979), propone un método tipo gradiente de la forma :

$$A_{ij} = \frac{\Delta Y}{\Delta X_i} \quad y \quad B = Y^O - AX^O \tag{2.32}$$

donde $\Delta Y_i / \Delta X_i$ es calculado mediante perturbación numérica del módulo de cálculo. La estructura computacional de la técnica modular simultanea lineal - con matriz de coeficientes tipo gradiente es ilustrada en la figura (13).

Si las relaciones entrada-salida del módulo unitario son expresadas en términos de la siguiente función residual

$$\delta (X) = Y - f (X) \tag{2.33}$$

entonces podemos ver que la solución de esta función residual en forma simultanea por el método de Newton es esencialmente el mismo que resolver el problema de simulación mediante la técnica simultanea lineal con matriz de coeficientes tipo gradiente.

Nishimura (1968) por otro lado utiliza expresiones analíticas para calcular los coeficientes de los modelos lineales.Las expresiones son únicas - para cada modelo de unidad y son funciones de las variables de las corrientes y los parámetros de equipo.

Umeda y Nishio (1972) compararón los métodos modulares secuenciales y - demostrarón que los modelos analíticos de Nishimura presentan mejores comportamientos en la convergencia que los modelos de fracción de separación,- los cuales muestran signos de inestabilidad. Esto último fue tambien confirmado por Mahalec (1979).

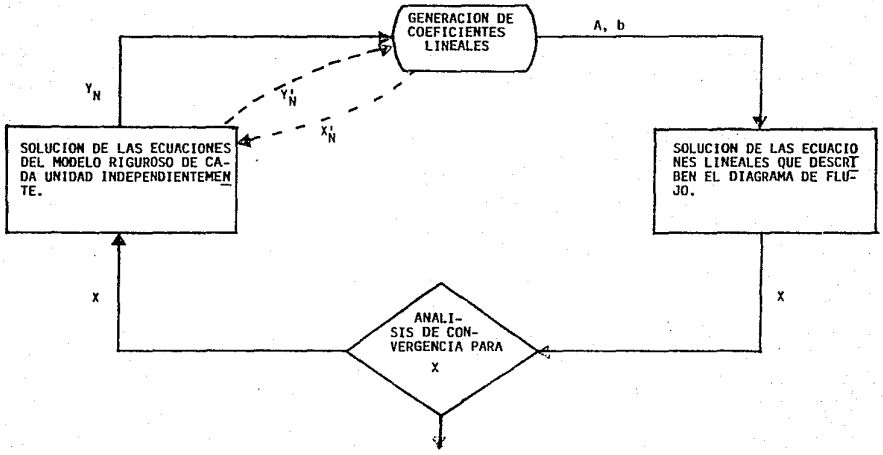


FIGURA 13.- MODELO DE CALCULO PARA EL ENFOQUE SIMULTANEO MODULAR CON MODELOS LINEALES.

Usando el hecho de que la matriz lineal del proceso es en esencia una - matriz jacobiana y tiene estructura dispersa, Mahalec,Klusik y Evans (1979)- emplearon el método de Broyden modificado por Schubert (1970) para aproximar la matriz de coeficientes lineales sin perturbación numérica, lo que reduce el número de ejecuciones de los modelos de cálculo riguroso en cada itera--- ción. Ellos demuestran que esta aproximación presenta una convergencia mucho más rápida para el caso del problema de Cavett (1963) comparada con el modular secuencial con Wegstein Acotado.

Se puede decir que el enfoque modular simultaneo toma las ventajas del - modular secuencial particularmente la de la Heurística para suponer estima-- dos iniciales y para el mejor de los casos en manejos especiales.

Este enfoque es muy especial, debido a la utilización de modelos simplificados para el análisis preliminar de un proceso, el cual es posteriormente verificado mediante el uso de modelos más rigurosos.

1.5.3 ENFOQUE ORIENTADO A ECUACIONES.

La idea básica de los modelos orientados a ecuaciones es, simplemente el agrupar todas las ecuaciones que describen el proceso y resolverlas como un gran sistema de ecuaciones algebraicas no-lineales.

Matematicamente el problema puede ser establecido como

$$\text{Resolver} \qquad F(X,U) = 0 \qquad (2.34)$$

donde $X =$ Vector de variables dependientes.
$U =$ Vector de variables independientes.

Las variables independientes podrán normalmente incluir todos los parámetros de equipos y las variables de las corrientes de alimentación y las - dependientes incluiran todas las variables de las corrientes de producto, - intermedias, internas y resultantes. Comparado con los enfoques modulares --

estos métodos son más flexibles debido a que diferentes sistemas de ecuaciones no-lineales deben ser resueltos, y ellos son potencialmente más eficientes debido a que eliminan la ineficiencia de la iteración de circuitos anidados. Como se muestra en la figura (14). Sin embargo las ventajas de los metodos modulares mencionados anteriormente se pierden, ya que los sistemas de ecuaciones son dispersos y estructurados se deben explotar tales caracterfsticas (Sargent (1980)).

El punto de partida para el método de ecuaciones orientadas en general es el mismo diagrama de bloques del modelo usado por el método secuencial modular (Figura (8)). Sin embargo en lugar de preparar módulos que calculen las variables de salida como función de las variables de entrada, el enfoque orientado a ecuaciones requiere de procedimientos que generen y representen las ecuaciones para cada módulo. Las ecuaciones pueden entonces ser alimentadas a un procedimiento eficiente de solución de ecuaciones.

Dentro de los métodos orientados a ecuaciones se pueden encontrar dos tendencias, una es la de las técnicas orientadas a ecuaciones con rompimiento, y otras con linearización simultanea.

En el enfoque orientado a ecuaciones con rompimiento, algunas ecuaciones son utilizadas para eliminar algunas variables con el fin de reducir el número de variables que pueden ser cortadas o iteradas.

El sistema de ecuaciones no-lineales reducido es entonces resuelto por medio del método de Newton o Quasi-Newton. Un sistema experimental que implemento este enfoque es Speedup (Perkins y Sargent (1982)). Como fue señalado por Lin y Mah (1977), debido a la larga cadena de calculos que tipicamente existen entre los valores de corte supuestos y los residuales en las ecuaciones de corte, problemas de sensibilidad que pueden ocacionar problemas de divergencia incluso para estimados iniciales muy cercanos a la solución. Tambien el trabajo de Stadtherr y Wood (1982) indica que el rompimiento puede únicamente reducir el número de ecuaciones que son resueltas simultaneamente por un factor de alrededor de cuatro.

En el enfoque orientado a ecuaciones con linerización simultanea, todas las ecuaciones que describen el proceso son resueltas simultaneamente

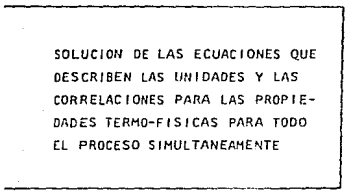


FIGURA 14.- ESQUEMA DE CALCULO PARA LOS ENFOQUES ORIENTADOS A ECUACIONES

usando los metodos de Newton o Quasi-Newton. Algunos sistemas experimentales que implementaron este enfoque son : Quasilin (Gorczynsky (1979), Ascend II (Benjamin (1980)) y Sequel (Stadtheir y Hilton (1982)). Debido a que el número de ecuaciones no-lineales a resolver simultaneamente es muy grande , todos estos sistemas usan técnicas para matrices dispersas para resolver los sistemas linearizados de ecuaciones lineales.

Debido a que todas las ecuaciones son resueltas simultaneamente,los problemas de sensibilidad del caso anterior son eliminados. Alternativamente, en los métodos orientados a ecuaciones, el problema puede ser formulado como un problema de optimización :

$$\text{Maximizar} \quad P(X,U)$$
$$\text{con} \quad F(X,U) = 0$$

más algunas restricciones impuestas por el problema de optimización. La restricción de igualdad $F(X,U) = 0$ es el mismo conjunto de ecuaciones como fue escrito anteriormente, pero en lugar de especificar arbitrariamente todas las variables independientes, ellas son seleccionadas para maximizar una función objetivo $P(X,U)$ apropiada. Westerberg (1979) señalo que es más natural formular los problemas de diseño como un problema de optimización.

Finalmente se puede decir que este enfoque orientado a ecuaciones no ha sido hasta la fecha utilizado en algun simulador industrial por varias de las siguientes razones :

1) Requiere de buenos estimados iniciales, y debido a la generalidad, es difícil generar buenos estimados iniciales automáticamente.
2) Este puede ser menos seguro debido a que sólo usa una sola técnica para resolver todas las ecuaciones.
3) No toma en cuenta las ventajas de los sistemas existentes.
4) Es más difícil de implementar debido al uso de técnicas de matriz dispersa.
5) Los requerimientos de memoria cuando es implementado es mayor que en los enfoques modulares.
6) Cuando los cálculos presentan dificultades la información disponible

es muy escasa para localizar la fuente de falla.

Los enfoques descritos anteriormente han sido implementados en sistemas para la simulación de procesos a régimen permanente y algunos tambien a la - simulación dinámica de procesos.

En el presente trabajo la mayoría de la información comenta que actualmente gran número de los simuladores comerciales son del tipo secuencial modular, en la figura (15) se comparan los procedimientos de solución contra la generalidad de los modelos para los enfoques anteriormente expuestos.

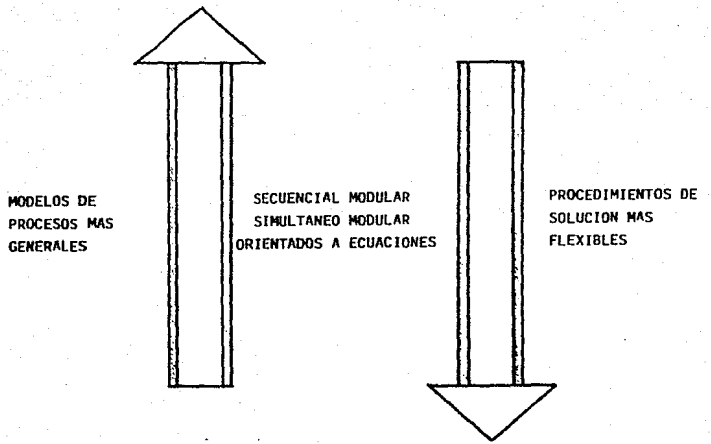


FIGURA 15 .- COMPARACION DE LOS PROCEDIMIENTOS DE SOLUCION VS GENERALIDAD DE LOS MODELOS PARA LOS ENFO - QUES UTILIZADOS EN LA SIMULACION DE PROCESOS.

C A P I T U L O I I

-- SIMULADORES --

11.1 Descripción general

A continuación se da una descripción de las características de los principales simuladores utilizados actualmente. Así mismo, en la tabla 1 se enlistan los programas desarrollados por instituciones académicas, y en la tabla 11 los desarrollados por empresas privadas.

A cada programa se le asigna una categoría consistiendo de una letra para indicar el desarrollo y uso presente, y un número indicando la cantidad de información disponible.

Desarrollo y uso:

a).- Presente en uso
b).- Uso no mayor ó suplantado por un programa reciente
c).- Uso incierto

Información disponible

1).- Escasa información
2).- Regular información
3).- Bastante información

La tabla 111 contiene detalles de los programas presentados en las tablas 1 y 11. Los símbolos estan definidos por las siguientes claves:

General

Tipo: B= Modo en lote
O= Modo en línea
M= Facilidad multicorrida

Area de aplicación:
PPD= Diseño de procesos preliminar
DD = Diseño detallado
OP = Planta en operación

PC = costo preliminar

DC = costo detallado

E = evaluación económica total

Problemas de tamaño máximo

Los problemas son normalmente dimensionados para manejar un cierto problema máximo.

U = Número de unidades

S = Número de corrientes

C = Número de componentes

Datos de entrada ;

Se pueden tener varias opciones dependiendo de la máquina a usar :

1.- Verificando la factibilidad, calculando una vez las variables

2.-Balanceando la información de los diagramas de flujo.

3.- Llevando los cálculos sobre el diagrama de flujo balanceado e impreso.

Convergencia de recirculación ;

SR = unidad de subrutina, que puede insertarse al control de convergencia.

EXEC = convergencia controlada por ejecutivo.

Unidad de subrutina ;

P = ejecución

D = puede usarse la subrutina de diseño

A = Subrutina para alguna combinación de variables

TS = área de almacenamiento temporal, creadas por corrientes y parámetros asociados con una unidad.

B = datos no requeridos fuera del bloque

Propiedades físicas ;

HK = manejando rutinas

PP = paquete de propiedades físicas

DB = banco de datos de componentes estandar

NO = todas las propiedades calculadas pueden escribirse en la unidad subrutina.

Facilidad de costo:

SE= Análisis económico simple
DE= Análisis económico detallado
NO= Todos los cálculos económicos en una subrutina

Optimización:

SE= Análisis de sensitividad automática
OPT= Facilidad de optimización automática

Fijo= Formato fijo
Libre= Formato libre
Conv = Convencional
Clave= Los vocables claves especifican el tipo de datos a seguir
Datos B= Datos del bloque, facilitan la existencia para la entrada de grandes segmentos de datos, tales como propiedades físicas.

Fichas= Basadas en fichas

Topología de la planta:

PM= Matriz de proceso
SCM= Matriz de conexión de corrientes

Verificación= Una verificación puede constar de unas pruebas lógicas para los datos de entrada, tal como la factibilidad de la lista del orden de cálculo. El nivel de salida requerido puede fijarse por el usuario.

Salida final:

U= Valores de las corrientes impresas basadas unidad por unidad
ST= Tabla de corrientes impresas
RG= Generadores de reportes especiales

Salida intermedia:

FI= Fijado sobre entrada a un caso particular
V= Variable durante la ejecución
E= Mensajes de error y precauciones durante la ejecución

Facilidad de orden:

R= Facilidad de reordenación
G= Facilidad de agrupación

TABLA 1

PROGRAMAS DESARROLLADOS POR INSTITUCIONES ACADEMICAS**

NOMBRE CORTO	NOMBRE TOTAL	FUENTE	CATEGORIA	R E F.
FACER	RUTINA EVALUADORA DE PROYECTOS Y PROGRAMAS	UNIVERSIDAD PURDUE	A3	(19)
MACSIM	————	UNIVERSIDAD MC. MASTER	B1	(19)
PACER (MAD)	————	UNIVERSIDAD DE HOUSTON	B1	(19)
PACER MK2 (PROTRAN)	————	UNIVERSIDAD	A1	(19)
GEMCS	SISTEMA DE COMPUTACION DE ADMON. E ING. EN GENERAL.	UNIVERSIDAD MC. MASTER (COL. IMP.)	A3	(19)
SPEED-UP (1964)	————	COLEGIO IMPERIAL	C1	(49)
CONCEPT	TECNOLOGIA Y COMPUTACION DE PROCESOS QUIMICOS SOBRE LINEAS	UNIVERSIDAD DE CAMBRIGDE (CAD)	A2	(19)
CHESS	SINTESIS DE SIMULACION EN ING. QUIMICA	UNIVERSIDAD DE HOUSTON	A2	(19)
SLED	LENGUAJE SIMPLIFICADO PARA ING. DE DISEÑO	UNIVERSIDAD DE MICHIGAN	A1	(19)
ASCEND	SISTEMAS AUTOMATICOS PARA DISEÑO EN ING. QUIMICA.	UNIVERSIDAD DE FLORIDA	A2	(19)
GDP	PROGRAMAS DE DISEÑO GENERAL	UNIVERSIDAD DE WESTERN ONTARIO	A1	(19)
GEPDS	————	UNIVERSIDAD DE OKLAHOMA	B1	(19)
ESSPROS	SIMULADOR DE PROCESOS EN EDO. ESTACIONARIO	UNIVERSIDAD DE SCOTLAN EDINBURGH	A1	(19)
ASPEN	UN SISTEMA AVNZADO PARA ING. DE PROCESOS	TECNOLOGICO DE MASSACHUSETTS	A3	(15)
SGP/ZAR	SIMULADOR GENERAL DE PROCESOS ZARAGOZA	UNIVERSIDAD NAL. DE MEX.	A2	

TABLA II

PROGRAMAS DESARROLLADOS POR COMPAÑIAS PRIVADAS

NOMBRE CORTO	NOMBRE TOTAL	FUENTE	CATEGORIA	R E F.
PACER 245	———	CORP. SISTEMA DIGITAL HANOVER	A2	(19)
NETWORK 67	-------	I.C.I LTD. CIRL	B3	(19)
FLOWPACK	-------	" " " " " "	A2	(19)
--------	PAQUETE DE EVALUACION Y EXPLORACION.	" " " " " "	A2	(19)
BASYS	------	" " " " " "	A1	(19)
GPS	SIMULADOR GENERAL DE PROCESOS	PHILLIPS	A1	(19)
PDA	PROGRAMA DE ANALISIS DE DISEÑO	" " " "	A1	(19)
FLOWTRAN	------	MONSANTO	A2	(55)
PROVES	SISTEMA DE EVALUACION Y ESTIMACION DE PROYECTOS	CORP. DIAMON SHAMROCK	A3	(19)
CHEOPS	SISTEMA DE OPTIMIZACION EN ING. QUIM.	COMP. DESARROLLADORA SHEL	C1	(19)
-------	DIAGRAMAS DE FLUJO FLEXIBLES	M. W. KELLOG CO.	B2	(19)
GFS	PROGRAMA GENERAL DE FLOWSHEETING	M. W. KELLOG CO.	A3	(19)
GIFS	SIMULACION DE FLUJO GENERALIZADA INTERRELACIONADA	CORPORACION DE SERVICIOS BUREAN	B1	(19)
CHIPS	SISTEMA DE PROCESO DE INFORMACION EN ING. QUIMICA	" " " " " " "	A1	(19)
-----	SISTEMA CHEVRON	CO. INVESTIGADORA CHEVRON	B1	(19)

TABLA II

NOMBRE CORTO	NOMBRE TOTAL	FUENTE	CATEGORIA	R E F.
CAPS	SIMULACION DE PROCE-- SOS AUTOMATICA CHEM SHARE	CORP. CHEM SHARE	C1	(19)
FLOWSIM	----------	CORP. DE SISTEMAS TECNICOS	C1	(19)
PROCESS	----------	SCIENCES OF SIMULA- CION INC.	A3	(5)
SIMPROC	SIMULADOR GENERAL DE PROCESOS	INSTITUTO MEXICANO- DEL PETRO LEO		(23)
DESIGN/2000	----------	CORP. CHEM SHARE	A3	(19)
ASPEN PLUS	SISTEMA AVANZADO PARA LA INGENIERIA DE PROCESOS	ASPEN TECNOLOGY INC.	A3	(19)

TABLA III

DATOS DE ALGUNOS PROGRAMAS PRESENTADOS EN LAS TABLAS I Y II

GENERAL

NOMBRE DEL PROGRAMA	1.- TIPO	2.- AREA DE APLICAC'
PACER (1968)	B,M	PPD,DD,OP
GEMCS	OL,B,M	PPD,DD,OP,PC
SPEED-UP	OL,B	DPD,DD,OP
NETWORK 67	B	-----
PAQUETE DE EXPLORACION y EVALUACION	B,OL,M	PPD,PC
CONCEPT	B,OL,M	PPD,DD,OP,PC,DC
PACER 245	B,OL,M	PPD,DD,OP,PC,DC
PROVES	B	PPD,DC,E
GPS	B,M	PPD,DD,OP
FLOWTRAN	B	PPD,DC
ASPEN	B,M	PPD,DD,OP,PC,DC,E
PROCESS	B	PPD,DD,OP
SIMPROC		PPD,DD,OP
DESIGN/2000	B,M	PPD,OP,DD

GENERAL

NOMBRE DEL PROGRAMA	3.- TAMAÑO DEL PROBLEMA MAXIMO
PACER (1968)	U = 20 S = 30 C = 10
GEMCS	U = 25 S = 50 C = 25
SPEED-UP	----------
CHESS	U = 100 S = 50 C = 20
NETWORK 67	U = 50 S x C = 500
PAQUETE DE EXPLORACION Y EVALUACION	U = 50 S = 150 C = 20
FLOWPACK	U = 100 S x (2+C) = 2000 C = 25
CONCEPT	U = 50 S = 150 C = 20
PACER 245	----------
PROVES	U = 20 S = 30 C = 20
GES	U = - S = - C = 48
FLOWTRAN	----------
ASPEN	ILIMITADO
PROCESS	U = 75 S = 150 C = 50
SIMTROC	----------
DESIGN/2000	----------

DATOS DE ENTRADA

NOMBRE	1.- TIPO	2.- TOPOLOGIA DE LA PLANTA	3.-VERIFICACION
PACER (1968)	FIJO	PM	NO
GEMCS	LIBRE,FIJO,CONV,FICHA	PM	NO
SPEED-UP	LIBRE,CLAVE,CONV.	SCM	ALGUNOS
CHESS	LIBRE,CLAVE,CONV.	PM	ALGUNOS
NETWORK 67	DEFINIDO POR EL USUARIO CON PALABRAS CLAVES ESTANDAR	SCM	ALGUNOS
FLOWPACK	LIBRE,CLAVE,B DATOS	SCM	COMPRENSIVO
PAQ. DE EXPL. Y EVALUACION	LIBRE.CONV,CLAVE,FICHA	PM	ALGUNOS
PACER 245	LIBRE,CONV,FICHA	PM	CPMPRENSIVO
PROVES	FIJO,CLAVE,FICHA	PM	--------
GFS	FIJO,CLAVE,FICHA	PM	NO
FLOWTRAN	LIBRE	PM	COMPRENSIVO
ASPEN	LIBRE	SCM	COMPRENSIVO
PROCESS	FIJO,LIBRE,CLAVE	----	COMPRENSIVO
SIMPROC	LIBRE	SCM	-----
DESIGN/2000	FIJO	SCM	NO

SALIDA

NOMBRE DEL PROGRAMA	1.- FINAL	2.- INTERMEDIA
PACER (1968)	ST	FI
GEMCS	U,RG	FI,E,U
SPEED-UP	---	----
CHESS	ST	FI,E
NETWORK 67	DEFINIDO POR EL USUARIO	
FLOWPACK	ST	FI,E
PAQ. DE EXPLORACION Y EVALUACION	ST,U	E,V
CONCEPT	ST,F	V,E
PACER 245	ST, F	V,E
PROVES	ST,R,G	NO
GFS	U	FI,E
FLOWTRAN	ST	—
ASPEN	ST	FI,V,E
PROCESS	ST	FI,V,E
SIMPROC	ST	FI,V
DESIGN/2000	ST	FI

FACILIDAD DE ORDENAMIENTO

NOMBRE DEL PROGRAMA	FACILIDAD DE ORDENAMIENTO
PACER (1968)	SIMPLE R (MAXIMO 3 LOOPS)
GEMCS	NO
SPEED-UP	COMPRENSIVO R,G,C
CHESS	NO
NETWORK 67	SIMPLE, G
FLOWPACK	COMPRENSIVO R,G
PAQ. DE EXPLORACION Y EVALUACION	NO
CONCEPT	COMPRENSIVO
PACER 245	COMPRENSIVO
PROVES	NO
GFS	NO
FLOWTRAN	-------
ASPEN	R,G,C
PROCESS	R,C,C
SIMPROC	R,G
DESIGN/2000	R,G

FASES DE CALCULO

NOMBRE	1.-FASES	2.-CONVERGENCIA DE RECIRCULACION	3.-METODOS DE CONVERGENCIA
PACER (1968)	2,3	EXEC	NO
GEMCS	2,3	SR	NO
SPEED-UP	----	EXEC	----
CHESS	2	EXEC	SIMPLE SECANTE LIMITADA NO INTERACTIVA
NETWORK 67	2	SR	NEWTON-RAPSON
FLOWPACK	1,2,3	EXEC	SECANTE APROX. SECANTE EVOL. ACC. REPSUB
PAQ. DE EXPL.	2,3	EXEC	NO
CONCEPT	2,3	EXEC	LINEARIZACION INTERPOLACION LINEAL
PACER 245	1,2,3	EXEC	SUPLIDA POR EL USUARIO
PROVES	2	EXEC	NO
GFS	---	SR	---
FLOWTRAN	---	----	SIMPLE LIMITADO SECANTE NO INTERACTIVA
ASPEN	2,3	EXEC	TODOS
PROCESS	2,3	EXEC	NEWTON-RAPSON
SIMPROC	1, 3	EXEC	SUSTITUCIONES SUCESIVAS
DESIGN/2000	---	EXEC	---

FASES DE CALCULO

NOMBRE DEL PROGRAMA	4.- UNIDADES LOGICAS
PACER (1968)	NO
GEMCS	NO
SPEED-UP	----
CHESS	NO
NETWORK 67	NO
FLOWPACK	NO
PAQ. DE EXPLORACION Y EVALUACION	NO
CONCEPT	----
PACER 245	5
PROVES	NO
GFS	4
FLOWTRAN	SI
ASPEN	SI
PROCESS	SI
SIMPROC	SI
DESING/2000	SI

UNIDADES DE SUBRUTINAS

NOMBRE	1.-BIBLIOTECA ESTANDAR	2.- ESTRUCTURA FLEXIBLE	3.- ACCESO DE DATOS
PACER (1968)	10	P	TS
GEMCS	6	P	TS
SPEED-UP	--	P,D,A	--
CHESS	10	P	TS
NETWORK 67	NINGUNO	P,D	LISTA DE ARGUMENTOS
FLOWPACK	17	P,D	B (UNIDADES DE DATOS)TS (CORRIENTES)
PAQ. DE EXPL. Y EVALUACION	6	P,D	LISTA DIRIGIDA
CONCEPT	10	P	B
PACER 245	150	P,D	T,S
GPS	11	P	---
PROVES	4	P	---
FLOWTRAN	42	P	----
ASPEN	SI	P,D,A	TS,B
PROCESS	SI	P,D,A	TS,B
SIMPROC	SI	P,D	TS
DESING/2000	SI	P,D,A	TS

PROPIEDADES FISICAS

NOMBRE DEL PROGRAMA	1.-FACILIDAD ESPECIAL	2.-SERVICIOS S/B
PACER (1968)	NO	NO
GEMCS	HR (SIMPLE)	NO
SPEED-UP	------	-----
CHESS	HR/DB/PP(62 HIDROCARBUROS)	6
NETWORK 67	NO	NO
PAQ. DE EXPL.	NO	NO
CONCEPT	HR/DB(63 PRINCIPALMENTE HIDROCARBUROS)	5
PACER 245	HR/DB/PP APROX. 100 COMPONENTES EN EL BANCO DE DATOS	20
PROVES	HR/SIMPLE (SOLO EN FASE DE DIMENSIONAMIENTO Y COSTO)	NO
FLOWTRAN	HR/DB/PP MAS DE 200 COMPONENTES	SI
GFS	HR/DB/PP 48 HIDROCARBUROS PRINCIPALMENTE	6
ASPEN	HR/DB/PP	SI
PROCESS	PP/DB 600 COMPONENTES	SI
SIMPROC	PP	NO
DESING/2000	PP	SI

ECONOMICOS

NOMBRE DEL PROGRAMA	1.-FACTIBILIDAD DE COSTO	2.-ANALISIS ECONOMICO	3.-OPTIMIZACION
PACER (1968)	SC	NO	NO
GEMCS	SC	NO	HOOKE,JEEVES SIMPLE
SPEED-UP	-----	-----	----
CHESS	NO	NO	OPT
NETWORK 67	NO	NO	NO
FLOWPACK	SC	NO	NO
PAQ. DE EXPL. Y EVALUACION	SC	NO	NO
CONCEPT	NO	NO	NO
PACER 245	DC	SE,DE	MODULAR
PROVES	DC	COMPRENSIVO	SA (SOBRE CIERTAS - CLAVES DE PARAMETROS ECONOMICOS
FLOWTRAN	DC	SI	NO
GPS	NO	NO	NO
ASPEN	DC	DE	SA
PROCESS	---	----	-----=
SIMPROC	---	----	------
DESING/2000	NO	----	------

II.1.1 FLOWTRAN

El sistema FLOWTRAN es un ajuste completamente integrado de programas escritos en FORTRAN.

El uso central del FLOWTRAN es la corriente de información la cual se indica en la figura (16), y los bloques de operación los cuales calculan las corrientes de salida a partir de las de entrada, cada uno de los bloques evalua una larga lista de propiedades físicas para llevar a cabo la determinación de punto de rocío y punto de burbuja.

El simulador FLOWTRAN se compone por :

1.- Simulador de procesos ; Este programa traslada la información sobre la descripción del FLOWTRAN a un diagrama de flujo de proceso dentro de un programa de computadora para ser ejecutado.

2.- Programa de propiedades físicas PROPTY ; Este programa toma datos de propiedades y calcula las constantes para las correlaciones de propiedades físicas, usadas en el simulador FLOWTRAN .

3.- Programa de equilibrio de fases líquido-vapor ; Calcula los parámetros para el coeficiente de actividad en fase líquida.

4.- Programa de recuperación de información ; Almacena las constantes de propiedades físicas del PROPTY en un orden público o privado, luego las recúpera para el simulador FLOWTRAN.

Para llevar a cabo la simulación FLOWTRAN, el nombre y tipo de cada operación unitaria que en el proceso se requiere, el orden de cálculo y el nombre de cada corriente de entrada y salida son necesarios:

1) Componentes químicos usados en el proceso.
2) El diseño de unidades y las variables de operación, tal como el número de platos en una torre y áreas en cambiadores de calor.
3) Composiciones y condiciones en cada corriente.

FIGURA 16

INDICE DE VECTOR	USO
1	Flujo del componente 1, lb mol / hr.
2	Flujo del componente 2, lb mol / hr.
	. . .
N	Flujo del componente N, lb mol / hr.
N+1	Flujo total, lb mol / hr.
N+2	Temperatura de la corriente, ^{o}F
N+3	Presión de la corriente, psia
N+4	Entalpia de la corriente, BTU / hr.
N+5	Fracción de vapor (molar)
N+6	Nombre de la corriente

NOTA.- El máximo número de componentes (N) es 25.

Información de la composición de una corriente.

Con esta información, el sistema FLOWTRAN calcula la operación de la - planta entera en estado estable. Se puede obtener el capital y costos de operación de cada pieza del equipo, junto con un estado de pérdidas y ganancias de la planta entera.

Unidades FLOWTRAN : Una línea es requerida para cada unidad, cada línea empieza con la palabra BLOCK y muestra el nombre de la unidad, el tipo de unidad y los nombres de las corrientes de entrada y salida, con cada nombre separado con uno o más espacios. Para los componentes que se incluyen - en el banco de datos sólo se necesitan sus nombres siendo precedidos por la palabra RETR, indicando que los datos son recuperados a el banco de propiedades físicas.

Todos los parámetros se introducen en líneas comenzando con las letras PARAM seguida por la unidad a la cual el parámetro aplica, cada bloque tiene un resúmen de parámetros que se listan en el resúmen del bloque. La composición, temperatura y presión de cada corriente se introducen en las líneas que cierran la información de parámetros, la velocidad de flujo se introduce en las líneas comenzando con las palabras POUND ó MOLES, seguidos - por el nombre de la corriente y el número del primer componente sobre la línea.

- Unidades de Operación FLOWTRAN:

Información de retroalimentación - la unidad de control ; Cuando el parámetro de diseño usado en las unidades de operación no corresponde al valor de diseño deseado, FLOWTRAN permite el uso de unidades de control, estas son unidades monitor de cualquier partida en una corriente y permite la manipulación de parámetros de cualquier subcorriente para lograr el valor - deseado en cualquier corriente monitoreada.

Unidad de convergencia : División de corriente ; FLOWTRAN requiere -- que el uso específico del orden de computación de las unidades usadas para simular el proceso. En el caso de las corrientes de recirculación, la mayoría se decide presentarlas, donde esta la división del sistema y haciendo - un estimado de las corrientes divididas. Puede entonces usarse una unidad -

de convergencia, estas unidades calculan nuevas corrientes de salida (valores estimados) basados sobre las corrientes de entrada (valores calculados), el procedimiento continua hasta que las corrientes de entrada igualan a las corrientes de salida. El usuario puede seleccionar los puntos de división con eficiencia, puesto que este conocimiento conduce a seleccionar las corrientes más factibles.

El preprocesador FLOWTRAN :

El usuario de FLOWTRAN usa un lenguaje simplificado para indicar el sistema con su configuración de diagrama de flujo, que parámetros se desean y que componentes se desean usar. El preprocesador toma esta información y la convierte dentro de un programa FORTRAN, compuesto principalmente por una serie de proposiciones CALL, entonces el programa es compilado y vinculado.

El preprocesador de propiedades físicas y la columna de propiedades físicas :

Probablemente el mayor tiempo consumido y la parte más difícil de la simulación es la recolección y desarrollo de los datos adecuados de propiedades físicas. Algunas características de este sistema son :

1) Una columna de propiedades físicas conteniendo datos de 180 compuestos orgánicos e inorgánicos.
2) Una recuperación automática de datos necesarios sobre el reconocimiento del nombre de un componente en una simulación FLOWTRAN.
3) Uso de correlaciones y ecuaciones avanzadas que pueden describir las propiedades de todas las clases de compuestos sobre un alto rango de temperatura y presión.
4) Capacidad y presición en el manejo de sistemas cuyas mezclas de líquidos se encuentran en el rango ideal, moderadamente no ideal, fuertemente no ideal y parcialmente inmiscible.

II.1.2FLOWPACK II

En el sistema FLOWPACK, dos áreas reciben gran énfasis :

La recuperación de energía y estudios de contaminación ambiental. El -- paquete puede ser fácil de usar para alguien no experto y tiene un rango de modelos no estandar, un sistema de banco de datos flexibles para propiedades físicas. Las rutinas para cálculos termodinamicos, pueden ser resultados -- confiables en todo caso, incluyendo las regiones críticas, retrogradas y fases multiples.

FLOWPACK II debe ser apto para una variedad de procesos, modelos grandes y pequeños, para manejar sistemas con varias fases incluyendo sólidos. Se -- pone particular atención en :

1.- Un lenguaje con entrada fácil de usar
2.- Un chequeo de datos comprensivo con diagnóstico significativo.
3.- Biblioteca de módulos de operaciones unitarias y correlaciones de -- propiedades físicas.
4.- Banco de datos fuerte y flexible de propiedades físicas.
5.- Una alta confiabilidad del sistema.
6.- Una armazón simple que permite la fácil adición de nuevos modelos de operaciones unitarias, correlaciones de propiedades físicas, etc..

La estructura adoptada por FLOWPACK II tiene tambien para la máxima eficiencia computacional :

Unidades de partición y separación .- Salvan tiempo, espacio y se asegura que se resuelven las mínimas ecuaciones independientes, esto reduce el -- riesgo de problemas computacionales de inconsistencia e interdependencia de ecuaciones :

-- Fuertes técnicas de solución de ecuaciones de primero y segundo orden

-- Control opcional sobre la estructura interna de la solución, por al-- teración de los datos antes que el programa.

-- Técnicas nuevas para calcular propiedades termofísicas y operaciones unitarias.

-- Una estructura de almacenamiento, flexible y compacta, la cual remueve las limitaciones sobre el número de unidades, corrientes y compues tos.

FLOWPACK II aparece ante el usuario como un sistema " secuencial modular", es la forma como un ingeniero concibe su planta, como una serie de operaciones unitarias conectadas entre si, los sistemas secuenciales modulares son extremadamente ineficientes y FLOWPACK II contiene un diseño muy - sofisticado para vencer esta desventaja :

-- Una subred con la cual se facilita la estructura computacional de -- los modelos de operaciones unitarias.

-- Además de proveer la opción de convergencia simultánea, permite una aproximación nueva a los cálculos de diseño.

II.1.3 PROCESS

PROCESS es un sistema integrado de métodos programados y datos para -- los cálculos de balances de masa y energía en estado estacionario en plan-- tas de proceso químicas y petroquímicas. El programa escrito en FORTRAN IV y conteniendo 250 subrutinas, esta diseñado para ser modular y de fichas o- rientadas.

El banco de datos contiene más de 600 componentes puros orgánicos e -- inorgánicos, estan incluidos los datos de más de 400 compuestos, adquiridos del sistema de datos de propiedades físicas, PPDS.

PROCESS tiene un extensivo rango de metodos para la predicción de valo- res de equilibrio (K), entalpía (H), entropía (S) y densidad, los metodos - disponibles se muestran en la tabla IV .

Los modelos de PROCESS para columnas pueden ser usados para absorbedo- res, fraccionadores criogénicos , fraccionadores tales como desmetanizado-- res multialimentación, columnas de destilación azeotrópica, torres de acei- te crudo y vació, cracking catalítico . La columna puede tener multiple ali mentación, productos, líquidos y vapores recirculados, cambiadores interfa- se y enfriadores. Los tipos de especificación incluyen :

-- Relación de productos.- Base molal, masa y volumen.
-- Pureza ó recuperación de componentes de producto uno, ó un grupo de - componentes sobre una base : molal, masa y volumen.
-- Cargas en platos.- Interfase vapor-líquido sobre una base : molal, ma sa y volumen.
-- Calentadores, enfriadores.

Utiliza el algoritmo de Newton-Rapson como técnica de convergencia, pa- ra metodos cortos en destilación PROCESS tiene dos modelos ; Fenske total - reflujo y Underwood mínimo reflujo, los dos modelos Fenske incluye un mode- lo regular fraccionador,adicionalmente un modelo multicolumna que es utili- zado para satisfacer requerimientos de fraccionamiento en la predicción de columna de petróleo ó vació.

TABLA IV

METODOS TERMODINAMICOS DE SISTEMAS DE HIDROCARBUROS

K-VALORES	ENTALPIA	ENTROPIA	DENSIDAD
CHAO-SEADER	CHAO-SEADER	REDLICH-KWONG	SOAVE MKR
GRAYSON-STREED	CURL-PITZER	CURL-PITZER	RICE PROPIERTIES
BRAUN K10	JOHNSON-GRAYSON	SOAVE MKR	API DATA BOOK
SOAVE MODIFICADO	SOAVE MKR	PENG-ROBINSON	RACKETT PARAMETER
REDLICH-KWONG	PENG-ROBINSON	RICE PROPIERTIES	LEE KESLER
PENG-ROBINSON	RICE PROPIER-	LEE KESLER	
K DELTA	LEE KESLER		

ESPECIFICACIONES GENERALIZADAS

Process permite al usuario realizar especificaciones sobre algunas de las unidades de operación en adición a las columnas de destilación. Las especificaciones generales pueden ser hechas sobre algún producto o sobre una base húmeda o seca en la forma de una cantidad absoluta, relación, suma ó diferencia, los tipos de especificación incluyen:

-- Relación de producto-base: molar, masa, volúmen
-- Propiedades del producto PM, ρ , μ , ϕ , etc.
-- Pureza o recuperación del producto
-- Curvas de destilación del producto
Subrutinas adheridas

La estructura de datos de Process permite al usuario la integración de subrutinas para la predicción de propiedades termodinámicas, cálculos de unidades de operación, reportes de salidas especiales.

Capacidad de los sistemas de simulación (Process Flowsheeting).

Los cálculos del flowsheeting normalmente involucran balances de masa y energía, el orden de cálculo puede ser descrito por el usuario o determinado por Process. Process contiene un analizador de circuitos de recirculación capaz de determinar las corrientes de corte y la secuencia de cálculo del enlace.

Lenguaje de entrada.

El lenguaje de entrada de Process ofrece una elección de entradas; formato, Hoja de datos de entrada o entrada con palabras clave en formato libre. Formato libre es particularmente conveniente para usuarios con teletipo, escritores o terminales CRT, la entrada es textual y emplea comunmente palabras clave entendibles para describir datos, procedimientos de cálculo y métodos.

Salidas

La salida en Process es sumarizada en las siguientes categorias; Entrada, resultados iterativos, sumario de soluciones a unidades de operación sumario de solución de corriente.

Impresión de entrada. Los datos del usuario son amplificados, los faltantes listados y los auxiliares derivados son impresos en forma de sumario.

Resultados iterativos.

Incluye historia de cálculo un trazo de la convergencia del enlace de recirculación y controlador de convergencia, varios niveles de las corrientes de recirculación y resultados de la iteración de columnas.

Resultados de la solución.

Existen numerosas opciones para seleccionar salidas para unidades de operación y corrientes:

-- Imprimen 10 niveles de una columna de platos
-- Curvas de condensación/vaporización para alguna corriente ó condensador/reboiler de columna o cambiador de calor.
-- Curva de calentamiento/enfriamiento para algunas corrientes de proceso.
-- Fracciones de composición para las corrientes de proceso.
-- Condiciones de flujo y propiedades en corrientes de proceso.
-- Conductividad térmica y viscocidad de corrientes de proceso.
-- Curvas de punto de burbuja y rocío en corrientes de proceso.

II.1.4 DESIGN/2000

Design/2000 es un programa extremadamente fuerte y comprensivo aplicado a simular y diseñar una gran variedad de procesos químicos e hidrocarburos.

Este programa no solo ejecuta cálculos de balance de masa y energía sino que también puede, diseñar cierto tipo de equipo y define composiciones de corrientes y propiedades. La versatilidad de Design/2000 es demostrada por una lista parcial de aplicaciones:

-- Completa la simulación del diagrama de flujo, que puede eliminar la necesidad de plantas piloto.
-- Cuestiona el tratamiento de "que es"
-- Rediseño de plantas existentes para optimización económica y conservación de energía.
-- Estudio detallado de expansión en plantas y nuevas alimentaciones.
-- Rápida evaluación de alternativas de diseño.

Una característica importante de Design/2000 es la flexibilidad requerida en subrutinas para propiedades de equipo especializado. Pueden ser insertadas subrutinas para cálculos tales como dimensionamiento de tuberías, propiedades de reactores ó económicos.

Características especiales.

Un módulo controlador que permite cambios cuando el programa es ejecutado, como si se trabajara en una planta piloto.

El programa ahorra el trabajo de conversión de un sistema de unidades a otro, permitiendo entrar con alguna combinación de todas las unidades, similarmente Design/2000 puede reportar los resultados en algún sistema de unidades.

El programa mismo maneja los mas complicados loops de recirculación, Design/2000 determina automáticamente y reporta el orden óptimo en que el equipo con loops de recirculación sera calculado, en edición reporta la identidad del mínimo número de corrientes para que los supuestos iniciales sean previstos.

Las fichas de entrada en Design/2000 se estructuran como sigue:

-- Title comand: nombre de la simulación.
-- Comando de módulo, equipo

-- Comandos generales
-- Comando final

II.1.5 SIMPROC

Es un programa de computadora digital que realiza los balances de materia y energía de diversos procesos y la evaluación de diversas propiedades termofísicas de las corrientes involucradas en los mismos, proporcionando información suficiente para llevar a cabo el diseño básico de los equipos.

La simulación de un proceso necesita de la información almacenada en los siguientes arreglos :

1.- Matriz de constantes termofísicas; Se tienen las constantes características de cada componente, tales como: T,P, volumen, Zc, constantes de H ideal, PM, w, parámetros de solubilidad, Tb normal etc.
2.- Matriz de datos de corrientes, en esta matriz se almacenan las principales propiedades de las corrientes de proceso, tales como; composición, T, P, flujo, H vaporización, ρ , PM.
3.- Vectores de Parámetros; Los parámetros relacionados a los diferentes módulos, se almacenan secuencial o indiscriminadamente en dos vectores de parámetros (reales y enteros).

   Entre estos parámetros se tienen: Los números de las corrientes relacionadas a cada módulo y los índices de posición, área, coeficientes de transferencia, P, cargas térmicas, etc.
4.- Matriz de secuencia de resolución de módulos; La secuencia de resolución de una simulación queda establecida mediante el orden en el cual el usuario suministra la información almacenandose en esta matriz el tipo de módulo y los indices de posición de los parámetros reales y enteros antes mencionados.

Módulos de cálculo

A continuación se enumeran los módulos de cálculo disponibles :

1.- Equilibrio físico líquido-vapor; Se dispone de 10 opciones de cálculo, dependiendo de que variable se calcula entre; P ó vaporización.

2.- Torres de destilación fraccionadoras; El usuario indica la separación requerida por medio de contaminaciones máximas y recuperaciones mínimas de los componentes claves, se disponen de 10 diferentes opciones para este módulo, las cuales resultan de las combinaciones; destilado líquido, vapor ó dos fases, tipo de rehervidor y si P de operación se calcula o es fijada por el usuario.

3.- Proceso isoentálpico e isoentrópico; Se dispone de cuatro opciones de cálculo dependiendo si se calcula; P ó T en un proceso adiabático ó isoentálpico.

4.- Compresores y turboexpansores; Se dispone de cuatro opciones de cálculo, dependiendo si se calcula; Potencia, presión de descarga, flujo admisible.

5.- Mezcla de dos corrientes; Opciones de cálculo ; mezcla de dos corrientes, división de una corriente por relación de flujos, para dar un flujo constante en masa o en volúmen o para reponer el consumo y pérdidas de un componente.

6.- Cambiadores de calor; Dispone de un gran número de opciones de cálculo, el simulador avisa cuando dependiendo de la opción no se puede llevar a cabo el intercambio térmico requerido.

7.- Procesos de separación física; Extracción, secado, endulzamiento, etc. .

8.- Torres absorbedoras y agotadoras con vapor; Efectua el cálculo para un número de platos teórico especificado.

9.- Reactores; Se tienen implementados módulos de simulación de reactores de hidrodesulfuración e hidrogenación de olefinas , se esta en etapa de implementación.

10.- Módulo de especificaciones de productos de petróleo, tales como; curvas de destilación, índices de octano, cetano, etc..

El simulador general de procesos, emplea para la convergencia del ciclo, El método de substitución sucesiva, con una suposición inicial de flujo cero.

El cálculo de propiedades termoffsicas se realiza internamente en el - simulador al efectuarse la escritura de las corrientes de proceso.

11.1.6 ASPEN

Un sistema avanzado para ingeniería de proceso. Este es un simulador y sistema de evaluación económica bajo el desarrollo del MIT. Este sistema es ta diseñado para tener mucha flexibilidad así que puede ser expandido a necesidades futuras para requerimientos de simulación.

A continuación se da una descripción del ASPEN.

En el ASPEN existen dos tipos de componentes : convencionales y no convencionales. Los componentes convencionales son componentes puros o pseudocomponentes que pueden ser caracterizados en términos de propiedades estandar como; PM, TC, PC, P^{o} y coeficientes de capacidad calorífica. Existen -- más de 200 propiedades de componentes puros (o parámetros de correlación) que pueden ser especificados para cada componente.

Los componentes no convencionales son componentes no puros o pseudocomponentes y estan caracterizados en términos de atributos no convencionales en relación con las propiedades estandar de componentes puros. Algunos ejemplos: incluyen carbón, cenizas y sólidos inertes.

En ASPEN una corriente representa el flujo de material y/o información de una unidad de proceso a la siguiente, las corrientes son subdivididas en subcorrientes.

Hay tres clases de subcorrientes :

Subcorrientes convencionales : describen el flujo de componentes convencionales, esta descrita por datos del proceso.

Subcorrientes no convencionales : Describen el flujo de componentes, esta descrito por la misma clase de datos que las corrientes convencionales.

Subcorriente de información : no involucra flujos de componentes, contiene solamente atributos y/o datos de proceso, un uso común de una subcorriente de información es describir el flujo de energía (calor o trabajo).

Un bloque en ASPEN se refiere al elemento del diagrama de flujo representando una ó más entradas ó una o más salidas. La primera función de una -

La entrada esta en formato libre

Arquitectura del sistema

ASPEN tiene adaptada una estructura de tipo " procesador " en el que - el traductor de entrada genera llamando un programa principal que es llevado a ejecutar la simulación particular, el flujo de información en ejecución de cálculos en ASPEN se muestra en la figura (17).

Un bosquejo de la versión de la simulación (programa) para un proceso a simular, un mezclador y dos unidades flash, se muestran en la figura (18) .

El traductor de entrada es completamente ejecutado en tabla, esto pretende que toda la información necesaria a las declaraciones de entrada del - proceso (nombre de las palabras claves, valores de datos faltantes, etc.) son almacenados en tablas en una ficha llamada " ficha de definición del - sistema " .

ASPEN utiliza una estructura de datos plex del tipo propuesto por Evans. La información es almacenada en bloques de localización contigua conocidos - como cuentas. Las cuentas pueden contener valores enteros, valores reales -- o fichas de caracter, todas las cuentas son identificadas y almacenadas por un número de cuenta determinado.

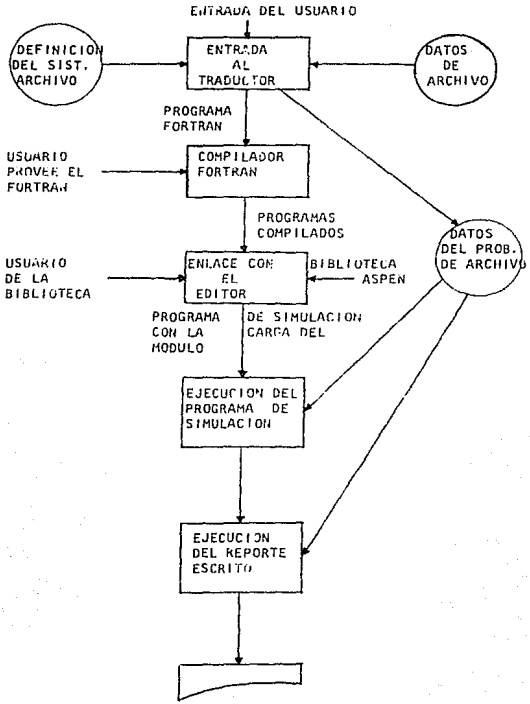


FIG. 17 - FLUJO DE INFORMACION EN ASPEN

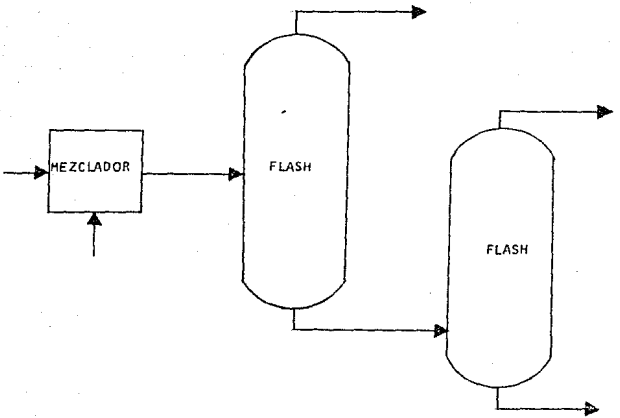


FIGURA 18

Versión de selección de un programa de simulación, usado como programa principal en la fase de simulación.

Los programas tienen acceso al satisfacer una cuenta usando su número de cuenta y subrutinas de administración de datos mediante su localización en la estructura PLEX.

ASPEN es un sistema de ficha - orientada, las fichas son usadas para almacenar tablas del traductor de entrada, resultados intermedios, programas y reportes.

ASPEN tienen un pequeño programa ejecutivo escrito en el lenguaje del sistema de operación, este control ejecutivo (trabajo de lenguaje de control para computadoras IBM), controla la ejecución de varios programas y - la creación y selección de fichas usadas por el sistema. Para una simulación típica este control ejecutivo muestra todos los pasos en la fig (17). El sistema ASPEN esta elaborado para constituirse de alrededor de 150,000 líneas de código.

ESTIMACION DE COSTOS Y EVALUACION ECONOMICA.

El sistema esta diseñado de tal manera que puede ser usado independientemente como un sistema solo con todos los tamaños de equipo y datos de proceso suplidos como entrada o puede ser corrido en conjunto como porción de la simulación de las unidades de operación de ASPEN. Esto provee un estudio preliminar económico con aproximación en un rango de más o menos un - treinta por ciento (30%).

C A P I T U L O I I I

METODOS DE SOLUCION DE ECUACIONES NO-LINEALES

Debido a la gran diversidad de problemas de simulación una gran mayoría presenta recirculaciones, de tal forma que no es posible efectuar el cálculo secuencial una sola vez; lo que implica la necesidad de un procedimiento iterativo que sea lo suficientemente eficiente para resolver los sistemas de ecuaciones que, en su mayoría, no son lineales y presentan dificultad en la solución; sin embargo con el avance de las técnicas numéricas, estos problemas han sido casi eliminados.

En el presente capítulo se analizán una serie de algoritmos encaminados a obtener la solución de sistemas de ecuaciones (2.3 y 2.4), con objeto de acelerar la convergencia para procesos con recirculaciones para obtener una solución con un mínimo número de iteraciones.

III. 1 METODO DE SUSTITUCION SUCESIVA.

Este método es el más simple y el más fácil de resolver, ya sea para una ecuación o para un sistema de ecuaciones. Como su nombre lo indica, usa el valor de la función corriente, para calcular la respuesta del siguiente :

$$x^{k+1} = f(x^k) \qquad 6.1$$

donde :

x^k = vector de variables de entrada.

k = número de iteración. k = 0,1,2, ...n

$f(x^k)$ = función representativa del proceso.

Es del tipo explícito, considerando como explícitos, aquellos métodos en los cuales para cada ecuación , se despeja una variable, llamandose éstas "variables de salida"; es fácil de programar, pero es muy lento en su convergencia debido a que es un algoritmo de primer orden.

Las etapas básicas de este método son :

1.- Definir k = 0, la tolerancia ξ y el estimado inicial x^0.
2.- Calcular $x^{k+1} = f(x^k)$
3.- Si $\| x^{k+1} - x^k \| \leq \xi$ el problema converge; de lo contrario hacer k=k+1 y regresar a 2.

La terminación del proceso iterativo debe estar de acuerdo con :

a) Un número de iteraciones fijas de antemano.
b) x^0, el valor inicial para la corriente recirculada, puede asumirse igual a cero(algunos procesos no convergen con esta suposición), o algunos conjuntos de valores finitos, a discreción del usuario.
c) Cuando el error relativo salga menor o igual que una tolerancia previamente especificada, teniendo en cuenta que ξ puede tener más que una forma; ya sea el cociente de las normas o el cociente de los valores absolutos o independientemente del tipo de problema es como se fija la tolerancia.

$$\|x^{k+1} - x^k\| \leq \varepsilon \quad ; \quad \|f(x^{k+1}) - f(x^k)\| \leq \varepsilon$$

$$\frac{\|x^{k+1} - x^k\|}{\|x^{k+1}\|} \leq \varepsilon \quad ; \quad \sum_{i=1}^{n} \frac{(x^{k+1} - x^k)_i^2}{(x^k)_i^2} \qquad \text{b.2}$$

La elección de cualquiera de estos criterios de convergencia, depende del tipo de problema; pero en la práctica se tienen otros restricciones que bien - pueden ser de tipo físico, químico o de seguridad(presión,= 0.5 psi temperatura. = 0.5°, entalpia = 5 BTU, Etc.), por lo que se pueden escoger otros criterios.

Debido a que el método de sustituciones sucesivas tiene una convergencia lineal(Local) en la vecindad de la solución, la relación del error entre la - iteración k y k+1, estará dada por :

$$\Delta x^k = J \Delta x^{k-1} \qquad \text{b.3}$$

donde :

$$\Delta x^k = x^{k+1} - x^k$$

$$\Delta x^{k-1} = x^k - x^{k-1}$$

$$J = \frac{\partial f_i}{\partial x_i}$$

J es la matriz del Jacobiano de f(x), que representa la relación lineal entre Δx^k y Δx^{k-1}. De hecho la relación de normas entre estos dos vectores - se puede expresar a través del valor propio de J de mayor magnitud. Como a cualquier método de convergencia se le pueden aplicar las pruebas para saber, si el estimado inicial es convergente, estas pruebas son a través de los valores propios dominantes, ya que se dice que cualquier matriz A con valores propios menores a uno es una matriz convergente, como se verá en el siguiente :

Teorema.- Sea J una matriz cuadrada de orden n, con su valor propio $/\lambda/^{max}$ de mayor magnitud y con su correspondiente vector propio v , que es de magnitud unitaria. Entonces si todos los vectores propios de J forman un conjunto linealmente independiente y se construye una secuencia de vectores $\{v^k\}_{k=0}^{\infty}$ de acuerdo con la regla :

$$v^k = \frac{Jv^{k-1}}{\|Jv^{k-1}\|} \qquad \text{b.4}$$

Iniciando con un vector v^0 arbitrario, se tiene que :

$$\lim_{k\to\infty} \quad v^k = v$$

$$\lim_{k\to\infty} /\!/ J v^k /\!/ = -/\lambda/^{max} \qquad b.5$$

dividiendo la ecuación b.3 entre $/\!/ \Delta x^k /\!/$ se tiene

$$\frac{\Delta x^k}{/\!/\Delta x^k/\!/} = \frac{J\Delta x^{k-1}}{/\!/\Delta x^k/\!/} = \frac{J\Delta x^{k-1}}{/\!/J\Delta x^{k-1}/\!/} \qquad b.6$$

que corresponde a b.4 con $v^k = \Delta x^k / /\!/\Delta x^k/\!/$

por lo tanto de b.5 es claro que:

$$\lim_{k\to\infty} \quad \frac{/\!/J\Delta x^{k-1}/\!/}{/\!/\Delta x^{k-1}/\!/} = /\lambda/^{max} \qquad b.7$$

por lo que en la vecindad de la solución α $/\lambda/^{max}$ esta dado por:

$$/\lambda/^{max} = \frac{/\!/\Delta x^k/\!/}{/\!/\Delta x^{k-1}/\!/} \qquad b.8$$

es decir por la relación de la norma de errores de la iteración k y k-1 aplicando sucesivamente b.8 en la forma:

$$/\!/\Delta x^k/\!/ = /\lambda/^{max} /\!/\Delta x^{k-1}/\!/ = (/\lambda/^{max})^2 /\!/\Delta x^{k-2}/\!/ = \ldots =$$

se puede demostrar fácilmente que para un error inicial Δx^0 se tiene :

$$/\!/\Delta x/\!/ = (/\lambda/^{max})^k /\!/\Delta x/\!/ \qquad b.9$$

deduciendose que una condición necesaria y suficiente para que converja el método de sustitución sucesiva es que $/\lambda/ < 1$, ya que entonces:

$$\lim_{K \to \infty} \left(|\lambda|^{max} \right)^K = 0 \qquad \text{b.10}$$

y por lo tanto la condición suficiente para la convergencia es que la suma de los valores absolutos de cada columna o renglon del jacobiano J sea menor a uno.

En la práctica las expresiones b.5 y b.9 son de gran utilidad ya que permiten predecir si un problema de un sistema de ecuaciones no-Lineales va a converger por el método de sustituciones sucesivas, y si es así aproximadamente en cuantas iteraciones. Para esto, en primer lugar es necesario determinar el valor de $|\lambda|^{max}$; aunque en teoría se podría hacer después de un número infinito de iteraciones a partir de x^0, en la práctica son aproximadamente 5, según - Shacham y Kotard, (1974). Si $|\lambda|^{max} < 1$, el problema convergerá y mientras el $|\lambda|^{max}$ tenga un valor cercano a uno, la convergencia será más lenta. Lo cual - se puede apreciar de b.9.

Para predecir el número total de iteraciones requeridas a partir de x^0 • se puede utilizar b.9; si ξ es la tolerancia final deseada, $\|\Delta x^m\|$ el error obtenido en la iteración m donde se hizó la estimación del valor absoluto de λ^{max}, el número total de iteraciones (NIT), estará dado por :

$$NIT = m + \left(\ln \frac{\xi}{\|\Delta x^m\|} \right) \Big/ \ln |\lambda|^{max} \qquad \text{b.11}$$

Finalmente, cabe señalar que la sustitución sucesiva ha sido uno de los métodos que más se ha usado en problemas de Ingeniería de Proceso, esto se debe en gran parte a la sencillez del método ya que requiere poca memoria de computadora. Sin embargo , es necesario señalar que con cierta frecuencia en las aplicaciones prácticas el método falla(pues $|\lambda|^{max} \geq 1$) o bien su convergencia es extremadamente lenta(si $|\lambda|^{max} \to 1$).

III.2 METODO DE RELAJACION.

Con objeto de acelerar la convergencia del método de sustitución sucesiva es posible definir el siguiente procedimiento iterativo.

$$y^k = f(x^k)$$

$$x^{k+1} = x^k + \omega I (y^k - x^k) \qquad b.12$$

donde :

y^k = valor calculado del vector de variables de entrada x^k

ω = escalar que representa el factor de relajación.

I = matriz identidad.

Notese que para $\omega = 1$, la ecuación b.12 se reduce al método de sustitu -- ción sucesiva. Es posible escoger valores de $\omega \neq 1$ que aumenten la rapidez de convergencia de b.12 o que inclusive hagan que b.12 sea convergente.

Para determinar el valor de ω que acelere la convergencia es necesario analizar los valores propios de la matriz B (matriz de relajación) en la expre sión siguiente :

$$\Delta x^k = B \Delta x^{k-1} \qquad b.13$$

donde :

$$B = \omega J - (\omega - 1) I$$

si θ son los valores propios de B, se tendrá por definición :

$$\text{Det} \left(B - \theta I \right) = 0$$

y sustituyendo el valor de B

$$\text{Det} \left(\omega J - (\omega - 1) I - \theta I \right) = 0 \qquad b.14$$

reacreglando b.14 como $Det(J-\gamma I) = 0$ se tiene que :

$$\gamma = \frac{\omega + \theta - 1}{\omega} \qquad b.15$$

Como para la matriz J, $Det(J-\lambda I) = 0$ se tiene que : $\lambda = \gamma$ y por lo tanto cada valor propio de θ de B estará dado por:

$$\theta_i = \omega(\lambda_i - 1) + 1 \qquad b.16$$

Lo cual se deduce de b.15. De b.16 es claro que $|\theta_i| < |\omega||\lambda_i| + |1-\omega|$ por lo cual se puede apreciar que existe una correspondencia directa entre las magnitudes de $|\theta_i|$ con las magnitudes de $|\lambda_i|$ Por lo tanto

$$|\theta|^{max} = |\omega(\lambda^{max} - 1) + 1|$$
$$|\theta|^{min} = |\omega(\lambda^{min} - 1) + 1| \qquad b.17$$

donde λ^{max} y λ^{min} corresponden a los valores propios de mayor y menor magnitud de la matriz J.

La elección óptima de ω estará dada por aquel valor que minimize el valor absoluto de θ^{max} y mantenga $|\theta|^{max} \geq |\theta|^{min}$ ya que de b.9 se puede ver que si el valor propio de mayor magnitud es pequeño, el error decrecerá más rapidamente.

Debido a que el valor de $|\lambda|^{min}$ es en general difícil de estimar en la práctica normalmente se consideran los dos casos siguientes:

a) Se supone $|\lambda|^{max} >> |\lambda|^{min}$ lo cual suele ser una suposición válida si hay involucradas muchas variables. Por lo que se tiene:

$$\omega^* = \frac{2}{2 - |\lambda|^{max}} \qquad b.18$$

b) Se supone $|\lambda|^{max} = |\lambda|^{min}$ lo cual suele cumplirse si el número de variables -

es pequeño ($n \leq 5$). En este caso:

$$\omega^* = \frac{1}{1 - |\lambda|_{max}} \qquad b.19$$

Cabe señalar que en general es preferible utilizar la expresión en b.18 ya que si $|\lambda|_{max} \doteq 1$ la expresión b.19 produce valores demasiado grandes de ω^* que es el factor de relajación obtenido por procedimiento numérico.

Las etapas básicas de este método son:

1.- Hacer unas 5 sustituciones sucesivas.
2.- Con b.8 evaluar $|\lambda|_{max}$
3.- Evaluar ω con b.18 ó b.19
4.- Evaluar x^{k+1} con b.12
5.- Si $\| x^{k+1} - x^k \| \leq \varepsilon$ el problema converje; si no regresar a 2.

Es desde luego posible fijar directamente un valor de ω y aplicar el procedimiento iterativo de relajación, con esto inclusive se puede lograr que el algoritmo converja aunque el método de sustituciones sucesivas diverja. Sin embargo en este caso, la selección de ω no es obvia y requiere de pruebas numéricas para su verificación.

III.3 METODO ORIGINAL DE LOS VALORES PROPIOS DOMINANTES (DEM ORIGINAL).

Como se puede apreciar en el método de relajación, el hecho de conocer $|\lambda|^{max}$, permite acelerar la convergencia. Es por esto que se podría pensar que si un método tomara en cuenta más de un valor propio con una magnitud significante, es decir que se tomaran los valores propios de J, sería posible obtener un método más eficiente para la aceleración de la convergencia.

Suponer que la ecuación :

$$x^{k+1} = x^k + G\,(y^k - x^k) = f(x^k) \qquad b.20$$

donde G = matriz de forzamiento.

Se puede aproximar por series de Taylor a términos de primer orden alrededor de un punto arbitrario x^o en la vecindad de x^k.

$$x^{k+1} = Jx^k + b \qquad b.20a$$

donde $b = F(x^o) - Jx^o$ b.21

La ecuación de iteración b.20a la cual es una ecuación de diferencias - lineales puede resolverse generalmente por el mismo valor inicial x^o así :

$$x^k = \sum_{j=1}^{m} C_j Z_j \lambda_j^k + x_s \qquad b.22$$

si todos los λ_j son diferentes. Entonces:

$$x_s = (I - J)^{-1} b \qquad b.23$$

$$C_j = \frac{\omega_j^T (x_o - x_s)}{(\omega_j^T Z_j)} \qquad b.24$$

x_s es la solución a estado estable de la ecuación b.20. Z_j y ω_j son los vectores e hileras propias para λ_j , el cual es el valor propio de la matriz J. Si los valores propios de J son puestos en orden descendente de magnitud absoluta, la condición necesaria y suficiente para que la ecuación b.20 converja

es que $|\lambda_1| < 1$ Como k se hace grande $\left|\left(\lambda_i/\lambda_1\right)^k\right|_J > 1$ decrece monotónicamente de manera que la ecuación b.22 se hace aproximadamente una progresión geométrica, en la diferencia de la solución :

$$x^k - x_s = C_1 Z_1 \lambda_1^k \qquad b.25$$

Por lo tanto la solución aparente se obtiene como sigue:

$$x'_s = x^{k-1} + \frac{\alpha\,(x^k - x^{k-1})}{(1 - \lambda_1)} \qquad b.26$$

con $0 < \alpha \leq 1$ incluida como un factor de amortiguamiento para suprimir la oscilación. Claramente si $|\lambda_1|$ es cercano a la unidad la convergencia de la ecuación b.25 es muy lenta y la corrección en b.26 es grande. Notesé también que si $\lambda_1 < 0$, la corrección en b.26 puede encontrarse entre los valores en x^k y x^{k+1}.

El valor aparente de λ puede determinarse de :

$$\Delta x^{k-1} = x^k - x^{k-1} = C_1 Z_1 (\lambda_1 - 1)\lambda_1^{k-1} \qquad b.27$$

por ejemplo para la relación de las normas:

$$\frac{\|\Delta x^{k-1}\|}{\|\Delta x^{k-2}\|} = |\lambda_1| \qquad b.28$$

El signo de λ_1 puede encontrarse de la relación de componentes de Δx^k y Δx^{k-1}.

El método DEM Original es más efectivo cuando la progresión geométrica - es lograda rapidamente y cuando $|\lambda_1|$ es cercano a la unidad, este método es - muy parecido al Wegstein, pero el DEM incorpora la interacción entre los componentes de x y suministra un criterio para decidir cuando se promueve la convergencia. La mayor ventaja del DEM sobre el Newton es que no es necesario la evaluación ni la inversión de una gran matriz.

Las etapas básicas de este método son :

1.- Evaluar con x^o $\left[dF / dx \right]_{x=x^o}$ y $b = F(x^o) - Jx^o$
2.- Evaluar b.23 y b.24
3.- Evaluar b.22
4.- Evaluar b.26
5.- Si $\| x'_s - x^k \| \leq \xi$ el problema converje; sino, hacer $x'_s = x^o$ y regresar al paso 1.

III.4 METODO GENERALIZADO DE LOS VALORES PROPIOS DOMINANTES.

El DEM original de Crowe (1971), fué efectivo para problemas multivariables. Crowe y Nishio (1975), desarrollan este método que es una extensión del DEM y el cual ofrece flexibilidad y usa más pasos efectivos de aceleración, es la idea fundamental predecir un valor a partir de $\Delta x^k = J\Delta x^{k-1}$, considerando que esta relación aplicará recursivamente a un número infinito de iteraciones y manteniendo la matriz J constante. Adicionalmente el método no requiere que se genere explícitamente la matriz J, ya que ésta se maneja indirectamente a través de sus valores propios dominantes.

Para deducir este método, considérese la ecuación característica de la matriz J y que está dada por :

$$\text{Det}\ (\lambda I - J) = \sum_{i=0}^{K} \mu_i \lambda^{K-i} = 0 \qquad b.29$$

donde μ_i representa los coeficientes de la ecuación característica de la matriz J y que está dado por un estimado del valor correcto μ_i y se define : $\hat{\mu}_o = \mu_o = 1$

En la ecuación característica b.29 los coeficientes se pueden relacionar con los valores propios por medio de la ecuación :

$$\mu_i = (-1)^i \sum \lambda_{l_1} \lambda_{l_2} \ldots \lambda_{l_i} \qquad b.30$$

donde $1 \leq l_1 \leq l_2 \leq \ldots \leq l_i \leq n$

Es decir, b.30 representa la suma de todos los productos posibles de "i" valores propios tomados como permutaciones ordinarias; la base del nuevo método es asumir que las iteraciones siguen aproximadamente una matriz lineal de ecuaciones diferenciales y calcular la solución aparente, usando estimados de los productos de los valores propios dominantes.

Es conveniente ahora referirse al teorema de Cayley-Hamilton.

Teorema.- Cualquier matriz cuadrado A satisface su ecuación característica en el sentido de que si $\text{Det}\,(\lambda I - A) = \sum_{i=0}^{K} \mu_i \lambda^{K-i}$

entonces la matriz A satisface $\sum_{i=0}^{K} \mu_i A^{K-i} = 0$

Aplicando este teorema a b.29 se tiene entonces:

$$\sum_{i=0}^{K} \mu_i J^{K-i} = 0 \qquad b.31$$

Si Δx^{m-k} es el error que se obtuvo en la iteración m-k ($m \geq k$) y se multiplica b.31 por este vector se tiene :

$$\sum_{i=0}^{K} \mu_i J^{K-i} \Delta x^{m-K} = 0 \qquad b.32$$

aplicando $\Delta x^k = J \Delta x^{k-1}$ recursivamente en $J^{k-i} \Delta x^{m-k} = J^{m-i+1} \Delta x^{m-k+1} = \Delta x^{m-i}$ por lo que b.32 se reduce a :

$$\sum_{i=0}^{K} \mu_i \Delta x^{m-i} = 0 \qquad b.33$$

Si los valores propios λ_i se ordenan en forma decreciente de magnitud y se supone que los primeros ν valores de λ_i son dominantes es decir tienen una magnitud mayor que el resto de los valores propios, entonces $\mu_i = \quad i = \nu+1 \ldots n$, debido a que por b.30 μ_i tendrá un valor muy pequeño para λ_i , $i > \nu$
De esta forma b.33 se puede reducir a :

$$\sum_{i=0}^{\nu} \mu_i \Delta x^{m-i} = 0 \qquad \text{b.34}$$

que implica básicamente que para el método iterativo el espacio n-dimensional se reduce a un espacio ν-dimensional.

Para obtener estos valores $\hat{\mu}_i$, $i = 1,2,\ldots\ldots \nu$ se minimizará el cuadrado de la norma Euclidiana de b.34 lo cual implica resolver el problema

$$(\mu_1 \min_{\mu_2 \ldots \mu_\nu}) \quad \phi = \left\| \sum_{i=0}^{\nu} \mu_i \Delta x^{m-i} \right\|^2 \qquad \text{b.35}$$

como es un factor escalar, es decir $\|x\|^2 = x^T x$, la función ϕ se puede escribir como :

$$\phi = \left(\sum_{i=0}^{\nu} \mu_i \Delta x^{m-i} \right)^T \left(\sum_{i=0}^{\nu} \mu_i \Delta x^{m-i} \right) \qquad \text{b.36}$$

Para determinar el mínimo de b.36 es necesario derivar esta expresión con respecto a μ_i e igualarla a cero. Expresando ϕ como un producto escalar:

$$\phi = \sum_{\ell=0}^{n} \left(\sum_{i=0}^{\nu} \mu_i \Delta x^{m-i}_{\ell} \right) \left(\sum_{j=0}^{\nu} \mu_j \Delta x^{m-i}_{\ell} \right) \qquad \text{b.37}$$

donde ℓ es el subíndice que está asociado a cada componente del vector Δx
Desarrollando b.37 se obtiene :

$$\phi = \sum_{i=1}^{\nu} \mu_i^2 (\Delta x^{m-i})^T \Delta x^{m-i} + 2 \sum_{i=1}^{\nu-1} \sum_{j=i+1}^{\nu} \mu_i \mu_j (\Delta x^{m-i})^T \Delta x^{m-j} \qquad \text{b.38}$$

derivando b.38 con respecto a μ_i e igualando a cero, se tiene:

$$\frac{\partial \phi}{\partial \mu_i} = 2\,\mu_i\,(\Delta x^{m-i})\;\Delta x^{m-i} + 2\sum_{\substack{j=0 \\ j\neq i}}^{\nu} \mu_j\,(\Delta x^{m-i})^T(\Delta x^{m-j}) = 0 \qquad b.39$$

de manera que para determinar los valores de μ_i hay que resolver el sistema lineal de ecuaciones:

$$\sum_{j=1}^{\nu} \hat{\mu}_j\, b_{ij} = -\mu_{ob\,i}$$
$$b_{ij} = (\Delta x^{m-i})\;(\Delta x^{m-j}) \qquad b.40$$

$i = 1,2,\ldots \nu$ $\qquad j = 0,1,2,\ldots \nu$ y $\mu_o = 1$ como se había definido.

Ahora se desea establecer, cual sería el valor de las variables y la relación en b.3 se aplicará un número infinito de veces teniendo a la matriz H como constante y manejada a través de sus valores propios dominantes.

Como en la iteración m se ha determinado $\Delta x^m = x^{m+1} - x^m$ el valor de x^α en función de x^m estará dado por :

$$x^\alpha = x^{m+1} + \sum_{i=m+1}^{\alpha} \Delta x^i \qquad b.41$$

sumando ahora sobre las i's en la ecuación b.34 se tiene:

$$\sum_{i=m+1}^{\alpha} \sum_{j=0}^{\nu} \mu_i\, \Delta x^{i-j} = 0 \qquad b.42$$

pero como $\mu_o = 1$

$$\sum_{i=m+1}^{\alpha} \Delta x^i = -\sum_{i=m+1}^{\alpha} \sum_{j=1}^{\nu} \mu_j\, \Delta x^{i-j} \qquad b.42a$$

por lo que al sustituir esta expresión en b.41 se tiene :

$$x^\alpha = x^{m+1} - \sum_{i=m+1}^{\alpha} \sum_{j=1}^{\nu} \mu_j\, \Delta x^{i-j} \qquad b.43$$

que al rearreglarse y reducirse se expresa como:

$$x^{\alpha} = \frac{\sum_{j=0}^{\nu} \mu_j \, x^{m+1-j}}{\sum_{j=0}^{\nu} \mu_j} \qquad \text{b.44}$$

ahora si se sustituye:

$$x^{m+1-j} = x^{m+1} - \sum_{i=0}^{j-1} \Delta x^{m-i}$$

se obtiene:

$$x^{\alpha} = x^{m+1} - \frac{\sum_{j=0}^{\nu} \mu_j \sum_{i=0}^{j-1} \Delta x^{m-i}}{\sum_{j=0}^{\nu} \mu_j} \qquad \text{b.45}$$

reduciendo b.45 y rearreglando, finalmente se tiene:

$$x^{\alpha} = x^{m+1} - \frac{\sum_{i=0}^{\nu-1} \left(\sum_{j=i+1}^{\nu} \mu_j \right) \Delta x^{m-i}}{\sum_{j=0}^{\nu} \mu_j} \qquad \text{b.46}$$

Es interesante notar que x^{α} al cual se le denomina "valor de promoción" está dado por una combinación lineal de los ultimos $\nu+1$ puntos, lo cual se aprecia claramente de b.44.

Un punto muy importante que hay que establecer, es cuando se va a calcular x^{α}, cuando se va a introducir como estimación dentro del método de sustitución sucesiva y como se va a estimar ν el número de valores propios dominantes.

Para los dos primeros puntos, Grossman y del Rosal (1978), han encontrado que la estrategia efectiva es la siguiente; El valor de x^{α} se calcula a cada iteración después de las primeras $\nu+1$ iteraciones. Con objeto de verificar que el valor de x^{α} es consistente se checa si :

$$\left\| x^{\alpha^{K+1}} - x^{\alpha^{K}} \right\| < \xi \qquad \text{b.47}$$

en cuyo caso $x^{\alpha^{K+1}}$ se acepta como una promoción para las sustituciones sucesivas. Hay que notar que cuando se utiliza una tolerancia ξ constante, este procedimiento puede ocasionar dificultades. Si ξ es muy pequeño, después de muchas iteraciones se aceptará x^{α} ; mientras que si ξ es relativamente grande x^{α} - se aceptara con demasiada frecuencia lo cual puede ocasionar oscilaciones en la vecindad de la solución. Para evitar este problema es conveniente utilizar una tolerancia ξ variable, según Grossmann y del Rosal, (1978).

$$\xi(K) = \max \left\{ \left(\frac{\delta - \xi}{NIT - 2} \right) K - (NIT) + \delta_1 \quad \| \Delta x^K \| \right\} \qquad b.48$$

con lo cual se asegura que $\xi(K)$ no llegue a ser demasiado pequeña para ya no - permitir una promoción.

Conviene ahora describir las etapas básicas de este método.

1.- Evaluar x^0, ν , Definir k=0, m= ν +1 y las tolerancias δ_1 ξ
2.- Efectuar ν +1 iteraciones por sustitución sucesiva y generar los valores Δx^{m-1} con los puntos x^k. Poner k = ν +1.
3.- Evaluar $|\lambda|^{max}$ y NIT con b.8 y b.11 respectivamente.
4.- Evaluar los coeficientes b_{ij} con b.40.
5.- Evaluar el sistema de ecuaciones

$$\sum_{j=1}^{\nu} \mu_j b_{ij} = -b_{io} \qquad \text{para determinar } \mu_j$$

6.- Evaluar la promoción x^{α^K} de b.46
7.- Si cumple b.48 poner $x^k = x^{\alpha^K}$
8.- Poner k = k+1 y Evaluar $x^k = g(x^{k-1})$
9.- Si $\| x^k - x^{k-1} \| < \xi$ el problema converge, de lo contrario poner m = m+1 actualizar Δx^{m-k} e ir al paso 4.

Con esto se puede apreciar que el método generalizado de valores propios es más difícil de implementar que el método de sustitución sucesiva o el de relajación. Sin embargo es necesario señalar también que estos dos metodos son - inferiores al generalizado de valores propios dominantes, particularmente si - se tiene un problema de convergencia lenta. Ahora, una reciente modificación -

o este método fué hecho por Soliman M.A. (1981), para la aceleración de la convergencia de sistemas cíclicos; en esta modificación la ecuación b.46 la derivan de una forma diferente a la presentada por Crowe y Nishio(1975). En la iteración m+1, la inversa del jacobiano J^{-1} satisface la siguiente ecuación:

$$J^{-1}Y = Q \qquad b.49$$

donde:

$$Y = \left[f_m - f_{m-\nu} \;\; \cdots \;\; f_m - f_{m-\nu+1} \;\; \cdots \;\; f_m - f_{m-1} \right]$$

$$Q = \left[x_m - x_{m-\nu} \;\; \cdots \;\; x_m - x_{m-\nu+1} \;\; \cdots \;\; x_m - x_{m-1} \right]$$

Por otro lado, las iteraciones presentadas por la ecuación b.20 pueden ponerse de la siguiente forma :

$$Q = G(Y - F) \qquad b.50$$

donde : $$F = \left[\sum_{j=m-\nu+1}^{m} f_j \;\; \cdots \;\; \sum_{j=m-\nu+2}^{m} f_j \;\; f_m \right]$$

de las ecuaciones b.49 y b.50 se obtiene:

$$(J^{-1} - G)Y = -GF \qquad b.51$$

esta ecuación no puede resolverse unicamente por la inversa del jacobiano, excepto si $\nu = n$ y $f_{m-\nu}$, $f_{m-\nu+1}$, ... f_m son linealmente independientes. Cuando $\nu < n$ la solución general es de la forma :

$$J^{-1} = G - GF(P^T Y)^{-1} P^T \qquad b.52$$

donde P es una matriz arbitraria $\quad n \times \nu$

Usando el nuevo estimado para J^{-1}, la iteración m+1 puede tomarse como :

$$x^a = x^m - J^{-1} F^m$$

$$x^a = x^{m+1} + GF(P^T Y)^{-1} P^T F^m \qquad b.53$$

se define el vector: $\nu \times 1$

$$\begin{pmatrix} \mu_\nu \\ \mu_{\nu-1} \\ \mu_1 \end{pmatrix} = (P^T Y)^{-1} P^T F^m \qquad b.54$$

de esta ecuación es claro que μ satisface la siguiente ecuación:

$$P^T\left[\sum_{j=1}^{\nu}\mu_j \quad (f_{m-j} - f_m) + f_m\right] = 0 \qquad b.56$$

En resumen, sustituyendo la ecuación b.54 en la b.53 se obtiene la fórmula del GDEM; en esta derivación se puede notar que ha sido obtenida una fórmula explícita para J^{-1} la cual es de esperar que sea una mejor aproximación a la inversa del jacobiano que G. Entonces, en la próxima secuencia de iteraciones, se puede usar J^{-1} en lugar de G.

La matriz arbitraria P es elegida tal que;

$$P = W\left[f_{m-\nu} \cdots f_{m-\nu+1} \cdots f_{m-1}\right] \qquad b.56a$$

donde la matriz de peso W puede tomarse como la matriz identidad.

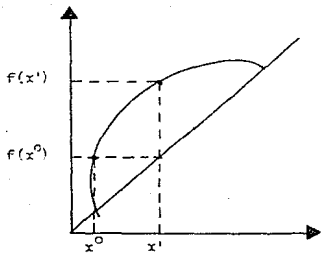
Las etapas básicas de esta modificación son :

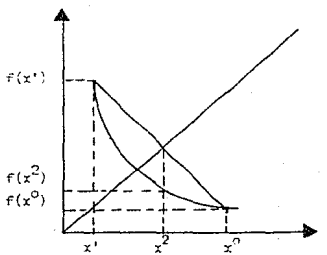
1.- Evaluar x^0, poner $k = 0$ y elegir un entero ν
2.- Use la fórmula de iteración $x^{k+1} = x^k - J^k F^k$ donde J es la matriz identidad o cualquier mejor cambio para la inversa del jacobiano.
3.- Cuando se cumplen ciertos criterios (Crowe y Nishio,1978), en la iteración m+1 evaluar $J^{k+1} = J^k - J^k F(P^T Y)^{-1} P^T$ donde P se obtiene de b.56 y hacer $x^\alpha = x^m - J^{k+1} F^m$
4.- Poner $k = k + 1$, $i = 0$ y $x^0 = x^\alpha$ y regresar al paso 2 si la solución no se obtiene dentro de la tolerancia prescrita.

La mejoría obtenida por esta modificación al GDEM, es a costa de un incremento en los requerimientos de almacenamiento de la memoria de computadora.

III.5 METODO ACOTADO DE WEGSTEIN.

Para acelerar el método de sustitución sucesiva; Wegstein (1958), propuso su método, que no es más que una extrapolación o interpolación de dos puntos anteriores iniciados estos con el método de sustitución sucesiva.





Para una ecuación simple no lineal se reemplaza lo (x) calculada por sustitución sucesiva con una ($\bar{x}$) donde :

$$\bar{x}^{k+1} = q(\bar{x})^k + (1 - q)x^k \qquad b.57$$

$$s \frac{f(x^k) - f(x^{k-1})}{x^k - x^{k-1}} \qquad b.58$$

siendo q = factor de peso = $\dfrac{1}{1 - s}$

Este método aunque no tiene una base teórica firme, produce buenos resultados, sobre todo en simulación de procesos; la idea fundamental del método acotado de Wegstein, es extender el método de Wegstein al caso multivariable.Es decir, el algoritmo es de la forma siguiente :

$$x_i^{k+1} = q_i^k x_i^k + (1 - q_i^k) f_c(x^k) \qquad i = 1,2,\ldots n \qquad b.59$$

donde :

$$q_i = \frac{f_i(x^{k-1}) - f_i(x^{k-2})}{x_i^{k-1} - x_i^{k-2}} \qquad b.60$$

Si se aplica este algoritmo directamente, se presentarán inestabilidades y divergencias con gran frecuencia. Para remediar esta situación Kliesch (1967) y Graves (1972), propusieron usar un valor de $q_i^k = 0$ en la ecuación b.59 a menos que q_i^k se encontrará dentro de ciertas cotas, o sea :

$$q^L < q_i^k < q^U \qquad b.61$$

De esta forma cuando $q_i^k = 0$ el algoritmo efectúa una sustitución sucesiva, lo cual tiende a estabilizar el método. Asimismo por esta razón es conveniente efectuar las primeras m iteraciones (típicamente m = 5).

Por lo que se refiere a la selección de q^L y q^U ésta es arbitraria y depende del problema. Sin embargo, generalmente se suele usar el valor $q^U = 0$ y para q^L un valor negativo. Concretamente para simulación de procesos Seader (1974), ha encontrado que valores adecuados son $q^L = -5$ y $q^U = 0$.

Los pasos generales del método acotado de Wegstein son :

1.- Evaluar x^0 y definir q^L, q^U, n, ξ y k = 0.
2.- Evaluar $f(x^k)$
3.- a) Si k < n, poner $q_i^k = 0$ $\qquad i = 1,2,\ldots,n$
    b) Si k > n, calcular q_i^k de b.60
    c) Si $q_i^k < q^L$ ó $q_i^k > q^U$ poner $q_i^k = 0$
4.- Evaluar $x_i^{k+1} = q_i^k x_i^{k-1} + (1 - q_i^k) f_c(x^k)$
5.- Si $\|x^{k+1} - x^k\| < \xi$ el problema converge; de lo contrario poner k = k + 1 e ir al paso 2.

Es interesante señalar que es posible establecer una conexión entre el método de Wegstein y el de los valores propios dominantes. Sustituyendo b.60 en b.59, suponiendo que los dos ultimas iteraciones fueron por sustitución sucesiva y recorriendo el indice k un valor hacia atras se tiene que el método de Wegstein se puede expresar como :

$$x_i^{K+1} = \left[\frac{1}{1 - \frac{\Delta x_i^{K-1}}{\Delta x_i^{K-2}}}\right] x_i^K + \left[1 - \frac{1}{1 - \frac{\Delta x_i^{K-1}}{\Delta x_i^{K-2}}}\right] x_i^{K-1} \qquad b.62$$

Por otro lado si se considera un sólo valor propio dominante en el método generalizado de los valores propios, al resolver H_i de b.40 y resolver en b.44 para este valor se tiene :

$$x^{\alpha} = \left[\frac{1}{1 - \frac{(\Delta x^{K-1})^T \Delta x^{K-1}}{(\Delta x^{K-2})^T \Delta x^{K-2}}}\right] x^K + \left[1 - \frac{1}{\frac{(\Delta x^{K-1})^T \Delta x^{K-1}}{(\Delta x^{K-2})^T \Delta x^{K-2}}}\right] x^{K-1} \qquad b.63$$

de b.62 y b.63 se puede ver que los cocientes de las diferencias representan en b.62 una estimación de $|\lambda|$.

Con esto se puede apreciar que ambos métodos son muy similares y que el de Wegstein viene a ser sólo una aproximación de un caso particular del método generalizado de valores propios dominantes por lo tanto sería de esperar que este método fuera más eficiente.

El método de Wegstein es recomendable para convergencia de balances de materia en procesos con recirculaciones, ya que estos generan una matriz diagonal, lo cual indica que no existe fuerte interación entre las variables.

Para algunos problemas específicos, es conveniente utilizar la norma λ en vez de s

$$\lambda = \frac{\| \Delta x^K \|}{\| \Delta x^{K-1} \|} \qquad q = \frac{1}{1 - \lambda}$$

Cada 3 iteraciones se puede encontrar λ y sustituir en Wegstein

$$\lambda = \left\| \frac{x^{K} - x^{K-1}}{x^{K-1} - x^{K-2}} \right\| \qquad 6.64$$

esta modificación, igualmente es válida para el método de Broyden.

III.6 METODO DE NEWTON

La idea básica de este método consiste en efectuar una aproximación lineal de f(x) en el punto donde se está iterando. Este genera la próxima suposición para el manejo de parámetros, dibujando una tangente a la curva como el punto corriente y extrapolando a cero; consecuentemente se da una nueva (x) dada por :

$$x^{k+1} = x^{k} - \frac{f(x^{k})}{f'(x^{k})} \qquad 6.65$$

donde : $$f'(x) = \frac{\partial f(x)}{\partial x}$$

Si se expande f(x) por series de Taylor en un punto x^k cercana al vector de solución real α y se considera el desplazamiento : $p = \alpha - x^k$ se tendrá

$$f(\alpha) = f(x^k + p) = f(x^k) + J(x^k)p + 0 \left\| p \right\|^2 \qquad 6.66$$

J es el jacobiano y $0 \left\| p \right\|^2$ son los términos de orden cuadrático en forma de orden de magnitud.

Como obviamente $f(\alpha) = 0$ pues α es la solución, y considerando que la magnitud de ρ es pequeña, se pueden entonces despreciar los términos de orden cuadrático con lo que de b.66 se tendrá :

$$f^k + J^k \rho = 0 \qquad \text{b.67}$$

donde $f^k = f(x^k)$.

Si el jacobiano es una matriz no-singular, el sistema lineal de ecuaciones en b.67 tendrá una solución única para ρ con el cual se puede obtener la solución α . El método de Newton consiste en aplicar la relación en b.67 de una forma iterativa, es decir :

$$J^k \rho^{k+1} = -f^k$$

$$x^{k+1} = x^k + \rho^{k+1} \qquad \text{b.68}$$

En lo que respecta a la velocidad de convergencia; esto es de segundo orden (cuadrática). De acuerdo con (2.17) basta demostrar que $\left(\frac{\partial q}{\partial x}\right)_{\alpha} = 0$. Para ésto, de b.68 es claro que $q(x)$ está dado por :

$$q(x) = x - J^{-1}(x) f(x) \qquad \text{b.69}$$

por lo que:

$$\frac{\partial q}{\partial x} = I - J^{-1}_{(x)} \frac{\partial f(x)}{\partial x} - \frac{\partial J^{-1}}{\partial x} f(x) \qquad \text{b.70}$$

como $f(\alpha) = 0$ y $J(x) = \frac{\partial f(x)}{\partial x}$ de b.70 se tiene :

$$\left(\frac{\partial q}{\partial x}\right)_{\alpha} = I - J^{-1}_{(\alpha)} J_{(\alpha)} = 0 \qquad \text{b.71}$$

Con objeto de aumentar la confiabilidad del algoritmo en b.68 se ha sugerido utilizar el paso ρ^{k+1} sólo como una dirección de búsqueda s^k y ajustar su tamaño con el escalar β^k . De esta forma b.68 queda :

$$J^k s^k = -f^k$$
$$x^{k+1} = x^k + \beta^k s^k \qquad 6.72$$

y β^k se escoge para controlar el error $f(x) = \frac{1}{2}f^T(x)D^2f$ donde D es una matriz diagonal, (La matriz identidad). Ahora bien β^k se puede escoger de tal forma que se cumpla $f(x^{k+1}) < f(x^k)$. La reducción de β^k se efectua con un factor θ constante (0.25), hasta que se cumpla $f(x^{k+1}) < f(x^k)$, con lo cual se garantiza que el error f(x) disminuya monotónicamente para la predicción de cada nueva dirección s^k. En general es necesario ajustar β^k cuando se está lejos de la solución, ya que en esas circunstancias $\|p\|^2$ tiene un valor significante. Por otro lado, cerca de la solución, este término se vuelve despreciable por lo que el valor de $\beta^k = 1$ producirá decrecimiento en el error.

Las etapas básicas de este método son :

1.- Evaluar x^o, definir ξ, $k = 0$, θ_c, D
2.- Evaluar $f(x^o) = f^o$ y $f(x^o) = f^{oT}D^2f^o$
3.- Evaluar $J(x^k)$
4.- Evaluar $\left[J(x^k)\right]^{-1}$
5.- Determinar la dirección $s^k = -(J^k)^{-1}f^k$
6.- Poner $\beta^k = 1$
7.- Evaluar $x^{k+1} = x^k + \beta^k s^k$
8.- Evaluar f^{k+1} y $f(x^{k+1}) = (f^{k+1})^T D^2 f^{k+1}$
9.- Si $f(x^{k+1}) \geq f(x^k)$ poner $\beta^k = \theta\beta^k$ e ir al paso 7.
10.- Si $f(x^{k+1}) < \xi$ el problema converge.
11.- Poner $k = k + 1$ e ir al paso 3.

Cabe señalar que una de las mayores ventajas del método de Newton es que tiene una aplicabilidad muy amplia pues sus condiciones de convergencia son menos restrictivas que las de los metodos con convergencia lineal. Por otro lado debido a la convergencia de tipo cuadrático del Newton, se requieren en general menos iteraciones que con los metodos de primer orden.

Es necesario también señalar algunos desventajas del método de Newton Uno de ellas es que si el Jacobiano es singular en un punto de iteración dado, el método diverge; hay que señalar sin embargo que ese caso es poco frecuente en la práctica. Otra desventajo es que hay que invertir la matriz del jacobiano en cada iteración lo cual puede requerir de un tiempo de computación relativamente grande. Sin embargo la mayor desventaja del método es que requiere la evaluacion del jacobiano, ya que en la práctica muchas veces no se cuenta con las expresiones analíticas o bien puede resultar tedioso el calcularlas. Para evitar el tener nue evaluar el jacobiano de una forma directa se han propuesto algunos métodos conocidos como quoasi - Newton que en algunos casos, aceleran la convergencia. Estos métodos se analizan en los siguientes secciones.

Finalmente para concluir conviene señalar, que cuando se cuenta con las expresiones analíticas de las ecuaciones y no resulta demasiado complicado el calcular el jacobiano; el método de Newton resulta sin duda, la mejor elección de todos los metodos disponibles.

III.7 METODO DE LA SECANTE GENERALIZADA.

Este método no requiere de la evaluación del Jacobiano en cada iteración como en el caso del método de Newton. La idea fundamental en este método consiste en aproximar el Jacobiano J con un hiperplano el cual pasa por n+1 valores de la función F(x); siendo Barnes(1965) uno de los primeros que propuso este método.

Suponer ℓ funciones F en ℓ variables x, también que el problema involucro solo circuitos implícitos. Se tratará de hallar la solución para encontrar las funciones lineales las cuales se aproximan a funciones no-lineales, se pueden ajustar a cero y resolver para sus raíces y esperar que estas se aproximen a las raíces de las funciones no-lineales. Suponer el funcionamiento del siguiente experimento numérico:

$$\begin{array}{ll} x^0 & e^0 = F(x^0) \\ x^1 & e^1 = F(x^1) \\ \vdots & \vdots \\ x^\ell & e^\ell = F(x^\ell) \end{array}$$

siendo la función error e=F(x estimada) a los l+1 valores diferentes para las variables x. Se puede escribir un modelo lineal aproximado, el cual de estos datos exactamente

$$e^* = A'x + b$$
$$e^\ell = A'x^\ell + b$$

Se tienen ℓ+1 ajustes de ℓ ecuaciones lineales y las incognitas son los $\ell x \ell$ elementos de A' y los elementos de b. Entonces se tiene $\ell^2+\ell$ ecuaciones lineales en $\ell^2+\ell$ incognitas y por lo tanto se puede resolver, resumiendo y eliminando b se obtiene:

$$\Delta e^\ell = e^\ell - e^{\ell-1} = A'(x^\ell - x^{\ell-1}) = A'\Delta x^\ell$$

entonces:

$$A' = \Delta E \Delta x^{-1} \qquad b.73$$

se encuentra A' sólo si Δx tiene inversa. Puesto que se puede elegir como se perturba x; se puede garantizar una inversa; por lo que b se obtiene de:

$$b = e^\ell - A'x^\ell \qquad b.74$$

ahora se puede estimar el valor de x^{nueva}, la cual puede hacer $e^* \to 0$.

Este paso involucra la resolucion de las ecuaciones lineales

$$e^{*}=0=Ax^{nueva}+b$$

ó $$x^{nueva}=-A^{-1}b=-A^{-1}(e^{\ell}-Ax^{\ell})$$

$$x^{nueva}=x^{\ell}-A^{-1}e^{\ell}$$

Las etapas básicas de este método son:

1) hallar $e=F(x)$ para $\ell+1$ valores diferentes de x
2) Use las ecuaciones (8.73) y (8.75) para evaluar una nueva x
3) hallar $e=F(x)$ para la x estimada en el paso 2
4) cheque si el problema tiene convergencia(Los errores en'e' son lo bastante pequeños y/o el valor de Δx también es pequeño ó continuo)
   a) Si no converge, reemplace una de las (x^{ℓ},e^{ℓ}) ajustadas, utilizadas en el paso 2, por la nueva e iterada del paso 2.
   b) De lo contrario, termina el proceso.

Este método original de la secante generalizado ha tenido algunas mejoras y aproximaciones, que han desarrollado unas ecuaciones nuevas mejoradas para - la secante generalizada; La aproximación desarrollada por Westerberg(1979), implica que a cada paso, se recalcule la matriz A del dato original:

$$A'=\Delta E\Delta X^{-1}$$

a) que se obtenga un nuevo Δe para un nuevo Δx
b) que se remplace una columna de ΔE y Δx con este nuevo punto
c) por último recalcular A

Esto es posible ya que se mejora A de una manera diferente; ahora asumir - que se ha evaluado $\Delta e'$ para un específico $\Delta x'$. Se puede tener una matriz A^2 tal que

$$\Delta e'=A^2\Delta x' \qquad b.76$$

aprovechando que es un término de "rango 1" mejorado. Ahora para obtener A^2 -

$$A^2=A'+u'(v')^T \qquad b.77$$

Note que la matriz $u'(v')^T$ es un producto exterior de los vectores u' y v'

$$u'(v')^T=\begin{bmatrix} u'v_1 & u_1v_2 \cdots\cdots & u_1v_t \\ u_2v_1 & u_2v_2 \cdots\cdots & u_2v_t \\ \vdots & & \vdots \\ u_tv_1 & u_tv_2 \cdots\cdots & u_tv_t \end{bmatrix}$$

y esta matriz tiene un rango de 1, sustituyendo (B.77) en (B.76)

$$\Delta e' = \left[A' + u'(v')^T \right] \Delta x' \qquad \text{B.78}$$

En este punto se elegirá arbitrariamente v' y se obtendrá una fórmula para u' resulta:

$$u' = \frac{\Delta e' - A'\Delta x'}{(v')^T \Delta x'} \qquad \text{B.79}$$

como v' no es ortogonal a $\Delta x'$ (es decir, el denominador $(v')^T \Delta x'$ no es cero) se tiene un vector definido u'. Entonces se puede convertir cualquier $\ell x \ell$ arbitrario a la matriz A' que satisfaga la ecuación (B.76) usando un rango de 1 actualizado. Note que si $\Delta e'$ es igual a $A' \Delta x'$ (es decir, el modelo linealizado estima correctamente $\Delta e'$) entonces u' es el vector cero y no actualiza resultados.

Ahora obtener Δe^2 para un paso Δx^2 (x^2 puede ser el resultado de resolver la ecuación $x^2 = x' - (A^2)^{-1} e'$, entonces $\Delta x^2 = x^2 - x^1$). Se puede entonces requerir que $\Delta e^2 = A^3 \Delta x^2 = \left[A^2 + u^2 (v^2)^T \right] \Delta x^2$ B.80

Se puede encontrar facilmente u^2 y v^2 como antes, pero ahora se requiere A^3 para satisfacer el primer punto

$$\Delta e^1 = A^3 \Delta x^1 = \left[A^2 + u^2 (v^2)^T \right] \Delta x^1$$

seleccionando u^2 ortogonal a $\Delta x'$ se encuentra el resultado, la ecuación para obtener v^2 es:

$$v^2 = \Delta x^2 - \left[\frac{\left[(\Delta x)^T \Delta x^2 \right] \Delta x^1}{(\Delta x^1)^T \Delta x^1} \right] \qquad \text{B.81}$$

Para obtener otro dato, Δe^3 vs. Δx^3 se requiere

$\Delta e^3 = A^4 \Delta x^3$ junto con $\Delta e^1 = A^4 \Delta x^1$

y para $A^4 = A^3 + u^3 (v^3)^T$

encontrando que u^3 puede seleccionarse siendo ortogonal a $\Delta x'$ y Δx^2

este resultado puede ser generalizado en una forma obvia:

$$A^{i+1} = A^i + \frac{(\Delta e^i - A^i \Delta x^i)(v^i)^T}{\Delta x^i (v^i)^T} \qquad \text{B.82}$$

donde v^i es Δx^i ortogonalizado con $\ell - 1$ vectores propios Δx^j.

Los etapas básicas de este método son:

1) Evaluar A' ($A' = I$)
2) Determine $\Delta x'$ usando (B.75), entonces $\Delta x' = x' - x^0$ y evaluar el resultado $\Delta e' = e' - e^0$

3) Seleccionar $v^1 = \Delta x^1$ y actualizar A^1, encontrando A^2

4) Evaluar Δx^2 usando (2.75) y evalue Δe^2

5) Seleccionar v^2 igual a el componente del vector de Δx^2, el cual es ortogonal a Δx^1 y evalue el resultado, actualize A^2, encontrando A^3

6) Evaluar Δx^3 y evalue Δe^3

7) etc. , Cuando se logra la tolerancia relativa preestablecida, el problema converge, de lo contrario, continuar el método.

III.8 METODO DE BROYDEN.

Este método,es uno de los más importantes de los QUASI-NEWTON y al igual que el método de la secante,este método no requiere de la evaluación del Jacobiano en cada iteración,como en el caso del método de Newton;La diferencia básica entre el Broyden y la Secante,es que en el primero,unicamente se toma en cuenta el último punto para predecir el Jacobiano,mientras que,en el segundo se utilizan los n+1 puntos anteriores.

El objetivo del método de Broyden(1965) es eliminar la inversión de la matriz producida por la secante generalizada.

El Broyden requiere:

$$A^{i+1}Z = A^{i}Z \qquad b.83$$

La ecuación,para todos los vectores Z,los cuales son ortogonales a Δx_i,la actualización requiere la nueva A,el último Δx_i dentro del resultado del Δe^i - tambien usando un rango de 1,se obtiene :

$$\Delta e^i = A^{i+1}\,\Delta x^i = \left[A^i + v^i(v^i)^T\right]\,\Delta x^i \qquad b.83a$$

La ecuación b.83 también dá :

$$\left[A^i + v^i(v^i)^T\right]Z = A^iZ \qquad b.83b$$

Para toda Z diferente de cero, la cual es ortogonal a Δx^i, claramente se selecciona $v^i = \Delta x^i$, entonces el producto interno $(v^i)^TZ = (\Delta x^i)^TZ = 0$, si , $Z \neq 0$ es ortogonal a $\Delta x^i = 0$ para este cambio:

$$v^i = \frac{(\Delta e^i - A\Delta x^i)}{(\Delta x^i)^T\Delta x^i} \qquad b.83c$$

y

$$A^{i+1} = A^i + \frac{(\Delta e^i - A\Delta x^i)(\Delta x^i)^T}{(\Delta x^i)^T\Delta x^i} \qquad b.84$$

La cual es la fórmula de Broyden.

Dando un estimado para A^{i+1} como la predicción del próximo paso es dado por :

$$A^{i+1}\,\Delta x^{i+1} = (\Delta e^{i+1})_{deseado} = -e^i \qquad b.85$$

Entonces se puede resolver para encontrar Δx^{i+1}.

Las etapas básicas de este método son :

1.- Evaluar $H^1 = (A^1)^{-1} = I^{-1} = I$ y x^0

2.- Evaluar e^0 como función de x^0. Si $(e^0)^T e^0$ es bastante pequeña, termina.

3.- Evaluar $\Delta x^1 = -H^1 e^0$

4.- Hacer $x^i = (x^{i-1}) + \Delta x^i$ y evaluar e^i

5.- Evaluar $\Delta e^i = e^i - e^{i-1}$

6.- Evaluar v^i (Δx^i para el Broyden).

7.- Actualizar A^i encontrando A^{i+1} usando la ecuación b.85.

8.- Evaluar el próximo $\Delta x^{i+1} = -A^{i+1} e^i$

9.- Ajustar $i = i+1$ y repetir desdeel paso 4.

La ventaja del método de Broyden sobre el método de la secante generalizada es que los vectores propios Δx^i nó necesitan almacenarse. La desventaja del método de Broyden es que tiene convergencia superlineal; como lo demuestran Dennis y More (1977), mientras que la secante generalizada tiene convergencia cuadrática.

III. 8.1 METODO DE BROYDEN-HOUSEHOLDER.

Otra forma de fórmula de Broyden, es con la ecuación de Householder para relacionar la inversa de A^{i+1} a la inversa A^i si las dos difieren por un rango de uno, haciendo : $H^{i+1} = (A^{i+1})^{-1}$

Si H^{i+1} se conoce, entonces la ecuación b.85 se convierte en :

$$\Delta x^{i+1} = -H^{i+1} e^i \qquad \text{b.85a}$$

la cual es una simple multiplicación de matriz, para encontrar Δx^{i+1}, eliminando la necesidad de resolver la ecuación b.85.

La fórmula de Householder es:

$$H^{i+1} = (A^{i+1})^{-1} = \left[A^i + u^i (v^i)^T \right]^{-1}$$

$$H^{i+1} = H^i - \frac{H^i u^i (v^i)^T H^i}{\left[1 + (v^i)^T H^i u^i \right]} \qquad b.86$$

y la fórmula mejorada para H es :

$$H^{i+1} = H^i - \frac{(H^i \Delta e^i - \Delta x^i)(v^i)^T H^i}{(v^i)^T H^i \Delta e^i} \qquad b.87$$

igualando el escalar C_{k+1} y los vectores u_{K+1} y P^T_{k+1}

$$C_{k+1} = \frac{1}{(\Delta x^i)^T \Delta x^i}$$

$$u_{k+1} = (\Delta e^i - A \Delta x^i)$$

$$P^T_{k+1} = \Delta x^T_i$$

Sustituyendo lo anterior en la ecuación b.84 se tiene :

$$J_{k+1} = J_k + u_{k+1} C_{k+1} P^T_{k+1} \qquad b.88$$

J_o es el estimado inicial de la matriz jacobiana con el cual el proceso iterativo se inicia :

$$J_o \Delta x_o = -f_o \qquad b.89$$

y

$$\Delta x_0 = -J_o^{-1} f_o \qquad b.90$$

Aunque la inversa de J_o aparece en esta ecuación, puede notarse que la expresión implícita de J_o^{-1} nunca necesita desarrollarse, sólo se requiere la factorizacion de LU. Si J_o es dispersa, J_o^{-1} no necesariamente es dispersa, pero la

factorización LU si es dispersa; por lo que, en el desarrollo restante, las inversas se muestran, pero las soluciones numéricas son encontradas por el uso de la factorización LU. Después de que Δx_0 ha sido usado para encontrar x_1; la matriz jacobiana mejorada J_1 se encuentra como sigue :

$$J_1 = J_0 + u_1 C_1 P_1^T \qquad b.91$$

Después de que Δx_1 ha sido usado para encontrar x_2; la matriz J_2 se encuentra como sigue :

$$J_2 = J_1 + u_2 C_2 P_2^T \qquad b.92$$

continuando este procedimiento, la matriz J_{k+1} se encuentra como sigue :

$$J_{k+1} = J_0 + \sum_{i=1}^{k+1} u_i C_i P_1^T \qquad b.93$$

De este modo se obtiene J_{k+1} en términos del estimado inicial J_0 y de las correcciones de Broyden para cada iteración sucesiva.

Las etapas básicas de este método se dan abajo para resolver las ecuaciones Newton - Raphson, usando sólo la factorización LU de J_0 y los términos mejorados Broyden, dados por las ecuaciones b.88, b.89 y b.90 .

1.- Hacer $j = 0$, resolver $J_0 \Delta x_0 = -f_0$

   a) Evaluar u_1, P_1^T

2.-Evaluar w y z, haciendo $J_0 w = -f_{k+1}$ y $J_0 z = u_{k+1}$

3.- Si $k = 0$ ir al paso 5.

4.- Hacer $j = 1,2,\ldots k$

$$\beta = \alpha_j P_j^T w$$
$$\gamma = \alpha_j P_j^T z$$
$$w = w + \beta v_j$$
$$z = z + \gamma v_j$$

5.- $v_{k+1} = z$

$$\alpha_{k+1} = \frac{1}{(1/C_{k+1}) + P_{k+1}^T v_{k+1}}$$

$$\beta = \alpha_{k+1} P_{k+1}^T w$$

$$\Delta x_{k+1} = w + \beta v_{k+1}$$

6.- Evaluar x_{k+1}, f_{k+1}, v_{k+1} y regresar al paso 2.

III.8.2 METODO BROYDEN-BENNETT.

En el método propuesto por Broyden, la fórmula de Householder fue usada para obtener la fórmula de la matriz inverso mostrada en la expresión 6.87; el segundo término del lado derecho de la expresión, contiene la corrección propuesta por Broyden. Ahora, en lugar de utilizar la fórmula de Householder, el cálculo de la inversa del jacobiano se evita completamente por el algoritmo propuesto por Bennett ( ), para mejorar los factores LU de la matriz jacobiano.

El algoritmo de Broyden mejora sucesivamente la matriz jacobiana usando la matriz de corrección $v_{k+1}c_{k+1}p_{k+1}$. Puesto que la matriz jacobiana puede establecerse en términos de los factores LU, la expresión 6.88 para J_{k+1} puede apoyarse en la siguiente forma :

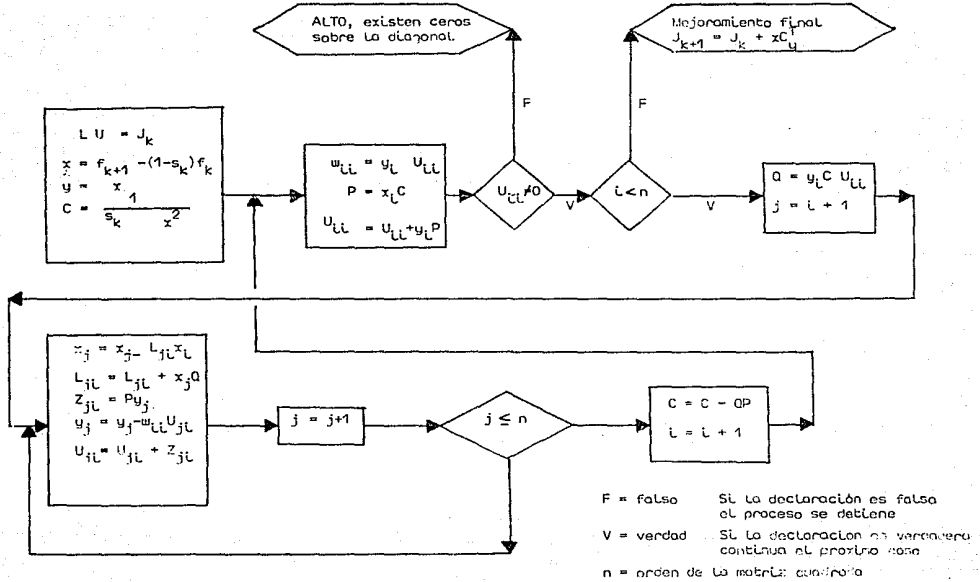
$$L_{k+1}U_{k+1} = L_kU_k + v_{k+1}c_{k+1}p_{k+1} \qquad 6.94$$

Bennett propone el algoritmo de la figura (19) para mejorar las matrices L_k y U_k y obtener las matrices mejoradas L_{k+1} y U_{k+1}. Cuando se usa el algoritmo de Bennett para la corrección del Broyden, se utiliza el siguiente procedimiento de cálculo :

1.- Asumir unos valores iniciales para las variables x_o y calcular $f_o=f(x_o)$

2.- Aproximar los elementos de $-J_o^{-1}$

a) Broyden obtiene una primera aproximación de los elementos de J_o por el uso de la fórmula de diferencias finitas de tal forma que cada elemento de J estará dado por :



ALgoritmo Bennett.
FIGURA 19

$$\frac{\partial f_i}{\partial x_j} \cong \frac{f_i(x^k + he_j) - f_i(x^k)}{h} \qquad b.95$$

donde h es el tamaño de perturbación (típicamente 1/1000 del valor de la variable perturbada) y e_j es la columna j de la matriz identidad.

b) Ahora, encontrar los factores L_o y U_o, tal que $J_o = L_o U_o$

3.- Teniendo la base del más reciente valor de L_k, U_k y f_k; calcular Δx_k y

$$L_k U_k \Delta x_k = -f_k$$

4.- Evaluar el factor de amortiguamiento S_k, tal que la norma Euclidiana de $f(x_k + S_k \Delta x_k)$ sea menor que la de $f(x_k)$. Primero probar $S_{k1} = 1$ y si la desigualdad se satisface :

$$\left[\sum_{i=1}^{n} f_i^2(x_k + S_k \Delta x_k)\right]^{1/2} < \left[\sum_{i=1}^{n} f_i^2(x_k)\right]^{1/2} \qquad b.96$$

se procede al paso 5; de otro modo calcular S_{k2}, utilizando la siguiente fórmula desarrollada por Broyden

$$S_{k2} = \frac{(1 + 6\eta)^{1/2} - 1}{3\eta} \qquad b.97$$

donde :

$$\eta = \frac{\sum_{i=1}^{n} f_i^2(x_k + S_k \Delta x_k)}{\sum_{i=1}^{n} f_i^2(x_k)} \qquad b.98$$

Si la norma no es reducida por el uso de S_{k2} después de un número específico de pruebas a través del procedimiento completo, regresar al paso 2; y reevaluar la derivada parcial de J_k sobre la base de x_k

5.- Probar f_{k+1} para la convergencia, si la convergencia no se ha logrado, calcular C_{k+1}, U_{k+1}, P^T_{k+1}

6.- Usar el algoritmo de Bennett para obtener las matrices mejoradas L_{k+1} y U_{k+1} de L_k y U_k y regresar al paso 3.

III.9 RELACION ENTRE LOS METODOS QUASI-NEWTON (QN) Y LOS METODOS DE LOS VALORES PROPIOS DOMINANTES (DEM).

En la simulación de procesos a estado estable, un procedimiento común es calcular en secuencia cada unidad de proceso, iniciando con estimados o supuestos iniciales y finalizando cada iteración con una reevaluación de esas variables. El procedimiento iterativo es equivalente a resolver un sistema de "m" ecuaciones algebraicas no-lineales. Para resolver este sistema de ecuaciones, los métodos QN son los más comunmente usados; en estos, una matriz es mejorada en cada iteración, lo que puede ocasionar una inestabilidad numérica cuando las ecuaciones son casi linealmente dependientes (8) o cuando el valor de la función $f_{i+1} >> f_i$ (43).

Otra posible forma de solución del sistema de ecuaciones algebraicas no lineales, es la que se obtiene con el uso del método de sustitución directa, en donde el valor de la variable calculada se utiliza para evaluar la próxima. La convergencia de este método puede acelerarse por el método de los valores propios dominantes, que pueden aplicarse cuando los estimados sucesivos de la solución aparente son lo suficientemente cercanos. Desde luego, también la aceleración frecuente ha conducido a inestabilidad numérica.

El trabajo realizado recientemente por Crowe (1984) muestra la comparación entre los métodos QN y DEM. Crowe muestra que para cualquier algoritmo QN existe un algoritmo DEM, tal que si la aceleración es aplicada a cualquier iteración, la secuencia de valores de las variables de corte pueden ser idénticas. Para establecer la relación entre los métodos QN y DEM, se necesita el siguiente :

Lema 1.- Para cualquier método QN

$$\Delta x_i = (Z_{i-1}^T f_{i-1}) \left\{ J_o f_i - \sum_{j=1}^{i-1} w_{j+1,i} \Delta x_j \right\} \qquad 6.99$$

para $i = 1,2,3,\ldots n$ y dado que solo la Z_i satisface la siguiente ecuación :

$$z_i^T \Delta f_j = q_i \Delta f_j = 1 \qquad (0 \leq i \leq m-1) \qquad \text{b.100}$$

por lo que :

$$W_{ji} \triangleq \frac{(z_j^T f_j)}{(z_j^T f_j)} \qquad \text{b.101}$$

La ecuación b.99 es también válida para i = 1 si se toma la sumatoria vacía

Prueba : De la fórmula de los QN de rango 2 se tiene:

$$J_{i+1} = J_i \left[I - \Delta f_i z_i^T - S_i f_i q_i^T \right] \qquad \text{b.102}$$

donde S es un escalar arbitrario diferente de cero. Los vectores Z_i y q_i son arbitrarios excepto para las condiciones de normalización de la ecuación b.100 Sustituyendo las ecuaciones b.100 y b.102 en:

$$x_{i+1} = x_i - S_i J_i f_i \qquad \text{b.103}$$

se obtiene :

$$\Delta x_i = J_{i-1} f_i (z_{i-1}^T f_{i-1}) \qquad \text{b.104}$$

ahora; considerando $Z_i = q_i$ y $S_i = 1$ de la ecuación b.102 en la fórmula de los QN de rango uno se tiene :

$$J_{i+1} = J_i \left[I - f_{i+1} z_i^T \right] \qquad \text{b.105}$$

ahora para $i \geq 1$ de las ecuaciones b.105; b.101 y b.104

$$J_\ell f_i = J_{\ell-1} f_i - W_{\ell-1,i} \Delta x_\ell \qquad \text{b.106}$$

para $\ell \geq 1$ y $i \geq 0$. Así, la sustitución de la ecuacion b.106 dentro de la ecuación b.104 resulta en la ecuación b.99.

Es aparente del Lema 1 que sólo los valores de W_{ji} con $j < i$ son necesarios para predecir Δx_i, dado que la ecuación b.100 se mantiene. De esta manera no es necesario elegir por anticipado, el vector Z_i ; antes de la ésima iteración, ni se necesitan las condiciones siguientes :

$$z_i^T \Delta f_j = q_i^T \Delta f_j = 0 \qquad (0 \leq j < i \leq m-1) \qquad \text{b.107}$$

para fijar z_i

Crowe establece el resultado principal como el siguiente:

Teorema.- Existe una relación uno a uno entre un conjunto de coeficientes

$$\left\{ \mu_j^i \quad : \quad 0 \leq j \leq i \; ; \quad i \geq 1 \right\}$$ para el DFM con la condición

$$\mu_0 \equiv 1 \quad (80) \qquad \sum_{j=0}^{s} \mu_j^{(s)} = 1 \quad (80)$$

y un conjunto de valores de $\left\{ W_{ji} : 0 \leq j \leq i \; ; \; i \geq 1 \right\}$ para los QN de rango - uno tal que cada algoritmo producirá exactamente la misma secuencia x_i del - mismo inicio x_0 y J_0 para un problema arbitrario. Esta relación es :

$$\mu_\ell^{(i)} = \mu_0^{(i)} (W_{i-\ell-1,i} - W_{i-\ell,i}) \qquad \text{b.108}$$

para $\ell = 1,2,\ldots,(i-1)$ y

$$\mu_\ell^{i} = -\mu_0^{(i)} w_{\ell i} \qquad \text{b.108a}$$

junto con $\mu_0^{(i)} = 1 \qquad (80)$ b.108b

$$\mu_0^{(i)} = -(z_{i-1}^T f_{i-1}) \qquad (81)$$

$$= 1 \Big/ (1 - w_{i-1,i}) \qquad \text{b.108c}$$

De la ecuación b.100. Notese que de la ecuación b.100 y b.101; otra vezso- lo se necesitan valores para W_{ji} para $0 \leq j \leq i-1$

$$z_{i-1}^T f_{i-1} = -\mu_0^{(i)} \Big/ \sum_{j=0}^{i} \mu_j^{(i)} \qquad \text{b.109}$$

para $i \geq 1$ y $$w_{\ell i} = -\sum_{j=i-1}^{i} \mu_j^{(i)} \Big/ \mu_0^{(i)} \qquad \text{b.109a}$$

para $0 \leq \ell \leq i-1$.

Prueba.- Si ON1 y DEM inician con el mismo x_0 y J_0 en un problema arbitrario- entonces, claramente Δx_0 es el mismo para cada método. Ahora; suponiendo que Δx_ℓ es el mismo para cada método por $\ell = 0,1,\ldots,i-1$, entonces Δx_i se-- rá el mismo si y sólo si los coeficientes de los términos correspondientes en las ecuaciones b.46 y b.99 son iguales.

Similarmente pero con relaciones más complicadas, pueden derivarse de las ecuaciones de los ON de segundo orden (ON2) de la forma de la ec. b.102

De acuerdo al Teorema 1, se necesita el siguiente :

Corolario : Para $\ell = 2,3,\ldots,(i-1)$

$$\mu_{i-1}^{(i)} = \frac{Z_\ell^T J_\ell^{-1}}{Z_\ell^T F_\ell} V_{\ell-1,i} \qquad b.110$$

donde :

$$V_{\ell i} \triangleq \mu_0^{(i)} \left[J_0 f_i + \sum_{k=1}^{\ell} w_{k-1,i} \Delta x_k \right] \qquad b.111$$

Las ecuaciones restantes para $\ell = 0,1$, proveen cualquier suma con su - límite superior menor que su límite inferior menor ajustado a cero.

Prueba.- De la ecuación b.106

$$J_\ell f_i = J_0 f_i - \sum_{k=1}^{\ell} w_{k-1,i} \Delta x_k \qquad b.112$$

premultiplicando por : $Z_\ell^T J_\ell^{-1} / (Z_\ell^T f_\ell)$ y con la ecuación b.101 se obtiene :

$$w_{\ell i} = \frac{Z_\ell^T J_\ell^{-1}}{Z_\ell^T f_\ell} \left[J_0 f_i - \sum_{k=1}^{\ell} w_{k-\ell,i} \Delta x_k \right] \qquad b.113$$

El resultado sigue de la ecuación b.108 usando $\Delta x_\ell = -\lambda_\ell f_\ell$. Entonces, los - coeficientes $\mu_{i-\ell}$ pueden calcularse en secuencia de la ecuación b.110 con - la ecuación b.109a, iniciando con $\ell = 0$.

Es conveniente proceder ahora a usar las relaciones desarrolladas antes entre los métodos ON y DEM para interpretar el método de Broyden en términos de su formulación equivalente en DEM. Otras formulaciones equivalentes de los

QN se encuentran en Crowe (1984).

Formulación equivalente del Broyden en DEM

Dela ecuación b.110 y sustituyendo para Z_{ℓ}

$$\mu_{i-\ell}^{(i)} = \frac{-(\Delta x_{\ell}^{T} v_{\ell-1,i})}{\Delta x_{\ell}^{T} \Delta x_{\ell}} \qquad b.114$$

para $\ell = 0,1,\ldots,(i-1)$ con $\mu_0 = 1$

Esto es equivalente a el siguiente problema :

$$\min_{\mu_{i-\ell}^{(i)}} \left\| \mu_{i-\ell}^{(i)} \Delta x_{\ell} + v_{\ell-1,i} \right\|^2 \qquad b.115$$

con

$$\| x \|^2 = x^T x \qquad b.116$$

Esta minimización es hecha sucesivamente por $\ell = 0,1,\ldots,(i-1)$ y se puede reemplazar la fórmula siguiente del DEM:

$$\mu_j^{(i)} \left\| \mu_0(-J_0 f_1) + \sum_{j=1}^{i} \mu_j^{(i)} \Delta x_{i-j} \right\|^2 \qquad b.117$$

Notese que la minimización de b.115 para $\ell = i-1$ no puede reducir el valor de la norma anterior la cual puede obtenerse por minimización de b.117 cambiando todos los coeficientes simultaneamente (con $\mu_0 = 1$).

El método de Broyden puede ser muy eficientemente implementado a través del uso de las ecuaciones b.114, b.111 y b.46 como se muestra enseguida:

Las etapas básicas de este nuevo método de Broyden son :

1.- a) Dar x_0, J_0; ajustar $i = 0$.
b) Evaluar $f_0 = f(x_0)$.
c) Evaluar $\Delta x_0 = -J_0 f_0$ y $x_j = x_0 + \Delta x_0$
d) Evaluar y almacenar $(\Delta x_0^T \Delta x_0)$

2.- Ajustar $i = i+1$

a) Evaluar $f_i = f(x_i)$

$$V_{oi} = - J_o f_i$$

$$W_{oi} = (\Delta x_o^T V_{oi}) / (\Delta x_o^T \Delta x_o)$$

Si $i = 1$ ir al paso 4.

3.- Para $j = 1,2,\ldots,(i-1)$

$$V_{ji} = V_{j-1,i} + W_{j-1,i} \Delta x_j$$

$$W_{ji} = (\Delta x_j^T V_{ji}) / (\Delta x_j^T \Delta x_j)$$

4.- $\Delta x_i = (V_{i-1,i}) / (1 - W_{i-1,i})$

Evaluar y almacenar $(\Delta x_i^T \Delta x_i)$

$x_{i+1} = x_i + \Delta x_i$

5.- Si $\| \Delta x_i \| <$ tolerancia$_1$ y $\| f_i \| <$ tolerancia$_2$. alto.
Más, si $i < i_{max}$ ir al paso 2; de otro modo alto.

Este algoritmo puede fácilmente adaptarse a otro método QN1 en el cual $z_i^T = v_i^T J_i$ para el mismo vector v_i e igualmente puede cubrir los metodos QN2 - tal como el método de Davidon - Fletcher - Powell; para la función de minimización. Este nuevo método fue más rápido que el método convencional de Broyden

III.10 MEJORAMIENTO PARA LA SOLUCION DE ECUACIONES ALGEBRAICAS NO-LINEALES.

Dentro del campo de los diferentes métodos de aceleración de convergencia, existe un continuo y dinámico desarrollo en este contexto tan es así que aún en estos momentos se encuentran por publicarse (o aún se encuentran en estudio) nuevos versiones o modificaciones de los metodos más usuales y, quizas métodos nuevos y novedosos. De hecho, recientemente, el trabajo de M.A. Soliman (1985), viene a enriquecer, todavia más este campo. Soliman lleva a cabo un mejoramiento a la fórmula de Davidon

Sea $f(x) = 0$ donde f es continuamente diferenciable. Asumiendo que se tiene una aproximación B a $\nabla f(\bar{x})$ donde $\bar{x}$ está dada por:

$$\bar{x} = x - tJf(x) \qquad b.118$$

donde $J = B^{-1}$ y t es un factor de amortiguamiento.

Davidon (14) sugiere la siguiente fórmula para mejorar B

$$\bar{B} = B - \frac{(Bq - Y)(Bq - Y)^T}{(Bq - Y)^T q} \qquad b.119$$

con la inversa $\bar{J}$ dada por :

$$\bar{J} = J - \frac{(JY - q)(JY - q)^T}{(JY - q)^T Y} \qquad b.120$$

donde :

$$Y = f(\bar{x}) - f(x)$$

$$q = \bar{x} - x$$

Este método tiene la propiedad de terminación cuadrática, cuando se aplica a la minimización de funciones cuadráticas. Sin embargo si sufre el hecho de que $(JY - q)$ y Y puedan ser ortogonal y así entonces la mejora no está definida a alguna iteración. Se han sugerido muchas mejoras sobre el algoritmo básico (81) pero esas mejoras aumentan el tiempo de cálculo.

La mejora de Soliman no satisface la ecuación Quasi-Newton $(\bar{B}q - Y) =$ pero elimina muchas de las dificultades asociadas con la mejora de Davidon, -

esto garantiza que se defina a cada iteración y sea siempre positivo.

El trabajo que se realiza sobre la formulación de Davidon, básicamente está dado por :

$$\bar{B} = B - \theta \frac{(Bq - Y)(Bq - Y)^T}{(Bq - Y)^T q} \qquad 6.121$$

donde θ es un parámetro que se elige para que se reduzca a la unidad en el caso unidimensional; esto provoca que el algoritmo se reduzca al método de la secante y tal que la mejora preserva a la propiedad de que sea siempre positivo.

Un posible cambio para θ cuando $t = 1$ es:

$$\theta = \frac{(Bq - Y)^T q \; q^T f(\bar{x})}{(Bq - Y)^T Jf(\bar{x}) q^T Bq} \qquad 6.122$$

En este caso la inversa de $\bar{J}$ está dada por :

$$\bar{J} = J - \frac{(JY - q)(JY - q)^T q^T f(\bar{x})}{(JY - q)^T f(\bar{x}) q^T Y} \qquad 6.123$$

otra posible eleccion para θ quedá una mejora menos eficiente es :

$$\theta = \frac{(Bq - Y)^T q \quad Y^T Jf(\bar{x})}{(Bq - Y)^T Jf(\bar{x}) \quad Y^T q} \qquad 6.124$$

Notesé que $\bar{B}$ no satisface la ecuación ON $Bq = Y$ excepto en el caso unidimensional.

El primer resultado teórico que se tiene es útil cuando la mejora se aplica para la solución de las ecuaciones resultantes de la aplicación de las condiciones de primer orden para minimización, donde se puede hacer que la dirección $-Jf(x)$ sea descendente. El segundo resultado teórico concierne a la convergencia del algoritmo.

Si B es simétrico y positivo, entonces $\bar{B}$ como lo define la ecuación b.121 es definitivamente positivo si y sólo si, $q^T Y > 0$ y t=1. $\bar{B}$ tiene un valor propio no-negativo, así si $|\bar{B}| > 0$, $\bar{B}$ es positiva exacta (12).

Ahora si t = 1

$$f(\bar{x}) = Y - Bq \qquad \text{b.125}$$

$$f(\bar{x}) = Y - Bq \qquad y \qquad |\bar{B}| = |B| \frac{q^T Y}{q^T B q} \qquad \text{b.126}$$

de este modo para $|\bar{B}| > 0$, se debe tener $q^T Y > 0$ y así se prueba lo anterior.

En el caso t = 1 pero t> 0, se puede asegurar la propiedad de exactitud positiva, introduciendo J_L en la b.123 en lugar de J, donde J_L es :

$$J_L = J + \frac{(1-t)}{t} \frac{qq^T}{q^T f(x)} \qquad \text{b.127}$$

De aquí se puede probar fácilmente, que si y sólo si, t> 0 es positiva exacta para J positiva exacta. Ahora Notesé que :

$$q = -Jf(x) = -tJf(x) \qquad \text{b.128}$$

De este modo, el factor de escala para J es la unidad y J_L mejorada usando la ecuación b.123 se preservará la exactitud positiva. Despues J_L es introducida de la ecuación b.127 dentro de la b.123 la mejora se convierte en :

$$\bar{J} = J + \frac{(1-t)}{t} \frac{qq^T}{q^T f(x)} - \frac{\lambda}{(\lambda - 1)} \frac{\left[JY + \left(\frac{\lambda (1-t)-1}{t}\right) q\right]\left[JY + \left(\frac{\lambda (1-t)-1}{t}\right) q\right]^T}{\left[JY + \left(\frac{\lambda (1-t)-1}{t}\right) q\right]^T f(\bar{x})}$$

donde : $\lambda = \dfrac{q^T f(\bar{x})}{q^T f(x)}$ b.129

Ahora, si $f(x) = Ax - b$, donde es simétrica A y exacta positiva, el algoritmo definido por las ecuaciones b.118 y b.129 es Globalmente y Superlinealmente convergente a $A^{-1}b$ si $(J_L - A^{-1})$ es semiexacta (positiva o negativa) La prueba de este Teorema la realizan Dennis y More (1977), Esta nueva mejora ha probado ser más confiable que otras, la principal desventaja de ella es que puede requerir de más evaluaciones de funciones.

III.11 METODO HIBRIDO DE POWELL.

El método del híbrido de Powell, es una combinación del método de Newton y del método del paso descendente y tiene la característica de evitar la divergencia del método de paso descendente y la rápida propiedad convergente del método de Newton cerca de la solución.

El Newton es cuadráticamente convergente si x^k es cercano a x^*. Si x^k no es cercano a x^k, no hay garantía de que x^{k+1} pueda ser mejor que x^k, y el método de Newton, no necesariamente converge: un método que garantiza que:

$$\|f(x^{k+1})\| < \|f(x^k)\| \qquad b.130$$

es el método de paso descendente donde $\|\cdot\|$ es la norma Euclidiana; este método es convergente muy lentamente cerca de la solución x^*.

La iteración clásica Newton-Raohson reemplaza un estimado x^k de la solución por el estimado :

$$x^{k+1} = x^k + \delta^k \qquad b.131$$

donde δ^k resuelve el sistema lineal :

$$f_i(x^k) + \sum_{j=1}^{n} J_{ij}^k \delta_j^k = 0 \qquad b.132$$

Una modificación diferente de la iteración clásica de Newton ha sido propuesta por Levenberg (1944) y Marquardt (1963), en ellas la ecuación b.131 es reemplazado por la iteración :

$$x^{k+1} = x^k + \eta^k \qquad b.133$$

donde η^k resuelve el sistema lineal:

$$\sum_{j=1}^{n}\left\{\mu^k I_{ij} + \sum_{t=1}^{n} J_{ti}^k J_{tj}^k\right\}\eta_j^k = -\sum_{t=1}^{n} J_{ti}^k f_t(x^k) \qquad i = 1,2,\ldots\ldots n \qquad b.134$$

La matriz I es la matriz identidad y μ^k es un parámetro no negativo cuyo valor es calculado para probar la desigualdad b.130 Notesé que cuando $\mu^k = 0$, se tiene la iteración clásica y si μ^k se hace grande, entonces la solución de las ecuaciones lineales, tienden al valor :

$$\eta_i^k \simeq -\sum_{\mu^k}^{n} \frac{J_{ij}^k f_j(x^k)}{\mu^k} = -\frac{\frac{1}{2}\left[\frac{\partial f(x)}{\partial x_i}\right]_{x=x^k}}{\mu^k} \qquad b.135$$

Lo cual muestra que η^k tiende a un pequeño múltiplo negativo del gradiente de $f(x)$. Por lo tanto, grandes valores de μ^k hacen la iteración b.133 similar al método clásico de paso descendente aplicado a la función $f(x)$; así que, a menos que x^k sea un punto estacionario de $f(x)$, se puede calcular el valor de μ^k para el cual la desigualdad b.130 se obtiene.

El algoritmo híbrido de Powell es muy parecido al de Levenberg/Marquardt la más importante diferencia, es que éste no requiere las expresiones implícitas para el jacobiano, en lugar de eso, éste usa los valores sucesivos de $f_i(x^k)$ para reforzar la aproximación numérica del jacobiano por la técnica de Broyden (1965). Sin embargo, retiene la característica importante del Levenberg/Marquardt, que si la corrección completa del Newton es grande o larga, entonces el desplazamiento de x^k es hacia la dirección del paso descendente de $f(x)$.

Para iniciar la k-ésima iteración, se requiere el estimado x^k, una longitud de paso Δ^k y dos números E y M. La longitud de paso puede cambiarse en cada paso de iteración, con el proposito de restringir la longitud del desplazamiento $(x^{k+1} - x^k)$ para que en cada iteración decrezca el valor de $f(x)$ y dado que $f(x)$ decrece sustancialmente, Δ^k tiene un valor grande. Los números E y M son valores fijos positivos puestos antes de iniciar la iteración y ellos gobiernan las condiciones para finalizar el proceso iterativo. Este finaliza si el valor de $f(x)$ se reduce menor que E o si el gradiente de $f(x)$ es tan pequeño que la distancia de x a una solución es predecible exceder M.

Por lo tanto usualmente E se ajusta a un valor muy pequeño con la condición :

$$\sum_{i=1}^{n} \left[f_i(x)\right]^2 < \xi \qquad b.136$$

implica que x es aceptablemente cercano a la solución y M es usualmente un sobre-estimado de la distancia de x^1 a la solución y la otra condición de paro

es obtenida sólo cuando x está cercana a un punto estacionario (usualmente un mínimo local) de f(x).

Primero la k-ésima iteración calcula los elementos de la matriz jacobiana a x^k y entonces se evaluan, la corrección completa N-ewton-Raphson δ^k (para resolver el sistema lineal b.132) y también el gradiente g^k de f(x) a x^k, - calculando los componentes:

$$g_i^k = \left[\frac{\partial}{\partial x_j} f(x)\right]_{x=x^k} = 2\sum_{i=1}^{n}\left[f_i(x)\,\frac{\partial}{\partial x_j} f_i(x)\right] \qquad \text{b.137}$$

esto entonces prueba :

$$f(x^k) \geq M\|g^k\|_2 \qquad \text{b.138}$$

y si se cumple ; se finaliza la iteración porque se tiene la probabilidad de que la secuencia de estimados x^k no sea convergente a la solución de las ecuaciones, pero si a un mínimo local de f(x). Esta prueba es apropiada porque $\|g^k\|_2$ es la pendiente del paso descendente de f(x) a x^k, y por lo tanto esto muestra que la longitud del cambio en x^k que es necesario para reducir — f(x) a cero tendría que exceder a M; lo cual es erroneo, si M es especificado en la forma recomendada. En este caso el valor de $\|g^k\|_2$ puede ser inusualmente pequeño, asi se sospecha que un punto estacionario cercano de f(x) es - el causante de las dificultades. Notesé que se eligio una prueba que no está influenciada por la singularidad del jacobiano.

Si la condición b.138 no se mantiene, entonces se calcula un desplazamiento $\hat{\delta}^k$ sumado a el vector x^k. Este desplazamiento es justo la corrección clásica δ^k si $\Delta^k \geq \|\delta^k\|_2$, pero si $\Delta^k < \|\delta^k\|_2$ b.139

entonces la longitud δ^k se hace igual a Δ^k. En este caso el desplazamiento tiene la forma :

$$\hat{\delta}^k = \alpha_1\,\delta^k + \beta_1 g^k \qquad \text{b.140}$$

donde α_1 y β_1 son escalares. De hecho se permite $\alpha_1 = 0$ si el paso a lo largo del vector del paso descendente de f(x);

$$\bar{\delta}^k = -\frac{\Delta^k g^k}{\|g^k\|_2} \qquad \text{b.141}$$

no va más allá del mínimo predicho de $f(x)$ a lo largo del vector de paso descendente de x^k. Este mínimo predicho es como el punto :

$$x^k - \left\{ \frac{1}{2}\|g^k\|_2^2 \Big/ \|J^k g^k\|_2^2 \right\} g^k \qquad \text{b.142}$$

así se prueba la condición :

$$\Delta^k \leq \frac{1}{2}\|g^k\|_2^3 \Big/ \|J^k g^k\|_2^2 \qquad \text{b.143}$$

y si se mantienen las desigualdades b.139 y b.143, entonces $\bar{\delta}^k$ se define por la ecuación b.141. En el caso que la condición b.139 se mantenga pero la condición b.143 no se satisface; se permite que el punto $(x^k + \bar{\delta}^k)$ sea una línea recta que una el punto b.142 a el punto $(x^k + \bar{\delta}^k)$, los componentes actuales de $\bar{\delta}^k$ son determinados usando la ecuación $\|\bar{\delta}^k\|_2 = \Delta^k$.

Se tiene ahora especificado $\bar{\delta}^k$ en todos los casos y notese que esto interpola entre el método de paso descendnate y la corrección clásica Newton. La próxima etapa de la iteración, es probar el estimado $(x^k + \bar{\delta}^k)$; así, se calculan las funciones $f_i(x)$ a este punto. Si la desigualdad esperada se mantiene :

$$f(x^k + \bar{\delta}^k) < f(x^k) \qquad \text{b.144}$$

entonces la iteración se define $x^{k+1} = x^k + \delta^k$ y se prueba la convergencia b.136 a x^{k+1}. Sin embargo, si la condición b.144 falla, se hace $x^{k+1} = x^k$ y la prueba b.146 conduce a una reducción en la longitud del paso Δ^k. De este modo, la iteración revisa el estimado x^k usando sólo un cálculo del lado izquierdo del sistema de ecuaciones algebraicas no-lineales $f_i(x) = 0$.

La última etapa de la iteración revisa la longitud de paso Δ^k. Este cálculo depende sólo del valor predicho de la suma de cuadrados residuales a $x^k + \bar{\delta}^k$, es decir :

$$\phi^k = \sum_{i=1}^{n} \left[f_i(x^k) + \sum_{j=1}^{n} J_{ij}^k \bar{\delta}_j^k \right]^2 \qquad \text{b.145}$$

el cual es menor que $f(x^k)$. Si se encuentra que esta predicción es más mala - que el valor actual de la suma de cuadrados, se satisface la desigualdad :

$$f(x^k + \bar{\delta}^k) > (1 - \xi)\, f(x^k) + \xi\, \phi^k \qquad \text{b.146}$$

donde ξ es una constante entre (0,1); entonces se juzga que las aproximaciones lineales de las funciones $f_i(x)$ derivadas del jacobiano no son adecuadas sobre la distancia $\| \delta^k \|_2$. Por lo tanto se reduce Δ^k y de hecho se multiplica esto por un factor constante μ, $0 < \mu < 1$.

Sin embargo si la desigualdad b.146 falla, entonces puede incrementarse Δ^k de acuerdo a la misma estrategia y si no, no se reduce esto. Sólo el valor de Δ^k cambia de iteración a iteración y se requiere un estimado inicial que sea finito y positivo Δ^1.

Las etapas básicas de este método son :

1.- Evaluar x^k, Δ^k y E,M

2.- Evaluar δ^k y g^k con b.132 y b.137

3.- Probar b.138 si converge alto; si no continue.

4.- Evaluar el desplazamiento $\bar{\delta}^k$ si $\Delta^k < \| \delta^k \|_2$ entonces $\bar{\delta}^k = \Delta^k$ y evaluar con b.141

5.- Probar el estimado $(x^k + \bar{\delta}^k)$ con la desigualdad b.144

6.- Probar la convergencia b.136

7.- Si b.144 falla; probar b.146 y si falla evaluar e incrementar Δ con - b.145 y regresar a 1.

III.12 METODO DE BROYDEN-SCHUBERT.

Este método se encuentra intimamente relacionado con el métdo de Broyden En la ecuación b.84 (la cual es la fórmula de Broyden), el segundo término de la derecha es un vector de producto-salida. Schubert propone una modificación al Broyden para el acomodamiento de sistemas de ecuaciones no-lineales con Jacobiano disperso; siendo que la iteración es implementada como la solución de un sistema lineal con matriz de coefi-cientes (A^i). Los sistemas de ecuaciones serán consideradas dispersos si el jacobiano dá menos del 10% de entradas diferentes de cero. La norma de las entradas cero y no-cero, o la norma de espar cimiento se fija para todas las (x).

La idea de Schubert es que el cálculo se haría en la dirección de una fi la para preservar la estructura dispersa del jacobiano. La mejora de Schubert está dada por :

$$A^{i+1} = A^i + \sum_{\substack{j=1 \\ \Delta x \neq 0}}^{n} e_j e_j^T \frac{(\Delta e^i - A\,\Delta x^i)}{(\Delta x^{iT})(\Delta x^i)} (\Delta x^{iT}) \qquad b.147$$

donde $\Delta x^i = D_{js}$. Note que para $\Delta x^i = 0$, la corrección cero es adicionada a la j-ésima fila de A^i. La notación de la pseudo-inversa sería usada para permi tir una presentación de b.147. Para un escalar : $\alpha \in \mathbb{R}$

$$\alpha^{i+1} = \begin{cases} \alpha^{-1}, & \text{si } \alpha \neq 0 \\ 0, & \text{si } \alpha = 0 \end{cases}$$

Por lo que la mejora de Schubert podría reescribirse como :

$$A^{i+1} = A^i + \sum_{j=1}^{n} (\Delta x^{iT} \Delta x^i)^{i+1} e_j^T (\Delta e^i - A^i \Delta x^i) e_j \Delta x^{iT} \qquad b.148$$

Lam (36) muestra la convergencia local y superlineal para el método de Schubert en el caso especial cuando $\Delta x^i \neq 0$ para $i = 1,2,\ldots n$ en cada iteración Esta suposición esta implícita, aunque no establecida, sin quedar fuera del método Lam;

$$A^{i+1} = A^i - \sum_{j=1}^{n} e_j e_j^T (A^i \Delta x^i - \Delta e^i) \frac{\Delta x^{iT}}{x^{iT} \Delta x^i} \qquad b.149$$

si no está definida, la iteración podría terminar.

Sea $x = (\xi_1, \xi_2)^T$ considere $f(x) = (\xi_1^2 - \xi_2, \xi_2)^T$ con $x^* = (1,0)^T$
Si $x_0 = (1 + \xi, 0)$ para $\xi > 0$, y $A_0 = J(x^*) = \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix}$
entonces $f(x_0) = (\xi^2 + 2\xi, 0)^T$
y un cálculo muestra que : $\Delta x_0 = (-\xi - \frac{1}{2}\xi^2, 0)^T$
entonces $D_1 \Delta x_0 = \Delta x_0$ y $D_2 \Delta x_0 = (0,0)^T$
Por lo tanto A^1 esta bien definida por b.148.

El siguiente teorema es el resultado de la convergencia local y superlineal por el método de Broyden.

Suponga a) f es continuamente diferenciable en una salida de series convexas D_0
b) Aquí existe un $x^* \xi D_0$ tal que $f(x^*) = 0$ y $J(x^*)$ no es singular.
por lo tanto existe una constante, $k > 0$ tal que :

$$\|J(x) - J(\bar{x})\|_f \leq k\|x - \bar{x}\|_2 \qquad \text{para toda } x, \bar{x} \in D_0$$

entonces aquí existe ξ $\delta > 0$, tal que si $x_0 \in D_0$ satisface
$\|x_0 - x^*\|_2 < \xi$ y $\|A_0 - J(x^*)\|_f < \delta$ para A_0 no singulares, entonces:

1.- El método de Broyden genera (A^i) con A^i no singular.
2.- (x^k) converge a x^* y
3.- La convergencia es superlineal en el sentido que:

$$\lim_{k \to \infty} \frac{\|x^{k+1} - x^*\|_2}{\|x^k - x^*\|_2} = 0 \qquad \text{b.150}$$

El teorema de deterioración limitada, genera su nombre de la hipótesis sobre el comportamiento de la aproximación al jacobiano. Para algún vector - norma se tiene :

$$\sigma(x^i, x^{i+1}) = \max(\|x^i - x^*\|, \|x^{i+1} - x^*\|) \qquad \text{b.151}$$

y se define :

$$E^i = A^i - J(x^*) \qquad y \qquad E^{i+1} = A^{i+i} - J(x^*) \qquad \text{b.152}$$

Suponiendo que existen constantes $\alpha_1, \alpha_2 \geq 0$, tal que para alguna matriz $\|\cdot\|$,

$$\|E^{i+1}\| \leq \left[1 + \alpha_1 \sigma(x^i, x^{i+1})\right] \|E\| + \alpha_2 \sigma(x^i, x^{i+1}) \qquad 6.153$$

El error $\|E^{i+1}\|$ puede ser peor que $\|E\|$, pero sólo en una ruta controlada. La dificultad en la aplicación del teorema de deterioración limitada, es el - mostrar que para una norma particular que el estimado 6.153 se satisface por el método mejorado, generando la secuencia de aproximación al jacobiano; lo - que prueba que el método de Schubert genera (A^i) con A^i no-singular y x^i converge localmente y linealmente a x^*.

Denniss y More (1977), muestran que el método satisface una condición - necesaria y suficiente:

$$\lim_{k \to \infty} \frac{\|A^k - J^*)(x^{k+1} - x^k\|_2}{\|x^{k+1} - x^k\|_2} = 0 \qquad 6.154$$

para lograr el perfil de convergencia superlineal.

Las ventajas del método de Schubert, son que los requerimientos de almacenamiento son reducidos; el trabajo para mejorar la aproximación al jacobiano está en función de la dispersidad del problema y para cada iteración se resuelve un problema lineal disperso.

III.13.1 EL ALGORITMO CONLES

Los sistemas de ecuaciones no lineales son difíciles de resolver por cualquiera de las siguientes razones:

1.- Algunas de las funciones no son definidas para ciertos valores de las variables.
2.- Algunas soluciones del sistema no son posibles físicamente.
3.- Las funciones son extremadamente no-lineales y mal escaladas.
4.- Los sistemas son muy grandes y dispersos.

Shacham (1985) muestra que existen métodos y software que no pueden resolver problemas con las dificultades mencionadas; a menos que el estimado inicial sea muy cercano a la solución.

El programa de computadora CONLES puede resolver problemas con y sin restricciones. El diagrama de flujo del programa se muestra en la figura (29') En el circuito externo el valor de $\Delta\theta$ es para el método de continuación. En el circuito interno la secuencia de problemas se resuelve usando los métodos de Newton o Broyden con longitud de paso restringida. Cuando la matriz Jacobiano se hace casi singular o singular, el algoritmo se cambia al método de Levenberg/Marquardt.

En un sistema de ecuaciones no-lineales, cualquiera o todas las variables pueden ser sujetas a restricciones del tipo:

$$g(x) \geq 0 \qquad \text{b.177}$$

donde g es un vector de m funciones.

Se pueden distinguir 2 tipos de restricciones:

Restricciones físicas.- Aquellas originadas de fenómenos físicos los cuales se representan por sistemas de ecuaciones y en los cuales se puede encontrar la solución matemática válida para el sistema, pero que no satisface las restricciones de b.177. En este caso la solución matemática es imposible; por lo que no se representa correctamente el fenómeno físico.

Restricciones absolutas.- Aquellas originadas del intervalo limitado de definición de algunas funciones matemáticas. Las funciones Log y SQRT, tienen intervalos donde ellas no estan definidas. Esto no indica justamente que la solución no pueda ser localizada dentro de la región no-factible; pero que cualquier intento --

para calcular el valor de la función en esta región puede conducir a un error -- ejecutivo. Para resolver las ecuaciones con restricciones; el sistema $f(x) = 0$ y las restricciones b.177 pueden modificarse.

Las restricciones de la forma $x_i \geq 0$ no requieren cambio pero las restricciones diferentes deben llevarse a la forma : $Z \geq 0$ b.178

donde $Z \in R^m$; introduciendo las restricciones originales dentro del sistema :

$$\phi(x, z) = \begin{bmatrix} f(x, z) \\ z - g(x) \end{bmatrix} \qquad \text{b.179}$$

el ejemplo siguiente muestra de una forma más clara lo anterior :

Las restricciones explícitas sobre x (de consideración física) son $x \geq 0$ y $x \leq 1$. Pero para la expresión $(\ln[0.4(1-x)/(0.4-0.5x)])$ se hace indefinida para $0.8 \leq x \leq 1$, así que el límite superior sobre x puede ser $x < 0.8$ y entonces $g(x) = 0.8 - x$.

Usando la transformación definida en b.179 se obtiene :

$$\phi(x, z) \quad \begin{bmatrix} x/(z + 0.2) - 5\ln[0.8(z+0.2)/z] + 4.45977 \\ z - 0.8 + x \end{bmatrix} \qquad \text{b.180}$$

Este sistema esta bien definido para cualquier $x \geq 0$ y $z > 0$. Notese que en este ejemplo hay algo burda, al hacer las transformaciones explícitas, pero en muchos de los casos prácticos, las restricciones son principalmente de la forma : $x \geq 0$ y las transformaciones explícitas no se requieren.

Primero se considera la solución de ecuaciones restringidas en presencia de restricciones absolutas. En tales problemas la solución no puede existir en la - región no-factible, así, se puede considerar cualquier iteración que resulte una violación de las restricciones, como proposarse de la solución; entonces la dirección es correcta pero la longitud de paso debe recortarse, así que las restricciones no se violan. Los métodos de Newton o Broyden pueden usarse para este propósito.

La longitud de paso λ^k se calcula de:

$$\lambda^k = \min_j \left| \alpha \, z_j^k / p_j^k \right| \qquad \text{b.1 M}$$

donde j representa los índices de todas las variables restringidas para la cual $p_j^k < 0$ y α es un número pequeño cercano a uno, α se usa para asegurar que una variable con restricción absoluta nunca no hacerse exactamente cero durante las iteraciones. Hay algunas funciones que no son definidas para $z_j \leq 0$ (como $1/z_j$, ln de (z_j), etc.) y hay otros que no son definidas solo para $z_j < 0$. El cálculo de λ^k de b.1M previene el rompimiento del proceso de solución si la función no es definida para $z_j = 0$, pero por el otro lado, ésto permite la convergencia a la solución $z_j = 0$ si la función es definida en este punto.

El comportamiento con restricciones físicas es más difícil que con restricciones absolutas, la razón para esto es que en tales casos la solución puede situarse en la región no-factible y todos los métodos de solución pueden dirigirse hacia la solución no-factible. El método de Newton (o Broyden) con longitud de paso restringida tiende a la convergencia en tales problemas en un punto donde uno de las variables restringidas se hace cero y no hay forma de moverlo lejos de este punto. Una opción a esto, es el uso del método de Continuación y Homotopía para resolver problemas con restricciones físicas; ya que genera una curva de solución que son los valores residuales y pueden continuamente decrecer. Si una función tiene dos ceros distintos, los valores absolutos de los residuales pueden incrementarse y decrecer otra vez a lo largo de la línea conectando estos dos ceros dado que el estimado inicial es lo bastante cercano a la solución, las oportunidades para que el método de continuación converja a la solución factible son completamente buenas; aunque esto se logre a expensas de cientos de iteraciones de Newton-Raphson (Shacham, 1980b).

Una aproximación diferente de las restricciones físicas es convertirlas a restricciones absolutas por el uso de funciones penalizadas:

$$p = \sum_{j=1}^{m} \ln(z_j) \qquad b.182$$

y formular el sistema de ecuaciones de b.180 a:

$$\emptyset_p(x,z) = \emptyset(x,z)\ p \qquad b.183$$

Este sistema tiene la misma solución como el sistema original en la región factible. Por el otro lado la función penalizada, maneja los valores de la función a infinito cuando se aproximan las restricciones. El sistema modificado puede ser eficientemente resuelto usando el método de Newton (o Broyden) con longitud de - paso restringido.

Shacham ( ) realiza la comparación del método de continuación con el -- uso de funciones penalizadas en 2 problemas prueba que se ajustán con soluciones no-factibles localizadas cerca de la región factible.

El uso de $\Delta\theta = 1$ del método de continuacion, para lograr la solución factible, fue suficiente; en estos casos el número de iteraciones es pequeño. La adición de las funciones penalizadas incrementa la no-linealidad y consecuentemente incrementa el número de iteraciones. Los casos donde el método de -- continuación usa $\Delta\theta < 1$, la adición de las funciones penalizadas reduce las iteraciones por factor de 5 a 20. En otro caso la convergencia del método de -- continuación fué lenta por la existencia de un punto singular en el camino de - la solución y la adición de las funiones penalizadas, evita la convergencia en el máximo número de iteraciones permitido.

Cuando se usa el algoritmo se deben proporcionar las funciones y los estimados iniciales para las variables y una indicación de que si la variable es restringida o no y si la restricción es absoluta o física. Los siguientes problemas deben ser resueltos por el usuario.

a) La longitud de paso para dividir la aproximación de diferencias del Jacobiano.
b) El escalamiento automático.
c) El criterio de convergencia.
d) La singularidad del Jacobiano o la divergencia.
e) Falta de progreso satisfactorio.

A continuación se discuten las decisiones con respecto a los problemas:

a) El programa CONLES puede usar funciones derivadas; si tales funciones son dadas por el usuario. Para funciones más complicadas es conveniente el uso de diferencias finitas; seleccionando el valor de h_i lo bastante pequeña, que preserve la velocidad de convergencia de segundo orden del método de Newton-Raphson; pero no lo suficientemente pequeño que evite dominación de acercamiento al error.

CONLES, adopta el cambio para h_i hecho por More y Cosnard (1979) :

$$h_i = \varepsilon(x_i) \qquad \text{si } x_i \neq 0$$
$$h_i = \varepsilon \qquad \text{si } x_i = 0 \qquad \text{8.184}$$

donde ε es la raíz cuadrada del número más pequeño para el cual $1 + \varepsilon^2 > 1$.

b) Algunos de los problemas no pueden resolverse a menos que las funciones sean escaladas; permitiendo que el usuario ponga el escalamiento, se hace apropiado para el estimado inicial; pero se hace inapropiado para valores diferentes de los parámetros que se ajustan o las diferentes --

variables; así que se prefiere el escalamiento automático. CONLES usa un método similar a Chen y Stadtherr (1981), para la función de escalamiento. Cada vez que la matriz jacobiano es recalculado, el mayor elemento (en valor absoluto) de cada hilera del jacobiano se almacena en las variables : $norm_i$. Cuando se usa el método de Newton, cada hilera de la ecuación :

$$J(x^k)p^k = -f(x^k)$$

se divide por el factor apropiado de normalización : $norm_i$. Cuando se usa el método de Broyden solo los valores de las funciones se dividen por $norm_i$, excepto en las iteraciones donde el jacobiano es recalculado.

c) El criterio de convergencia depende de la medida del error relativo entre dos iteraciones consecutivas. Pero si la solución para algunos elementos del vector x es cero, entonces el error relativo no puede y podría ser no ser usado CONLES calcula el vector de error $\mathcal{E}$ elemento por elemento :

$$\mathcal{E}_i = \frac{\left| x_i^{k+1} - x_i^k \right|}{\left| x_i^k \right|} \quad \text{si } x_i^k \geq tol \quad \text{si no}$$

$$\mathcal{E}_i = \left| x_i^{k+1} - x_i^k \right| \qquad \text{b.185}$$

donde "tol" es la tolerancia del error especificada por el usuario. El primer criterio para convergencia es que $tol \geq \|\mathcal{E}\|_2$ b.186

El método de Newton, rápidamente converge; este criterio usualmente garantiza los $-\log_{10}(tol)$ dígitos significativos para todos los elementos de x^{k+1}; el criterio b.186 puede fallar en dos casos; si el factor de amortiguamiento se hace muy pequeño, el criterio se satisface lejos de la solución; la otra posibilidad es que el algoritmo Levenberg/Marquard, converja a un mínimo local de F(x), en tal caso el criterio b.186 puede también satisfacerse.

Para determinar que x^{k+1} es una solución verdadera, los valores residuales tienen que ser checados; sin embargo son afectados por el escalamiento de F. Para obtener una medida, lo cual es independiente del escalamiento, los valo-

res residuales son divididos por los factores de normalización $norm_i$, entonces el criterio adicional para convergencia es el siguiente :

$$\left\{\sum_{i=1}^{n}\left[f_i\,(x^{k+1})/norm_i\right]^2\right\}^{1/2} < 10.tol \qquad b.187$$

Cuando se usa el método de Broyden, el jacobiano y los factores de normalización son recalculados antes de usar el criterio b.187. Si b.186 y b.187 son satisfechos para x^{k+1}, se asume que es una solución verdadera. Si b.186 es satisfecha pero b.187 no, el algoritmo converge a un mínimo local. La subrutina puede intentar encontrar la solución verdadera regresando al estimado inicial y usando el método de continuación con pequeños valores de $\Delta\theta$.

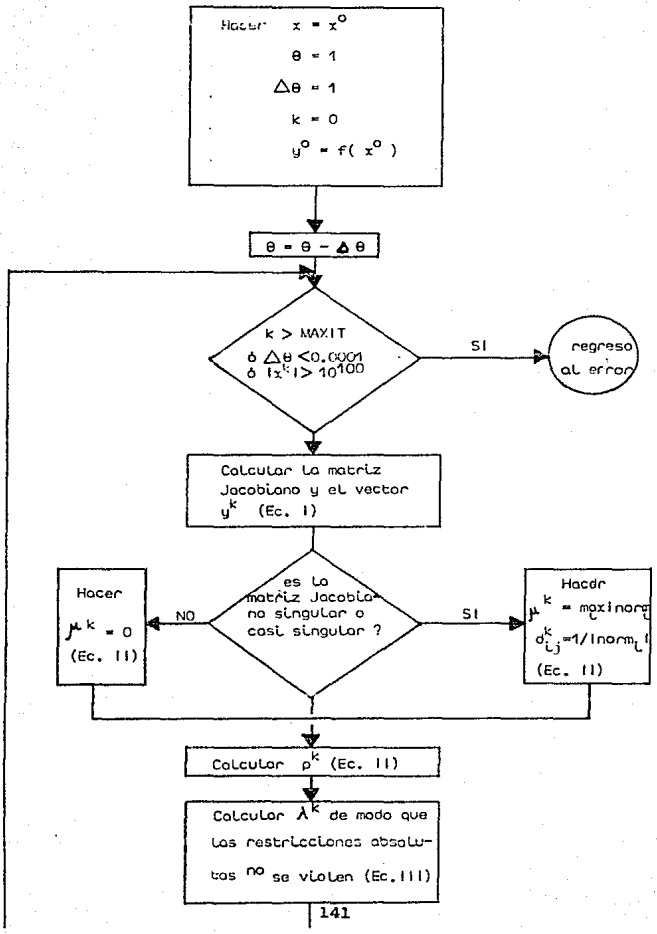
d) En el caso del jacobiano singular o casi singular se usa el algoritmo Levenberg/Marquard para moverse lejos del punto singular mientrás que se reduce la suma de cuadrados de los valores residuales.

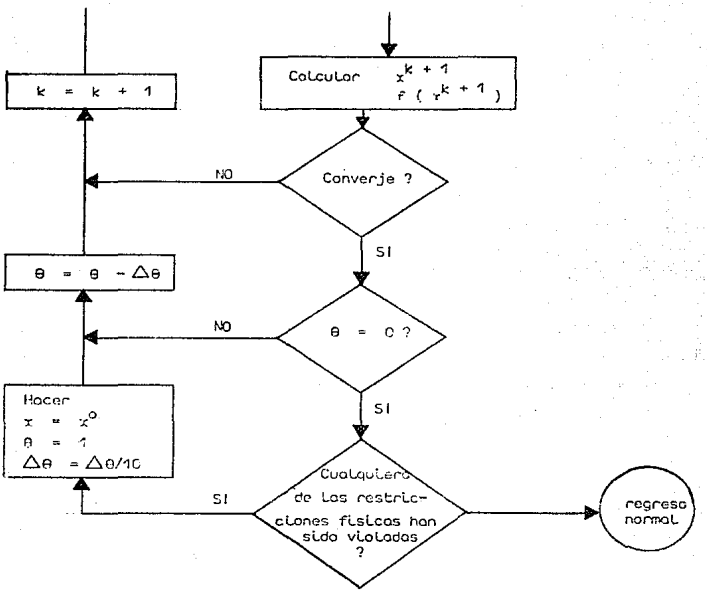
Para matrices casi singulares el valor de p^k tiende a ser más grande que x^k para detectar tal situación se hace lo siguiente, para los elementos del vector de corrección :

$$|p^k| < 10.|x_j^k|\,, \quad \text{si } |x_j^k| > 0.1, \quad \text{si no } |p^k| < 1.0 \qquad b.188$$

Si todos los elementos del vector de corrección p^k satisfacen la ecuación anterior, la corrección de Newton o Broyden es aceptada; de lo contrario, es recalculada usando el algoritmo Levenberg/Marquard. Puesto que este algoritmo es usado solo como una medida de emergencia, se minimiza el esfuerzo computacional asociado con èste y se intentan algunas opciones diferentes en la selección de v^k igual al mayor elemento (en valor absoluto) de J que aparece como el mejor cambio. La entrada diagonal de D^k contiene la inversa de los factores respectivos de normalización $1/norm_i$.

e) La falta de progreso normalmente conduce a la satisfacción del criterio b.186 sin satisfacer el criterio de b.187; en tal caso la subrutina puede probar un camino diferente de la solución usando el método de continuación en vez de detener las iteraciones cuando no hay progreso.


Hacer $x = x^0$
$\theta = 1$
$\Delta\theta = 1$
$k = 0$
$y^0 = f(x^0)$
$\theta = \theta - \Delta\theta$
$k > MAXIT$
ó $\Delta\theta < 0.0001$
ó $|x^k| > 10^{100}$
SI
regreso al error
Calcular la matriz Jacobiano y el vector y^k (Ec. I)
es la matriz Jacobiana singular o casi singular ?
NO
Hacer $\mu^k = 0$ (Ec. II)
SI
Hacer $\mu^k = \max_i \text{norm}_i$ $d^k_{ij} = 1/\text{norm}_i$ (Ec. II)
Calcular p^k (Ec. II)
Calcular λ^k de modo que las restricciones absolutas no se violen (Ec. III)



Ecuaciones :

$$y^k = f(x^k) - \theta\, y^0 \qquad \text{I}$$

$$\left[\mu^k D^k + J^T J\right] p^k = -J^T y^k \qquad \text{II}$$

$$x^{k+1} = x^k + \lambda^k p^k \qquad \text{III}$$

FIGURA 29

Diagrama de flujo y principales ecuaciones del Algoritmo CONLES.

El usuario especifica el número máximo de las iteraciones, si el progreso en la solución es lento. En cualquier caso el valor de x (para el cual la norma Euclidiana de los valores residuales es mínima) se almacena.

El algoritmo CONLES ha sido extensivamente probado con los problemas de -- Moré (1981). El funcionamiento de CONLES que usa el método de Newton fué comparado con el funcionamiento de otros códigos para ecuaciones no lineales. Los resultados detallados se dan en Shachom (1982b). La conclusión de este estudio es que CONLES es comparable o mejor que otros códigos en la resolución de problemas sin restricciones y que es el único que puede manejar problemas con restricciones.

III.13 METODOS DE CONVERGENCIA GLOBAL

III.13.1 CONTINUACION Y HOMOTOPIA DIFERENCIAL

Una gran cantidad de proyectos de investigación se han iniciado para desarrollar un verdadero método de continuación y homotopía diferencial; debido a que se encontró, que estos métodos son Globalmente convergentes. Desde 1976 cuatro rutinas han predominado: Kubicek (1976), Abbott (1975), Watson (1980) y Rheinboldt y Burkardt (1983). La mayoría de ellos proceden de Davidenko - (1953), y posteriormente desarrollados por Klopfenstein (1961).

Como se muestra en la tabla (3); las ecuaciones f(x) se introducen en una homotopía, h(x) con otro ajuste de funciones g(x) cuya solución es conocida o fácilmente determinada y un parámetro de homotopía t, el cual varía de 0 a 1. La solución se obtiene siguiendo una forma diferencial de la trayectoria de la ruta de g(x) a f(x).

Considere la aplicación del método de Kubicek (1976), a dos ecuaciones:

$$f_1(x_1, x_2) = x_1^2 + x_2^2 - 17 = 0 \qquad 3.15a$$

$$f_2(x_1, x_2) = 2x_1^{1/2} + x_2^{1/2} - 4 = 0 \qquad 3.15b$$

Se desea encontrar las ajustes de raíces de las ecuaciones, usando un estimado inicial de $x_1^0 = 10$, $x_2^0 = 10$, lo cual provoca que el método de Newton y algunos del híbrido de Powell, fallen. El método de continuación y homotopía diferencial, el cual se resume en la tabla (3); inicia con la construcción de una homotopía, h a relacionar la función f, por la ruta de otra función g, cuya solución es conocida o puede obtenerse fácilmente. Si una homotopía que es lineal se forma en un parámetro de homotopía t, :

$$h_1(x_1, x_2, t) = tf_1(x_1, x_2) + (1-t)g_1(x_1, x_2) = 0 \qquad 3.16a$$

$$h_2(x_1, x_2, t) = tf_2(x_1, x_2) + (1-t)g_2(x_1, x_2) = 0 \qquad 3.16b$$

cuando $t = 0$, al inicio de la ruta, $h = g = 0$, donde la solución conocida o - es x_1^0 o x_2^0. Cuando $t = 1$, $h = f = 0$, donde la raíz x_1, x_2 será determinada. Si la función de homotopía es continuamente diferenciable y la matriz de derivadas es invertible, el teorema de función implícita, garantiza la exis--

tencia de un camino continuo que enlace el punto inicial x_1^o, x_2^o a la solución deseada (x_1, x_2). Entonces sólo se necesita seleccionar un apropiado g_1 y g_2 y entonces plantear un esquema para apoyar sobre la ruta mientras se mueve de $t = 0$ a $t = 1$.

Un cambio posible para g es ($f - f^o$), el cual cuando se inserta en las ecuaciones b.157, b.158, se obtiene lo que se conoce como homotopia Newton:

$$h_1(x_1, x_2, t) = f_1(x_1, x_2) - (1-t)f_1(x_1^o, x_2^o) = 0 \qquad \text{b.159}$$

$$h_2(x_1, x_2, t) = f_2(x_1, x_2) - (1-t)f_2(x_1^o, x_2^o) = 0 \qquad \text{b.160}$$

donde x_1^o y x_2^o pueden seleccionarse arbitrariamente; con $x_1^o = 10$ y $x_2^o = 10$ -- (Seader, 1985) y la función como se da en las ecuaciones b.155 y b.156; las ecuaciones b.159 y b.160 hacen :

$$h_1(x_1, x_2, t) = x_1^2 + x_2^2 - 200 + 183t = 0 \qquad \text{b.161}$$

$$h_2(x_1, x_2, t) = 2x_1^{1/5} + x_2^{1/4} - 7.4711 + 3.4711t = 0 \qquad \text{b.162}$$

Una gráfica previa de la ruta para x_2 como una función de t se muestra en la figura (28).

En el método de continuación clásica (discreto), una secuencia de valores se selecciona para el parámetro de homotopia t, y las ecuaciones b.161 y b.162 se resuelven para cada valor sucesivo de t, iniciando a $t = 0$. Desafortunadamente, esta forma de continuación puede ser ineficiente e insegura y no se recomienda porque no sigue la curvatura real de la ruta.

El conjunto de ecuaciones no-lineales de acuerdo a las ideas de Davidenko (1953), es reformulada como un conjunto de ecuaciones simultaneas diferenciales ordinarias por diferenciación con respecto a la Longitud del arco, p de la ruta. Entonces el método es según Wayburn y Seader (1984), como "continuación y homotopia diferencial de Longitud del arco". Afortunadamente, las ecuaciones diferenciales, nunca son rígidos y por lo tanto pueden resolverse fácilmente por métodos bien conocidos.

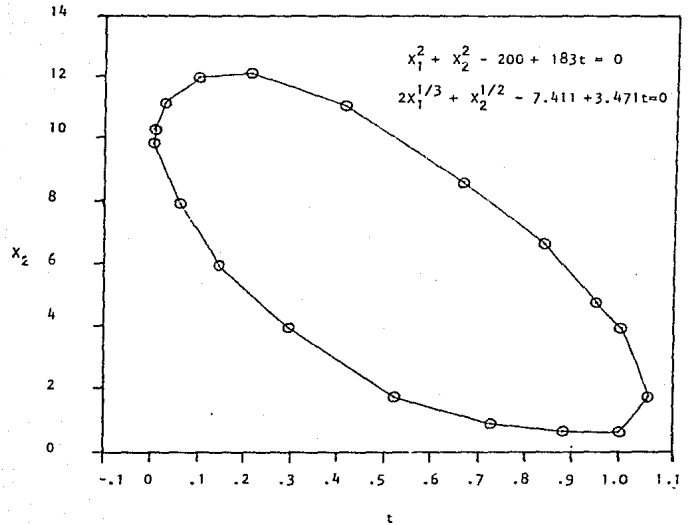


figura (28).- Ejemplo de ruta de Homotopia para ecuaciones no-lineales.

$f_i(x_1, x_2, \ldots, x_n) = 0$

$i = 1, 2, \ldots, n$

construir la homotopía con parámetro 't'

$h_i(x_1, x_2, \ldots, x_n, t) = 0 \quad i = 1, 2, \ldots, n$

donde $t^o \leq t \leq t_f$

con $x_i = x_i^o$ conociendo x_i^o a t^o

$h_i = f_i$ a t_f

convertir el problema de valor inicial en ecuaciones diferenciales ordinarias en términos de longitud de arco, haciendo $t = x_{n+1}$

$$\sum_{j=1}^{n+1} \frac{\partial h_i}{\partial x_j} \frac{dx_j}{d\rho} = 0$$

$$\sum_{j=1}^{n+1} \left(\frac{dx_j}{d\rho}\right)^2 = 1$$

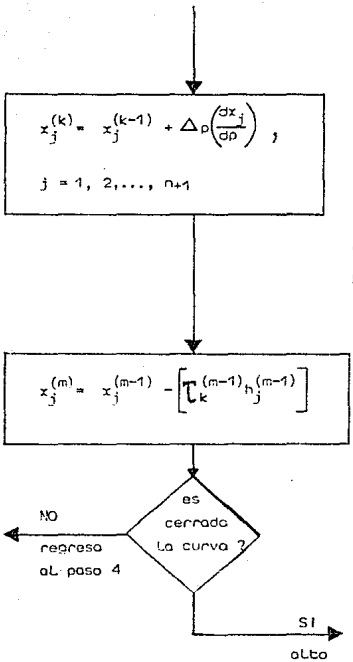
seleccionar la variable independiente 'k' para evitar la singularidad del jacobiano y combinar las ecuaciones para obtener las derivadas en forma explícita.

$$\left(\frac{dx_k}{dt}\right) = \pm \left[1 + \sum_{\substack{j=1 \\ j \neq k}}^{n+1} \beta_j^2\right]^{-\frac{1}{2}} \quad \text{con} \quad \beta_j = -\tau_k^{-1} \frac{\partial h_i}{\partial x_k}$$

τ_k^{-1} = Jacobiano de $\frac{\partial h_i}{\partial x_j}$ sin $\frac{\partial h_i}{\partial x_k}$

$$\frac{dx_i}{dt} = \beta \frac{dx_k}{dt}, \quad j = 1, 2, \ldots, k-1, k+1, \ldots, n+1$$

calcular Euler (ó Adams-Bashforth) para dar $\Delta\rho$.



resolver las ecuaciones de homotopia a x_k fija para x_j, $j=1,2,\ldots,k-1, k+1,\ldots,n+1$. Por el método de -- Newton para corregir o minimizar la truncación del error en el paso de integración.

TABLA 8

Resumen del método de Continuación y Homotopia.

La diferenciación de las ecuaciones b.161 y b.162 por la regla de la cadena (notese que x_1, x_2 y t, todas dependen de ρ) dá :

$$2x_1\frac{dx_1}{d\rho} + 2x_2\frac{dx_2}{d\rho} + 183\frac{dt}{d\rho} = 0 \qquad \text{b.163}$$

$$2/3x_1^{-2/3}\frac{dx_1}{d\rho} + \tfrac{1}{4}x_2^{-1/4}\frac{dx_2}{d\rho} + 3.4711\frac{dt}{d\rho} = 0 \qquad \text{b.164}$$

La forma multidimensional del teorema de Pitágoras se aplica, dando :

$$dx_1^2 + dx_2^2 + dt^2 = d\rho^2$$

$$\left(\frac{dx_1}{d\rho}\right)^2+\left(\frac{dx_2}{d\rho}\right)^2+\left(\frac{dt}{d\rho}\right)^2 = 1 \qquad \text{b.165}$$

Entonces, ahora se pueden resolver 3 ecuaciones diferenciables simultaneas ordinarias, lo cual constituye un problema de valor inicial, donde las condiciones iniciales son :

$$x_1 = x_1^o\,, \quad x_2 = x_2^o\,, \quad t = 0 \;\; o \;\; \rho = 0$$

Si un método explícito , tal como el Euler o una extensión del multipaso de orden superior el Adams - Boshforth, es usado para integrar las ecuaciones b.163, b.164 y b.165 que son manipulados por simple reducción, como sigue Para obtener las expresiones explícitas para las 3 derivadas con respecto a la longitud del arco . En forma de matriz, las ecuaciones b.163 y b.164 pueden escribirse como :

$$\begin{bmatrix} 2x_1 & 2x_2 \\ 2/3x_1^{-2/3} & \tfrac{1}{4}x_2^{-1/4} \end{bmatrix} \cdot \begin{bmatrix} \frac{dx_1}{d\rho} \\ \frac{dx_2}{d\rho} \end{bmatrix} = - \begin{bmatrix} 183\,\frac{dt}{d\rho} \\ 3.4711\,\frac{dt}{d\rho} \end{bmatrix} \qquad \text{b.166}$$

ó

$$\begin{bmatrix} \frac{dx_1}{d\rho} \\ \frac{dx_2}{d\rho} \end{bmatrix} = - \begin{bmatrix} 2x_1 & 2x_2 \\ 2/3x_1^{-2/3} & \tfrac{1}{4}x_2^{-1/4} \end{bmatrix}^{-1} \begin{bmatrix} 183 \\ 3.4711 \end{bmatrix} \frac{dt}{d\rho} \qquad \text{b.167}$$

La matriz inversa es el Jacobiano .Si éste es casi singular (tiene un - determinante casi cero) entonces dt/dp es casi cero ; por lo que se está cerca de un punto decisivo en la ruta , donde t puede cambiar de dirección ; este punto se observa en la figura (28) . Es muy importante que sea capaz para - seguir la ruta al-rededor del punto de decisión como se describe enseguida. Pero primero se resuelve la ecuacion b.166 para obtener :

$$\frac{dx_1}{dp} = -\frac{\left[3.4711(2x_1) - 183\ \ ^2/_3x_1^{-2/3}\right]}{\left[(\frac{1}{2}x_2^{-\frac{1}{2}})(2x_1)^{2/3}x_1^{-2/3}(2x_2)\right]}\frac{dt}{dp} = \beta_1\frac{dt}{dp} \qquad \text{b.167}$$

$$\frac{dx_2}{dp} = -\frac{\left[183(\frac{1}{2}x_2^{-\frac{1}{2}}) - 3.4711(2x_1)\right]}{\left[(\frac{1}{2}x_2^{-\frac{1}{2}})(2x_1) - (^2/_3x_1^{-2/3})(2x_2)\right]}\frac{dt}{dp} = \beta_2\frac{dt}{dp} \qquad \text{b.168}$$

Sustituyendo las ecuaciones b.167 y b.168 en la ecuación. b.165 se tiene:

$$\frac{dt}{dp} = \left[1 + \beta_1^2 + \beta_2^2\right]^{-\frac{1}{2}} \qquad \text{b,169}$$

Los valores de β dependen sólo de valores de x al punto previo en la ruta y (dt/dp) = 0.00002389 se calcula fácilmente de la ecuación b.169.

El signo de la ecuación b.169 determina la dirección inicial que se toma a lo largo de la ruta. Desafortunadamente, este valor de dt/dp es muy pequeño y es una consecuencia del jacobiano casi-singular. Por lo tanto los valores de β_1 y β_2 son calculados muy grandes en valores absolutos. Sobre estas condiciones, esto es mejor para seleccionar una variable diferente de t. Esta selección es hecha por Kubicek con la reducción de Gauss-Jordan con pivoteo- sobre la matriz jacobiana aumentada, la cual es :

$$\begin{bmatrix} 2x_1 & 2x_2 & 183 \\ ^2/_3x_1^{-2/3} & \frac{1}{2}x_2^{-\frac{1}{2}} & 3.4711 \end{bmatrix}$$

La columna que no suministra un pivote hace la variable independiente

En este caso, x_2 (antés que t) es seleccionado como el principio de la ruta y por lo tanto en lugar de resolver la ecuación anterior, se resuelve un cambio de esto:

$$\begin{bmatrix} \frac{dx_1}{d\rho} \\ \frac{dt}{d\rho} \end{bmatrix} = -\begin{bmatrix} 2x_1 & 183 \\ 2/3x_1^{-2/3} & 3.4711 \end{bmatrix}^{-1} \begin{bmatrix} 2x_2 \\ \frac{1}{3}x_2^{-\frac{1}{3}} \end{bmatrix} \frac{dx_2}{d\rho} \qquad b.170$$

El nuevo Jacobiano está lejos de ser singular y por lo tanto se obtiene :

$$\beta_1 = -0.93855 \qquad y \qquad \beta_t = -0.0034653, \quad \text{entonces:}$$

$$\frac{dx_2}{d\rho} = \frac{1}{\left[1 + \beta_1^2 \quad \beta_t^2\right]^{\frac{1}{2}}} = 0.72916 \qquad b.171$$

ahora, tomando el signo positivo de la ecuación b.169;

$$\frac{dx_1}{d\rho} = \beta_1 \frac{dx_2}{d\rho} = (-0.93855)(0.72916) = -0.68435 \qquad b.172$$

$$\frac{dt}{d\rho} = \beta_t \frac{dx_2}{d\rho} = (-0.0034653)(0.72916) = -0.0025268 \qquad b.173$$

Si se aplica el método simple de integración explícita de Euler, con un tamaño de paso para $\Delta\rho = 0.05$

$$x_1 = x_1^o + \Delta\rho\left(\frac{dx_1}{d\rho}\right)^o = 10 + (0.5)(-0.68435) = 9.65783 \qquad b.174$$

$$x_2 = x_2^o + \Delta\rho\left(\frac{dx_2}{d\rho}\right)^o = 10 + (0.5)(0.72916) = 10.36458 \qquad b.175$$

$$t = t^o + \Delta\rho\left(\frac{dt}{d\rho}\right)^o = 0 + (0.5)(-0.0025268) = -0.0012634 \qquad b.176$$

ahora, del error truncado en la integración numérica, los valores de x_1 y t son corregidos por el método de Newton para ecuaciones no-lineales en la forma de -- homotopía como las ecuaciones b.161 y b.162. Para mantener la variable independiente seleccionado x_2 a 10.36458 y usando x_1 a 9.65783 y t = -0.0012634 como -

los estimados iniciales; unicamente se requieren 2 ó 3 iteraciones de Newton, por que solo se necesito apoyar para cerrar la ruta de la homotopia. Cuando se alarga t = 1.0 donde la raíz deseada se encuentra, entonces se pueden usar un número suficiente de iteraciones de Newton para lograr la convergencia adecuada. Entonces el procedimiento alterna entre un Euler o un paso de integración de Adams, como los que predicen el camino y unas pocas iteraciones de Newton como las que corrigen. El resultado es un punto sobre o cercano a la curva de homotopia como se muestra en la figura

La eficiencia del método de Kubicek puede incrementarse por :

1.- Usando métodos de matrices dispersas con comportamiento de Jacobiano.

2.- Dando un algoritmo del tamaño de paso de integración como el discutido por Wayburn y Seader(1984)

Con el tiempo, se desarrollaran sistemas expertos para determinar automaticamente la naturaleza de las ecuaciones a resolverse y seleccionar el método más apropiado.. EL programa de Williams (1982) TKISOLVER es un programa desarrollado con este enfoque.

APLICACION :

En esta sección se comparan algunos de los metodos de convergencia comunmente utilizados y que se presentaron en el capítulo III, dichos metodos son: Método de Wegsteing, Secante generalizada, Newton-Raphson, Broyden, Broyden-Bennett, Broyden-Schubert.

Para la aplicación y comparación de estos metodos se utiliza el problema de Cavett (1963), el cual consiste de un sistema de destilación flash, compuesto por 4 (cuatro) flash isotérmicos; con 3 (tres) corrientes de recirculación y 2 (dos) lineas de mezclado. En el sistema se manejan 16 (dieciseis) componentes en la alimentación y se mencionan a continuación :

	COMPUESTO	FLUJO (lbmol/ hr)
1)	Nitrógeno (N_2)	358.2
2)	Dióxido de carbono (CO_2)	4,965.6
3)	Acido sulhídrico (H_2S)	339.4
4)	Metano (CH_4)	2,995.5
5)	Etano (C_2H_6)	2,395.5
6)	Propano (C_3H_8)	2,291.0
7)	Isobutano ($i\text{-}C_4H_{10}$)	604.0
8)	n-Butano ($n\text{-}C_4H_{10}$)	1,559.0
9)	Isopentano ($i\text{-}C_5H_{12}$)	790.4
10)	n-Pentano ($n\text{-}C_5H_{12}$)	1,129,9
11)	n-Hexano (C_6H_{14})	1,764.4
12)	n-Heptano (C_7H_{16})	2,606.7
13)	n-Octano (C_8H_{18})	1,844.5
14)	n-Nonano (C_9H_{20})	1,669.0
15)	n-Decano ($C_{10}H_{22}$)	831.7
16)	n-Undecano ($C_{11}H_{24}$)	1,214.5

Para la solución del problema se utilizó el simulador SGP/ZAR que emplea el enfoque modular secuencial.

El criterio de convergencia utilizado es el siguiente

$$\text{error} = \left| \frac{x^{k+1} - x^k}{x^{k+1}} \right|$$

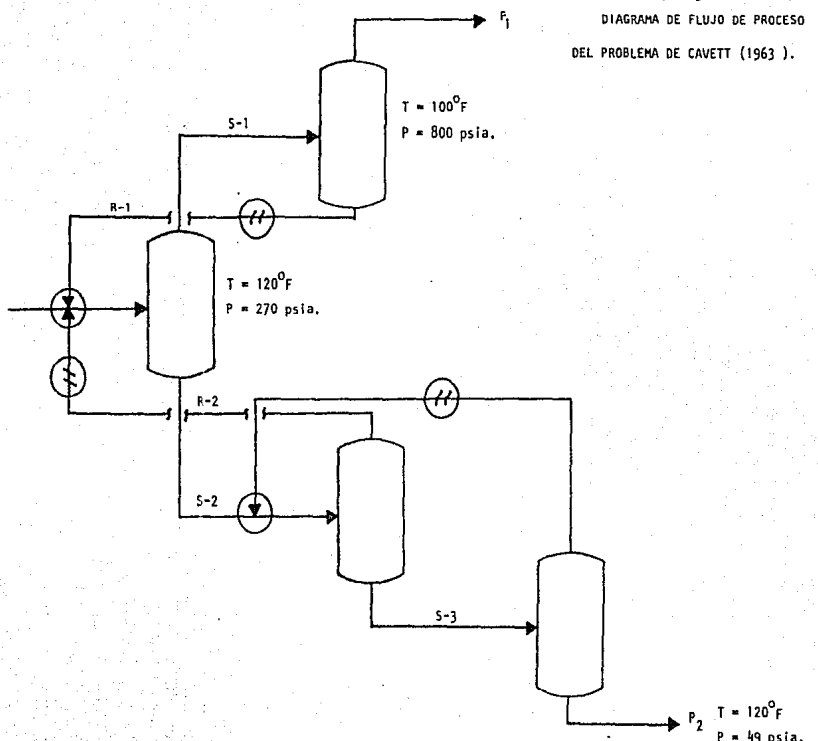
El cual es fijado anticipadamente con una tolerancia máxima de 0.01. Con la finalidad de comprobar la comparación teórica realizada en el Capítulo III se establece que el estimado inicial sea el mismo para todos los metodos.

Los diagramas de flujo de proceso y el diagrama de bloques para la simulación del proceso se presentan en las figuras (30 y 31) respectivamente.

FIGURA 30

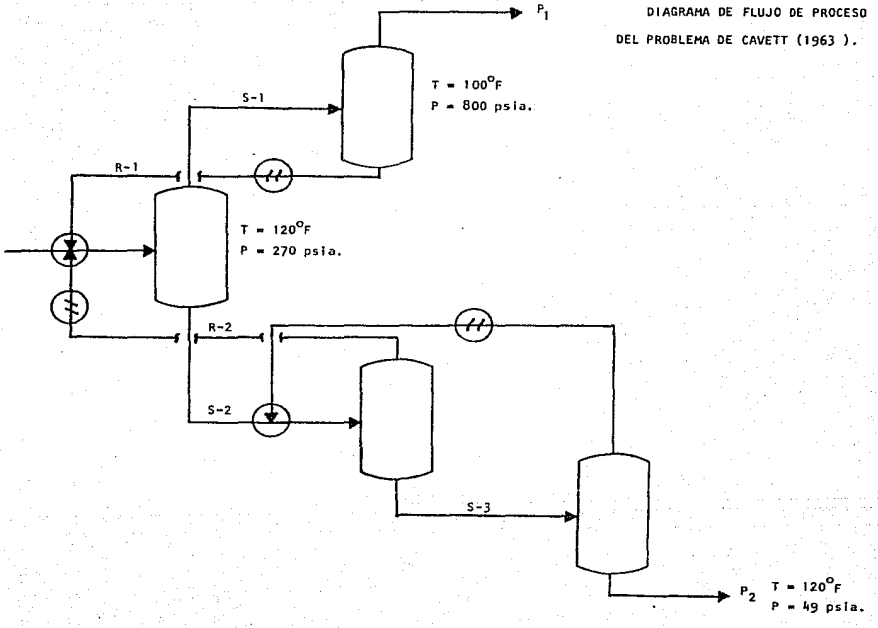
DIAGRAMA DE FLUJO DE PROCESO

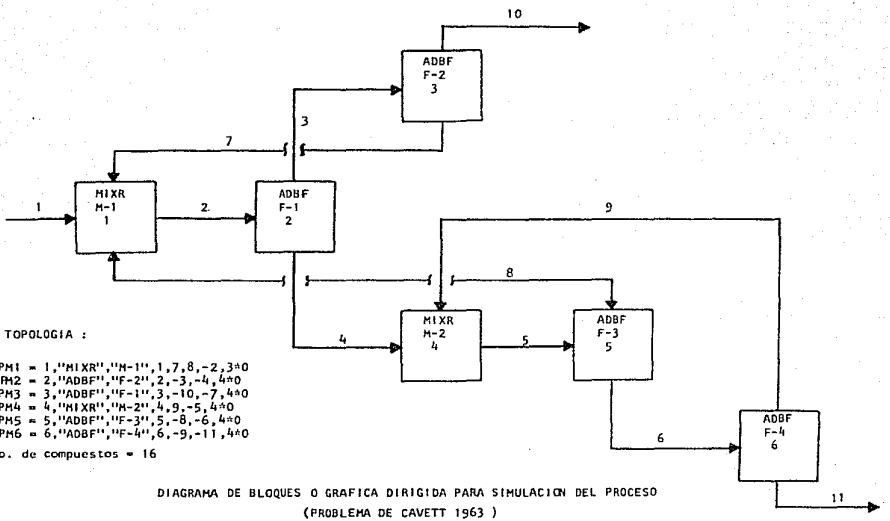
DEL PROBLEMA DE CAVETT (1963).



155

FIGURA 30
DIAGRAMA DE FLUJO DE PROCESO
DEL PROBLEMA DE CAVETT (1963).





TOPOLOGIA :

```
KPM1 = 1,"MIXR","M-1",1,7,8,-2,3*0
KPM2 = 2,"ADBF","F-2",2,-3,-4,4*0
KPM3 = 3,"ADBF","F-1",3,-10,-7,4*0
KPM4 = 4,"MIXR","M-2",4,9,-5,4*0
KPM5 = 5,"ADBF","F-3",5,-8,-6,4*0
KPM6 = 6,"ADBF","F-4",6,-9,-11,4*0
```

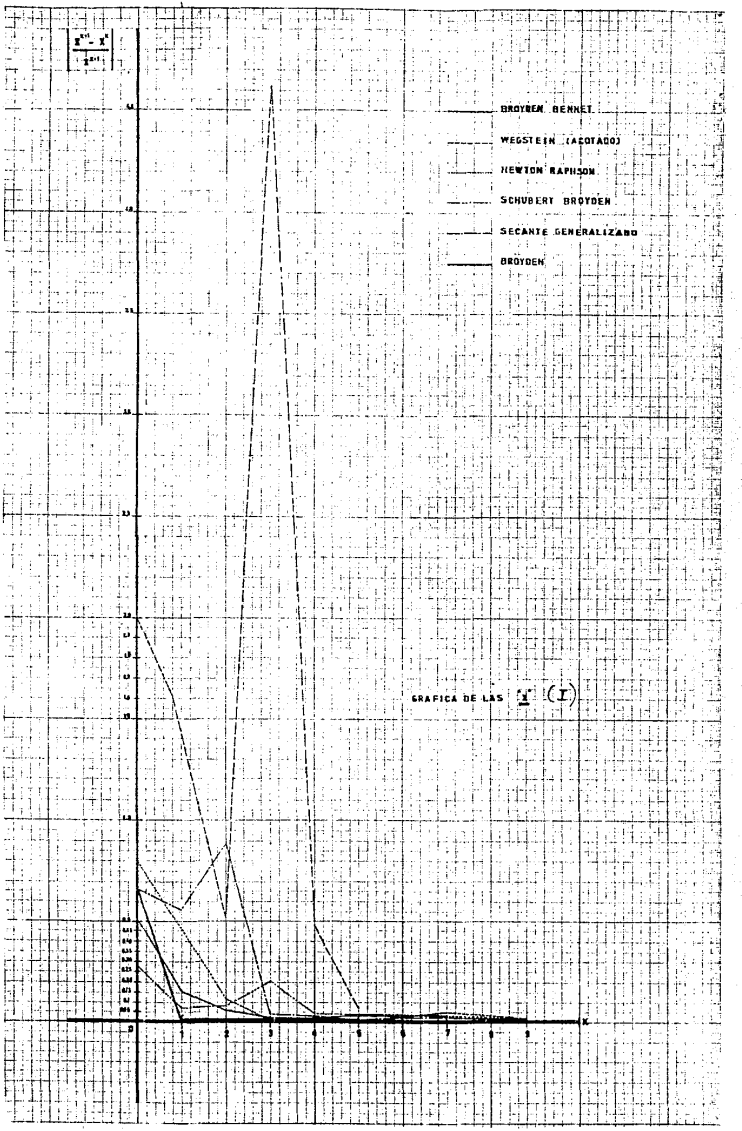
No. de compuestos = 16

DIAGRAMA DE BLOQUES O GRAFICA DIRIGIDA PARA SIMULACION DEL PROCESO
(PROBLEMA DE CAVETT 1963)
FIGURA 31.

TABLA V

RESULTADOS : Los resultados obtenidos de la simulación del problema de Cavett, se presentan en la siguiente tabla y se representan en la gráfica (I).

ITERACION	METODO DE WEGSTEING	SECANTE GENERALIZADA	NEWTON RAPHSON	BROYDEN	BROYDEN BENNETT	BROYDEN SCHUBERT
1	2.00	0.66	0.80	0.66	0.51	0.28
2	1.61	0.57	0.47	0.02	0.06	0.08
3	0.525	0.89	0.12	0.01	0.02	0.21
4	4.718	0.03	0.01	0.01	0.025	0.04
5	0.48	0.01	0.01		0.04	0.03
6	0.07	0.015			0.02	0.02
7	0.035	0.02			0.05	0.03
8		0.01			0.03	0.02
9					0.01	0.0


$\frac{\|x^{k+1} - x^k\|}{\|x^{k+1}\|}$
BROYDEN BENNET
WEGSTEIN (ACOTADO)
NEWTON RAPHSON
SCHUBERT BROYDEN
SECANTE GENERALIZADO
BROYDEN
GRAFICA DE LAS $\frac{\|x^{k+1} - x^k\|}{\|x^{k+1}\|}$ (I)
K

ANALISIS DE RESULTADOS :

En base a los resultados obtenidos podemos observar con claridad que el método que requiere de un menor número de iteraciones para lograr la convergencia es el Método de Broyden, ya que cumple el valor del error máximo permitido (0.01) a partir de la 3^a iteración. Esto puede deberse a la eliminación de la obtención de la inversa del Jacobiano y en consecuencia de la evaluación de derivadas parciales; ya que el Método de Broyden tiene convergencia superlineal en comparación con los otros metodos que son de convergencia cuadrática, tal como el Newton-Raphson que generalmente en problemas no muy complejos tiene una convergencia rápida; tal y como se aprecia en los resultados anteriores (logra la convergencia a partir de la cuarta iteración).

El método que no logra la convergencia hasta el límite de error máximo, es el Método de Wegsteing; esto es debido a que este método se utiliza para variables que no tienen una fuerte interacción entre las variables, dado que en el problema de Cavett las ecuaciones resultantes si presentan fuerte interacción, en este caso se presenta el problema de la divergencia.

Finalmente los metodos de mejora del Broyden requieren de un número mayor de iteraciones; lo cual es debido al mayor número de evaluaciones de funciones generadas por las matrices que se manejan en estos metodos.

CONCLUSIONES

En base al análisis realizado en este trabajo; para los diferentes enfoques de simulación de procesos, se puede concluir que en procesos cuyos circuitos de recirculación son independientes, el enfoque secuencial modular funciona satisfactoriamente, aplicando el método de convergencia más apropiado (por ejemplo el Broyden, Schubert, Newton, Wegsteing, etc..), - como se vera más adelante. Pero para sistemas complejos con circuitos de recirculación anidados, así como con reestructuraciones de diseño, la técnica Secuencial Modular presenta un pobre comportamiento en la convergencia por lo que es preferible utilizar el enfoque Modular simultaneo, ya - que este enfoque toma las ventajas del Secuencial Modular, principalmente de la Heuristica para suponer estimados iniciales, además de que el sistema de ecuaciones generado se resuelve globalmente y no por modulos independientes como en el enfoque Secuencial Modular, con lo que se evita tener que resolver muchos sistemas de ecuaciones independientes.

Por lo que respecta a los metodos de aceleración de la convergencia; - es conveniente aclarar que para un problema específico es necesario hacer un análisis para determinar cual es el método más conveniente, en base a las características de éste y a la naturaleza del problema a resolver; - de tal modo que se obtenga la solución con el menor número de iteraciones. Sin embargo, los siguientes criterios ayudaran en la mejor elección del - método de aceleración de convergencia; por ejemplo, para el método de Newton se tiene que una de las mayores ventajas, es que tiene una aplicabilidad muy amplia, pues sus condiciones de convergencia cuadrática son menos restrictivas que las de los metodos de convergencia lineal (Sustitución sucesiva, Acotado de Wegsteing, etc.). Por lo que, para concluir conviene señalar que cuando se cuenta con las expresiones analíticas de las ecuaciones y no resulta demasiado complicado el calcular el Jacobiano, el Método de Newton resulta sin duda, la mejor elección de todos los metodos disponobles. Como pudo constatarse al realizar el ejemplo de aplicación - con el problema de Cavett; en éste el segundo método que logró la convergencia en menos iteraciones fue el Método de Newton.

Sin embargo es necesario tambien señalar algunas desventajas del Método de Newton; una de ellas es que si el Jacobiano es singular en un punto de iteración dado, el método diverge; pero ese caso es poco frecuente en

la práctica. Otra desventaja que hay, es que hay que invertir la matriz del Jacobiano en cada iteración, lo cual puede requerir de un tiempo de computación relativamente grande. Sin embargo la mayor desventaja del método es que requiere la evaluación del Jacobiano; ya que en la práctica muchas veces no se cuenta con las expresiones analíticas o bien puede resultar tedioso el calcularlas. Para evitar tener que calcular el Jacobiano se analizarón los metodos llamados "Quasi-Newton", como el Broyden-Householder; Broyden-Bennett; Schubert; etc. que en principio, por sus condiciones de convergencia superlineal, se podría pensar que son menos eficientes que el Newton, lo cual es falso, ya que el Método de Broyden al no requerir de la evaluación del Jacobiano en cada iteración, como el Newton, permite lograr la convergencia en un menor número de iteraciones. Como se observa claramente en la tabla(V) de donde se deduce que el Método de Broyden es el más rápido de los metodos analizados. Asimismo, el Método de Schubert permite que los requerimientos de almacenamiento sean reducidos y el trabajo para mejorar la aproximación al Jacobiano, esta en función de la dispersidad del problema y de aquí que, para cada iteración se resuelve un problema lineal disperso. Tambien es importante destacar que otros metodos (que no fue posible analizarlos en el presente trabajo) tales como la Mejora de Soliman y la Relación entre los metodos Quasi-Newton y el Método de los valores propios dominantes; que al igual que los anteriores son de convergencia superlineal y por lo tanto tambien presentan desventajas; de las cuales se puede decir que la principal es ; que se requiere de un mayor número de evaluaciones de funciones. Finalmente, es necesario puntualizar que existen algunos metodos que, en teoría , son mejores que los anteriores, tales como el Híbrido de Powell, con una región expandida de la convergencia; el Algoritmo Conles y el Método de Continuación y Homotopia que son globalmente convergentes para cualquier estimado inicial.

En cuanto al análisis de los simuladores realizado en este trabajo para nivel académico e industrial, se puede concluir que el mejor simulador de procesos actualmente es el ASPEN ya que presenta muchas ventajas potenciales con respecto a los demás simuladores analizados en este trabajo; por ejemplo, la mayoría de los simuladores tiene un máximo de compuestos a manejar (Flowtran (35), Design 2000 (35), Process (50)); limitación que no tiene el ASPEN; ya que el puede manejar un número ilimitado de compuestos.

Finalmente para la solución de grandes conjuntos de ecuaciones lineales, que resultan de los diagramas de simulación de procesos y que se resuelven mediante la aplicación de matrices, se cuenta con tecnicas que -- tratan de reducir el tiempo de cálculo y el espacio de almacenamiento; -- dentro de estas tecnicas, se encontró que las que manejan la estructura - específica de los problemas en forma óptima y reducen potencialmente el - espacio requerido para almacenamiento, son las llamadas CBS y RANKI. Actualmente se estan haciendo mejoras a algunas de estas tecnicas, pero esto queda fuera del alcance de este trabajo.

A P E N D I C E A

I N T R O D U C C I O N.

La necesidad de resolver grandes conjuntos de ecuaciones lineales dispersas, surge en muchos campos del comportamiento de las ciencias aplicadas.

Los efectos combinados del rápido incremento, en la capacidad desarrollada en el área de las computadoras y el desarrollo de las técnicas de matrices dispersas, han proporcionado la capacidad para poder resolver distintos tipos de problemas en estos campos, cuya solución anteriormente resultaba inimaginable. En Ingeniería Química la necesidad de resolver grandes conjuntos de ecuaciones, surge como resultado de tratar de encontrar la solución de ecuaciones diferenciales y la de ecuaciones no-lineales, ésto ocurre principalmente en problemas de simulación de procesos.

La técnica de matrices dispersas proveen la capacidad de hacer de las técnicas de solución de problemas, una técnica de diferencias finitas.

Los métodos de matrices dispersas para la solución de sistemas de ecuaciones, surgen de ecuaciones diferenciales, si se explora la estructura y/o las propiedades numéricas del sistema. Con estas técnicas se logra una reducción en el tiempo de cómputo, comparado con el requerido en el empleo de las técnicas de matrices totales; además la red en el tiempo de cómputo es generalmente coincidente con la red en el almacenamiento (pocos elementos sobre los que se opera.).

En este capítulo se definen algunos métodos y se hace mención de sus ventajas y desventajas.

MATRICES

La teoría de la matriz dispersa es un cuerpo del conocimiento que permite resolver una gran variedad de problemas, las técnicas para su solución son métodos simples que exploran la estructura cero/no-cero de grandes conjuntos de ecuaciones lineales. La filosofía básica de la teoría de las matrices dispersas tiene dos formas :

1) Almacenar sólo datos diferentes de cero, esto es reducir los requerimientos de memoria.

2) Ejecutar operaciones sólo sobre operandos diferentes de cero, lo cual significa un ahorro en el tiempo de cálculo.

Además la reducción en el tiempo de cálculo generalmente coincide con la reducción en el almacenamiento, algunas técnicas reducen uno a expensas del otro.

La dispersidad de una matriz esta definida en términos de densidad, la relación cero/no-cero dentro de la matriz total, quiza un método más relevante esta en términos del número de nó-ceros promedio por ecuación.

En general las ecuaciones que resultan por la aplicación de diferencias finitas o elementos finitos, para diseño de procesos y problemas de simulación tiene relativamente un número fijo de no-ceros por ecuación, no importa cuan grande sea el sistema.

En otras palabras el número de no ceros es una función lineal de N (el número de ecuaciones) y no de N^2 (el número de entradas), para cada sistema teniendo de 1 a 30 ecuaciones, en promedio de 3 a 15 es común.

La técnica de una matriz dispersa es la explotación de tres simples características aritméticas:

i) $a + 0 = a$

ii) $a \cdot 0 = 0$

iii) $a \cdot 1 = a$

La característica iii) es usada en menor grado que i) y ii). En este capítulo se hace mención de algunos de los metodos más significativos para contar con herramientas que nos permitan de alguna forma el poder resol--

ver en general grandes sistemas de ecuaciones lineales.

I.1 ESTRUCTURA DE MATRICES

Es razonable asumir que las ecuaciones para cada módulo unidad serán generadas simultaneamente;esto es las ecuaciones y variables para un modulo en particular serán adyacentes mutuamente.De esto se puede obtener la estructura de la matriz (la matriz de ocurrencia) para tener agrupados o en bloques los elementos no-cero.Tambien ya que cada módulo unidad esta conectada solo en algunas otras (usualmente entre 1 y 4) se puede esperar que la matriz sea muy dispersa,con pequeños grupos relativamente densos - de elementos no-ceros.

Como un ejemplo considerar el siguiente diagrama de flujo:

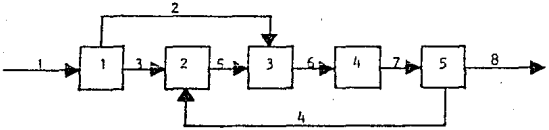


Fig. 19 Diagrama de flujo simple

en el cual hay cinco unidades módulo,cada uno conectado al siguiente excepto para una derivacion hacia adelante y una corriente de recirculación.

Para propositos demostrativos,si las ecuaciones son generadas unidad - por unidad y las variables de las corrientes son ordenadas tal como ellas ocurren en las corrientes de salida(las variables de la corriente 5 seran numeradas siguiendo a las variables de la corriente 8 y no a las variables de la corriente 4),la matriz resultante es la siguiente:

	1	2	3	4	6	7	8	5
1	X	X	X					
2		X	X	X				X
3		X		X	X			
4					X	X		
5						X	X	X
							X	X

M-1

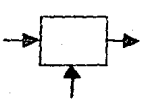
Las X's representan bloques de no-ceros de ecuaciones y variables. Los números horizontales son números de corrientes y los verticales son números de unidades. Las unidades pueden consistir de un conjunto de ecuaciones para cada corriente de salida. Por eso las unidades 1 y 5 tienen cada una dos conjuntos de ecuaciones, la matriz no es aún cuadrada y no es posible resolverla así, - si se considera el problema de simulación (como opuesto al problema de diseño) por especificación de las variables de la corriente 1, el subsistema resulta :

	1	2	3	4	6	7	8	5	
S	X								
1	X	X	X						
2		X	X						
3			X	X			X		N-2
4		X		X	X				
5				X	X				
				X	X	X			

Ahora definiendo las corrientes de salida de una unidad particular como "perteneciendo" a esa unidad, conseguiremos que surga el cuadrado entre las ecuaciones de unidades y bloques de variables. Ordenadas en esta forma las X's bajo los bloques de la diagonal cuadrada representan información de alimentación adelante (anterior) y las X's sobre los bloques de la diagonal representan información de recirculación.

La estructura de los bloques es independiente sobre que tipo de unidad - de operación esta representada.

A continuación se consideran 6 (seis) diferentes tipos de unidades, en esta se consideran como variables de corrientes ; flujo molar de cada componente, entalpia y presión, se pueden utilizar otras variables como : fracciones mol, y flujo total en vez del flujo molar, temperatura en vez de entalpia.



	m	m	m	H	P	m	m	m	H	P	m	m	m	H	P
1	X					X					X				
2		X					X					X			
3			X					X					X		
4				X					X					X	
5					X					X					X

Fig. 20 Estructura de un mezclador

	m	m	m	P	H	m	m	m	P	H	m	m	m	P	H	P	T	Q
1	X					X					X							
2		X					X					X						
3			X					X					X					
4					X					X					X			X
5									X					X				
6									X							X		
7						X	X	X			X	X	X			X	X	
8						X	X	X			X	X	X			X	X	
9						X	X	X			X	X	X			X	X	
10						X	X	X		X							X	
11											X	X	X				X	

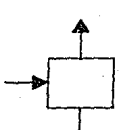


Fig. 21 Estructura de una unidad de flash.

	m	m	m	P	H	m	m	m	P	H	m	m	m	P	H	S
1	X					X										X
2		X					X									X
3			X					X								X
4				X	X				X							X
5										X						X
6	X										X					X
7		X										X				X
8			X										X			X
9				X										X		X
10					X										X	X

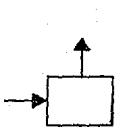


Fig. 22 Estructura de un separador.

	m	m	m	P	H	m	m	m	P	H	Q
1	X					X					
2		X					X				
3			X					X			
4				X					X		
5					X					X	X

Fig. 23 Estructura de un cambiador de calor.

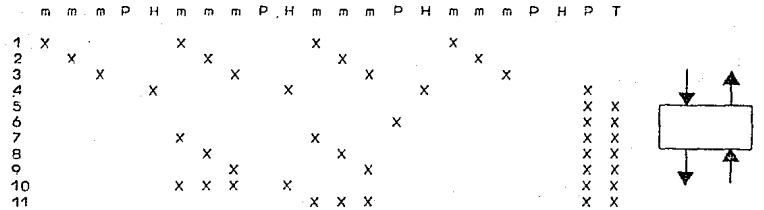


Fig. 24 Estructura de una cascada ideal.

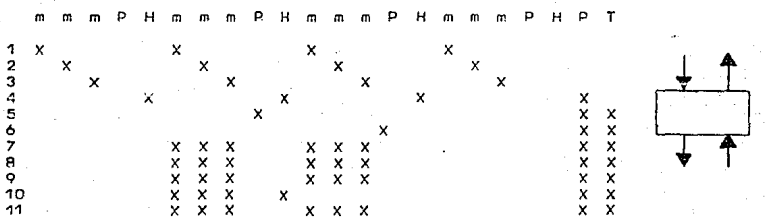


Fig. 25 Estructura de una cascada no ideal.

Cada una de estas unidades contiene de 2 a 8 bloques no-cero. El tamaño de los bloques son generalmente de tamaño 2 + número de componentes, 2 + nc; nc relación de flujo molar, m;presión, p; y entalpia H.

La excepción ocurre en las unidades que contienen variables internas. Por ejemplo en un cambiador de calor,la carga total es una variable interna, el cambiador de calor entonces contfenedos bloques; uno con nc + 2 y otro con - nc + 3 variables.

En la figura 26 se muestran algunas estructuras de matrices dispersas - que se obtienen con métodos directos, principalmente basados sobre la eliminación Gaussiana.

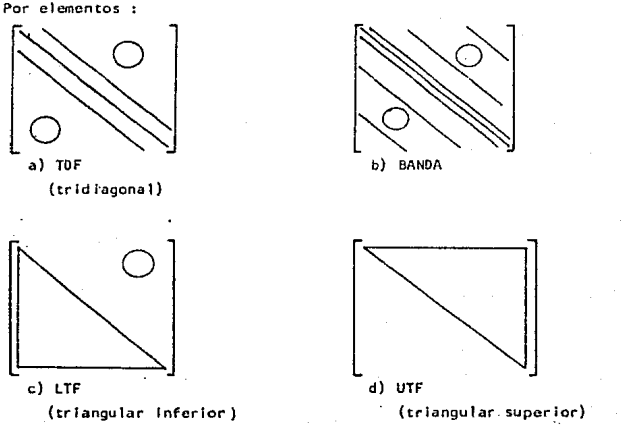
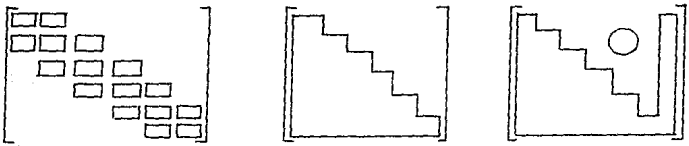


FIG. 26

ESTRUCTURAS DE MATRICES DISPERSAS



e) BTDF (bloque triangular) f) BLTF (bloque triangular inferior) g) BBLTF (bloque triangular inferior bordeada)

FIGURA 27). Algunas estructuras organizadas de matrices dispersas.

TECNICAS DE ALMACENAMIENTO

Existen muchas tecnicas de almacenamiento disponibles ; algunos exploran la dispersidad general mientras exploran especificamente la estructura de los no-ceros exhibidos por el conjunto de ecuaciones.

Las ecuaciones que resultan de los problemas de simulación de procesos - se prestan mejor a las técnicas de almacenamiento de matrices dispersas.

Aunque la estructura general de tales ecuaciones es muy similar de problema a problema, esto no es predecible para obtener algún beneficio.

Dos técnicas usadas extensamente para almacenar tales sistemas son :

Un esquema utiliza una lista de estructura eslabonada. Este método requiere almacenar dos vectores : una fila índice, RLI (I), de longitud N y un vector LINK (L), de longitud 3 π, donde π es el número de no-ceros.

Dando el número de fila I, al índice de localización, regresa al sitio en LINK, del primer no-cero en la fila. Cada no-cero en LINK requiere tres sitios de almacenamiento ; son número de columna, su valor y el sitio en LINK del siguiente valor no-cero en esa fila o un "0" si es el elemento último en la fila. Considerando la siguiente matriz :

-1	0	2		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
-3	1	0	RLI	1	7	13												
0	-4	0	LINK	1	-1	4	3	2	0	1	-3	10	2	1	0	2	-4	0

M-3 Lista de la estructura eslabonada para almacenamiento.

Este esquema es particularmente usado si durante los cálculos un cero se hace no-cero. Por ejemplo después del primer paso de eliminación Gaussiana el elemento a_{23} se convierte en (-6) . El nuevo valor de a_{23} es almacenado al final de la lista y eslabonado al final de la fila 2 LINK(12) es fijado a 16.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	-1	4	3	2	0	1	-3	10	2	-1	16	-2	-4	0	-3	-6	0

M-4

De esta manera creando nuevamente no-ceros pueden ser eslabonados en algún orden conveniente a esa fila y almacenado al final de la lista.

Otra aproximación involucra (n+1) + 2τ sitios de almacenamiento. Este esquema es usualmente implementado en tres vectores, una fila de índices de localización, RLI (I), de longitud N + 1 ; una columna índice CI (L), de longitud τ ; y un vector de elementos A (L), de longitud τ. Similarmente a la aproximación anterior dando el número de fila I, RLI (I) regresa a la colocación en CI y A del primer elemento no-cero en la fila. Existe una correspondencia uno a uno entre CI y A en la que CI contiene el número de la columna no-cero A(L) ya que las filas son almacenadas, consecutivamente la localización del último no-cero en la fila I es uno menos que la localización del primer no-cero de la siguiente fila, RLI(I +1) - 1 . Este método se demuestra con el ejemplo anterior.

	1	2	3	4	5
RLI	1	3	5	6	
CI	1	3	1	2	2
A	-1	2	-3	1	-4

M-5

- Un ejemplo de una técnica de almacenamiento 2τ

La estructura presentada por Gustavson (1973) en forma de fila de una matriz A esta dada por tres arreglos unidimensionales. Dos arreglos enlistan los índices de columna (JA) y los valores numéricos (A) de las entradas NA de los elementos diferentes de cero ya que integran la matriz. El orden

de estos elementos es en forma de fila similar a los mecionados anteriormente, el tercer arreglo es un conjunto de filas (dirigidas) indices (IA) donde el iésimo elemento de la JA es el indice en ambos JA y A del primer elemento no-cero de la iésima fila de A, por ejemplo :

7	0	-3	0	-1	0
2	8	0	0	0	0
0	0	1	0	0	0
-3	0	0	5	0	0
0	-1	0	0	4	0
0	0	0	-2	0	6

M-6

Una representación en forma de una fila dispersa de "A" está dada por :

IA	1			4		6	7		9		11		13	
	↓			↓		↓	↓		↓		↓		↓	
JA	3	5	1	2	1	3	1	4	2	5	6	4		M-7
A	-3	-1	7	8	2	1	-3	5	-1	4	6	-2		

La cuarta fila de A inicia en la posición IA(4)= 7 y termina en la posición IA (4 + 1) - 1 = 8 del arreglo JA y A .Así la cuarta fila de A ,tiene dos no-ceros en las posiciones (4,1) y (4,4) cuyos valores son (-3) y (5) -- respectivamente. Notesé que las columnas indices en JA estan desordenadas -- dentro de una fila dada. Una representación dispersa en forma de fila de "A" es ordenada si para cada fila i $1 \leq i \leq r$ se tiene que $1 \leq j_1 < j_2 < \ldots j_r \leq S$ -- para las columnas índice JV de los no-ceros r_i de la fila i, por ejemplo la matriz A no esta ordenada porque las columnas índice en las filas 1,2 y 6 no estan en orden ascendente.

Una representación en forma de fila dispersa es muy pobre cuando uno desea información de una columna específica.

Gustavson representa dos algoritmos que usan esta estructura de datos - el primer algoritmo calcula PAQ^{-1}, donde P y Q son matrices de permutación y A es dispersa .El segundo algoritmo calcula el producto C de las matrices -- dispersas A y B , estos algoritmos son óptimos en tiempo de ejecución y almacenamiento.

El algorítmo para calcular PAQ^{-1}, es obtenido por aplicación de una rutina llamada HALFPERM, éste es un algoritmo de una matriz transpuesta dispersa que ha sido modificada para calcular $(PA)^t$ a partir de A^t, para mayores detalles ver Gustavson (1973,1978).

Un algoritmo transpuesto :

Una representación en fila de A es igual a una representación en columna

A^t la matriz $(PA)^t$ sería representada por IAT, JAT y AT, el costo de encontrar una representación en forma de columna, es económico cuando se utiliza el vector de permutación P, tambien IAT, JAT y AT serán una representación ordenada en forma de fila dispersa de $(PA)^t$, esta característica ilumina el estilo natural del algorítmo transpuesto por ejemplo: Fijando P=I y aplicando el algorítmo a la matriz anterior (M6) tenemos el arreglo :

IAT	1			4		6		8		10		12	13
	↓			↓		↓		↓		↓		↓	↓
JAT	1	2	4	2	5	1	3	4	6	1	5	6	
AT	7	2	-3	8	-1	-3	1	5	-2	-1	4	6	

M-8

SISTEMAS LINEALES COMPUESTOS

Considerando la forma estandar para representar algún conjunto de ecuaciones lineales :

$$A\,x = b$$

donde A es una matriz de coeficientes(n x n) y (x) y (b) son vectores columna de longitud (n). Correspondientemente la solución "x" esta representada por :

$$x = A^{-1}\,b \qquad (1)$$

La forma explícita de la inversa de A, A^{-1} para matrices completas es generalmente calculada usando eliminación Gaussiana o Gauss Jordan, sobre la matriz aumentada $[A|I]$ para reducir esto a $[I\,|\,A^{-1}]$.

Sin embargo normalmente es calculada explícitamente sólo cuando soluciones repetidas para varios vectores(b) son deseados y cuando A es suficientemente pequeña. Si éste no es el caso , la eliminación Gaussiana ó algun método tal como el de Crout, es normalmente empleado.

La fase de la eliminación Gaussiana resulta en :

$$U\,x = b$$

donde U es triangular superior de n x n, la solución es entonces calculada por sustitución inversa.

Para grandes sistemas se requiere generalmente pivoteo para mantener la estabilidad numérica através de los cálculos. El pivoteo total involucra la selección del mayor elemento en la matriz activa o resultante como pivote. Los algoritmos de matrices pueden emplear pivoteo parcial, que consiste como el elemento pivote al elemento en la fila pivote (o columna) -- con el mayor valor absoluto.

En la implementación de todas las técnicas de matrices totales se requieren (N^2) localidades de almacenamiento y en adición 2N localidades para las matrices P y Q de permutación si se emplea el pivoteo total. Los 2N sitios se reducen a N en la aplicación del pivoteo parcial. Si se emplea eliminación Gaussiana, los requerimientos de cálculo son del orden de (N^3)/3 cálculos (multiplicaciones y divisiones para grandes sistemas).

MULTIPLICACION DE MATRICES

UN ALGORITMO PARA MULTIPLICACION DE MATRICES DISPERSAS

Construcción de algún algoritmo con tiempo de corrida proporcional al número de multiplicaciones. Asumiendo que A y B son matrices P X Q y Q X R, dandolas en forma de fila, la tabla V indica estos arreglos, la expresión dentro del paréntesis de los arreglos es el tamaño de arreglo, por ejemplo :

A usa 2NA + P + 1, celdas de memoria para su descripción donde NA es el número de elementos $\neq 0$ en A. La matriz resultante C= AB esta descrita por IC (P+1), JC (NC) y C (NC). El objetivo es determinar "C" en O(H) operaciones donde M, $0 \leq M \leq PQR$, es el número no trivial de multiplicaciones para formar "C".

Nombre de la matriz	Nombre del arreglo fila indicadora	Nombre del arreglo columna índice	Nombre del arreglo valor numérico
A	IA (p+1)	JA (NA)	A (NA)
B	IB (q+1)	IB (NB)	B (NB)

Tabla V . Matrices A y B en formato de forma, de fila dispersa.

a_{ij} \ b_{ij}	0	1
0	0	0
1	0	1

Tabla VI . Cuatro tipos de multiplicación.

Por la definición de multiplicación de matrices tenemos.

$$C_{ij} = \sum_{v=1}^{q} a_{iv} b_{vj} \quad \text{para} \quad 1 \leqslant i \leqslant p \quad y \quad 1 \leqslant j \leqslant r \qquad \text{A-1}$$

Existen (pqr) posibles multiplicaciones, la general es $a_{iv}b_{vj}$, existen - cuatro clases de partición de multiplicaciones dependiendo de los valores de los operandos, estas cuatro clases se muestran en la tabla VI. Si A y B son dispersas muchos de los operandos estaran bajo la clase 0,0 , las siguientes clases más populares son las de (0,1) y (1,0), la última clase (1,1) es la - que tiene mayor interés. Un algoritmo previo de MC Namec (1971) para multi-- plicacion de matrices dispersas fue basado sobre las ecuaciones A-1 , ya que B será almacenada en forma de fila, un algorítmo transpuesto será utilizado al inicio para tener una representación de columna.El costo principal - de Mc Nnames (1971) esta en la unión de la la fila (i) de A con la fila (j) de B^t, estas operaciones bajo las clases 0,1 y 1,0 tabla VI, puede ser - de un orden de magnitud mayor que las operaciones en la clase (1,1) ,la --- ecuación (A.1) puede ser escrita como :

$$C_j = \sum_{a_{iv} \neq 0} a_{ij} b_v \qquad \text{para} \quad 1 \leqslant i \leqslant p \qquad \text{A.2}$$

Esta ecuación sólo esta referida a las filas, es decir que las filas - ith de C son una combinación lineal de estas v filas de B para la cual $a_{iv} \neq 0$ Además si A y B son almacenadas en formato de forma de fila, entonces sólo los elementos de la clase (1,1) ocurren en la ecuación A.2

El algoritmo propuesto por Gustavson se basa en la ecuación A.2 y tiene la propiedad de que la cantidad de operaciones puede ser proporcional a H.

UNA FORMA CANONICA PARA MULTIPLICACION DE MATRICES DISPERSAS :

En esta forma canónica se demuestra la naturaleza de las operaciones "tipo-cero" que requiere la multiplicación de matrices dispersas orientadas en fila.

Suponiendo que A tiene $p_3 \leq p$ filas vacias y $q_3 \leq q$ columnas vacias y que B tiene $r_3 \leq r$ columnas vacias y $q_2 \leq q$ filas vacias que no pertenecen a las columnas asociadas a q_3 de A, dejando que q_i sean las columnas restantes de A que serán tambien las filas restantes de B, i,e., $q = q_1+q_2+q_3$, haciendo p_2 la fila de A que son cero evaluadas en las columnas q_1 y similarmente dejando r_1 que sean las columnas de B que son cero evaluadas en las filas asociadas con q_1 P,Q y R serán las matrices de permutación que respectivamente ordenen las filas de A en tres grupos : p_1, p_2, y p_3, las columnas de A y las filas de B en tres grupos q_1,q_2 y q_3, y las columnas de B en tres grupos r_1,r_2 y r_3. Haciendo $A=PAQ^t$,$B= QBR^t$ y $C=PCR^t$, a esta forma se le llama la forma canónica para multiplicación de matrices dispersas A, B y C (fig.M.9).Los bloques x_1,x_2,x_3 y x_4 de las matrices " no son de cuidado " sus argumentos acompañantes durante la multiplicación de matrices son siempre matrices cero y ello produce matrices cero como producto.

La matriz $A(p_1 \times q_1)$ y la matriz $B(q_1 \times r_1)$ son matrices dispersas con la propiedad de que cada fila y columna no esta vacía. Su producto es una matriz dispersa $p_1 \times r_1$ con la misma propiedad se cancelaran si se disgregan accidentalmente. Note tambien que alguno de los p_i, q_i y r_i, con $i= 1,2,3$, pueden ser igual a cero.

Usando la fig. M.9 y la ecuación A.2 se pueden encontrar las operaciones del algoritmo de matrices dispersas. Primero se hacen pruebas a p_3 para descubrir que filas p_3 de A estan vacias, se examina cada elemento diferente de cero, de los bloques de matrices x_1 y x_2 hasta el total de elementos $nx_1 + nx_2$. Para cada no cero tal (x_{ij}) se encuentra la columna indice (j) perteneciente al conjunto q_2, no existen multiplicaciones porque alguna fila q_2 de B es una fila cero.

Finalmente si $p \neq 0$, $q_1 \neq 0$ y $r_1 = 0$ esto implica que $A=0$ y $B_1=0$, entonces se examinan todos los NA, no ceros de A con multiplicaciones no existentes en realidad $NI = 0$, en este caso, cuando ambos A_1 y B_1, son ma

trices no-cero, cada elemento a_{ij} de A produce al menos una multiplicación ya que cada fila (j) de B es $\neq 0$; es decir, la característica de que A tiene no-ceros junto con B, que multiplicando A x B es O(M).

Una rutina de multiplicación de matrices de columna orientada :

$$C.j = \sum_{b_{vj}\neq 0} a_{.v}\, b_{vj} \qquad \text{para } 1 \leq j \leq r \qquad A.3$$

En términos de la fig. 3 y la ec. 3 nosotros tenemos una situación análoga observando las operaciones sobrantes. Hay $r_3 + r_3 + r_4$ operaciones sobrantes.

En adición si $A_1 = 0$ y $B \neq 0$, entonces se examinan todas MB, elementos de B con multiplicaciones no resultantes .

$$\begin{array}{c} \\ P_1 \\ P_2 \\ P_3 \end{array}\!\!\begin{array}{c} \begin{array}{ccc} q_1 & q_2 & q_3 \end{array} \\ \begin{bmatrix} A_1 & X_1 & 0 \\ 0 & X_2 & 0 \\ 0 & 0 & 0 \end{bmatrix} \end{array} \times \begin{array}{c} \begin{array}{ccc} r_1 & r_2 & r_3 \end{array} \\ \begin{bmatrix} B_1 & 0 & 0 \\ 0 & 0 & 0 \\ X_3 & X_4 & 0 \end{bmatrix} \end{array}\!\!\begin{array}{c} \\ q_1 \\ q_2 \\ q_3 \end{array} = \begin{array}{c} \\ P_1 \\ P_2 \\ P_3 \end{array}\!\!\begin{array}{c} \begin{array}{ccc} r_1 & r_2 & r_3 \end{array} \\ \begin{bmatrix} A_1B_1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \end{array}$$

$\overline{A}$ $\overline{B}$ $\overline{C}$

M-9 Matrices $\overline{A}$, $\overline{B}$ y $\overline{C}$

FORMA DE BLOQUE TRIANGULAR INFERIOR DE UNA MATRIZ

El problema de encontrar la forma de bloque triangular inferior de una matriz dispersa. El problema consiste en arreglar una matriz M, así que la forma de partición es triangular inferior.

En el diagrama siguiente se muestra un ejemplo donde la partición se hace en tres bloques

$$\begin{bmatrix} A & 0 & 0 \\ X & B & 0 \\ Y & Z & C \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix}$$

M-10

Suponiendo que se requiere resolver Mu=V donde M es n x n y $n=n_1+n_2+n_3$. Si A.1 representa la forma de bloque triangular inferior de M entonces se puede considerar resolver el sistema simple:

$$A\,x = \alpha$$

$$B\,y = \beta - X\,z$$

$$C\,z = \gamma - Y\,x - Z\,y$$

$$A = \cdot$$

Uno puede encontrar la forma de bloque triangular inferior en dos -- fases :

El primer paso es encontrar una asignación para la matriz dispersa. Una vez que la asignación ha sido encontrada, la matriz puede ser considerada una matriz de gráfica directa.

El segundo proceso es encontrar los componentes firmes de la gráfica directa asociada. Los componentes firmes y sus conexiones son isomorficos a la forma de bloque triangular inferior, como ejemplo considere la siguiente matriz de 6 x 6 :

$$A = \begin{bmatrix} \otimes & \times & \cdot \\ & \otimes & \\ \times & & \otimes \\ & \times & & \otimes \\ \times & \otimes & \\ & \times & & & \otimes \end{bmatrix} \qquad \text{M-11}$$

Los elementos encerrados en circulos constituyen una asignación, ésta puede ser descrita por la permutación Q= 152364 que define un arreglo de - las filas de A, así que la diagonal no tiene ceros. Note que si una asignación no existe para A entonces A es singular, ya que todos los productos - n! de la expansión del determinante de A son cero y esto es det A = 0

Haciendo $B = QA$

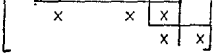
$$B = \begin{bmatrix} \times & & \times & & \times \\ \times & \times & & & \\ & & \times & & \\ \times & & & \times & \\ & \times & & & \times \\ & & & \times & & \times \end{bmatrix} \qquad \text{M-12}$$

El algoritmo de componentes firmes de Tarjan (1972) es O(n,N) donde - N es el número de no-ceros de A , (n) es el orden de A y A es alguna matriz dispersa con asignación (se ha notado que la forma de bloque triangular inferior es independiente de la asignación usada para generar la grá

fica directa asociada (ver Duff (1977), George y Gustavson (1980)). La -- gráfica asociada con A tiene (n) nodos y N flechas, hay una flecha del nodo (i) al nodo (j) si y sólo si a_{ij} de A es $\neq 0$. Un componente firme de - una gráfica es un conjunto máximo de nodos ,tales que para cada dos nodos i,j en el conjunto haya una ruta en la gráfica de i a j y de j a i .

Para la matriz esto es equivalente a encontrar una permutación P tal que PAP^T sea en forma de bloque triangular inferior. El algorítmo de Tarjan aplicado a B produce la permutación P = 325146. Ahora $N = PBP^T$ es la siguiente matriz :

M =



M-13

De este modo la matriz A tiene cuatro bloques diagonales de tamaño - 1,3,1y 1 .

Existen muchos algorítmos para encontrar una asignación,el mejor según la complejidad es el algorítmo ó $(n^{1/2}, N)$ de Hopcroft y Karp(1973) - que encuentra una máxima combinación sobre una gráfica bipartita.

Gustavson (1976) tiene un gran algorítmo de asignación llamado "Asignación de fila" (ASSIGN ROW) , este algorítmo y el de Tarjan son ejemplos del DFS ó más comunmente trayectoria inversa. Este procedimiento es un - método particular para la exploración de flechas y esta basado sobre la -- siguiente regla de elección : cuando seleccionamos una flecha a recorrer, siempre eligiendo una arista procediendo del vértice más recientemente alcanzado que tiene vertices fijos no explorados, el conjunto de vertices antiguos con posibles vértices no explorados, puede ser almacenado sobre una fila. Así el DFS es muy fácil de programar.

El algorítmo ASSIGN ROW llama dos subrutinas ; CHEAP ASSIGN y DFS.

Cinco características del algorítmo son listadas (Gustavson (1976))

1) El uso de DFS para extender la longitud por uno.
2) El uso de CHEAP ASSIGN.
3) El uso del arreglo indicador de fila auxiliar para limitar el -- costo total de todas las ejecuciones de CHEAP ASSIGN a muchas -- comparaciones NT.

4) El uso de la columna de arreglo , es utilizada para reducir -- sustancialmente el costo de DFS.
5) El uso del arreglo Boleano fila dada para un circuito corto a-- bortivo probado en la DFS.

Otro algorítmo de rastreo inverso es el algorítmo BLTF para la forma de bloque triangular inferior, este es una mejora del algorítmo STCO. Gustavson (1976), el algorítmo MC13D (DUFF y REID (1978)) y Strong Connect, - Tarjan (1972)).

El algorítmo BLTF y el MC13D estan estrechamente relacionados, ya que ambos son una mejora de STRONG CONNECT. Sin embargo el algorítmo BLTF es - superior a MC13D en los siguientes términos:

1) Ejecución rápida.
2) Utiliza menor requerimiento de almacenamiento.
3) Calcula exactamente las mismas salidas como STRONG CONNECT.
4) De este modo la translación hacia el lenguaje FORTRAN esta completamente estructurada.
5) El loop interior es más eficiente.

El algoritmo STCO de Gustavson (1976) es tambien una traslación estructural del algorítmo de Tarjan.

Mas adelante se describen algunos métodos desarrollados por Wood(1979) en la tabla (VII) se muestran algunos métodos directos para matrices dispersas.

EL CASO SIMETRICO

En este punto el objetivo es almacenar solamente la matriz triangular superior y hacer la mitad de las operaciones requeridas por el caso general, el tomar la ventaja de la simetría tanto en almacenaje como en operaciones estimadas ya que $(A = LU = U^t DU)$, advirtiendose que las columnas de acceso a los datos de filas orientadas serán necesarios.

Para matrices generales se inicializa $X = i^{th}$ fila de sobre y a lo -- largo de la diagonal, la operación general es :

$$X-X - l_{iv}(\text{fila V de U}) \quad l_{iv} \neq 0 \qquad A\text{-}5$$

METODOS DIRECTOS PARA MATRICES DISPERSAS

TABLA V II

CODIGO	REFERENCIA
1.- TRGB	BENDIG Y HUTCHISON- 1973
2.- SLMATH	GUSTAVSON- 1977
3.- MA28A Y MA28AD	RUFF Y REID- 1977
4.- NSPIV	SHERMAN- 1978
5.- Y12M- SSLEST ES SU ANTERIOR	ZLATEV, WASNIEUSKI Y SCHANBURG- 1979
6.- CLARK	CLARK- 1980 BAJO LA DIRECCION DE WESTERBERG
7.- SPAR2PAS	STADTHEIR Y WOOD- 1983

METODOS DIRECTOS PARA MATRICES DISPERSAS

TABLA V II

CODIGO	REFERENCIA
1.- TRGB	BENDIG Y HUTCHISON- 1973
2.- SLMATH	GUSTAVSON- 1977
3.- MA28A Y MA28AD	RUFF Y REID- 1977
4.- NSPIV	SHERMAN- 1978
5.- Y12M- SSLEST ES SU ANTERIOR	ZLATEV, WASNIEWSKI Y SCHANBURG- 1979
6.- CLARK	CLARK- 1980 BAJO LA DIRECCION DE WESTERBERG
7.- SPAR2PAS	STADTHEIR Y WOOD- 1983

Ya que (fila ν de U)= (col' ν de L)β_ν donde $\beta_\nu = D^{-1}{}_\nu u$ y $U_{\nu i} = l_{i\nu}\beta_\nu$ se puede remplazar A.5 con

X----X----$U_{\nu i}$(fila de ν de L^t)$U_{\nu i} \neq 0$ A.6

Localizando un arreglo dimensional IY(ILT),JU(JLT) y UN(LTN), que toma las filas índices,columnas índices y valores numéricos de U(L^t) - los arreglos ILT,JLT, y LTN son equivalentes a IV,JV y UN respectiva-- mente y ambos no toman almacenamiento adicional.

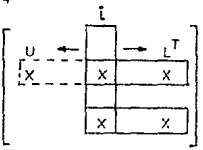
En la fig. M-14 se muestra un almacenamiento rápido durante el procesamiento de la fila (i) de A hacia la fila de L^t.Los elementos de la columna (i) son inicialmente cambiados de elementos de L^t hacia elementos de U.

La multiplicación (multiplicadores l_{iu}) son disponibles en arreglos JLT y LTN y serán directos para un arreglo IUP(ILTP) de dimension (n). El elemento $l_{iu} \neq 0$ si JLT(ILTP(u))=i.

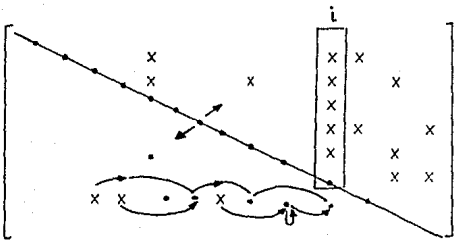
La fig. M-15 muestra una mejor solución,observese que en M-15 aplicado a matrices generales,hay un inherente orden en el que las operaciones elementales son ejecutadas; i.e, =1,...,i-1,$l_{iv} \neq 0$.La razón para esto es que l_{iu} es independiente de valores previos de l_{iu}, =1,..., -1. Esto no es cierto en A.6 cuando todos los multiplicadores han sido calculados.Estas características permiten las operaciones elementales (02) en algún orden.En la fig. M-15 los índices $k_1 \ldots k_4$,son las columnas índices de los elementos no-cero de la fila (i) de A izquierdo a la diagonal.Ahora se puede demostrar que en un subconjunto de estos índices puede ser usado para puntualizar el conjunto total de multiplicadores no-- cero de (3),cada índice K , =1,.....,4 da origen a una cadena de elementos : (K_1,K_1),(K1,l1),(l1,l1),(l1,l2),(l2,l2),(l2,i)(i,i),(K2,Ka), (K2,l1),(l1,l1),(K3,K3),(K3,l3),(l3,l3),(l3,i)(i,i) y (K4,K4).
El conjunto de columnas índices menores que (i) en este conjunto de la - cadena es una secuencia total de las operaciones elementales de f(3); -- i.e.,(k,l,l_2,K_3,K_4,l_3,j) donde $l_3=K_4$.Seleccionando el inicio lógico de la cadena ,suponiendo un elemento general de alguna cadena,ejemplo: el elemento(l_1,l_1) de la cadena K_1,para obtener el siguiente elemento observamos el primer elemento,Ul_1l_2 de la fila l_1 de U,por que de la simetría $l_{12}l_1 \neq 0$.Ahora considerando alrededor de las operaciones que ocurren cuando la fila l_2 de A fuese procesada a la fila l_2 de L y U.

Un múltiplo de la fila l_1 será substraído de la fila l_2 y el elemento Ul_{2i} recibiendo una contribución no-cero de $-l_{12,11}U_{11}$.

De este modo conociendo $Ul_{2i} \neq 0$ y por simetría así es $l_i l_2$ que dice que la fila l_2 se integra en A-6 y ahí se puede definir la siguiente entrada en la cadena. La cadena termina cuando se alcanza un múltiplo que anteriormente ha sido usado (elemento (l_1, l_1) en la cadena k_2, y el elemento (k_4, k_4) en cadena K_4) ó cuando se alcanza la diagonal.(Elemento (i, i) en cadenas k_1 y k_3)



M-14



M-15

FORMA DE ELIMINACION DE LA INVERSA (EFI)

Dos formas de la inversa no-explícita más comunmente usadas son : La forma del producto de la inversa (PFI) y la forma de la eliminación de la inversa (EFI) .

La PFI tiene características benéficas cuando es empleada en la solución de conjuntos de ecuaciones con tipos de estructura específica . La EFI que es la más popular involucra la determinación de 2n matrices factores - tales que :

$$A^{-1}=U_1U_2\ldots U_{n-1}U_nL_nL_{n-1}\ldots\ldots L_2L_1 \qquad A.7$$

donde las $U_{k's}$ son matrices triangulares superiores de orden n, las $L_{k's}$ son matrices triangulares inferiores de orden n_1 y A es la matriz coeficiente. Las $L_{k's}$ y $U_{k's}$ son usualmente determinadas por el uso de la eliminación Gaussiana como se describe a continuación :

Iniciando con el conjunto inicial de ecuaciones,

$$Ax=b \qquad (1)$$

L_1 esta definido, así que premultiplicando (1) por L_1 es equivalente a el primer paso de la eliminación Gaussiana. Por ejemplo :

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

L_1 seria

M-16

$$L_1 = \begin{bmatrix} 1/a_{11} & 0 & 0 \\ -a_{21}/a_{11} & 1 & 0 \\ -a_{31}/a_{11} & 0 & 1 \end{bmatrix}$$

M-17

el producto $L_1 A$ es :

$$L_1 A \begin{bmatrix} 1 & a_{12}/a_{11} & a_{13}/a_{11} \\ 0 & a_{22} - a_{21}\frac{a_{12}}{a_{11}} & a_{23} - a_{21}\frac{a_{13}}{a_{11}} \\ 0 & a_{32} - a_{31}\frac{a_{12}}{a_{11}} & a_{33} - a_{31}\frac{a_{13}}{a_{11}} \end{bmatrix}$$

M-18

Este es el mismo resultado que se tendría de la eliminación Gaussiana. U_1 es definida tal que en el producto L_1AU_1 la fila 1 es definida a hacerse idéntica a la primera fila de la matriz identidad, esto es:

$$U_1 \begin{bmatrix} 1 & -a_{12}/a_{11} & -a_{13}/a_{11} \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

M-19

tomando L_1AU_1 como A' ; $L_1AU_1 = A'$ entonces

$$A^1 = L_1AU_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & a_{22}^{(1)} & a_{23}^{(1)} \\ 0 & a_{32}^{(1)} & a_{33}^{(1)} \end{bmatrix} \quad \text{donde} \quad a_{Lj}^{(1)} = a_{Lj} - a_{L1}\frac{a_{1j}}{}$$

M-20

continuando el proceso con A' dado :

$$A^2 = L_2A^1U_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{a_{22}^{(1)}} & 0 \\ 0 & -\frac{a_{32}}{a_{32}^{(1)}} & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & a_{22}^{(1)} & a_{23}^{(1)} \\ 0 & a_{32}^{(1)} & a_{33}^{(1)} \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & -\frac{a_{23}}{a_{22}^{(1)}} \\ 0 & 0 & 1 \end{bmatrix}$$

M-21

$$A^{(2)} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & a_{33}^{(2)} \end{bmatrix} \quad \text{y finalmente para } A^{(3)} \qquad \text{M-22}$$

$$A^{(3)} = L_3 A^{(2)} U_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1/a_{33}^{(2)} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & a_{33}^{(2)} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad \text{M-23}$$

$$A^{(3)} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad \text{M-24}$$

Notese que en el caso general n-dimensional $A^n = I$ se tiene

$$L_n L_{n-1} \ldots L_2 L_1 A U_1 U_2 \ldots U_{n-1} U_n = I \qquad \text{A.8}$$

Resolviendo A.8 , para A^{-1} por la 1ª premultiplicación, ambos lados - por $L_{n-1}^{-1} \ldots L_1^{-1}$, A^{-1} respectivamente da;

$$U_1 U_2 \ldots U_{n-1} U_n = A^{-1} L_1^{-1} L_2^{-1} \ldots L_{n-1}^{-1} L_n^{-1} \qquad \text{A.9}$$

Esto es lo mismo que (a'). La solución puede ahora ser representada como una serie de matrices factoradas :

$$X = A^{-1} b = U_1 U_2 \ldots U_{n-1} U_n L_n L_{n-1} \ldots L_2 L_1 b \qquad \text{A.10}$$

Usando notación estandar, las matrices factoradas son denotadas como

$$U^{-1} = U_1 U_2 \ldots U_{n-1} U_n \qquad \text{A.11}$$

$$L^{-1} = L_n L_{n-1} \ldots L_2 L_1 \qquad \text{A.12}$$

y la solución se hace :

$$X = U^{-1} L^{-1} b \qquad \text{A.13}$$

Comparando este sistema con II el producto $U^{-1}L^{-1}$ es A^{-1}

por lo tanto :

$$A^{-1}=U^{-1}L^{-1} \qquad A.14$$

$$A=LU \qquad A.15$$

De este modo, el uso de la forma de eliminación de la inversa es equivalente a la descomposición LU ; esto es, es así equivalente a factorizar A hacia un producto de una matriz triangular inferior y una matriz triangular superior. Así se notara tambien que los factores L_k representan una eliminación columna por columna, durante la primera fase de la eliminación Gaussiana y los factores U_k representan la eliminación fila por fila durante la substitución inversa.

Resolviendo las ecuaciones A-14 y A-15 para U y L respectivamente dadas:

$$U=U_n^{-1}U_{n-1}^{-1}.....U_2^{-1}\,U_1^{-1} \qquad A.16$$

$$L=L_1^{-1}L_2^{-1}.....L_{n-1}^{-1}L_n^{-1} \qquad A.17$$

Para el ejemplo dado en M-16 hasta M-24 L es evaluada o calculada por la aplicación de las multiplicaciones de matrices indicadas por M-16 tenemos :

$$L = L_1^{-1}L_2^{-1}L_3^{-1} = \begin{bmatrix} a_{11} & 0 & 0 \\ a_{22} & 1 & 0 \\ a_{31} & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 \\ 0 & a_{22}^{(1)} & 0 \\ 0 & a_{32}^{(1)} & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & a_{33}^{(2)} \end{bmatrix}$$

M-25

$$L = \begin{bmatrix} a_{11} & 0 & 0 \\ a_{21} & a_{22}^{(1)} & 0 \\ a_{31} & a_{32}^{(1)} & a_{33}^{(2)} \end{bmatrix}$$

M-26

similarmente para U :

$$U = U_3^{-1}U_2^{-1}U_1^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & a_{23}/a_{22}^{(1)} \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & a_{12}/a_{11} & a_{13}/a_{11} \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

M-27

$$U = \begin{bmatrix} 1 & \frac{a_{12}}{a_{11}} & \frac{a_{13}}{a_{11}} \\ 0 & 1 & \frac{a_{23}^{(1)}}{a_{22}^{(1)}} \\ 0 & 0 & 1 \end{bmatrix} \qquad \text{M-28}$$

Los elementos de la inversa de cada L_k son calculados los elementos diagonales y multiplicando los elementos fuera de la diagonal por el inverso negativo del elemento diagonal en esa columna. La inversa de cada U_k es calculada similarmente excepto que las columnas son remplazadas por las filas en la anterior declaración. Notese que si $A^{(0)}=A$ $\|$ a_{ik} th columna de L es la K^{th} columna de $A^{(k-1)}$ y la K^{th} fila de U es la mismisma que la K^{th} fila de $L_k A^{(k-1)}$.

Visualizando la forma de eliminación inversa, los elementos no triviales de los factores L_k y U_k pueden ser compactados en una matriz A^1 :

$$A^1 = \begin{bmatrix} 1/a_{11} & -a_{21}/a_{11} & -a_{31}/a_{11} \\ -a_{21}/a_{11} & 1/a_{22}^{(1)} & -a_{23}^{(1)}/a_{22}^{(1)} \\ -a_{31}/a_{11} & -a_{32}^{(1)}/a_{22}^{(1)} & 1/a_{33}^{(2)} \end{bmatrix}$$

M-29

donde los elementos de las $L_{k's}$ ocurren sobre y bajo la diagonal y los elementos de las $U_{k's}$ ocurren sobre la diagonal. Así, la información contenida en A^1 es la necesaria para calcular X para un vector (b) dado. El A_{ij} para A^1 esta determinado por la siguiente ecuación :

$$a'_{ik} = \frac{-a_{ik}^{k-1}}{a_{kk}^{k-1}} \qquad \text{para } i \; k$$

$$a'_{kk} = \frac{1}{a_{kk}^{k-1}} \qquad a'_{kj} = \frac{-a_{kj}^{k-1}}{a_{kk}^{k-1}} \qquad \text{para } k \; j$$

A.18

Las razones para representar la inversa en forma factorada es que para matrices dispersas esto reduce el "fill-in" significativamente. Como un ejemplo considerar la siguiente matriz 10 x 10 con 37 no-ceros:

$$A = \begin{bmatrix} X & X & X & & & & & & & \\ X & & & & X & & & & & X \\ X & X & & X & & & X & & & X \\ & & & X & & X & & & X & \\ X & & & & X & & & & & \\ & X & X & X & X & & X & X & & \\ & & X & X & & & & & X & X \\ X & & & & & & & & & X \\ & X & X & & X & X & & & & \\ X & & & X & X & X & X & & & \end{bmatrix}$$

M-30

Donde las X's representan los elementos no-ceros y los espacios son ceros. Usando la eliminación estandar (Gaussiana) e ignorando la accidental posibilidad de creación de ceros, la forma explícita de la inversa es completamente llena.

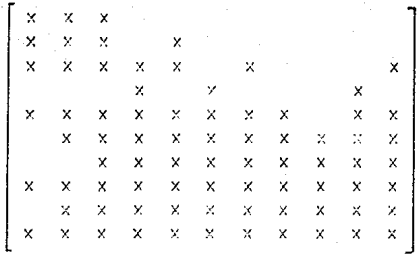
$$A^{-1} = \begin{bmatrix} X & X & X & X & X & X & X & X & X & X \\ X & X & X & X & X & X & X & X & X & X \\ X & X & X & X & X & X & X & X & X & X \\ X & X & X & X & X & X & X & X & X & X \\ X & X & X & X & X & X & X & X & X & X \\ X & X & X & X & X & X & X & X & X & X \\ X & X & X & X & X & X & X & X & X & X \\ X & X & X & X & X & X & X & X & X & X \\ X & X & X & X & X & X & X & X & X & X \\ X & X & X & X & X & X & X & X & X & X \end{bmatrix}$$

M-31

Se requieren 577 operaciones (multiplicaciones y divisiones) para obtener A^{-1} y asumiendo un vector (b) total, una adición de 100 operaciones para calcular la solución.

Ahora la forma compacta de la EFI, A^I :

$A^1 =$



M-32

Esta forma requiere 139 operaciones para evaluar y una adición de 73 operaciones para obtener la solución si (b) esta completa.

SELECCION DEL PIVOTE PARA MANTENER LA DISPERSIDAD-

Los metodos de selección de pivote para minimizar el "fill-in" caen - en una de dos grandes categorias. Los metodos locales reevaluan la matriz después de cada paso de la eliminación y elige un pivote que tienda a minimizar el "fill-in" para el siguiente paso ó pasos.

Los metodos " A priori " reordenan las columnas y/o filas anteriores a los cálculos y generalmente permiten el pivote parcial. Es generalmente entendido que un método a priori no son casi tan efectivos como los metodos locales en la preservación de la dispersidad o reducción del número de cálculos. En la aplicación de los metodos locales no es posible conocer -- donde ocurrira el fill-in, mientras que por el otro lado en la aplicación de los metodos a priori es posible predecir los sitios del fill-in, de este modo es posible emplear 2 ó fijar 1 esquemas de almacenamiento dis-- perso.

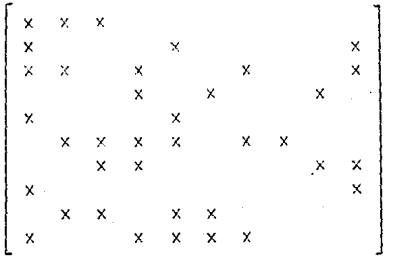
Algunas técnicas de ordenación local se describen a continuación; Después de analizar las técnicas a priori más utilizadas.

-- SELECCION DE PIVOTE MARKOWITZ --

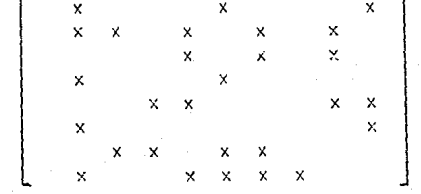
A el paso $(K+1)^{th}$ de la eliminación, el algoritmo de Markowitz selecciona un elemento a_{ij}^{k} $(i>k, j>k)$ que minimize el producto del número de otros no-ceros en la columna (j) al tiempo el número de otros en la fila (i). Este producto es el máximo fill-in que podría ocurrir posiblemente durante el paso si, es elegido como un pivote.

Simbólicamente este método selecciona un pivote no-cero, que minimiza $(r_i^k - 1)(c_j^k - 1)$ donde r_i^k y c_j^k son el número de elementos no-ceros restantes en la fila (i) y columnas (j) al final del paso k.

Como un ejemplo, la matriz (M-30) es reintroducida:

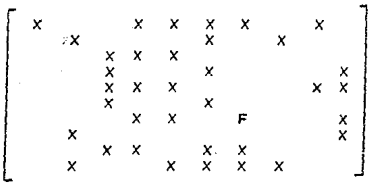


Después de examinar $a_{68}^{(0)}$ es encontrada para ser el pivote de acuerdo al criterio. Aquí, $r_6=6, c_2=1$ y $(6-1)(1-1)=0$. De este modo no resultara fill-in. Reacomodando el pivote $a_{68}^{(0)}$ en la posición (1,1) y ejecutando el primer paso de la eliminación Gaussiana de la siguiente forma para A^1 :



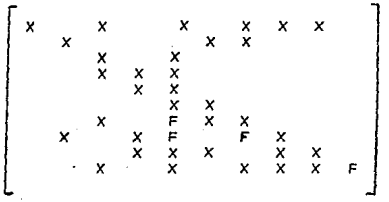
M-33

Ahora considerando solamente de las filas y columnas 2 hasta las 10 - dea$^{1}_{50}$ es elegida como el nuevo pivote. El siguiente paso de la eliminación resulta en la matriz A^2



M-34

Después del segundo paso de la eliminación, ahí aparece un fill-in - como el denotado por la "F". Finalmente através de n-pasos resulta la si-- guiente estructura cero/ no-cero para A^1:



M-35

La forma compacta de la EFI usando el criterio de Markowitz requiere 34 operaciones para calcular. Una adición de 42 operaciones (igual al número de no-ceros en A^1) son requeridos para obtener la solución dando un vector (b).

Steward (1962,1965) puntualizó que suponiendo un corte, una aproximación iterativa hacía resolver problemas lineales o no lineales dispersos. Considerar el siguiente conjunto de ecuaciones :

$$\begin{aligned} a_{11}x_1 &+ 0 + 0 + a_{14}x_4 &= b_1 \\ a_{21}x_1 &+ a_{22}x_2 + 0 + 0 &= b_2 \\ 0 &+ a_{32}x_2 + a_{33}x_3 + 0 &= b_3 \\ 0 &+ 0 + a_{43}x_3 + a_{44}x_4 &= b_4 \end{aligned} \qquad \text{M-36}$$

Si x_4 fuese conocida, el resto del sistema podría ser resuelto por sustitución directa. Una posible solución iterativa sería : Suponer x_4; resolver la ecuación para x_1 por sustitución directa de x_4, resolver la ecuación para x_2 por sustitución directa de x_1; resolver la ecuación 3 para x_3 por sustitución directa de x_2; resolver la ecuación 4 para x_4 por sustitución directa de x_3; comparar el valor calculado de x_4 con el valor supuesto; repetir el proceso usando el nuevo valor, hasta que x_4 converga.

En este sistema x_4 es el corte (variable) y la ecuación 4 es la ecuación de corte. El " conjunto de salida " consiste del par ecuación-variable que indica que ecuación será resuelta, para que variable, el orden de procedencia ; es la secuencia en que los pares ecuación-variable serán resueltos. Tambien - Steward remarcó la importancia de partir o dividir los sistemas hacia subsistemas pequeños que pueden ser resueltos independientemente unos de otros sin necesidad de cortar. Uno de tales sistemas es :

$$\begin{bmatrix} X & & X & & & & \\ X & X & & & & & \\ & X & X & & & & \\ & & & X & & X & \\ & & & X & X & & \\ X & & & & X & X & \end{bmatrix} \qquad \text{M-37}$$

En el que las X's representan las variables de ocurrencia en una ecuación. En este sistema las tres primeras ecuaciones pueden ser resueltas fue---

ra de la consideración de las tres últimas ecuaciones y las tres últimas - tambien pueden ser resueltas independientemente de las tres primeras,una - vez que éstas han sido resueltas.

-- PROCEDIMIENTO DE PIVOTEO PRE-ASIGNADO PARTICION (P4) --

Muchos métodos primero ordenan la matriz tanto como sea posible en una forma triangular inferior.Esto se da durante la triangulacióhacia arriba y triangulación hacia abajo,en las fases de ordenación.

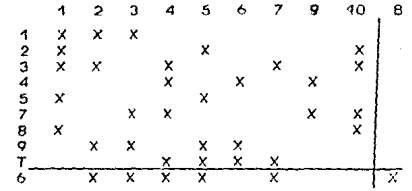
Como un ejemplo,considerando la matriz de 10 X 10:

	1	2	3	4	5	6	7	8	9	10
1	X	X	X							
2	X				X					X
3	X	X		X			X			X
4				X		X			X	
5	X				X					
6		X	X	X	X		X	X		
7			X	X					X	X
8	X									X
9		X	X		X	X				
10	X			X	X	X	X			

M-38

La triangularización hacia abajo (o inversa) involucra encontrar repetidamente una columna con un simple no-cero,eliminando la fila con que ésta - se intersecta y la columna, y colocando la fila y columna en la última posición disponible de la matriz reordenada,la triangularización finaliza cuando todos los elementos tienen dos o mas elementos.La triangularización hacia -- arriba procede similarmente excepto que las filas con un sólo no-cero son encontradas y las columnas que se intersectan con ellas son colocadas en la -- primera posición disponible de la matriz ordenada.

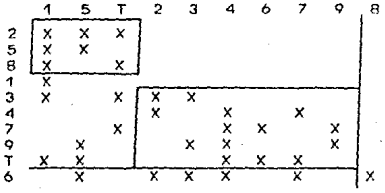
En este ejemplo la columna 8 tiene un simple no-cero de este modo la columna 8 y la fila 6 colocadas al final de la matriz reordenada:



M-39

Las líneas separan la parte ordenada de la desordenada de la matriz. Hasta este punto no hay más columnas con un no-cero. El procedimiento entonces continua a la triangularización hacia arriba. Sin embargo, no hay filas con un sólo elemento no-cero, así que la triangularización termina.

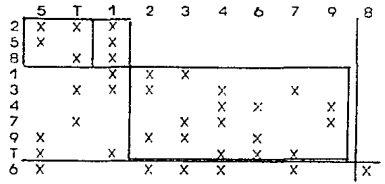
El algorítmo P4 ahora parte la matriz hacia bloques irreducibles



M-40

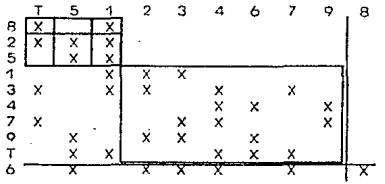
Los cuadros indican los bloques irreducibles. P4 procesa un bloque a un tiempo, la siguiente fase de la ordenación es la "selección del pico" El objetivo aquí es elegir una columna que dada su supresión daría la mayor oportunidad de reordenar la parte restante del bloque usando triangularización hacia arriba. P4 selecciona como un pico la columna que intersecta con más filas de mínima cuenta de filas. Considerando el bloque de 3 x 3 en M-40 la mínima cuenta de filas es 2. La columna 1, que intersecta con dos filas de cuenta dos, es seleccionada como el pico. Notese que

si la columna (1) es eliminada habrá dos filas con 1,dando en P_4 dos - cambios que permitan la reordenación completa por triangularización hacia adelante.La columna 1 es eliminada y cambiada temporalmente a la ultima posicion en el bloque y se asignará una fila que este vacía,anterior en la reordenación.La situación actual esta marcada como:



M-41

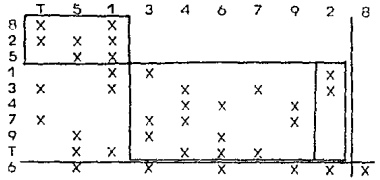
P_4 ahora continua con triangularizacion hacia arriba dentro del primer - bloque.Las filas 5 y 8 tienen cada una un simple elemento,eligiendo arbitrariamente la fila 8.La fila 8 y la columna son entonces colocadas en la primera posición disponible de la matriz reordenada :



M-42

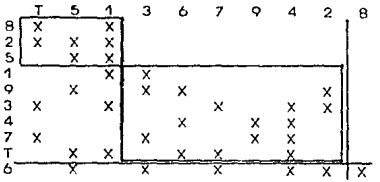
Eliminando la fila 8 y la columna T permitiendo dos filas,cada una - con un solo elemento.Seleccionando la fila dos y la columna 5 como el siguiente pivote.Esto permite que la fila 5 sin elementos,así sea asignada a la columna pico,columna 1.

Continuando con el siguiente bloque,la columna dos es seleccionada como la columna pico:



M-43

La triangularización hacia arriba continua con la fila 1 y la columna 3 y con la fila 9 y la columna 6.Además todas las filas en la matriz restante tienen dos ó más elementos.La columna 4 es seleccionada como un pico:



M-44

Cuando aplicamos la eliminación hacia adelante,P_4 intentará encontrar - una fila,tal que cuando corresponda a una columna sea eliminada,permitiendo más de una fila sin elementos.En este caso aplica a todas las filas restan-- tes.Por instancia,eligiendo la fila tres y la columna 7 como el siguiente pivote,dejando la fila T que no tiene elementos,por lo tanto el pico 4 puede - ser asignado a la fila T:

	T	5	1	3	6	7	4	9	2	8	
8	X		X								
2	X	X	X								
5		X	X								
1			X	X					X		
9		X		X	X				X		
3	X					X	X		X		M-45
T		X	X		X	X	X				
4					X		X	X			
7	X			X			X	X			
6		X		X		X	X		X	X	

La reordenación es terminada seleccionando la fila 4 y la columna 9 como el siguiente pivote, así dejando la fila 7 sin elementos. Esta es asignada a la columna pico 2 . La matriz M45 es la reordenación final.

Calculando el EFI para esta matriz resulta la siguiente estructura para la forma compacta de la inversa A^{-1}:

	T	5	1	3	6	7	9	2	8	4	
8	X		X								
2	X	X	X								
5		X	X								
1			X	X					X		
9		X		X	X				X		
3	X					X	X		X		M-46
T		X	X		X	X	X		X		
4					X		X	X	X		
7	X			X			X	X	X		
6		X		X		X	X		X	X	

La EFI que es obtenida de el P4 requiere 19 operaciones a calcular. Uno adicional de 40 operaciones son requeridas para obtener la solución dado un vector b . Esta forma de la inversa tiene 40 elementos comparados con las - 100 en la forma explícita de la inversa y 73 en la EFI con pivoteo no es especial para mantener la dispersidad.

--- SELECCION DEL PIVOTE PARA MEJORAR LA ESTABILIDAD ---

Tradicionalmente el pivoteo ha sido empleado para mantener la estabilidad, esto es mediante la selección como pivote del elemento con el mayor valor absoluto en la fila o en la columna (pivoteo parcial) o con el mayor valor absoluto en la matriz activa (pivoteo completo). Si alguno de estos métodos se aplicara a una matriz grande dispersa, todas las ventajas de usar un esquema de ordenación se perderian. Sin embargo, si los esquemas de ordenación se aplicaran estrictamente a reducir el fill-in, y surge la inestabilidad numérica.

Este dilema es resuelto usando el método llamado "pivoteo inicial", en la aplicación de este método algunos elementos cuyo valor absoluto es mayor que la tolerancia pivote, PTOL, a veces el máximo valor absoluto en la fila (o columna) es un pivote aceptable, la tolerancia del pivote, es una fracción tal que 0 PTOL 1 . En la aplicación de estas a la técnica de Markowitz, un pivote es elegido con el más bajo $(r_i^{k}-1)$ $(c_j^{k}-1)$ que tambien satisface el criterio de principio.

En la aplicación de la técnica a la forma pico de la matriz,el pivoteo ocurre sólo si los elementos de la diagonal no sarisfacen la prueba de principio. Cuando esto pasa la columna cercana a la diagonal en la que el elemento no-cero satisface la prueba.

FASE DE ORDENACION

A continuación se describen brevemente algunas técnicas de ordenación - comparadas por Wood (1979).

P4

1.- Triangularización inversa, triangularización hacia adelante.
2.- Partición de la matriz restante, e introducción de bloques irreducibles.
3.- Seleccionar una columna pico y colocarla sobre la fila pico. Seleccionar como columna pico, la columna que tiene el máximo número de intersecciones con fila de número mínimo de fila. Si el máximo número de tales es - uno y hay más de una columna teniendo una intersección con una fila de - mínimo número, entonces se elige por tomar la columna que intersecta -- con más filas del siguiente número mínimo de fila.
4.- Reajustando filas contadas a numerar para remover el pico y triangularizar hacia adelante, si una fila con elementos no-ceros existe, asignarla a la columna pico a lo alto de la pila. Sacar la anterior de la pila y colocar la pila asignada en la nueva posición en el bloque reordenado.
5.- Vaya al paso 3.
6.- Saque el bloque siguiente y procese, vaya al paso 3, si no hay bloque - sobre la pila vaya al paso 7.
7.- Fin.

P4 es una modificación de P3, la principal modificación es la inclusión de partición en P4, además P4 incluye algunas observaciones, la selección - del pico y la reordenación estan basadas principalmente en consideraciones locales considerando fuera la estructura total de la matriz reordenada.

b) --- HP 11 ---

1) Triangularización adelante, triangularización inversa.
2) Partición de la matriz resultante y colocar bloques irreducibles sobre la pila de bloques. Procesar el primer bloque sobre la pila procediendo al paso (3).
3) Hacer la selección de la columna pico usando el mismo criterio que en P4.
4) Ajustar las filas numeradas y triangularizar adelante.
5) Hacer la selección inicial de la fila spike. Los criterios de selección son los transpuestos de los usados en el paso 3 (remplazar filas con columnas y viceversa en los criterios de selección de columnas spike.
6) Ajustar las columnas numeradas y triangularizar (hacia atras) inversamente.
7) Partición del bloque restante hacia subbloques irreducibles. Calcular el índice P, definido como la suma de los cuadrados del orden de subbloques.
8) Encontrar la combinación de columna y fila pico que de el mismo P aplicando un criterio de exclusión para reducir el número de combinaciones examinadas (cada combinación requiere la partición en orden a evaluar P.
9) Reexaminar todas las columnas con columnas enumeradas mayores que esa columna pico seleccionada en (8). Entonces es elegida como pico la columna con el mayor número. Si algunos subbloques irreducibles son encontrados como un resultado de la partición del bloque restante después de la selección final de la fila y columna pico, entonces pongase éste sobre la pila de bloques.
10) Tome el bloque siguiente de la pila de bloques y procese pasando al paso (3), si no hay más bloques sobre la pila pase al paso (11).
11) Fin.

c) --- HP 10 [11] ---

Es el mismo que HP excepto que en el paso (8), el número de combinaciones de columna y fila pico es reducida. Esto incrementa la velocidad de reordenación, pero el valor de P logrado no será mayor que un valor absoluto mínimo.

--- HP 20 [11] ---

Este es el mismo que HP, sin embargo es omitido el paso (8) para adicionar una velocidad mayor de reordenación relativa a HP [10].

--- HP 30 [11] ---

Wood (1979) propone este algoritmo como una modificación adicional - de HP, es el mismo que HP excepto que los pasos (8 y 9) son omitidos.

d) --- SPK 1 ---

1) Triangularización hacia adelante, triangularización inversa.
2) Participación de la matriz restante e introducir los bloques irreducibles sobre la pila de bloques. Procesar el primer bloque sobre la pila para - proceder al paso (3).
3) Los criterios para la selección del pico son:
   a) Encontrar la fila con mínimo número de fila, si es un límite, entonces para cada fila dividida teniendo no-ceros, sumar los números de columna. Seleccionar la fila con la mayor suma.
   b) Ahora considerar sólo las columnas teniendo elementos no-cero en la fila seleccionada en (3-a). La fila seleccionada es asignada a la columna con la numeración más pequeña y son colocadas en las posiciones --- abiertas en el bloque reordenado, la columna con el mayor número de co-

lumna es entonces puesta sobre la pila pico, seguida por la columna con el siguiente número mayor de columna,etc., hasta el total número de columnas restantes con elementos no-ceros en la fila seleccionada son colocadas sobre la pila pico. Seleccionar las columnas pico y colocarlas sobre la pila pico.

4) Reajustando filas numeradas y triangularizando hacia adelante. Si una fila con no-ceros aparece, asignar a la columna sobre lo alto de la pila pico sacando los picos de la pila y colocando ahí las filas asignadas hacia las siguientes posiciones en el bloque reordenado, si esta completo el bloque vaya hacia el paso 6.

5) Vaya al paso 3.

6) Coloque el siguiente bloque desde el bloque pico y procese en el paso 3.

7) Fin.

SPK1 es menos sofisticado que P4, esto permite aumento en la velocidad de reordenación.

e) --- SPK 2 ---

Este algoritmo esta basado sobre las ideas usadas en el algoritmo de partición descrito por Stadtheir (20) que tiende a localizar pequeños subbloques (2 x 2 en este caso), SPK 2 es el mismo que SPK 1 excepto en el paso 3.

3) Seleccionar una columna pico y colocarlos sobre la pila.
Los criterios para la selección del pico son :

a) Encontrar la fila con mínimo número de filas, si esta es un enlace, entonces para cada fila contenida, el número de filas de mínimo número de fila que resultara si todas las columnas no-cero en la fila contenida fueran eliminadas. Eligiendo la columna que complazca la mayor parte de tales filas de mínima cuenta de fila.

b) Si hay un enlace fijo de filas con número mínimo de filas, use el procedimiento de enlace en (3-a) SPK 1.

c) Ahora considerando solamente la columna, teniendo no-cero en fila seleccionada en (3-a y 3-b) asignamos la fila seleccionada y la columna correspondiente a la matriz reordenada y a la pila pico. como en el paso (3-b) de SPK 1.

f) --- BLOQUES ----

Este algoritmo es para usarse en sistemas con una relativa dispersidad en matrices de bloques que aparecen y pueden ser construidas de la siguiente forma :

1) Identificar la estructura de bloques de la matriz y construir la matriz correspondiente, bloques ocurriendo. Para las matrices de flowsheeting este procedimiento es más detallado en 2 .
2) Aplicar SPK 2 a la matriz de bloques ocurriendo.
3) Expender la matriz de bloques-ocurriendo, se ordena hacia una matriz - ordenada sobre el nivel de ecuaciones variables.
4) Triangularizar hacia adelante, triangularizar inversamente. conjunto -- k=1.
5) Definir como activo para la selección del pivote las filas remanentes en el k-th bloque-fila en el bloque de la matriz de ocurrencia. Si no - hay filas restantes en el K-th bloque-fila vaya a (7).
6) Aplique los pasos del (3-5) del SPK 1, excepto la restricción del paso (3-a) de SPK 1 de las filas correspondientemente definidas como activas. Notese que en el paso 4 de SPK1, todas las filas son consideradas para triangularizar inversamente, no exactamente esas activas para la selección del pivote. Cuando las filas activas han sido reordenadas - fuera del paso 4 de SPK 1 y procedidas al paso de abajo (5) antes que al paso (6) de SPK 1.
7) Si K=N donde N= número de fila-bloque vaya al paso (8) de otro modo -- conjuntando K= K + 1 vaya al paso (5).
8) Fin.

A.15 FASE NUMERICA

La segunda fase del método en dos pasos es la solución numérica de la matriz reordenada, en esta fase una consideración primaria es el evitar la creación de elementos no-cero (fill-in). Se han desarrollado una gran cantidad de algoritmos (programas) para la solución de sistemas lineales resultantes en problemas de flowsheeting de procesos.

A continuación se describen algunos de los algoritmos que se han generado :

a) Substitución inversa continua (CBS).

CBS toma ventaja especial de la forma pico de la matriz, este método elimina repetidamente adelante através de cada fila pico y elimina inversamente la columna pico correspondiente. De esta manera el fill-in potencial es restringido al área en la columna pico bajo el mayor no-cero y sobre o arriba de la diagonal. Recordando que en la descomposición LU el - fill-in ocurre en cualquier sitio en la columna pico bajo el mayor no-cero.

La forma factorizada de la inversa que resulta de CBS es una forma modificada del producto de la inversa.

El procedimiento es demostrado usando la matriz de 10 x 10 (la estrategia es una modificación de la de Stadtheir y Wood (1980)).

	1	2	3	4	5	6	7	8	9	T
1	X	X	X							
2	X				X					X
3	X	X		X			X			X
4				X		X			X	
5	X				X					
6		X	X	X	X		X	X		
7			X	X					X	X
8	X									X
9		X	X		X	X				
T	X			X	X	X	X			

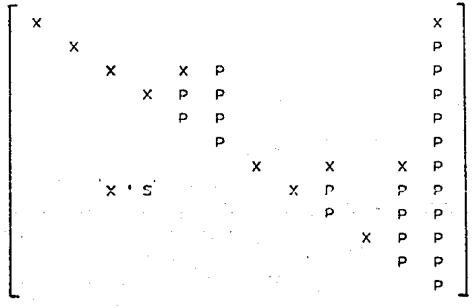
M-47

Aplicando la técnica de reordenación SPK 2 (M-47) da :

accidentales). Tambien ha sido observado que en la mayoría de las matrices reordenadas por los otros metodos ocurre completo fill-in. Sin embargo, como se mostro anteriormente, usando una cantidad moderada de pivoteo(para - mantener la estabilidad numérica) causa considerablemente más fill-in en -- CBS que en la descomposición LU.

El completo fill-in que ocurre cuando se aplica CBS sin embargo, sugiere otra ventaja potencial. Porque las columnas pico en muchos casos se llenan completamente, ellos pueden ser almacenados eficientemente usando todas las técnicas de almacenamiento; esto es, sólo se necesita un sitio de almacenamiento necesario para distribuir por no-cero (una técnica así llamada LT). Indicando por la localización y el número de fila del mayor no-cero en el pico , todos los otros elementos en el pico pueden ser situados directamente.

En el caso de la solución de un simple vector (b), las columnas pico - que son eliminadas inversamente no necesitan ser retenidos, con el libre almacenamiento para picos encontrados en cálculos anteriores. Esta prioridad es particularmente importante cuando la estructura condiderada esta presente, la cual es ilustrada abajo :



M-63

Las X's representan no-ceros y las P's representan no-ceros potenciales.

	T	5	1	2	6	9	4	7	3	8
8	X		X							
2	X	X	X							
5		X	X							
1			X	X					X	
9		X		X	X				X	
7	X					X	X		X	
4					X	X	X			
T		X	X		X		X	X		
3	X		X	X			X	X		
6		X		X			X	X	X	X

M-48

CBS inicia con eliminar una fila hacia adelante, através de la primera fila pico fila 3. Por conveniencia, se enumeran las filas y columnas en -- forma ascendente. El resultado de este paso es :

	1	2	3	4	5	6	7	8	9	T
1	1		X							
2	0	1	X							
3		0	1							
4			X	X					X	
5		X		X	X				X	
6	X					X	X		X	
7					X	X	X			
8		X	X		X		X	X		
9	X		X	X			X	X		
T		X		X			X	X	X	X

M-49

Ahora la eliminación inversa es ejecutada sobre la columna 3 :

	1	2	3	4	5	6	7	8	9	T
1	1		0							
2		1	0							
3			1							
4			X	X					X	
5		X		X	X				X	
6	X					X	X		X	
7					X	X	X			
8		X	X		X		X	X		
9	X		X	X			X	X		
T		X			X		X	X	X	X

M-50

La eliminación adelante continua através de la siguiente fila pico, - fila 7 :

	1	2	3	4	5	6	7	8	9	T
1	1									
2		1								
3			1							
4			0	1					X	
5		0		0	1				X	
6	0					1	X		X	
7					0	0	1		F	
8		X	X		X		X	X		
9	X		X	X			X	X		
T		X		X			X	X	X	X

M-51

Esto ha causado fill-in en las posiciones (7,9). La columna pico 7 es ahora eliminada inversamente :

	1	2	3	4	5	6	7	8	9	T
1	1									
2		1								
3			1							
4				1					X	
5					1				X	
6						1	0		X	
7							1		F	
8		X	X		X		X	X		
9	X		X	X			X	X		
T		X		X			X	X	X	X

M-52

Continuando la eliminación através de la ecuación 9 resulta :

	1	2	3	4	5	6	7	8	9	T
1	1									
2		1								
3			1							
4				1					X	
5					1				X	
6						1			X	
7							1		F	
8		0	0		0		0	1	F	
9	0		0	0			0	0	F	
T		X		X			X	X	X	X

M-53

Esto causa fill-in en posiciones (8,9) y (9,9), finalmente la columna 9 es eliminada inversamente y la fila T es eliminada adelante. Esto permite una unidad de matriz diagonal. El vector (b) ahora contiene la solución.CBS automáticamente explora la estructura de bloques de la matriz.

La forma factorizada que resulta de emplear CBS consite de una serie de producto de matrices 2n-1 que transforman el coeficiente de matriz A, hacia la matriz identidad :

$$P_{2n-1} \cdot P_{2n-2} \cdots\cdots\cdots P_2 \cdot P_1 \cdot A = I$$

posteriormente multiplicando por A^{-1} tenemos :

$$P_{2n-1} \cdot P_{2n-2} \cdots\cdots\cdots P_2 \cdot P_1 = A^{-1}$$

$X = P_{2n-1}P_{2n-2}\cdots P_2P_1b$

sustituyendo en la solución estandar $X = A^{-1}b$:

$$P_{2n-1}P_{2n-2}\cdots\cdots P_2P_1 \cdot A^{-1}$$

Un ejemplo de este método es el siguiente :

$$A = A^{(0)} = \begin{bmatrix} a_{11}^{(0)} & & a_{13}^{(0)} & a_{15}^{(0)} \\ a_{21}^{(0)} & a_{22}^{(0)} & a_{23}^{(0)} & a_{25}^{(0)} \\ a_{31}^{(0)} & a_{32}^{(0)} & a_{33}^{(0)} & a_{35}^{(0)} \\ a_{41}^{(0)} & a_{42}^{(0)} & a_{43}^{(0)} a_{44}^{(0)} & a_{45} \\ a_{51}^{(0)} & a_{52}^{(0)} & a_{53}^{(0)} a_{54}^{(0)} & a_{55}^{(0)} \end{bmatrix}$$

M-54

El primer pivote es a_{11} y P_1 divide la fila 1 por $a_{11}^{(0)}$

$$P_1 = \begin{bmatrix} \frac{1}{a_{11}^{(0)}} & & & & \\ & 1 & & & \\ & & 1 & & \\ & & & 1 & \\ & & & & 1 \end{bmatrix}$$

M-55

El resultado de multiplicar $A^{(0)}$ por P_1 es $A_1^{(1)} = P_1A^{(0)}$

$$A_1^{(1)} = \begin{bmatrix} 1 & & \frac{a_{13}^{(0)}}{a_{11}^{(0)}} & & \frac{a_{15}^{(0)}}{a_{11}^{(0)}} \\ a_{21}^{(0)} & a_{22}^{(0)} & a_{23}^{(0)} & & a_{25}^{(0)} \\ a_{31}^{(0)} & a_{32}^{(0)} & a_{33}^{(0)} & & a_{35}^{(0)} \\ a_{41}^{(0)} & a_{42}^{(0)} & a_{43}^{(0)} & a_{44}^{(0)} & a_{45}^{(0)} \\ a_{51}^{(0)} & a_{52}^{(0)} & a_{53}^{(0)} & a_{54}^{(0)} & a_{55}^{(0)} \end{bmatrix}$$

M-56

Redefiniendo todos los elementos de esta fase a $a_{ij}^{(1)}$, reflejando sus valores después de las operaciones indicadas, la fila 2 es eliminada con el producto de la matriz P_2 :

$$P_2 = \begin{bmatrix} 1 & & & & \\ & \frac{1}{a_{21}^{(0)}} & & & \\ & & 1 & & \\ & & & 1 & \\ & & & & 1 \\ & & & & & 1 \end{bmatrix}$$

M-57

El resultado de hacer P_2A^1 es A^2 :

$$A^{(2)} = \begin{bmatrix} 1 & & & & \\ 0 & a_{22}^{(1)} & a_{23}^{(1)} - a_{21}^{(1)}a_{13}^{(1)} & & a_{25}^{(1)} - a_{21}^{(1)}a_{15}^{(1)} \\ a_{31}^{(1)} & a_{32}^{(1)} & a_{33}^{(1)} & & a_{35}^{(1)} \\ a_{41}^{(1)} & a_{42}^{(1)} & a_{43}^{(1)} & a_{44}^{(1)} & a_{45}^{(1)} \\ a_{51}^{(1)} & a_{52}^{(1)} & a_{53}^{(1)} & a_{54}^{(1)} & a_{55}^{(1)} \end{bmatrix}$$

M-58

Dividiendo la fila 2 por $a_{22}^{(2)}$ con P_3 y eliminando la fila 3 con P_4, - la columna 3 es eliminada por premultiplicación por P_5. El resultado es -

$A^{(5)} = P_5 A^{(4)}$, donde :

$$A^{(4)} = \begin{bmatrix} 1 & & a_{13}^{(4)} & & a_{15}^{(4)} \\ & 1 & a_{23}^{(4)} & & a_{25}^{(4)} \\ 0 & 0 & a_{33}^{(4)} & & a_{35}^{(4)} \\ a_{41}^{(4)} & a_{42}^{(4)} & a_{43}^{(4)} & a_{44}^{(4)} & a_{45}^{(4)} \\ a_{51}^{(4)} & a_{52}^{(4)} & a_{53}^{(4)} & a_{54}^{(4)} & a_{55}^{(4)} \end{bmatrix} \quad P_5 = \begin{bmatrix} 1 & & -\frac{a_{13}^{(4)}}{a_{33}^{(4)}} & & \\ & 1 & -\frac{a_{23}^{(4)}}{a_{33}^{(4)}} & & \\ & & \frac{1}{a_{33}^{(4)}} & & \\ & & & 1 & \\ & & & & 1 \end{bmatrix}$$

M-59

$$A^{(5)} = \begin{bmatrix} 1 & & 0 & & a_{15}^{(4)} - a_{35}^{(4)} a_{13}^{(4)} / a_{33}^{(4)} \\ & 1 & 0 & & a_{25}^{(4)} - a_{35}^{(4)} a_{23}^{(4)} / a_{33}^{(4)} \\ & & 1 & & a_{35}^{(4)} / a_{33}^{(4)} \\ a_{41}^{(4)} & a_{42}^{(4)} & a_{43}^{(4)} & a_{44}^{(4)} & a_{45}^{(4)} \\ a_{51}^{(4)} & a_{52}^{(4)} & a_{53}^{(4)} & a_{54}^{(4)} & a_{55}^{(4)} \end{bmatrix}$$

M-60

M-61

Continuando, P_6 elimina la fila 4 y P_7 elimina la columna 4, ya que es to no es una cantidad pico para dividir la fila por $a_{44}^{(6)}$. Finalmente P_8 e limina adelante la fila 5 y P_4 elimina atras la columna pico 5. El producto de matrices resultante puede ser condensado hacia una forma de producto modificado de la inversa :

$$P' = \begin{bmatrix} 1/a_{11}^{0} & & -a_{13}^{4}\, a_{33}^{4} & & -a_{15}^{8}\, a_{55}^{8} \\ -a_{21}^{1} & 1/a_{22}^{2} & -a_{23}^{4}\, a_{33}^{4} & & -a_{25}^{8}\, a_{55}^{8} \\ -a_{31}^{3} & -a_{32}^{3} & 1/a_{33}^{4} & & -a_{35}^{8}\, a_{55}^{8} \\ -a_{41}^{5} & -a_{42}^{5} & -a_{43}^{5} & 1/a_{44}^{6} & -a_{45}^{8}\, a_{55}^{8} \\ -a_{51}^{7} & -a_{52}^{7} & -a_{53}^{7} & -a_{54}^{7} & 1/a_{55}^{8} \end{bmatrix}$$

M-62

Este producto de la forma estandar de la inversa contiene (n) matrices factorizadas cada una eliminando una columna entera, así causando fill-in a través de cada columna pico. sobre o arriba de la diagonal y bajo el mayor no-cero en la columna, note además que el índice sobreescrito de los elementos bajo la diagonal han sido cambiados para reflejar el estado de la matriz a las diversas fases de la combinación, estos elementos son los negativos de los unos correspondientes en la matriz original. El a_{ij} para P' son definidos por las siguientes ecuaciones :

$$a'_{kk} = \frac{1}{a_{kk}^{(2k-2)}} \qquad a'_{ik} = \frac{-a^{(2k-2)}}{a_{kk}^{(2k-2)}} \quad \text{para } i < k$$

$$a_{kj} = -a_{kj}^{(2k-3)} \qquad \text{para } i > k$$

La ventaja esperada de usar CBS es que el área de fill-in potencial es substancialmente reducido en comparación con la descomposición LU.

En realidad se ha observado completo fill-in cuando CBS es aplicado a matrices reordenadas por los algoritmos HP, (excepto para creación de ceros

Excepto para los pico ellos son ceros sobre la diagonal. Durante la solución de un vector simple (b) el almacenamiento requerido para las columnas pico es el máximo de la suma de las longitudes del pico local presente en alguna ecuación. En la matriz 63 en la fila 3 hay tres picos ; las columnas 5,6 y 12 respectivamente. La suma es con sus respectivas longitudes 3,5 y - 12 igual a 20, ya que la eliminación es fila x fila, esto no permite ver -- las columnas pico 9 y 11 hasta que se llegue a la fila 7. Para el tiempo - en que el pivoteo alcanza la fila 7, las filas 4 y 6 han sido eliminadas inversamente y sus lugares de almacenamiento pueden ser sobreescritos con las columnas pico 9 y 11. La suma de las longitudes de los picos locales en - la ecuación 7 es 20. El máximo almacenamiento requerido es 20, en vez de los 28 que se requerirían si la inversa fuerá retenida. Como se mostro anteriormente, la reducción de almacenamiento es más substancial para matrices de - diseño de procesos grandes que tienen una considerable estructura jerárquica de bloques.

En la aplicación de CBS el pivoteo esta dado sólo para mantener la estabilidad numérica y evitar pivotes cero. Lin y Mah (1977) han previsto que - cuando un algoritmo de ordenación HP es aplicado, todos los elementos cero de la diagonal se harán no-cero o de otro modo es numéricamente singular y no puede ser resuelta. En su trabajo ellos parecen no requerir pivoteo para mantener la estabilidad, en realidad un mal pivote puede causar una solución inadecuada. Para evitar tales problemas se puede emplear "pivoteo inicial". Después la fila pivote es eliminada adelante (reduciendo los elementos a - la izquierda de la diagonal a cero), la fila es buscada por el mayor valor absoluto. Los no-ceros en la fila son colocados sobre la diagonal y en columnas pico de la que no ha sido aún tomado un pivote. Si el valor absoluto - del elemento diagonal no es mayor que la tolerancia del pivote en vez del - valor absoluto mayor, ocurre un cambio de columna.

La columna diagonal es entonces cambiada con una columna pico cercana a la diagonal en el cual el elemento de la columna satisface el criterio inicial.

b) SUBSTITUCION INVERSA IMPLICITA "RANKI"

Ranki calcula sucesivamente la variable diagonal de cada fila spike co-

mo una función de la variable pico en esa fila, remplaza la fila pico original con la nueva fila calculada, reordena la matriz tal que haya una columna pico menos, y repite los cálculos con la siguiente fila pico. Al final de la fase de este cálculo la forma pico de la matriz ha sido transformada hacia una forma triangular inferior.

El procedimiento es demostrado con la matriz :

	1	2	3	4	5	6	7	8	9	T	
1	X		X								b
2	X	X	X								b
3		X	X								b
4			X	X					X		b
5		X		X	X				X		b
6	X					X	X		X		b
7					X	X	X				b
8		X	X		X		X	X			b
9	X		X	X			X	X			b
T		X		X			X	X	X	X	b

M-64

El vector (b) es incluido para propositos de ilustración. El algoritmo inicia calculando la variable 3 como una función de la variable pico en la fila pico 3. Ya que no hay otra columna pico en la fila 3, la variable 3 es calculada directamente. Esto es para reducir los elementos a la izquierda de la diagonal en la fila 3 a cero. La eliminación procede iniciando con la fila 2 usando el elemento (2,2) como pivote para eliminar el elemento en la posición (3,2).

	1	2	3	4	5	6	7	8	9	T	
1	X		X								b
2	X	X	X								b
3	T	0	X								b
4			X	X					X		b
5		X		X	X				X		b
6	X					X	X		X		b
7					X	X	X				b
8		X	X		X		X	X			b
9			X	X			X	X			b
T		X		X			X	X	X	X	

M-65

Esto causa temporal fill-in en la posición (3,1). Las X's indican los -- elementos que han sido cambiados de sus elementos correspondientes en la matriz original. El fill-in temporal es eliminado usando la fila 1, con (1,1) como pivote :

	1	2	3	4	5	6	7	8	9	T	
1	X		X								b
2	X	X	X								b
3	0		X'						X		b
4			X	X					X		b
5		X		X	X				X		b
6	X					X	X				b
7					X	X	X				b
8		X	X		X		X	X			b
9	X		X	X			X	X			b
T		X		X			X	X	X	X	b

M-66

La eliminación procede en sentido inverso así que todos los calculos a la fila pico tres, excepto para esta fila, todos los elementos en la matriz no -- cambian de su valor inicial. La nueva fila tres calculada en efecto reemplaza la fila original tres y los calculos inician sobre esta, designando la posición de la fila del mayor no-cero en la columna pico como ITOP, la posición corriente (1,1), es simplemente movida a la posición (ITOP,ITOP). El elemento diagonal (3,3) es por tanto movido a la posición (1,1) por el cambio de la apropiada y - los elementos diagonales interviniendo son movidos una posición bajo la diagonal;

	1	2	3	4	5	6	7	8	9	T	
1	X'										b'
2	X	X									b
3	X	X	X								b
4	X			X					X		b
5			X	X	X				X		b
6		X				X	X		X		b
7					X	X	X				b
8	X		X		X		X	X			b
9	X	X		X			X	X			b
T			X	X			X	X	X	X	b

M-67

Esto tiene el efecto de reducir el número de picos por uno como se muestra en (M- 68) . El procedimiento continua con la fila pico 7, en este caso la variable 7 es calculada como una función de la variable 9. Como antes de proceder la eliminación en un sentido inverso iniciando con la fila 6 y procediendo a la fila 3.

	3	1	2	4	5	6	7	8	9	T	
3	X'										b'
1	X	X									b
2	X	X	X								b
4	X			X					X		b
5			X	X	X				X		b
6		X				X	X		X		b
7	0	0	0	0	0	0	X'		F'		b
8	X		X		X		X	X			b
9	X	X		X			X	X			b
T			X	X			X	X	X	X	b

M-68

De nuevo, llevando la eliminación fuera de esta forma de restricciones todos los cálculos se hacen a la fila 7 pico, todos los otros elementos restantes son mantenidos. La diagonal (7,7) es movida a la posición (6,6) y (6,6) es movida a una posición abajo.

	3	1	2	4	5	7	6	8	9	T	
3	X'										b'
1	X	X									b
2	X	X	X								b
4	X			X					X		b
5			X	X	X				X		b
7						X'			F'		b'
6		X				X	X		X		b
8	X		X		X	X		X			b
9	X	X		X		X		X			b
T			X	X		X		X	X	X	b

M-69.

Finalmente el procedimiento elimina la fila 9, calculando directamente la variable 9. La diagonal (9,9) es movida a (4,4) :

	1	2	3	4	5	6	7	8	9	T	
1	X		X								b
2	X	X	X								b
3		X	X'								b
4			X	X					X		b
5		X		X	X				X		b
6	X					X	X		X		b
7					X	X	X'		F'		b
8		X	X		X		X	X			b
9	X		X	X			X	X	F'		b
T		X		X			X	X	X	X	b

M-70

En este punto se ha llevado a la forma triangular inferior la matriz. - La solución es obtenida por eliminación adelante, la forma no explícita de - la inversa que resulta de aplicar RANKI es difícil describir en una forma -- convencional factorizada. Esto es más fácil si se considera en forma bordea da bloquetriangular.

El fill-in para el RANKI inverso potencialmente ocurre a las intersec-- ciones de las filas pico con las columnas pico locales. Esto cuantifica - una considerable reducción en fill-in potencial cuando se compara con CBS o descomposición LU.

El pivoteo para mantener la estabilidad numérica en el sentido conven-- cional es difícil de aplicar en el empleo de RANKI. Si un elemento diagonal falla el criterio inicial, su fila y columna correspondiente son designadas y tratadas como fila y columna pico.

c) MA-28

Emplea una aproximación en un sólo paso, usando descomposición estandar LU en la fase numérica y el criterio de Markowitz en la fase de ordenación. En esta implementación MA-28 intenta colocar la matriz en la forma de bloque triangular y entonces procesa cada bloque irreducible separadamente.

El arreglo de almacenamiento requerido por MA-28 consiste de arreglo -- REAL de longitud LINC + N y arreglos INTEGER de longitud LIRN + LINC + 13N, donde N es el orden de la matriz. LINC deberá ser al menos igual al número de no-ceros en la eliminación de la forma inversa. LIRN deberá ser igual al menos al número de no-ceros en la parte activa de la matriz, cualquiera que sea mayor. La parte activa de la matriz es esa parte del bloque irredu cible que hasta ahora será descompuesto. Compilando con CDCFTN4.80TP=2 com pilador el conjunto entero de subrutinas es de 3981 palabras de longitud.

NSPIV 3

Emplea una eliminación Gaussiana-fila con pivoteo parcial. El pivoteo parcial consiste en seleccionar como pivote el elemento en la fila corrien- te con el mayor valor absoluto. NSPIV está hecho para resolver un vector -

simple b y no retiene la inversa. Así sólo el factor U en la descomposición Lu necesita ser retenida durante el cálculo, ademas Sherman provee una subrutina simple de ordenación a priori, PREORD, el sugiere que el usuario provea su conocimiento preordenado en la matriz si es posible.

El arreglo de almacenamiento requerido consiste de REAL de longitud 3N + NZ + NU y arreglo INTEGER de longitud 6N + NZ +NU donde: ZN es el número de no ceros en la matriz original y NU es el número de no-ceros en la parte de U de la matriz. Compilando con el compilador CDCFTN4.80PT=2, la subrutina - NSPIV es de 337 palabras de longitud.

d) LUISOL

Es una modificación de NSPIV diseñado para explorar la reordenación i-Bo-TF provisto de pivoteo inicial. El pivoteo inicial elige como pivote el elemento pivote que esta cercana a la diagonal y cuyo valor absoluto excede a - la vez la tolerancia pivote, el mayor valor absoluto en la fila.

En otros aspectos LUISOL es el mismo que NSPIV, el arreglo requerido es - REAL de longitud total 3N + NZ + NU y arreglo INTEGER de longitud 6N + NZ + NU, compilado con el compilador CDCFTN4.80PT=2. La subrutina LUISOL tiene -- 444 palabras de longitud.

e) LUIOUT

Esta modificación de LUISOL toma la matriz original sobre el almacenamiento apoyado y procesando una fila a la vez. Esto provee más almacenaje en esencia para para el factor U.

El arreglo de almacenamiento requerido consiste de REAL de longitud 3N + NU y arreglo INTEGER de longitud SN + NU . Compilado con FTNOPT=2. Esta subrutina tiene 442 palabras de longitud.

CBS

Este código es un implemento del algoritmo CBS descrito anteriormente. Esta hecho para la solución de un vector b simple y no retiene la inversa. El arreglo de almacenamiento requerido es : REAL 2N + NZ + NU y INTEGER 6N + NZ. Puesto que CBS no calcula un factor U en el sentido convencional, NU hace sim

ple referencia al espacio que debe ser colocado para almacenar los no-ceros sobre la diagonal. Compilado con CDCFTN4.80PT=2 . La subrutina tiene 659 palabras de longitud.

f) CBSOUT

Es una modificación de CBS sobre el almacenaje inverso y procesa una fila a la vez. Como en el caso de LUIOUT, esto provee mas espacio en esencia para almacenamiento de no-ceros sobre la diagonal. El arreglo de almacenamiento requerido es: REAL de longitud 2N +NU, INTEGER 5N donde NU es el mismo que en el caso de CBS. Compilado con CDCFTN4.80PT=2 . La subrutina tiene 621 palabras de longitud.

g) RANKI

Este código implementa el algoritmo RANKI, la versión esta hecha para la solución de un vector simple, el código retiene la información de la inversa que así puede ser fácilmente adoptada a la solución de múltiples vectores-b. El arreglo de almacenamiento requerido es: REAL de longitud 3N +NZ +NU y INTEGER de longitud 4N + 2N + NV. Donde NU tiene la misma intención que el caso de el código CBS. Compilado con CDCFTN4.80PT=2. La subrutina tiene 363 palabras de longitud.

BIBLIOGRAFIA

1.-Abbott,J.P. "Algoritm 110,Computer Journal," 23 pp.85-89 (1980)

2.-Barnes,J.G.P.,"An Algorithm for Solving Nonlinear Equations Based On the Secant Method",Computer Journal,V-8,p 65 (1965)

3.-Barchers,D.,"Optimal Convergence of Complex Recycle Process System" Ph. D. Thesis in Chemical Engineering,Oregon State University (1975)

4.-Benjamin D.R.,M.H. Locke,and A.W.Westerberg,"Interactive Programs for Process Design",Pres.At AICHE Meeting Detroit (1980)

5.-Brannock N.F.,"Process^{sm} Simulation Program",Computers & Chemical Engineering 3 pp529-352 (1979)

6.-Bridell,Talbolt,E."Process Desing by Computer",Chemical Engineering,Feb. 4,March 4,April (1974)

7.-Broyden C.G.,J.E. Donnis & Jorge J. More "On de Local and Super Linear Convergence of Quasi-Newton Methods",J.Inst.,Maths. Applies (1973) 12,pp.223-245

8.-Broyden C.G. Math. Comput. 21,pp368 (1967)

9.-Cavett,R.H.,"Applications of Numerical Methods to the Convergence of Simulated Processes Involving Recycle Loops",American Petroleum Institute,n-04-63(1963)

10.-Chen,H.S. y M.A. Stadtherr, "A Modification of Powell's Dodleg Method for Solving Systems of Nonlinear Equations",Computers and Chem. Eng. V.5 p.143 (1981)

11.-Davidenko,D.I.,"On a New Method of Numerical Solution of Systems of Nonlinear Equations",Doklady Akad,Nauk.SSSR,(NS),V.88,p.601-602 (1953)

12.-Dennis J.R.,& Moré,J."Quasi-Newtons Methods,Motivation and Theory",SIAM - Review Vol. 19 No. 1 (1977) pp.96-98

13.-Dennis,J.R.,"On the Convergence of Broyden's Method for nonlinear Systems of Equations",Math. Comp. Vol. 25 No. 115 pp.559-567 (1971)

14.-Davidon W.C.,"Variable Metric Method for Minimization",A.E.C.,Research and Development Report,ANL-5990 (Rev. TID-4500,14th.ed.) (1959)

15.-Evans L.B.,"ASPEN: An Advanced System for Process", Computers & Chemical - Eng. 3 pp. 319-327 (1979)

16.-Evans L.B.,Seider W.D.,"The Requeriments of an Advanced Computing System" Chemical Eng. Progress pp. 80,June (1976)

17.-Evans L.B.,Steward D.G.,Sprague C.R.,"Computer Aided Chemical Process Design",Chemical Engineering Progress 64 No. 4 pp. 39-46 (1968)

18.-Franks R.G.,"The Evolution of Digital Simulation Programs",Chem. Eng. Progress, Vol. 63 No. 4 (1967) pp. 68-78

19.- Flower J.R.,Witchead B.D.,"Computer -Aided Design: A Survey of Flowsheeting Programs", Chem. Eng. pp. 208 April (1973)

20.-Gjumbir M. & Zarko. O.,"Effective ways to salve single nonlinear Equations" Chem. Eng. (1984),pp. 51-56

21.-Gorczynski,E.W. y H.P.Hutchinson,"Towards a Quasilinear Process Simulator: I.-Fundamental Ideas",Comp. Chem. Eng. V.2 pp. 169 (1978)

22.-Gross H.,Kaijaluoto,S. Mattson,"Some New Aspect on Partition and Tearing in Steady state Process Simulation in Computer Applications in the Analysis of Chemical Data and Plants",Science Press,Princeton(1979)

23.-Grosman y del Rosal "Simulador general de procesos IMP",Instituto Mexicano de Ingenieros Quimicos 1975 pp. 32-36

24.-Gustavson F.G.,"Some Aspects of Computation With Sparce Matrices",Foundations of Computer-Aided Chem. Process Design,Vol. 1e,Engineering Foundation N.Y. (1981) pp 78.

25.-Henley,E.J.:Williams,R.A.,"Graph Theory in Modern Engineering"Academic Press New York (1973)

26.-Hilton,C.H.,"Numerical Studies in Equation-Based Chemical Process Flowsheeting" Tesis de Doctorado,Massachusetts,Institute of Technoloty MIT(1982)

27.-Hlavacek,V.,"Analysis of a complex plant-steady state and transient behavior", Computers and Chemical Engineering 1, pp. 75-100 (1977)

28.-Hlavacek,V.,"Simulation of the Steady-State Process ",Journal Review 1,pp 81 (1978)

29.-Hlavacek,V.,Kubicek M.,Prochska F.,"Global Modular Newton-Raphson Technique for Simulation", Chemical Engineering Science,31,pp.277-284(1976)

30.-Holland C.D.,Gallum S.E."Modifications of Broyden's Methods for the Solution of Problems Destillation Involving Highly Non-ideal Solutions",Paper Presented of Houston AICHE Meeting (1979)

31.-Kehat,E & Shacham M."Chemical Process Simulation Programs",Process Technology International Vol. 18,No. 1/2,3 and 4/5 (1973)

32.-Kliesch,H.C.,"An Analysis of Steady-State Process Simulation:Formulation and Convergence", Ph.D. Thesis,Tulane,University,New Orleans,La. (1967)

33.-Kloptestein,R.W.,"Zeros of Nonlinear Functions",J. Assoc. Comput. Mach.& pp. 366-373(1961)

34.-Kluzick H.A.,"A Study of the Simultaneous Modular Convergence of chemical Process Flowsheets",Tesis de Maestria,Massachusetts Institute of Technology",ACM Trans. on Math. Software V.2,p. 98 (1976)

35.-Kubicek M. "Algorithm 502.- Dependence of Solution of Non-linear - Systems on a Parameter", ACM trans. on Tath software, V.2 p.98[1976].

36.-Lam B. "On the convergence of a Quasi-Newton for sparse nonlinear - Systems", Math. Comp., 32 p.447-451 [1978].

37.-Levenverg K.-"A method for the solution of certain nonlinear problems in least squares", Quart Appl. Math., 2 p. 164-168 [1944].

38.-Lin T.D. y R.S.H. mah. "Hierarchical Partition.- A new optimal Pivoting Algotithm", Math. Programming, V.12 p. 260 [1977].

39.-Mahalec V.H. Kluzik y L. B. Evans.- "Simultaneous Modular Algorithm for Steady State Flowsheet Simulation and Design", 12 th. Eur. Sym. Computers in Chem. Eng. Montreux, Suiza [1979].

40.-Marquardt, D.W.-"An algorithm for least squares estimation of non-linear parameters", J.Soc. Indust, Appl. Math., 11, p. -- 431-441.

41.-Marwil Earl .-"Convergence results for Schuberts Method for solving sparce nonlinear equations", SIAM J. Numerical Analysis V. 16, No. 4, p. 588-604, [1979].

42.-Matulsky M.-"Chemshare Process Simulation Programs", Computer Aided Process Plant Design, p. 489 [ ].

43.-Metcalfe S.R. & J.D. Perkins.- Trans. Inst. Chem. Eng. 56 p. 210 -- [1978].

44.-More J.J. y M.Y. Consnard.- "Numerical Solution of nonlinear Equation , ACM Trans. Math. Software, V.5, p. 65 [1978].

45.-Motard R.L., N. Shacham y E.M. Rosen.- "Steady State Chemical Proce ss Simulation", AICHE J., V.21, p. 417 [1975].

46.-Naphtali L.M.- "Process heat and Material Balances", Chem. Eng. Prog. V.60, p. 9 [1964].

47.-Nishio M. , M. Eng thesis, M.C. Master University, Hamilton , Ontario [1973].

48.-Nishimura H. Y. Hiraizumi y S. Yagi, " kagaku KogaKu", V.31, p. 183, [1967].

49.-Perkins J.D. & R.W.H. Sargent.- "Speed-up: A computer program for -- steady State and Dynamic Simulation and Design of Chemical Process", presented at AICHE fall Meeting, New Orleans. p. 8-12 [1982].

50.-Orbach O. & Crowe C.M. "Convergence Promotion in the simulation of chemical process with recycle the dominant eigenvalue method", Canadian Journal of Chem. Eng. V.49, p. 509-513 , - [1971].

51.-Powell M.J.D.- " A hybrid method for nonlinear equation", Numerical methods for nonlinear algebraic equation, por P. Robinowits e. Gordon & Branch, N. York [1980].

52.-Rheinboldt W.C. and J.V. Burkardt.-"Algorithm 596.- A program for a locally parameterized continuation process", ACM. Trans. on Math Software 9, No. 2 p. 236 [1983].

53.-Rosen E.M.- "A machine computation method for performing material balances", Chem. Eng. Prog., V. 58, p. 10 (1962).

54.-Rosen E.M.- "Steady state chemical process simulation state of the art review", Computer aplications to Chemical Eng., HCS Symposium Series, V. 124, p. 3 (1980).

55.-Rosen E.M. Pauls A.C.- "Computers aided chemical process design: - The FLOWTRAN System", Comp. Chem. Eng. V.1, p. 11 (1977).

56.-Sargent R.W.H.- "A review of methods for solving nonlinear algebraic equations", Foundations of Computers-aided chemical Process Design, V.1, e Eng. Foundations (1981).

57.-Sargent R.W.H. and A.N. Westerberg Trans. Inst. Chem Eng. 42, 190, No. 179, p. 190 (1964).

58.-Seader J.D. - "Computer Modeling of chemical process", AICHE Monograph Series, V.81, No. 15 (1985).

59.-Seider W. Evans L.B. Joseph Wong E., Jirapongphan - "Routing of -- calculations in process simulation", Ind. Eng. Chem. process Des. Dev. 18, No. 2, p. 292 (1979).

60.-Schubert L.K.-" Modification of a Quisi-Newton method for nonlinear equations with a sparse jacobian", Math. Comput., V.25 -- p. 27 (1970).

61.-Shacham M.-" Recent developments in solutions techniques for systems of nonlinear equations", Proceding of the second international conference on foundations of computer aided process design", CACHE publications (1985-b).

62.-Shacham M.-"Comparing software for the solution of systems of nonlicear equation arising in chemical eng.", Comp. & Chem. Eng. submitted (1982-b).

63.-Shacham M. and R.L. Motard.- "Applications of the theory of linear recycle systems", paper presented on 75th. National Mtg.Am. Inst. Chem. Eng. Salt Lake Chey (1974).

64.-Soliman M.A.- "Quasi-Newton methods for convergence acceleration of cyclic systems", Can. J. Chem. Eng., V.57, p.643 (1979).

65.-Soliman M.A.- "A new update for the solution of nonlinear algebraic equations", Comp. Chem. Eng. V. 9, p. 407 (1985).

66.-Stadtherr M.A. y C.H. Hilton.-" Development of a new equation based process flowsheeting systems: Numerical Studies", Selectd topics on Computer-aided process Design and analysis, AICHE Symposium Series, V. 78, p. 12 (1982).

67.-Stadtherr M.A. y C.H. hilton.-" On efficent solution of large scale Newton-Raphson based flowsheeting problems in limited core" Comp. Chem. Eng. , V. 6, p. 115 (1982).

68.-Tarjan R.-"Depth-First Search in linear graph algorithm", SIAM J. -- Comput., V. 1 , p. 146 (1972)

69.-Umeda T. y M. Nishio.-"Comparasion between Sequential and Simultaneous aproaches in process simulation", Ind. Eng. Chem. Process Design, V 11, p. 153 (1972).

70.-Updhye R.S. y E.A. Grens.- "Selection of descomposition for chemical process simulation", AICHE Journal, V. 21, p. 136 (1975).

71.-Watson L.T. and Denner.-"Algorithm 555.- Chow-yorke algorithm for -- fixed point or zeros of C. maps.", ACM Trans. on Math. soft ware 6, No.2 , p. 252 (1980).

72.-Wayburn T.L. and L.D. Seader.-" Solution of systems of internaked distillation columns by differential homotopy continuation methods." Proceding of the second international conference on foundanors of computers-aided process design" Westerberg A.W. and H.H. Chien editors , CHACHE, Ann Arbor (1984).

73.-Wegstein, J.H..-"Accelerating convergence of iterative process", Comunication association computing machines, V.1, p.9 (1958).

74.-Westerberg A.W., H.P. Hutchinson, R,L. Motard y P. Winter.-" Process flowsheeting", c. Cambridge University Press, Cambridge(1979).

75.-Williams G. " Software arts TK' solver", Byte 7, 85-92 (oct-1982).

76.-Wood E.S.-"Two pass strategies for sparse matrix computations in Chemical process flowsheeting problem", Tesis de doctorado, University of Illinois (1982).

77.-Worley F.L. , Motard R.L.-" Information system in chemical eng. Design" Process analysis and design p. 4 ( ).

78.-Yudth V.S. Eakman J.M.-" Identification of process flow networks". Process analysis design p. 40 ( ).

79.-Zepeda R. Cano J. Rosal R.-" Simulador general de procesos IMP", IMIQ p. 32 ( ).