



**UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO**

**Escuela Nacional de Estudios Profesionales**

**\* ZARAGOZA \***

**SIMULACION DE PROCESOS A  
ESTADO ESTACIONARIO**

**T E S I S**

QUE PARA OBTENER EL TITULO DE:

**INGENIERO QUIMICO**

P R E S E N T A

**J. MARIANO RAMOS OLMOS**

**MEXICO, D. F.**

**1984**



Universidad Nacional  
Autónoma de México



**UNAM – Dirección General de Bibliotecas**  
**Tesis Digitales**  
**Restricciones de uso**

**DERECHOS RESERVADOS ©**  
**PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

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

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

INDICE GENERAL:PAGINA:

CAPITULO I	Introducción General .....	1
CAPITULO II	Generalidades .....	4
2.1	Introducción .....	5
2.2	Modelos Matemáticos .....	11
2.3	Algoritmos .....	19
2.3.1	Simulación secuencial modular .....	19
2.3.2	Métodos de ecuación-orientada .....	21
2.3.3	Simulación por métodos lineales .....	23
2.3.4	Simulación cuasi-lineal .....	25
2.3.5	Simuladores simultáneos modulares ...	25
2.4	Software .....	26
2.4.1	Aspectos de organización en el desa- rrollo del software .....	30
2.4.2	Aspectos históricos de la simulación- de procesos a estado estacionario ...	33
2.5	Interfase con el usuario .....	36
CAPITULO III	Identificación de procesos .....	38
3.1	Introducción .....	39
3.2	Identificación .....	41
3.2.1	Método de Kehat y Shacham .....	43
3.2.2	Método de Jain y Eakman .....	49
3.2.3	Método de Westerberg .....	65
3.3	Rompimiento de ciclos .....	69
3.4	Cálculos de convergencia .....	79
3.4.1	Métodos de linearización directa ....	82
3.4.2	Métodos indirectos .....	87
3.4.2.1	Substitución sucesiva .....	87
3.4.2.2	Método de Wegstein .....	89
3.4.2.3	Método de los valores pro- pios dominantes .....	92

INDICE GENERAL - (Continuación)

PAGINA:

	3.4.2.4	Método generalizado de la- secante .....	95
	3.4.2.5	Método de Broyden .....	96
CAPITULO	IV	Estructura del programa .....	102
	4.1	Estructura del programa .....	103
	4.2	Descripción del programa .....	103
	4.2.1	Programa principal LOOPS .....	103
	4.2.2	Subrutina FLOOPS .....	104
	4.2.3	Subrutina LOOPCH .....	105
	4.2.4	Subrutina ATEAR .....	106
	4.2.4	Subrutina ATFARI .....	106
CAPITULO	V	Resultados .....	109
	5.1	Ejemplo No. 1 .....	110
	5.2	Ejemplo No. 2 .....	113
	5.3	Ejemplo No. 3 .....	116
	5.4	Ejemplo No. 4 .....	119
	5.5	Problema de Cavett .....	123
CAPITULO	VI	Conclusiones .....	141
BIBLIOGRAGIA		.....	144

## INDICE DE FIGURAS

FIG. No	TITULO	Página
2.1	Etapas que intervienen en el desarrollo de un nuevo proceso .....	7
2.2	Estructura de las actividades del diseño de nuevos procesos .....	8
2.3	Construcción de bloques de un sistema computacional para la ayuda en el análisis.....	9
2.4	Estructura esquemática de un modelo matemático de una unidad de proceso.....	15
2.5	Posibles especificaciones para un modelo de intercambiador de calor a contra corriente.....	17
2.6	Diagrama de flujo para ilustrar el analisis de la estructura computacional.....	22
2.7	Algoritmo de doble rompimiento.....	27
2.8	Comparación del procedimiento de solución vs.: generalidad de los modelos de los algoritmos usados en los simuladores de proceso .....	29
2.9	Flujo de información de un sistema avanzado de simulación .....	32
3.1	Diagrama de flujo de un proceso con recirculación .....	40
3.2	Gráfica dirigida de un proceso.....	45
3.3	Problema de Cavett .....	53
3.4	Diagrama de flujo para la multiplicación de matrices .....	56
3.5	Aplicación de la estructura de arboles en la detección de ciclos .....	59
3.6	Rastreo de las trayectorias de la estructura de arbol para la identificación de un ciclo máximo .....	64

INDICE DE FIGURAS (Continuación)

FIG. No	TITULO	Página
3.7	Algoritmo de Westerberg (1977) para la identificación de ciclos .....	70
3.8	Algoritmo de Mahalec (1978) para detectar un conjunto de corte .....	80
3.9	Linearización de una función de una sola variable .....	84
3.10	Métodos indirectos .....	88
3.11	Ortogonalización de $V^{(2)}$ con respecto a $\Delta x^{(1)}$ a través de $\Delta x^{(2)}$ .....	98
5.1	Ejemplo No 1 .....	111
5.2	Ejemplo No 2 .....	114
5.3	Ejemplo No 3 .....	117
5.4	Ejemplo No 4 .....	120

## INDICE DE TABLAS

TABLA No.	TITULO	Página
2.1	Clasificación de los sistemas de simulación de procesos, de acuerdo al tipo de algoritmo de solución que usan.....	28
3.1	Matrices de adyacencia y de índices.....	46
3.2	Eliminación de columnas y su respectiva matriz de índices .....	47
3.3	Eliminación de hileras y su respectiva matriz de índices .....	48
3.4	Identificación de ciclos por medio del método de Kehat y Schacham.....	50
3.5	Matriz adyacente y vectores de corrientes e identificación .....	54
3.6	Propiedades del parámetro de aceleración $q$ .....	91
5.1	Comparación del orden de cálculo y de los conjuntos de corte para las recirculaciones que presenta el problema de Cavett cuando existen corrientes preferidas de corte .....	140

# CAPITULO I

## INTRODUCCION



## I. INTRODUCCION GENERAL:

En el presente trabajo serán estudiados los elementos básicos que constituyen a un simulador de procesos a estado estacionario así como los problemas asociados a una simulación de procesos químicos.

Como parte del análisis de los componentes de un programa de cómputo para la simulación de procesos, se estudiarán aspectos tales como:

- 1) Características y necesidades de los modelos matemáticos que se emplean en una simulación.
- 2) Principios, ventajas y desventajas de los diferentes algoritmos utilizados para resolver el problema matemático generado por la simulación de un proceso.
- 3) Requerimientos y características necesarias para la creación de nuevos simuladores.

Por otra parte, será tratado el problema matemático generado durante la simulación de un proceso; será analizada la estructura de un proceso para determinar las recirculaciones existentes y la secuencia de cálculo de las unidades de proceso en las recirculaciones.

Los objetivos que se persiguen son; hacer un análisis y descripción de:

- a) El estado actual de la simulación de procesos químicos a estado estacionario.
- b) Las técnicas aplicadas a la identificación y rompimiento de ciclos.
- c) Los métodos de convergencia utilizados para resolver los problemas matemáticos generados por el rompimiento de los ciclos.

Como complemento de la investigación bibliográfica, se apli-

can algunas de las técnicas más avanzadas en identificación y rompimiento de ciclos a la solución de problemas ilustrativos.

En el capítulo II, se describe el estado actual de la simulación de procesos en función de los componentes principales de los simuladores de procesos; dentro de estos componentes destacan:

- a) Modelos matemáticos.
- b) Algoritmos para la simulación.
- c) Software\* computacional.
- d) Interfase con el usuario.

La descripción de cada uno de estos componentes se presenta lo más detallada posible, con el objeto de que las personas que consulten el presente trabajo puedan visualizar el alcance de cada uno de ellos.

En el capítulo III se describen las técnicas de simulación - aplicadas a la identificación y rompimiento de ciclos y los métodos de convergencia. Se hace un énfasis especial en los métodos que serán seleccionados para utilizarlos en la construcción del programa computacional.

En el capítulo IV, se describe la estructura del programa -- computacional que se construyó para determinar el número de recirculaciones, los conjuntos de corrientes de corte y el orden de -- cálculo de las recirculaciones.

En el capítulo V, se presentan los resultados obtenidos al - aplicar el programa construido a la solución de problemas de inte - rés.

Por último, en el capítulo VI se presentan las conclusiones - obtenidas en el desarrollo del presente trabajo.

(\*) Software no tiene traducción al español.

## CAPITULO II

### GENERALIDADES

## 2.1 INTRODUCCION:

La ayuda computacional en el análisis y diseño de los procesos químicos, surge como una gran disciplina de la Ingeniería Química. El alcance de la Ingeniería Química es muy amplio, por lo que aquí, la función de la misma será definida como; "Análisis, diseño, construcción y operación de plantas de procesos". Para complementar esta función, el Ingeniero Químico, debe poder cuantificar y evaluar los efectos de las leyes físicas y químicas gobernantes de los procesos, así como también debe relacionarse la economía de los mismos.

La habilidad del Ingeniero para crear procesos más eficientes se ve limitada por el esfuerzo tan grande requerido para evaluar las diferentes alternativas de un proceso.

Es por esto, que se recomienda hacer uso de un sistema de cálculo efectivo para el análisis de procesos que permita a los ingenieros evaluar más configuraciones de proceso de las que actualmente son posibles.

En las figuras 2.1 y 2.2 se ilustran las etapas y actividades involucradas en el diseño de nuevos procesos. Los bloques -- marcados con un asterisco (\*), representan las etapas en las cuales puede asistir la computadora. La computadora sirve como una herramienta que facilita al diseñador refinar su entendimiento de un proceso. Los diferentes tipos de análisis mostrados en la figura 2.1 pueden ser asistidos por un sistema de cálculo comprensivo que permita una interacción cerrada entre el usuario y la computadora. Una de estas etapas es la simulación de procesos a estado estacionario que es el tema central en el presente trabajo.

Motard, et.al. (1975), define a la simulación de procesos químicos como: "La representación de un procesos químico mediante modelos matemáticos que son resueltos para obtener información

acerca del funcionamiento del mismo".

Escencialmente, un sistema de simulación de procesos, es un programa de computadora comúnmente escrito en lenguaje de alto nivel (FORTRAN ó PL/1), con una interfase con el usuario que permite al ingeniero concentrarse en los componentes y especificaciones del proceso.

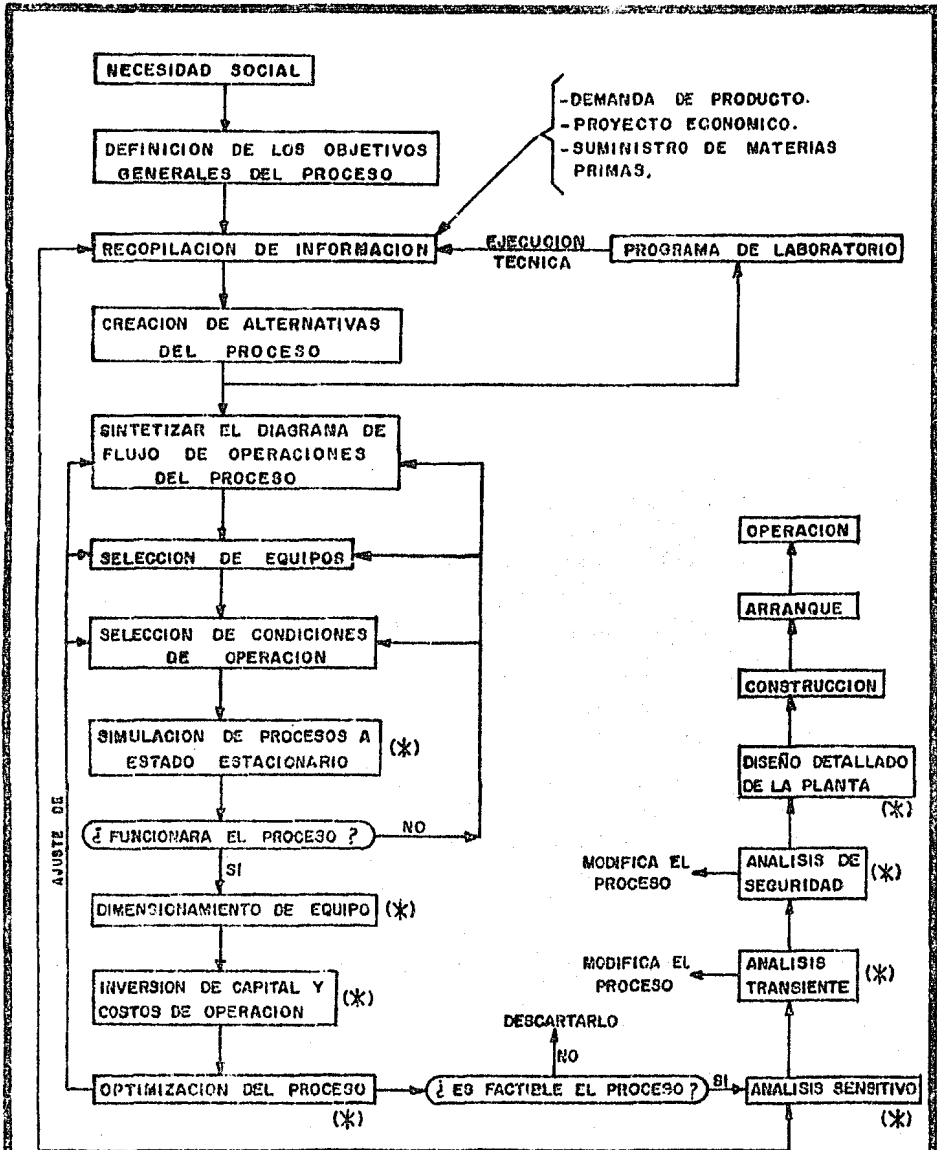
Los elementos principales de cualquier sistema de cálculo a gran escala para la ayuda en el análisis ingenieril son mostrados en la figura 2.3.

La base de cualquier sistema computacional son los Modelos, a su vez, constituyen la base del análisis. Los modelos de un proceso químico usados por un sistema de simulación, son todas las relaciones matemáticas derivadas a partir de las leyes de conservación, ecuaciones de velocidad, relaciones de propiedades físicas, termodinámicas y de transporte y de las restricciones de diseño y control.

Estos modelos matemáticos toman la forma de ecuaciones algebraicas y diferenciales que describen el proceso. La característica más importante de los modelos matemáticos es que sean lo más apropiados para su uso en términos de rigor, nivel de detalle, exactitud, valides y generalidad.

El siguiente bloque está formado por los ALGORITMOS que operan sobre los modelos para producir los resultados requeridos. Estos algoritmos resuelven el problema matemático generado por los modelos; el tipo de problema matemático que debe ser resuelto depende del tipo de análisis que interese pero, normalmente incluye la solución de ecuaciones algebraicas, diferenciales y programación no lineal.

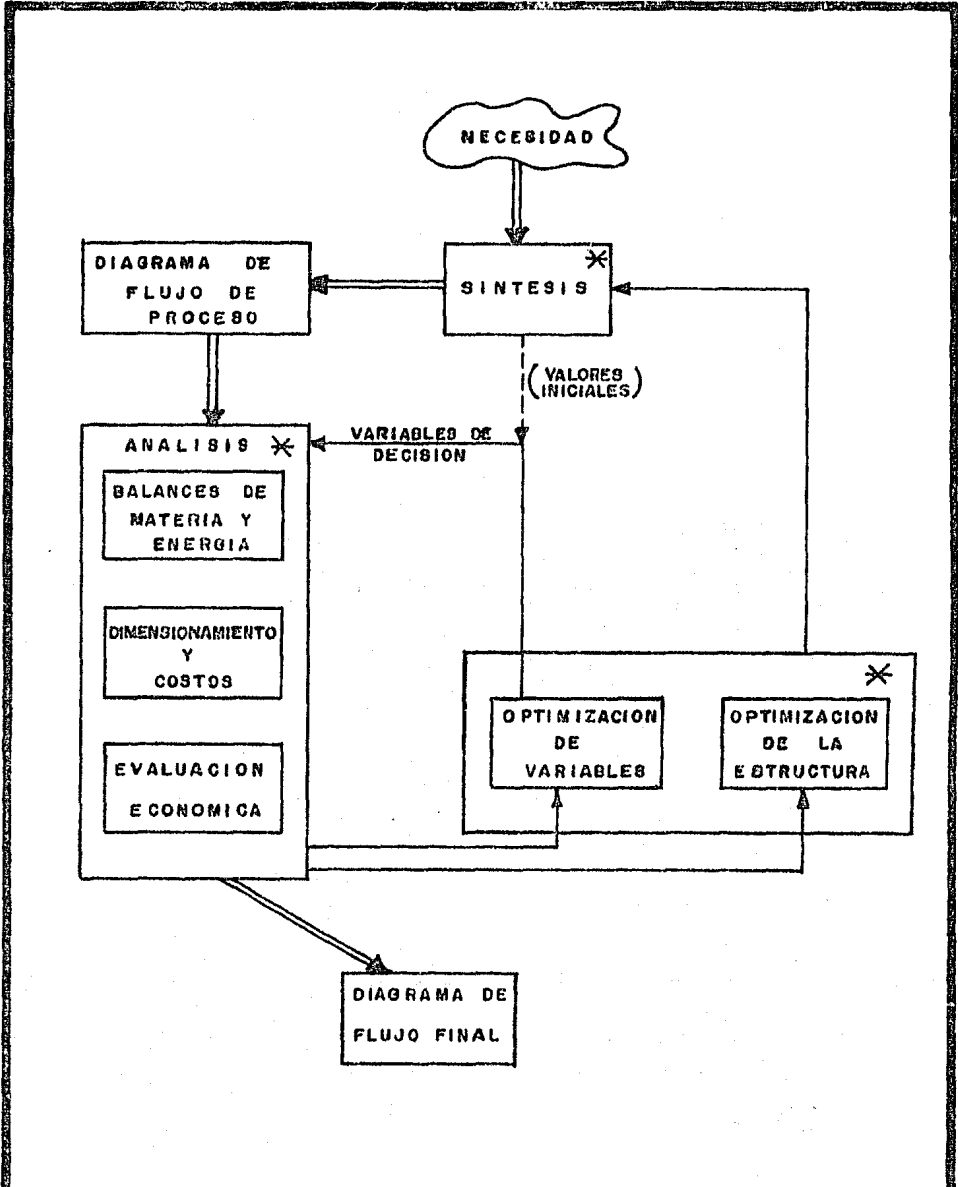
Los requerimientos de un buen algoritmo son: debe ser lo más general que sea posible, lo más eficiente en términos de ejecu



- DEMANDA DE PRODUCTO.
- PROYECTO ECONOMICO.
- SUMINISTRO DE MATERIAS PRIMAS.

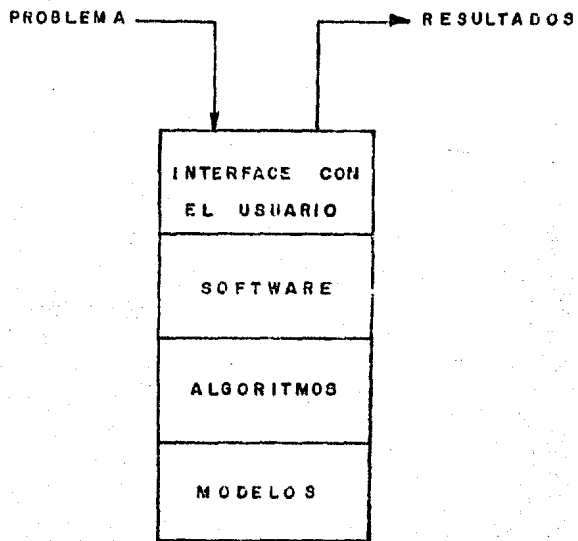
TITULO: FIGURA 2.1  
 ETAPAS QUE INTERVIENEN EN EL  
 DESARROLLO DE UN NUEVO PROCESO

TESIS PROFESIONAL  
 JOSE MARIANO RAMOS OLMOS  
 E.N.E.P. ZARAGOZA U.N.A.M.



TITULO:  
 FIG. 2.2 ESTRUCTURA DE LAS ACTIVIDADES DEL DISEÑO DE NUEVOS PROCESOS.

TESIS PROFESIONAL  
 JOSE MARIANO RAMOS OLMOS  
 E.N.E.P. ZARAGOZA U.N.A.M.



TITULO:

FIG. 2.3 CONSTRUCCION DE BLOQUES DE UN SISTEMA COMPUTACIONAL PARA LA AYUDA EN EL ANALISIS

TESIS PROFESIONAL

JOSE MARIANO RAMOS OLMOS

E.N.E.P. ZARAGOZA U.N.A.M.



ción y almacenamiento, claro y elegante. Frecuentemente estos - requerimientos están en conflicto y deben de hacerse compatibles.

El siguiente bloque consiste del SOFTWARE computacional. - El software incluye todas las cosas requeridas para implementar - los algoritmos en una computadora y en un sistema de operación - particular. Incluidos dentro de la categoría del software están:

- 1) La arquitectura del programa y del sistema.
- 2) Estructura de datos.
- 3) Interfase con el archivo del sistema.
- 4) Lenguaje de programación.
- 5) Documentación del programa.

Los requerimientos para un buen software son: Deben ser - fáciles de entender, fáciles de mantener y modificar y lo más -- transparentes que sea posible.

Finalmente, en lo más alto de la estructura está la Inter- fase con el usuario. La interfase con el usuario incluye el len guaje de entrada por medio del cual el usuario describe su pro- blema, los reportes que contendrán los resultados, la documenta- ción que explique al usuario cómo debe usar el sistema y los pro tocolos para que sirvan de interfase con otros programas y siste- mas. Una buena interfase con el usuario debe aceptar la entrada en la forma más natural que sea posible y debe presentar los re- sultados en la forma más legible.

Idealmente, el software y los algoritmos deben ser lo más- transparentes que sea posible para que el usuario esté interac- tuando directamente con los modelos a través de la interfase con el usuario.

Actualmente, los grandes procesos no pueden ser construidos sin haber sido simulados previamente el proceso en una computadora

mediante el uso de modelos matemáticos.

Los simuladores de procesos químicos manejan y resuelven los siguientes tipos de ecuaciones:

- 1) Ecuaciones de conservación de materia y energía.
- 2) Interconexión de corriente basada en la topología del diagrama de flujo del proceso.
- 3) Ecuaciones de operaciones específicas a cada unidad.
- 4) Ecuaciones de propiedades físicas, termodinámicas y de transporte.
- 5) En algunos casos ecuaciones de dimensionamiento y costos.

## 2.2 MODELOS MATEMATICOS:

Un modelo matemático es una colección de ecuaciones que describen algunos aspectos del comportamiento del sistema que se está investigando.

Generalmente, los modelos matemáticos nos brindan:

- 1) Bases para una teoría.
- 2) Habilidad para predecir el comportamiento de situaciones que aún no han sido estudiadas.
- 3) Bases para el diseño y/o análisis de procesos.
- 4) Bases para el desarrollo de un programa computacional.

Las etapas que se necesitan cubrir en la construcción de un modelo matemático son:

- 1) Establecimiento de los objetivos.
- 2) Selección de las ecuaciones.
- 3) Construcción del modelo.
- 4) Definición de variables.
- 5) Ajuste del modelo.

La primera etapa de la construcción de un modelo matemático es la de decidir el tipo de sistema que va a ser modelado. Los sistemas de mayor importancia en Ingeniería Química son:

- 1) Estado estacionario
- 2) Dinámico

El éxito de la aplicación del modelaje a cualquier proyecto, depende de la capacidad del mismo para anticipar las preguntas que pueden ser formuladas durante el desarrollo del proyecto, entonces decidir si el modelo propuesto es el adecuado. Los diferentes tipos de modelo se pueden resumir como a continuación se indica:

- 1) Modelos estáticos / experimentales.
- 2) Modelos conceptuales.
- 3) Modelos combinados (usados en identificación de procesos).

En esta etapa, es irrelevante la complejidad del proceso - debido a que, un proceso simple, requerirá un modelo simple y un proceso complejo requerirá un modelo lleno de aproximaciones para poder hacerlo accesible. Independientemente de la complejidad del modelo, es importante hacer notar que un modelo matemático es específico para el problema que se está investigando.

Una vez que se ha decidido la forma del modelo matemático, la siguiente etapa es la de coleccionar todas las ecuaciones matemáticas que representan al sistema con el objeto de construir el modelo. Los diferentes tipos de ecuaciones que se tienen que resolver así como sus características se enlistan a continuación:

a) TIPOS DE ECUACIONES:

- 1) Algebraicas
- 2) Diferenciales ordinarias (EDO)

## 3) Diferenciales parciales (EIP)

## b) Características de las ecuaciones:

- 1) Lineales
- 2) No lineales
  - a) Trascendentales
  - b) Discontinuas
  - c) Implícitas
  - d) Discretas
  - e) Enteras

El tipo y características de las ecuaciones depende directamente del sistema que se está estudiando.

Una vez que se ha establecido la forma de las ecuaciones - que constituyen el modelo, la siguiente etapa es combinarlas de tal manera que se puedan resolver y, de esta forma, poder obtener las respuestas requeridas por el mismo; por ejemplo, un modelo matemático de un proceso puede calcular los cambios termofísicos de las corrientes de proceso cuando éstas pasan a través de las unidades de proceso o, los parámetros físicos de las unidades de proceso basado en las condiciones termofísicas de las corrientes del mismo.

La estructura de un modelo matemático se ilustra en la figura 2.4. Las ecuaciones que constituyen el modelo están formadas por variables y constantes numéricas; generalmente los modelos generan un conjunto de relaciones algebraicas no lineales de la forma:

Variables de salida = f (variables de entrada)

Las variables de entrada son las variables de las corrientes de entrada y los parámetros de los modelos. Las variables de salida son las variables de las corrientes de salida, las variables resultantes y las variables internas o de retención. -- Las variables internas son valores intermedios no requeridos en-

el diagrama de flujo de proceso que pueden ser salvadas para ser usadas como valores iniciales en la siguiente iteración si la unidad está en una recirculación.

A su vez, las variables se clasifican en dos grupos:

- 1) Variables independientes
- 2) Variables dependientes

Es de vital importancia saber distinguir estos dos grupos de variables debido a que, de esto, depende el buen manejo y entendimiento de los modelos.

Las variables independientes son aquellas que especifican el problema y cuyos valores son dados como datos de entrada para cualquier problema y que, por lo general se clasifican en tres grupos:

- 1) Variables de corrientes de entrada
- 2) Variables del equipo
- 3) Parámetros del modelo

Las variables dependientes son aquellas cuyos valores son determinados por medio de las ecuaciones del modelo. Dentro de esta categoría están:

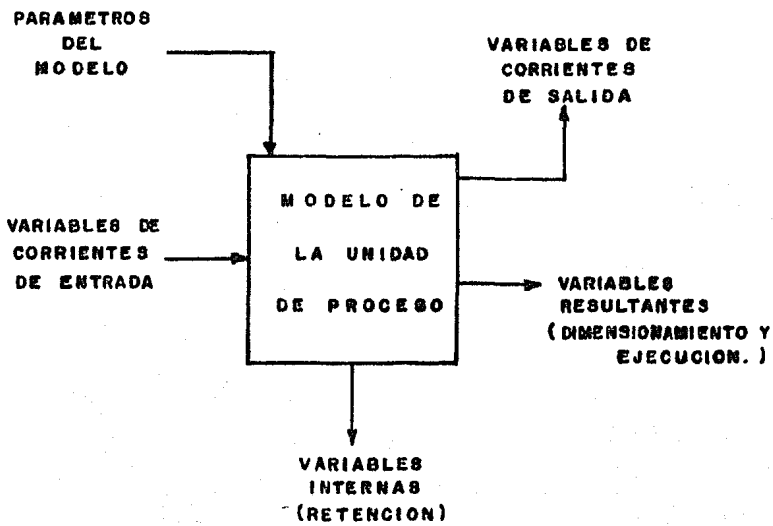
- 1) Variables de corriente de salida
- 2) Variables resultantes
- 3) Variables intermedias

Funcionalmente, un modelo de una unidad de proceso puede ser expresada como:

$$f(\underline{u}, \underline{x}, \underline{y}, \underline{z},) = 0 \quad (2.1)$$

Donde:  $\underline{u}$  = Vector de parámetros del modelo

$\underline{x}$  = Vector de variables de corriente de entrada



TITULO:  
 FIG. 2.4 ESTRUCTURA ESQUEMATICA DE UN  
 MODELO MATEMATICO DE UNA  
 UNIDAD DE PROCESO.

TESIS PROFESIONAL

JOSE MARIANO RAMOS OLMOS

E.N.E.P. ZARAGOZA U.N.A.M.

$\underline{y}$  = Vector de variables de corriente de salida

$\underline{x}$  = Vector de variables internas

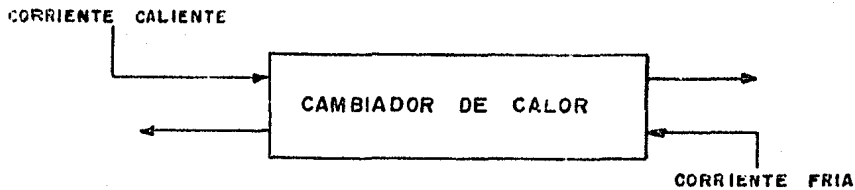
En la simulación de un proceso, el número de ecuaciones es igual a la suma del número de variables de corrientes de salida, variables internas y variables resultantes y, el número de grados de libertad, es igual a la suma de los parámetros de los modelos y de las variables de corrientes de entrada. El número total de ecuaciones es más grande que el número de grados de libertad debido a que existe una gran cantidad de variables internas.

Existe una gran variedad de posible combinaciones de parámetros de modelos que pueden ser usados para especificar una unidad de proceso. Por ejemplo, consideramos el cambiador de calor a contracorriente ilustrado en la figura 2.5. En dicha figura, se muestran tres conjuntos de especificaciones que son:

- 1) Especificar el coeficiente global de transferencia de calor y el área de intercambio. Dadas las dos corrientes de entrada, quedan totalmente determinadas las dos corrientes de salida.
- 2) Especificar el coeficiente global de transferencia de calor y el  $T$ .
- 3) Especificar el coeficiente global y la carga total  $Q$ .

Cualquiera de estas tres combinaciones puede obtener la convergencia del problema y, una vez obtenida, las variables restantes serán fáciles de calcular.

La primera opción, permite efectuar cálculos para evaluar si un cambiador de calor existente cumple o no con ciertas especificaciones, mientras que la segunda y tercera opción pueden ser utilizadas para el diseño de un equipo de intercambio de calor, ya que éstas, especifican el funcionamiento requerido y permiten determinar el tamaño del cambiador de calor una vez que se satisfacen los balances de masa y energía.



		1	2	3
COEFICIENTE	U	*	*	*
AREA INTERC.	A	*		
APROXIMACION	$AT_{FRIO}$		*	
CARGA TOTAL	Q			*

TITULO:

FIG. 2.5 POSIBLES ESPECIFICACIONES PARA UN MODELO DE UN INTERCAMBIADOR DE CALOR A CONTRA CORRIENTE.

TESIS PROFESIONAL

JOSE MARIANO RAMOS OLMOS

E.N.E.P. ZARAGOZA U.N.A.M.



El efecto de seleccionar diferentes conjuntos de parámetros de un modelo para definir el cambiador de calor se refleja en las variables de transferencia entre los parámetros del modelo y las variables resultantes. En el caso 1,  $A$  es un parámetro del modelo y  $\Delta T_m$  y  $Q$  son variables resultantes. En los casos 2 y 3  $A$  llega a ser una variable resultante y  $\Delta T_m$  o  $Q$  es un parámetro del modelo.

En general, el número total de variables en un modelo de una unidad de proceso, permanecerá igual para diferentes conjuntos de especificaciones pero, su clasificación variará.

La última etapa en la construcción de un modelo la constituye el ajuste del mismo; para lograrlo es necesario ajustar los parámetros del modelo en base a datos experimentales o reportados en la literatura.

### 2.3 ALGORITMOS:

El problema matemático que se presenta durante una simulación de procesos a estado estacionario, es el de resolver grandes sistemas de ecuaciones algebraicas no lineales. Un proceso simple puede requerir fácilmente la solución de más de 10,000 ecuaciones. Por lo regular, alrededor de un 80% de estas ecuaciones describen las relaciones termodinámicas (Westerberg, et. al. (1979)). Sargent (1980) ha elaborado una revisión de los métodos generalizados para resolver sistemas de ecuaciones no lineales.

Para resolver estos sistemas de ecuaciones, los algoritmos generalmente son divididos en la aproximación secuencial modular y en varios tipos de solución simultánea.

#### 2.3.1 SIMULACION SECUENCIAL MODULAR:

El término secuencial modular fué usado por Westerberg, et. al. (1979) para describir la aproximación usada actualmente por los simuladores industriales. En esta aproximación, el programa es dividido en varias subrutinas o módulos y, cada módulo corresponde a una o más etapas de la planta de procesamiento. Existe un programa ejecutivo que arma el diagrama de flujo de proceso y establece el orden de cálculo.

En esta aproximación se presentan dos opciones diferentes de cálculo que son:

- 1) Cálculos orientados al funcionamiento
- 2) Cálculos orientados al diseño

En la primera opción, se desarrolla una subrutina para cualquier tipo de unidad y de esta forma, se calculan las variables de las corrientes de salida como una función de las varia-

bles de las corrientes de entrada y los parámetros de la unidad. Los cálculos proceden de módulo en módulo para simular el proceso. Aquí, generalmente el flujo de información es paralelo al flujo del proceso que está siendo simulado.

En el caso de usar un simulador como medio de diseño, es necesario hacer un diseño más detallado de los modelos de los módulos y tratarlos como en la primera opción. Para resolver este problema, el procedimiento de solución es alterado debido a que las variables fijadas en la primera opción, ahora actuarán como grados de libertad. Este procedimiento es denominado flujo de información inverso.

En la aproximación secuencial modular, todas las corrientes, excepto las corrientes de conectividad, están contenidas en el módulo. Las corrientes de recirculación necesitan ser identificadas y rotas para posteriormente hacer que cumplan las condiciones de convergencia mediante un proceso iterativo. Los estimados iniciales de las variables de una corriente de recirculación deben ser dadas (ya sea por el usuario o asignadas por la computadora) y las especificaciones (restricciones) también deben ser resueltas iterativamente.

Para resolver una simulación por el método secuencial modular, es necesario identificar las particiones del diagrama de flujo, seleccionar las corrientes de corte, almacenar los cálculos y determinar la secuencia de cálculo. Para ejemplificar lo anterior, consideremos el diagrama de flujo ilustrado en la figura 2.6. Este es un diagrama esquemático de un proceso que contiene mezcladores (M), reactores (R) y separadores (D).

Existe una especificación (ESPEC), que consiste en recuperar un componente por medio del separador D1 y ajustarlo para encontrar la concentración deseada en la corriente. Según el flujo de información I1, indica que la recuperación en el separador (parámetros del bloque I1) necesita ser determinado mediante las especificaciones de ESPEC (ver Fig. 2.6).

Las particiones, identifican aquellas colecciones de bloques (referido como un subsistema cíclico máximo o red irreductible) que deben ser resueltos al mismo tiempo.

El rompimiento determina aquellas corrientes o flujos de información que deben ser rotos para dar un sistema acíclico, en el subsistema I, la corriente S4 y el flujo de información II -- fueron seleccionados.

La agrupación de los cálculos, determina cuáles corrientes de corte pueden satisfacer las condiciones de convergencia simultáneamente y el orden en el cual la colección de corrientes de corte también las alcanzará. En el subsistema I, es posible agrupar la convergencia de las variables de la corriente de recirculación dentro de la convergencia de los valores de las especificaciones de diseño y viceversa o, la corriente de recirculación y las especificaciones de diseño pueden converger simultáneamente. Una vez que la agrupación de corrientes de corte ha sido especificada, es necesario determinar la secuencia de cálculo y, finalmente, usar un algoritmo numérico adecuado.

Los puntos expuestos anteriormente serán tratados con mayor amplitud en el capítulo III.

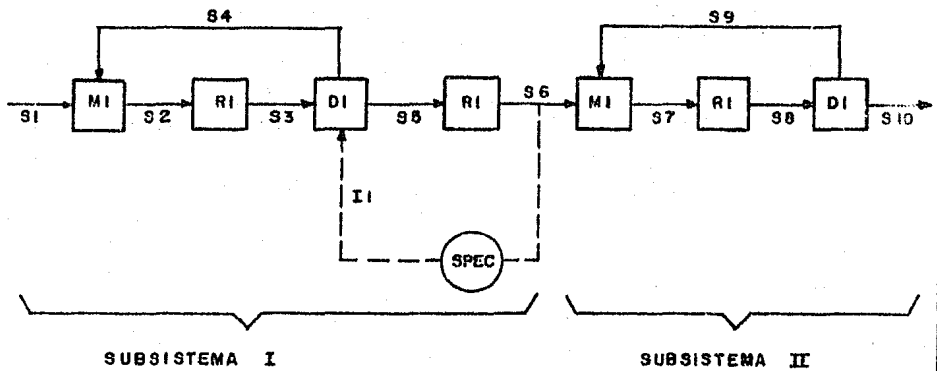
### 2.3.2 MÉTODOS DE ECUACION - ORIENTADA:

La idea básica del método ecuación orientada es: Colectar todas las ecuaciones que describen al diagrama de flujo del proceso y resolverlas como un sistema de ecuaciones algebraicas no-lineales. Matemáticamente el problema puede ser planteado como:

$$\text{Resolver } F(\underline{x}, \underline{u}) = 0 \quad (2.2)$$

Donde:  $\underline{X}$  = Vector de variables dependientes.

$\underline{U}$  = Vector de variables independientes.



TITULO:

FIG. 2.6 DIAGRAMA DE FLUJO PARA  
ILUSTRAR EL ANALISIS DE LA  
ESTRUCTURA COMPUTACIONAL.

TESIS PROFESIONAL

JOSE MARIANO RAMOS OLMOS

E.N.E.P. ZARAGOZA U.N.A.M.

El punto de partida para los métodos de ecuaciones orientadas, es la partición del diagrama de flujo como el usado en el método secuencial modular (ver fig. 2.6). No obstante, en vez de hacer uso de subrutinas para calcular las variables de salida como una función de las variables de entrada, estos simuladores, necesitan de ciertos procedimientos que generen y representen las ecuaciones para cualquier bloque y, una vez generadas las ecuaciones, pueden ser alimentadas a un método eficiente para su solución.

A su vez, en un simulador del tipo solución de ecuaciones, el problema puede ser formulado como un problema de optimización:

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

Más, algunos límites y restricciones impuestas por el mismo problema de optimización. La restricción de igualdad  $F(\underline{X}, \underline{U}) = 0$  es el conjunto de ecuaciones descrito anteriormente y, la diferencia consiste en que ahora las variables independientes son especificadas arbitrariamente con el objeto de maximizar la función objetivo  $P(\underline{X}, \underline{U})$ .

Una de las limitaciones para poder formular el problema del método de solución de ecuaciones como un problema de optimización, ha sido la falta de algoritmos de programación no lineal que puedan tratar efectivamente con un gran número de restricciones de igualdad.

Los algoritmos de solución de ecuaciones han sido usados para modelar diferentes tipos individuales de equipos, para resolver problemas de simulación especializados que involucren conjuntos de unidades del mismo tipo (tales como columnas de destilación o cambiadores de calor) y, hasta el momento, no se han usado como parte de un simulador industrial.

### 2.3.3 SIMULACION POR METODOS LINEALES:

De acuerdo a Westerberg, et.al. (1979) la simulación lineal es aplicada correctamente después de que los balances de masa y energía han sido establecidos, ambos de datos de planta o de simulación rigurosa. Cualquier perturbación pequeña del caso disponible puede ser simulada mediante el uso de relaciones lineales de entrada-salida para cada módulo y combinarlos con ecuaciones lineales de balances de masa y energía.

Rosen (1980) explica que las especificaciones deben ser escritas como a continuación se indica:

$$\text{Ecuaciones de Módulo} \quad \underline{Y} = \underline{A}\underline{X} + B \quad (2.3)$$

$$\text{Ecuaciones de Conservación} \quad \underline{X} = \underline{C}\underline{Y} + F \quad (2.4)$$

(Conexión)

Donde:  $\underline{X}$  = Vector de corrientes de entrada

$\underline{Y}$  = Vector de corrientes de salida

$\underline{F}$  = Vector de corrientes de alimentación

Eliminando las variables de entrada, el vector  $\underline{Y}$  queda expresado como:  $\underline{Y} = (\underline{I} - \underline{A}\underline{C}^{-1})(\underline{A}\underline{F} + B)$  (2.5)

En esta aproximación, el diagrama de flujo es resuelto solamente una vez. Debido a que los modelos lineales son resueltos exactamente, debe tenerse cuidado en que las ecuaciones lineales del módulo sean lo bastante precisas en el rango de interés.

Aunque los paquetes de simulación lineal son nuevos - - - Hutchinson (1974), esta aproximación tiene una larga historia de aplicaciones a los problemas de refinerías donde, muchas corrientes de recirculación conectan módulos que son difíciles de simular, excepto como aproximaciones lineales. Además, la simulación puede ser realizada por la adición de una función objetivo lineal y algunas restricciones de diseño lineales. Entonces, el problema puede ser resuelto como un problema de programación lineal a gran escala.

#### 2.3.4 SIMULACION CUASI - LINEAL:

El procedimiento de solución para esta aproximación, es -- una continuación de los métodos de simulación lineal. Aquí, los modelos lineales del módulo son los Jacobianos de Y con respecto a X y el sistema, es resuelto por el método de Newton donde la -- matriz es descompuesta y resuelta en cada iteración.

Los algoritmos Cuasi - Lineales, han sido aplicados con -- éxito a la simulación de una o más columnas de destilación. Aquí, las recirculaciones deben ser resueltas en cada etapa y pa -- ra el balance de masa y energía total. Además, la mayoría de -- las ecuaciones tales como las relaciones de conservación y equi -- librio son lineales con respecto al flujo de masa.

La incorporación de la aproximación Cuasi - Lineal dentro -- de un paquete de simulación está todavía en desarrollo (Wester--- berg, et.al., 1979), ya que existen problemas de tamaño de alma--- cenaje y manipulación de la matriz.

#### 2.3.5 SIMULADORES SIMULTANEOS MODULARES;

Estos simuladores son híbridos de la aproximación secuen -- cial modular y del tipo de solución de ecuaciones y, también es -- conocido como algoritmo de doble rompimiento Rosen (1980). Westerberg, et.al. (1979) han demostrado que esta aproximación -- es superior a las otras cuando existen restricciones de diseño.

En esta aproximación, la mayoría de las ecuaciones conti -- núan siendo internas a los módulos de proceso, pero las ecuacio -- nes de conectividad, son resueltas simultáneamente en lugar de -- secuencialmente.



El algoritmo de solución de esta aproximación se ilustra en la figura 2.7. Los modelos rigurosos son los modelos de las operaciones unitarias usados en la aproximación secuencial modular. Estos modelos, son usados para determinar los parámetros de los modelos linearizados que son representados como ecuaciones.

Los modelos linearizados son resueltos por cualquier algoritmo de solución de ecuaciones para determinar los nuevos valores de las variables de las corrientes de entrada.

Umeda y Nishio (1972), compararon los métodos secuencial modular y simultáneo modular en dos problemas de ejemplo. Mientras se hacía uso de substitución sucesiva para obtener la convergencia para el secuencial modular, ellos implementaron el modelo de fracción de rompimiento de Rosen así como el coeficiente lineal de Nishimura en el esquema simultáneo modular. En ambos problemas la aproximación simultáneo modular requirió de menos iteraciones para obtener la convergencia.

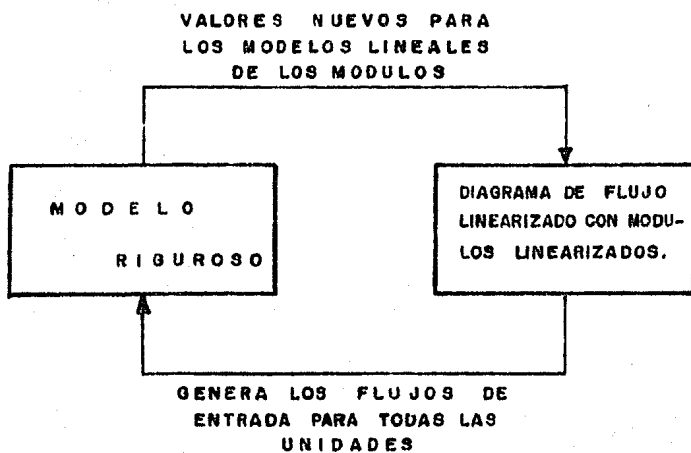
En la tabla 2.1 se ilustran los tipos de algoritmos que usan los simuladores de proceso a estado estacionario, por último, en la figura 2.8 se comparan los procedimientos de solución contra la generalidad de los modelos para los algoritmos descritos anteriormente.

#### 2.4 SOFTWARE:

El campo del software en Ingeniería Química surgió en la década de los 60's y, sistemáticamente, está siendo transformada para producir nuevos sistemas software.

Originalmente, el desarrollo de la simulación de procesos estaba limitado a tres fuentes:

- 1) Instituciones Académicas



TITULO:

FIG. 2.7 ALGORITMO DEL DOBLE  
ROMPIMIENTO

TESIS PROFESIONAL

JOSE MARIANO RAMOS OLMOS

E.N.E.P. ZARAGOZA U.N.A.M.

TABLA 2.1 Clasificación de los Sistemas de Simulación de --  
Procesos, de acuerdo al tipo de algoritmo de so--  
lución que usan.

ALGORITMO:SISTEMA:

Secuencial Modular

Aspen  
Capes  
Desing  
Flowpack II  
Flowtran  
Gps  
Process  
Spad

Simultáneo Modular

Dis.  
Tisflo

Lineal

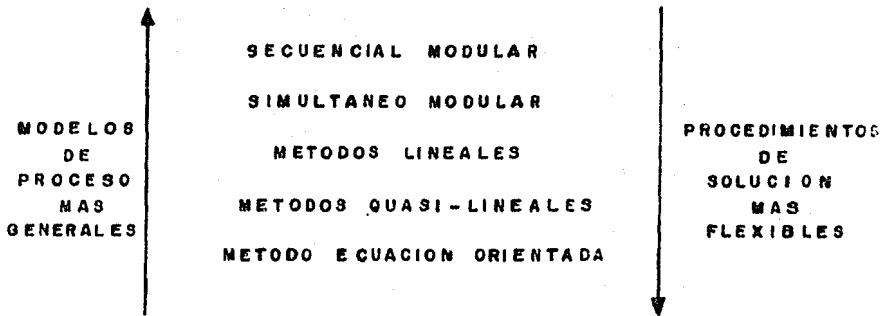
MPB II  
Symbol

Quasi-lineal

Multicol  
Quasilin

Ecuación Orientada

Gender  
Sped-up



TITULO:  
FIG. 2.8 COMPARACION DEL PROCEDIMIENTO DE  
SOLUCION V.S GENERALIDAD DE LOS  
MODELOS DE LOS ALGORITMOS USADOS  
EN LOS SIMULADORES DE PROCESO.

TESIS PROFESIONAL

JOSE MARIANO RAMOS OLMOS

E.N.E.P. ZARAGOZA U.N.A.M.

- 2) Firmas de Consultoría
- 3) Compañías de Proceso Químico y Petroquímico

Debido a que, tradicionalmente sólo las grandes compañías han podido realizar grandes inversiones para el desarrollo de -- sistemas de simulación de procesos, el acceso a la información y uso de los simuladores de proceso a nivel industrial se ha limitado, la mayoría de las veces, a gente de la compañía que financió el desarrollo de los mismos.

Lo anteriormente expuesto, ha repercutido fuertemente en el desarrollo de nuevos sistemas de simulación debido a que, en algunas ocasiones, se cae en el problema de repetición por desconocer lo que se ha investigado y desarrollado hasta el momento.

Para evitar la repetición y alentar el desarrollo de nuevos sistemas de simulación, se han definido ciertos aspectos de organización para el desarrollo de nuevos sistemas software.

#### 2.4.1 ASPECTOS DE ORGANIZACION EN EL DESARROLLO DEL SOFTWARE:

Existen cinco etapas en la vida de cualquier programa computacional a gran escala:

- 1) Definición
- 2) Diseño
- 3) Implementación
- 4) Prueba
- 5) Transferencia

La primera etapa es la DEFINICION de las especificaciones funcionales o criterios de diseño que plantean lo que el sistema podrá hacer pero, no indican cómo. Las especificaciones funcionales sirven para contestar preguntas tales como: -

¿Qué tipos de modelos ingenieriles serán incluidos?

¿Qué tipos de análisis serán ejecutados?

¿Cuáles características de la interfase con el usuario son importantes?

El diseñador debe tener una buena visión de lo que hará el sistema. Durante la etapa de definición es muy importante realizar un estudio del software y de los sistemas existentes disponibles para evitar la repetición de información.

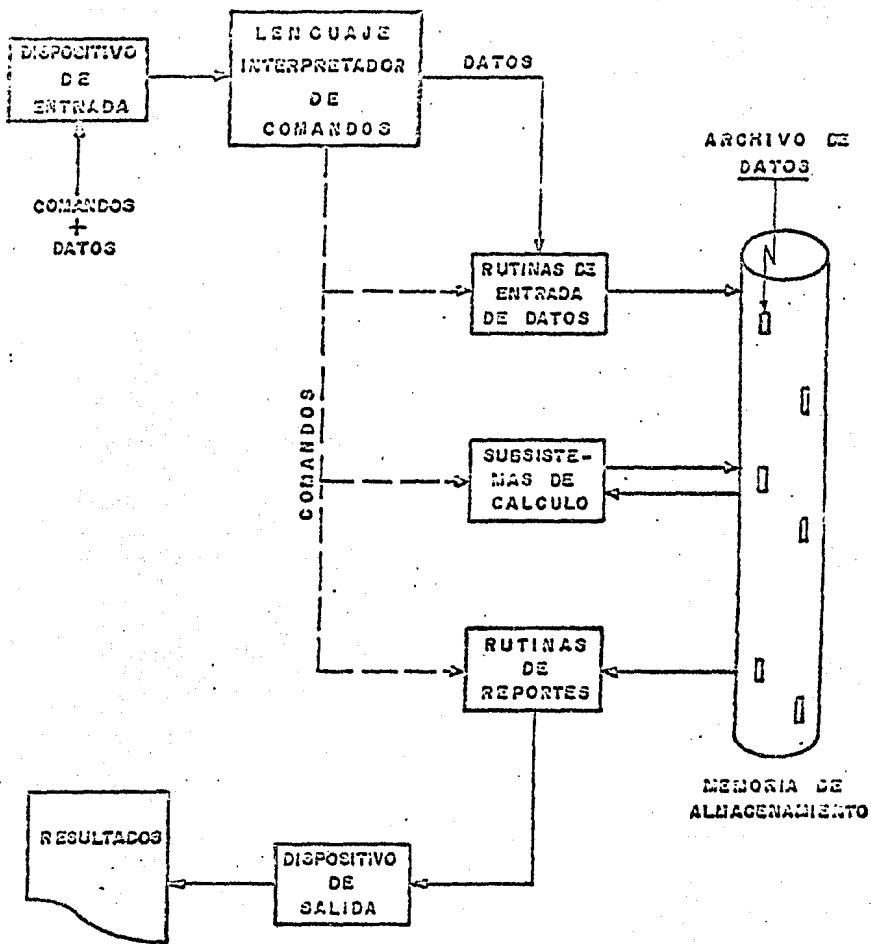
Una de las etapas más críticas de la etapa de definición es la comunicación de los diseñadores con los usuarios potenciales. Esta comunicación se puede establecer por medio de cuestionarios en donde se les muestre a los usuarios las principales características del nuevo sistema y estos a su vez, den sus opiniones o comentarios del mismo.

Una vez que se han definido los requerimientos funcionales del sistema, la siguiente etapa corresponde al DISEÑO del sistema. El diseño, especifica detalladamente la forma como serán encontrados los requerimientos funcionales. En esta etapa se definen:

- a) Arquitectura del sistema
- b) Estructura de Datos
- c) Lenguaje de programación, etc.

El diseño del sistema no es más que una síntesis de proceso en donde se necesitan considerar y evaluar diferentes alternativas. En la figura 2.9 se ilustra la estructura de un sistema avanzado de simulación.

Una vez que se ha diseñado el sistema, se inicia la etapa de la IMPLEMENTACION. La implementación es el proceso que permite transcribir el sistema diseñado a un lenguaje computacional; en caso de no haber seleccionado un lenguaje de programación en la etapa de diseño, se debe seleccionar para poder iniciar la --



TITULO:  
FIG. 2.9 FLUJO DE INFORMACION DE UN SISTEMA AVANZADO DE SIMULACION.

TESIS PROFESIONAL  
JOSE MARIANO RAMOS OLMO  
E.N.E.P. ZARAGOZA U.N.A.M.

implementación.

Regularmente, se seleccionan lenguajes de alto nivel tales como FORTRAN o PL/1 que son bastante comunes y con las características necesarias para poder manejar este tipo de problemas.

La cuarta etapa es la PRUEBA, la cual resulta ser una de las etapas más costosas y de las que consume más tiempo. En esta etapa se demuestra si el programa es confiable y si realmente está libre de fallas.

La etapa final en la vida del sistema es la fase de TRANSFERENCIA en la cual el programa, se empieza a usar continuamente para resolver problemas. En esta última etapa es de suma importancia establecer las estrategias para el mantenimiento y futuras expansiones.

#### 2.4.2 ASPECTOS HISTORICOS DE LA SIMULACION DE PROCESOS A ESTADO-ESTACIONARIO:

Las raíces de los sistemas de simulación en computadora -- pueden ser rastreadas a partir de la publicación del primer programa FLEXIBLE FLOWSHET en 1958 (Kesler y Kesler (1958)). A partir de esta fecha se inicia la la. generación de los simuladores de proceso.

La característica más importante de los simuladores de la. generación es que, básicamente, estaban enfocados hacia la - simulación individual de las operaciones unitarias.

Estos programas fueron desarrollados por innovadores entusiastas que tenían que ser expertos en Ingeniería Química, Matemáticas y en los nuevos lenguajes de computadora. Afortunadamente hubo muchas de estas personas multitalentosas que pudieron adaptar los simuladores a las nuevas máquinas y usarlos a nivel industrial. Los éxitos (aunque fueron pocos) de estos programas, - fueron lo suficientemente notables para garantizar el soporte -- continuo de este campo recién formado.



Los sistemas de simulación industrial más importantes fueron desarrollados en la década de los 60's. Esta 2a. "Generación" de programas, radicalmente no exhibieron cambios y generalmente fueron construídos haciendo uso de la tecnología de los programas existentes. A continuación se presentan las diferencias entre los programas de la 1a. y la 2a. generación:

<u>1a. Generación</u>	<u>2a. Generación</u>
1.- De 1 a 5 años-hombre para desarrollarlos	De 20 a 60 años-hombre para desarrollarlos
2.- Costo menor a \$200,000.00	Costo mayor a \$1'000,000.00
3.- La simulación es realizada a partir de la interpretación de la descripción del proceso	La simulación se lleva a cabo mediante la ejecución de un programa formado a base de palabras clave
4.- Capacidades muy limitadas de predicción de propiedades físicas	Extenso banco de datos de información de propiedades físicas
5.- Archivos de subrutinas de equipo muy limitado	Extenso archivo de subrutinas de equipos
6.- Las nuevas subrutinas de equipo sólo pueden ser adicionadas e implementadas por el programador del sistema	Las nuevas subrutinas de equipo pueden ser adicionadas e implementadas por el usuario
7.- Capaz de simular solamente del modo "funcionamiento"	Capaz de resolver problemas "diseño" por medio de simulación controlada
8.- Generalmente la convergen	Técnicas sofisticadas para la

- cia de las recirculacio-- convergencia de recirculacio--  
 nes se lleva a cabo por - nes  
 sustitución directa
- 9.- Limitada a sistemas líquido --  
 vapor
- 10.- Usa estructuras fijas de arre-  
 glos de datos
- 11.- Todas las corrientes deben ser  
 descritas por el mismo conjun-  
 to de variables
- 12.- Métodos de alimentación de da-  
 tos por listas - fijas o pala-  
 bras clave

Aquí los términos 1a. y 2a. generación se manejan en el as  
pecto evolutivo. Históricamente, nosotros podemos visualizar el  
 surgimiento de los programas de la primera generación tales co--  
 mo: CHEVRON, FLEXIBLE FLOWSHFET, PACER Y CHESS. Los simuladores  
 de la segunda generación fueron construídos tomando como base la  
 tecnología de los de la 1a. generación. Ejemplos de los simula-  
 dores de la 2a. generación son: FLOWTRAN, CAPEP, IPES, CPES, --  
 etc.

A pesar de las impresionantes mejoras obtenidas en los nue  
vos modelos de simulación, permanecen ciertas deficiencias. De-  
 bido al alto costo del desarrollo, la mayoría de los paquetes in  
dustriales sólo son para uso privado. Este aislamiento resulta-  
 no solamente en duplicación de esfuerzos sino que inhibe cual- -  
 quier mejora de los mismos. El mantenimiento y actualización de  
 los grandes sistemas de simulación requiere de mucha dedicación-  
 y dinero que, en muchas ocasiones, es difícil de justificar. --  
 Asociado con los problemas de mantener un sistema, está el pro-  
 blema de expandir un sistema. Independientemente de que se cuen  
te con recursos y tiempo para desarrollar el sistema inicial, --

posteriormente, las adiciones y modificaciones frecuentemente -- son precipitadas, contribuyendo así al deterioro gradual de todo el sistema lógico.

En las etapas iniciales del proyecto ASPEN, Evans y Seider (1976), enumeraron los criterios para los sistemas de simulación de la 3a. generación. Entre estos estuvieron:

- 1.- El sistema debe poder analizar los diagramas de flujo de proceso incorporando diferentes tipos de corrientes, en especial, las corrientes que contengan sólidos.
- 2.- El sistema debe aceptar diagramas de flujo de proceso a diferentes niveles de detalle.
- 3.- El usuario debe poder manejar directamente la configuración del diagrama de flujo de proceso.
- 4.- El sistema debe estar bien documentado y diseñado para comodidad de uso.
- 5.- El sistema debe expandirse y ser modificado fácilmente.
- 6.- El sistema de salida debe ser orientado al usuario.
- 7.- El sistema debe tener capacidad de simular unidades de proceso que manejen sólidos.
- 8.- Costo superiores a los \$5'000,000.00
- 9.- Requerirá de 20 - 60 hombre / año durante 3 - 5 años.

## 2.5 INTERFASE CON EL USUARIO:

Se podría decir que la interfase con el usuario es la ventana por la cual el usuario puede observar detenidamente el sis-

tema. Incluye el lenguaje de entrada y otros parámetros por medio de los cuales el usuario describe su problema.

Las características de la interfase con el usuario que deben cubrir los simuladores de la 3a. generación se enlistan a -- continuación:

- 1) Deben tener formatos libres.
- 2) Debe estar organizado por medio de párrafos, oraciones y palabras.
- 3) Debe permitir diferentes tipos de entrada:
  - a) Por medio de palabras clave
  - b) En forma posicional
- 4) Debe proporcionar valores por default en los lugares - que el usuario los solicite.
- 5) El usuario debe especificar totalmente las unidades de entrada y salida dentro de un párrafo o, para datos -- aislados.
- 6) Debe orientar al usuario acerca de la forma de introducir los datos.
- 7) El usuario puede tener acceso para hacer cualquier - - transformación de las variables de los modelos mediante el uso de proposiciones FORTRAN.
- 8) La descripción del diagrama de flujo de proceso será - independiente del esquema de convergencia.

Por lo regular, muchos de estos requerimientos están en -- conflicto y deben ser tratados a fondo para obtener el mayor provecho posible de la interfase.

C A P I T U L O     I I I

IDENTIFICACION DE PROCESOS:

### 3.1 INTRODUCCION:

En la simulación modular se emplean diferentes procedimientos de cálculo para cada unidad de un proceso. Generalmente el programa es dividido en varios módulos. Cada módulo corresponde a una o más etapas de la Planta y están basados en los datos suministrados por el usuario.

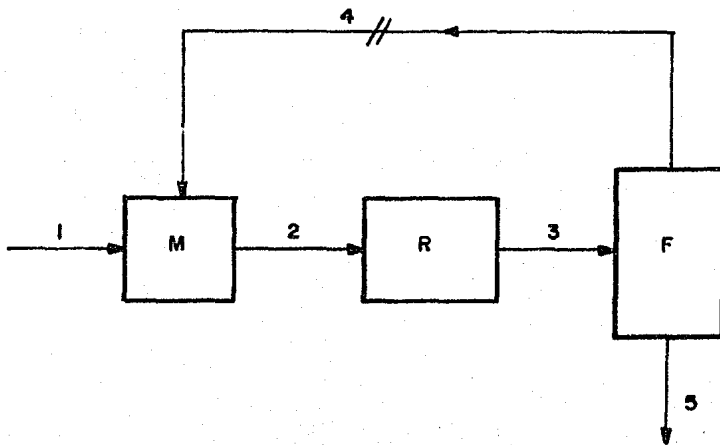
Una vez que se tienen disponibles los módulos de las unidades de proceso, uno puede decidir la forma en que se va a simular el proceso. Una de estas formas puede ser el algoritmo secuencial modular.

Para ilustrar el algoritmo de solución del secuencial modular, analizaremos la figura 3.1. El proceso representado en dicha figura, puede ser simulado mediante la secuencia de pasos mostrada a continuación:

- 1) Suponga la corriente de recirculación 4
- 2) Calcule la corriente de salida del mezclador M
- 3) Calcule la corriente de salida del reactor R
- 4) Calcule la corriente de salida del separador F
- 5) Compare el valor calculado para la corriente 4 con el valor supuesto, si no son iguales dentro de la tolerancia especificada continúa, de lo contrario termina.
- 6) Tomando como base el valor calculado para la corriente 4, suponga un nuevo valor para esta corriente y regrese a 2.

Usualmente un proceso complejo contiene varias recirculaciones y, las corrientes de una recirculación son calculadas mediante métodos iterativos. Al analizar los diagramas de flujo de estos procesos, se identifican los subsistemas que contienen las recirculaciones que comúnmente se conocen como ciclos.

Un procedimiento general de cálculo para un diagrama de --



TITULO:

FIG. 3.1 DIAGRAMA DE FLUJO DE UN  
PROCESO CON RECIRCULACION.

TESIS PROFESIONAL

JOSE MARIANO RAMOS OLMOS

E.N.E.P. ZARAGOZA U.N.A.M.

flujo de proceso es:

- 1) Identificar y determinar el orden de precedencia de -- los ciclos.
- 2) Determinar los valores de las variables que serán usados para obtener la convergencia de cualquier ciclo.
- 3) Seleccionar un método para hacer converger a dichos valores.

La identificación de una red de flujo de un proceso incluye la determinación de la secuencia de cálculo de los equipos, - conjuntos de recirculación y la localización de equipos y co- -- rrientes que forman estos conjuntos.

En la práctica, el hecho de determinar las variables que - serán usadas para obtener la convergencia se le conoce como rompimiento. En el ejemplo descrito anteriormente, son supuestos - los valores para las variables de la corriente 4 y por consi- -- guiente, a esta corriente se le llamará corriente de corte. En los grandes procesos donde, usualmente existen varias recirculaciones, existen varias corrientes de corte. El conjunto de co- -- rrientes seleccionadas para romper todas las recirculaciones se le conoce como conjunto de corte.

Por último, para resolver los sistemas de ecuaciones simultáneas no lineales generados por cada ciclo en un número finito de aproximaciones sucesivas, se requieren los métodos de convergencia.

### 3.2 IDENTIFICACION:

Una unidad de proceso puede ser simulada si todos los parámetros de la unidad y las corrientes de entrada son conocidos. - Algunas unidades pueden ser ordenadas fácilmente en una secuen- -- cia, no obstante, las unidades involucradas en una recirculación no pueden ser simuladas individualmente debido a que, sus entra-



das dependen de otras unidades y en general, las unidades involucradas en una recirculación son tratadas como un bloque en la secuencia de cálculo.

Existen diferentes técnicas para la identificación de las redes de flujo de proceso. Norman (1965), Himmelblau (1970) y Mah (1974) hacen uso de las Técnicas Matriciales. En la aproximación de Norman, los ciclos son tratados conforme se van identificando, las unidades involucradas en los ciclos son condensadas en pseudounidades y las redes máximas se definen cuando ya no existen más ciclos en la red condensada.

Existe una segunda técnica denominada Rastreo de Pasos o Trazo de la Trayectoria. En este procedimiento la red de flujo de proceso es analizada tomando como base a una unidad y rastreando el flujo de información a las unidades subsecuentes.

Sargent y Westerberg (1964) proponen un método en el cual el flujo de información de una gráfica dirigida es rastreado en sentido inverso al del flujo. El rastreo se inicia desde una unidad inicial continuando hasta que este se vuelve a localizar, todas las unidades involucradas en el rastreo forman un ciclo y son condensados en una pseudounidad. El procedimiento se continúa hasta que ya no se localizan más ciclos y todas las unidades han sido rastreadas. Este método presenta dificultades para ser programado y los autores, utilizan un lenguaje basado en listas de información para su algoritmo.

Christiansen y Rudd (1969) simplificaron el algoritmo de Sargent y Westerberg eliminando las unidades que no tienen corrientes de entrada y a las que no tienen corrientes de salida. Esta eliminación se hace debido a que estas unidades no pueden estar incluidas en alguno de los ciclos identificados.

De las técnicas mencionadas anteriormente, las técnicas de rastreo de pasos presentan ventajas que la hacen preponderante sobre las técnicas matriciales; entre las más importantes están:

- 1) Fácil eliminación de las unidades no involucradas en las recirculaciones.
- 2) Ahorro de memoria durante el almacenamiento.
- 3) Menor tiempo de cómputo.

Estas dos técnicas constituyen la base de todos los demás métodos de identificación, algunos son modificaciones a éstas y otros son combinaciones de ambas.

### 3.2.1 METODO DE KEHAT Y SHACHAM (1973):

Básicamente una de las desventajas de las técnicas matriciales es que, requieren de grandes espacios de memoria y de un gran número de operaciones en la elevación de las potencias de las matrices. Para evitar esto, estos autores proponen una combinación de las ventajas de los algoritmos desarrollados por Norman (1964) y Himmelblau (1966) reduciendo en gran parte los requerimientos de memoria y el tiempo de cómputo.

El método consiste en formar una matriz de índices, en donde serán almacenados los índices de las hileras y columnas que contengan a elementos de la matriz adyacente de un proceso cuyo valor sea diferente de cero. Una vez formada la matriz de índices, solamente será necesario efectuar comparaciones y sustituciones para el cálculo de las potencias de la matriz.

Los pasos para aplicar el algoritmo de Kehat y Shacham se describen a continuación:

- a) Construir la matriz adyacente y la matriz de índices de la gráfica dirigida del diagrama de flujo de un proceso.
- b) Eliminar las columnas e hileras de la matriz adyacente cuyos elementos sean igual a cero. Para llevar a cabo la eliminación, se tienen que localizar las columnas que no tengan elementos con valor de 1 (unidades de --

proceso que no tienen corrientes de entrada) eliminándolas con sus respectivas hileras. Posteriormente, se localizan las hileras que no tengan elementos con valor de 1 (unidades de proceso que no tienen corrientes de salida) y se eliminan con sus respectivas columnas. Este procedimiento se continúa hasta que todas las columnas e hileras con las características mencionadas anteriormente son eliminadas construyendo una nueva matriz adyacente y una de índices en cada eliminación. En la figura 3.2 y en las tablas 3.1, 3.2 y 3.3 se ilustra con un ejemplo el proceso de eliminación. Al analizar la tabla 3.2, se observa que originalmente se requerían 49 espacios de memoria para almacenar a la matriz adyacente y 14 espacios en la matriz de índices, lográndose disminuir a 16 y 8 espacios respectivamente después de haber eliminado las columnas e hileras cuyos elementos eran igual a cero (tabla 3.3).

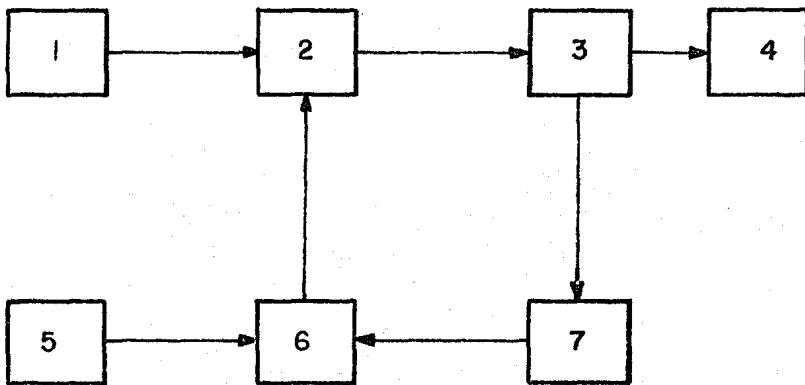
- c) Una vez obtenida la matriz de índices reducida, se eleva a potencias sucesivas con el objeto de identificar los ciclos que forman al proceso.

La máxima potencia a la que se va a elevar la matriz de índices es igual al número de unidades que forman la matriz adyacente reducida. Para hacerlo hay que proceder como se indica a continuación.

Tomando como base la matriz de índices reducida de la tabla 3.3, se localiza el número del lado derecho del primer par de elementos original (2, 3), entonces, se busca en el lado izquierdo de cada par de elementos de la matriz I el número 3, en nuestro caso es el par (3, 7), entonces, se procede a reemplazar el 7 por el 3 en el par de elementos original. El par resultante en la matriz  $I^2$  es (2,7).

Este procedimiento se continúa para todos los pares de la matriz de índices original y para cada una de las potencias sucesivas tomando siempre como base a la matriz de índices reducida original.

Los ciclos serán identificados por los conjuntos de elementos de la diagonal principal de la matriz adya-



TITULO:

FIG. 3.2. GRAFICA DIRIGIDA DE  
UN PROCESO

TESIS PROFESIONAL

JOSE MARIANO RAMOS OLMOS

E.N.E.P. ZARAGOZA U.N.A.M.

TABLA 3.1 Matrices de adyacencia y de índices.

$$A = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{matrix} & \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \end{matrix}$$

MATRIZ ADYACENTE

$$I = \begin{pmatrix} 1 & 2 \\ 2 & 3 \\ 3 & 4 \\ 3 & 7 \\ 5 & 6 \\ 6 & 2 \\ 7 & 6 \end{pmatrix}$$

MATRIZ  
DE  
INDICES

TABLA 3.2 Eliminación de columnas y su respectiva matriz de índices.

$$\begin{array}{c}
 \begin{array}{ccccccc}
 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\
 1 & \left( \begin{array}{ccccccc} 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{array} \right) & & & & & & \\
 2 & \left( \begin{array}{ccccccc} 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{array} \right) & & & & & & \\
 3 & \left( \begin{array}{ccccccc} 0 & 0 & 0 & 1 & 0 & 0 & 1 \end{array} \right) & & & & & & \\
 A = 4 & \left( \begin{array}{ccccccc} 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array} \right) & & & & & & \\
 5 & \left( \begin{array}{ccccccc} 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{array} \right) & & & & & & \\
 6 & \left( \begin{array}{ccccccc} 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{array} \right) & & & & & & \\
 7 & \left( \begin{array}{ccccccc} 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{array} \right) & & & & & & \\
 \end{array}
 \end{array}
 \quad
 \begin{array}{c}
 I = \left( \begin{array}{cc} 1 & 2 \\ 2 & 3 \\ 3 & 4 \\ 3 & 7 \\ 5 & 6 \\ 6 & 2 \\ 7 & 6 \end{array} \right)
 \end{array}$$

$$\begin{array}{c}
 \begin{array}{ccccc}
 & 2 & 3 & 4 & 6 & 7 \\
 2 & \left( \begin{array}{ccccc} 0 & 1 & 0 & 0 & 0 \end{array} \right) & & & & \\
 3 & \left( \begin{array}{ccccc} 0 & 0 & 1 & 0 & 1 \end{array} \right) & & & & \\
 A = 4 & \left( \begin{array}{ccccc} 0 & 0 & 0 & 0 & 0 \end{array} \right) & & & & \\
 6 & \left( \begin{array}{ccccc} 1 & 0 & 0 & 0 & 0 \end{array} \right) & & & & \\
 7 & \left( \begin{array}{ccccc} 0 & 0 & 0 & 1 & 0 \end{array} \right) & & & & \\
 \end{array}
 \end{array}
 \quad
 \begin{array}{c}
 I = \left( \begin{array}{cc} 2 & 3 \\ 3 & 4 \\ 3 & 7 \\ 6 & 2 \\ 7 & 6 \end{array} \right)
 \end{array}$$

TABLA 3.3 Eliminación de hileras y su respectiva matriz de índices.

$$\begin{array}{r}
 \\
 \\
 \\
 \\
 \\
 \\
 \end{array}
 \begin{array}{c}
 2 \quad 3 \quad 4 \quad 6 \quad 7 \\
 A = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}
 \end{array}
 \qquad
 \begin{array}{c}
 \\
 \\
 \\
 \\
 \\
 \\
 \end{array}
 \begin{array}{c}
 I = \begin{pmatrix} 2 & 3 \\ 3 & 4 \\ 3 & 7 \\ 6 & 2 \\ 7 & 6 \end{pmatrix}
 \end{array}$$

$$\begin{array}{r}
 \\
 \\
 \\
 \\
 \\
 \\
 \end{array}
 \begin{array}{c}
 2 \quad 3 \quad 6 \quad 7 \\
 A = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}
 \end{array}
 \qquad
 \begin{array}{c}
 \\
 \\
 \\
 \\
 \\
 \\
 \end{array}
 \begin{array}{c}
 I = \begin{pmatrix} 2 & 3 \\ 3 & 7 \\ 6 & 2 \\ 7 & 6 \end{pmatrix}
 \end{array}$$

cente cuyo valor sea igual a 1 y a su vez, están representados por los pares de números iguales en la matriz de índices.

La identificación de los ciclos del proceso ilustrado en la figura 3.2 se ilustran en la tabla 3.4.

En caso de que se presenten más pares idénticos que la potencia a la cual fué elevada la matriz de índices, - uno de estos pares es eliminado y se continúa elevando a las potencias sucesivas. El par eliminado se combina para formar una pseudo-unidad, sin embargo; si el número de pares idénticos es igual a la potencia de la matriz, las unidades se condensan formando de esta manera una pseudo-unidad y se repiten los pasos b) y c) - ya descritos anteriormente.

### 3.2.2 METODO DE JAIN Y EACKMAN (1972):

Estos autores desarrollan una nueva técnica basada en una combinación de las técnicas matriciales y de la técnica de rastreo de pasos. Hacen uso de matrices dispersas (matriz de adyacencia) y de multiplicación de matrices en forma compacta. La técnica del rastreo de pasos solamente se utiliza para identificar las unidades que forman un ciclo máximo.

En este método, el diagrama de flujo de proceso es tratado como una gráfica dirigida y es representada por la matriz de adyacencia correspondiente. Los ciclos son identificados por la presencia de trayectorias que involucran a más de una unidad y son identificadas a partir de la matriz adyacente elevada a la potencia correspondiente. El ciclo máximo se determina cuando ya no son detectados más ciclos en la red condensada por medio de la matriz de alcance.

La matriz de alcance es la suma booleana de las potencias sucesivas de la matriz de adyacencia hasta que se obtiene una matriz constante.



TABLE 3.4 Identificación de ciclos por medio del método de Kehat y Schacham.

$$A = \begin{matrix} & 2 & 3 & 6 & 7 \\ \begin{matrix} 2 \\ 3 \\ 6 \\ 7 \end{matrix} & \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \end{matrix}$$

$$A_2 = \begin{matrix} & 2 & 3 & 6 & 7 \\ \begin{matrix} 2 \\ 3 \\ 6 \\ 7 \end{matrix} & \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \end{matrix}$$

$$A_3 = \begin{matrix} & 2 & 3 & 6 & 7 \\ \begin{matrix} 2 \\ 3 \\ 6 \\ 7 \end{matrix} & \begin{pmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{pmatrix} \end{matrix}$$

$$A_4 = \begin{matrix} & 2 & 3 & 6 & 7 \\ \begin{matrix} 2 \\ 3 \\ 6 \\ 7 \end{matrix} & \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \end{matrix}$$

$$I = \begin{pmatrix} 2 & 3 \\ 3 & 7 \\ 6 & 2 \\ 7 & 6 \end{pmatrix}$$

$$I_2 = \begin{pmatrix} 2 & 7 \\ 3 & 6 \\ 6 & 3 \\ 7 & 2 \end{pmatrix}$$

$$I_3 = \begin{pmatrix} 2 & 6 \\ 3 & 2 \\ 6 & 7 \\ 7 & 3 \end{pmatrix}$$

$$I_4 = \begin{pmatrix} 2 & 2 \\ 3 & 3 \\ 6 & 6 \\ 7 & 7 \end{pmatrix}$$

El método propuesto, está formado por la combinación de -- los siguientes pasos:

- 1) Elaboración de la matriz de adyacencia y su almacenam--  
amiento en forma compacta.
- 2) Determinación de los ciclos máximos que incluye:
  - a) Localización de ciclos
  - b) Separación de ciclos
  - c) Condensación de ciclos
- 3) Ordenar las unidades y bloques de unidades.
- 4) Determinación de los subciclos en un ciclo máximo.

### 3.2.2.1 MATRIZ DE ADYACENCIA Y SU ALMACENAMIENTO:

La matriz adyacente se forma a partir de la información ob--  
tenida de las corrientes de la red de flujo de proceso. Un ele--  
mento  $a_{i,j}$  representa el número de corrientes que están fluyendo  
de la unidad  $i$  a la unidad  $j$ . La matriz de adyacencia es almace--  
nada por medio de dos vectores.

- a) Vector de corrientes
- b) Vector de identificación

La dimensión del vector de corrientes, es igual al número--  
de corrientes de la red, mientras que la del vector de identifi--  
cación es igual al número de unidades. Para construir ambos vec--  
tores, es necesario proceder como se indica a continuación:

Se inicia con la unidad designada con el número 1 en la --  
gráfica dirigida del diagrama de flujo del proceso. Se enlistan  
en el vector de corrientes en orden ascendente los números co--  
rrespondientes a las unidades que reciben flujo de información -  
de la primera unidad. En caso de que una unidad sea receptora -  
de  $n$  corrientes paralelas, el número correspondiente a esa uni--  
dad, será listado sucesivamente tantas veces como corrientes re--  
ciba.

Una vez que han sido listadas todas las corrientes que tie--  
nen como origen la primera unidad, el valor del último índice --

del vector de corrientes, es almacenado como el primer elemento del vector de identificación. Este procedimiento se repite para las corrientes que tienen como origen la segunda unidad, tercera etc., donde los números correspondientes a las unidades receptoras de cualquier corriente, son enlazadas a aquellas que ya están incluidas en el vector de corrientes.

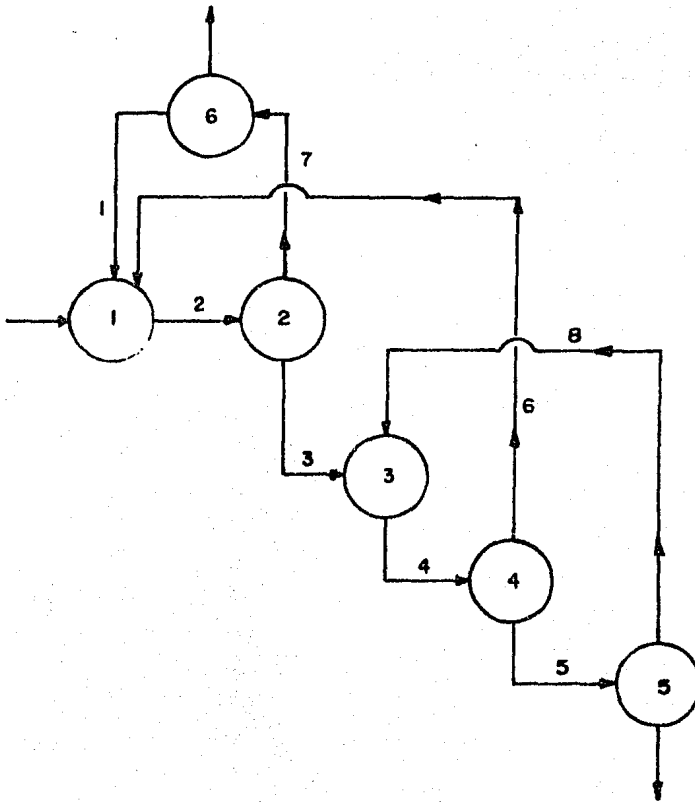
En cualquier caso, el índice del vector de corrientes correspondiente a la última corriente cuyo origen es la unidad  $i$ , es almacenado en la  $i$ -ésima posición del vector de identificación. Por último, en caso de que existan unidades que no den origen a ninguna corriente y, sean solamente receptoras, no podrán ser consideradas como elemento del vector de corrientes, mientras que en el vector de identificación, tomarán el valor del índice de la última unidad registrada en el vector de corrientes.

En la tabla 3.5, se ilustran la matriz de adyacencia y los vectores de almacenamiento de la gráfica dirigida de la figura 3.3. El número de espacios de memoria requeridos por esta técnica es igual a la suma del número de corrientes y el número de unidades del proceso. La técnica convencional requiere el cuadrado del número de equipos de espacios de memoria para su almacenamiento.

### 3.2.2.2 DETERMINACION DE CICLOS MAXIMOS:

Para definir un ciclo máximo, es necesario referirnos a la definición de un ciclo. Un ciclo es una trayectoria cerrada que consiste en una sucesión de unidades que aparecen en el orden de flujo. La colección más grande de estas unidades se llama ciclo máximo, cualquier ciclo puede ser un ciclo máximo o formar parte de éste.

La determinación de ciclos máximos consta de tres etapas que son aplicadas repetitivamente:



TITULO:

FIG. 3.3 PROBLEMA DE CAVETT

TESIS PROFESIONAL

JOSE MARIANO RAMOS OLMOS

E.N.E.P. ZARAGOZA U.N.A.M.

**TABLA 3.5** Matriz adyacente y vectores de corrientes e identificación.

$$A = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{matrix} & \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{matrix}$$

Indices                    1 2 3 4 5 6 7 8

Vector de Corrientes      2 3 6 4 1 5 3 1

Vector de Identificación  1 3 4 6 7 8

- a) Localización de ciclos: Se lleva a cabo mediante una-examinación de las potencias sucesivas de la matriz adyacente.
- b) Separación de ciclos: Se realiza al especificar las - unidades involucradas en un ciclo por medio de la técnica de estructuración de árboles.
- c) Condensación de ciclos: Se utiliza para formar pseudo unidades y de esta forma, representar el ciclo máximo.

a) Localización de ciclos:

La existencia de ciclos en un proceso, se determina cuando al analizar las potencias sucesivas de la matriz adyacente, se detectan elementos diferentes de cero en la diagonal principal. Las potencias requeridas se obtienen por medio de un algoritmo para la multiplicación de matrices compactas.

Este algoritmo, hace uso de la forma en que se almacena la matriz adyacente. Los elementos del vector de corrientes son ordenados en hileras desarrollando de esta forma las hileras de la matriz resultante. Los vectores de identificación de corrientes son desarrollados simultáneamente. En la figura 3.4 se ilustra el diagrama de flujo para la multiplicación de matrices.

La forma para multiplicar es:

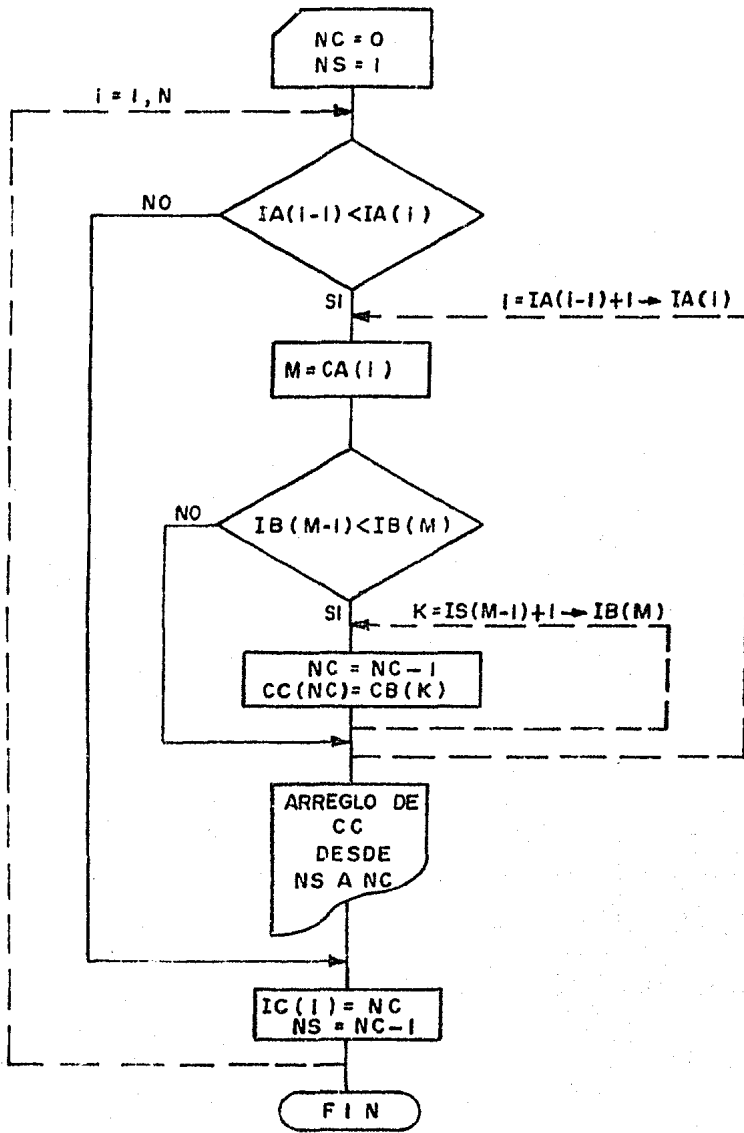
$$\underline{C} = \underline{A} \times \underline{B} \quad (3.1)$$

Donde: C = Matriz con vector de corrientes CC y vector de identificación IC

B = Matriz con vector de corrientes CB y vector de identificación IB

A = Matriz con vector de corrientes CA y vector de identificación IA

N = Número de hileras de la matriz A



TITULO:  
 FIG. 3.4. DIAGRAMA DE FLUJO PARA LA  
 MULTIPLICACION DE  
 MATRICES

TESIS PROFESIONAL  
 JOSE MARIANO RAMOS OLMOS  
 E.N.E.P. ZARAGOZA U.N.A.M.

NC = Contador de los elementos ocupados en el vector CC  
 NS = Posición del primer elemento en CC para la hilera actual

b) Separación de ciclos:

Los ciclos del mismo tamaño producen simultáneamente elementos diferentes de cero en la diagonal principal de las potencias de la matriz adyacente correspondientes. Estos ciclos involucran conjuntos de unidades diferentes, sin embargo; el conjunto de estos equipos no pueden ser determinados a partir del análisis de esta matriz. La separación de ciclos resulta más sencilla si ésta se define como la determinación de un ciclo simple de tamaño mínimo y las unidades involucradas en éste mientras que los otros ciclos serán localizados en los pasos subsecuentes.

Las unidades involucradas en un ciclo simple, son condensados para formar pseudo-unidades dando como resultado la formación de una nueva matriz adyacente. Los ciclos de tamaño mínimo que forman un subconjunto de un conjunto de unidades dadas, son localizados utilizando estructuras de árboles mediante la construcción de trayectorias de longitud mínima entre éstas. Para lograrlo, se inicia con alguna unidad del conjunto que se clasifica en el nivel cero de la estructura y se enlistan en el primer nivel de la estructura todas las unidades que reciben flujo de información del nivel cero. En el segundo nivel se enlistarán todos los equipos del ciclo que reciben flujo de información de los equipos del primer nivel y que, todavía no han sido enlistados. Este procedimiento se puede continuar para enlistar todos los equipos del ciclo en la estructura de árbol.

El nivel en el cual son enlistados los equipos, es la trayectoria más corta que existe desde el nivel cero hasta el



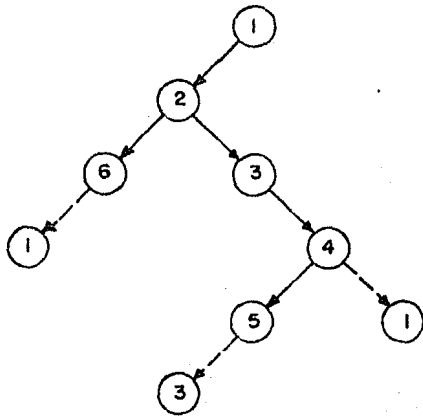
equipo considerado. Todos los equipos estarán involucrados al menos en un ciclo de tamaño determinado y, debe incluirse al equipo inicial. Lo expuesto anteriormente implica que el equipo inicial recibirá flujo desde uno de los equipos del nivel N-1. Al localizarse esta unidad, la estructura es rastreada en sentido inverso para identificar a todos los equipos involucrados en el ciclo. Para poder efectuar este rastreo, es necesario conocer la unidad que va a dirigir flujo hacia el nivel cero.

Esta información es almacenada en el vector de rastreo durante la construcción de la estructura. El elemento  $j$  del vector de rastreo contiene a  $i$ , que es el número de la unidad que dirigió información hacia la unidad  $j$ .

Un ejemplo de la aplicación de las estructuras de árboles para identificar los equipos de tamaño mínimo se ilustra en la figura 3.5. En este ejemplo se resuelve la gráfica dirigida del proceso ilustrado en la figura 3.3.

Si se selecciona al equipo 1 como equipo inicial (nivel cero), el equipo 2 recibe flujo de esta unidad motivo por el cual, será ubicado en el primer nivel y en el vector de rastreo, se asignará el valor de 1 al segundo lugar. En el nivel 1 existe flujo desde el equipo 2 a los equipos 6 y 3 que hasta el momento no han sido listados, ubicándolos de esta forma en el nivel 2. El número del equipo base que será asignado a los lugares 6 y 3 será el 2. El flujo del equipo 6 se dirige al equipo 1 que ya ha sido listado en el nivel cero detectando de esta forma un ciclo que incluye a los equipos 1, 2 y 6. Se deberá asignar el 6 a la primera casilla del vector de rastreo.

En el mismo segundo nivel, el equipo 3 origina flujo de información hacia el equipo 4, por lo que éste se ubicará en el nivel 3, el número del equipo base asignado al lugar 4 será 3. Si continuamos este procedimiento se detectará la



	1	6
---NIVEL 0	2	1
---NIVEL 1	3	2
---NIVEL 2	4	3
---NIVEL 3	5	4
---NIVEL 4	6	2

Nº CICLO	EQUIPOS INVOLUCRADOS	TAMAÑO
1	1, 2, 6	3
2	1, 2, 3, 4	4
3	3, 4, 5	3

TITULO:

FIG. 3.5 APLICACIÓN DE LA ESTRUCTURA DE ARBOLES EN LA DETECCIÓN DE CICLOS.

TESIS PROFESIONAL

JOSE MARIANO RAMOS OLMOS

E.N.E.P ZARAGOZA U.N.A.M.

existencia de 3 ciclos que están formados por los equipos: (1, 2, 6), (1, 2, 3, 4) y (3, 4, 5).

c) Condensación de ciclos:

Se puede afirmar que todo ciclo forma parte de un ciclo máximo y como consecuencia, las unidades que forman a éste.- El desarrollo de los ciclos máximos se inicia suponiendo - que las unidades de un ciclo formarán parte de un ciclo máximo.

Debido a que la gráfica dirigida de un diagrama de flujo - de proceso puede contener varios ciclos, el conjunto de -- equipos que se designó a un ciclo, se incrementará en pa-- sos sucesivos hasta que éste se convierte en un ciclo máxi-- mo. La determinación de los ciclos por medio de las técni-- cas matriciales, requerirá la localización de trayectorias de diferentes tamaños y algunas de las trayectorias pueden cruzar a una unidad más de una vez, lo cual implicará que-- estos ciclos no sean independientes. Tales ciclos incre-- mentan el tamaño del problema y no interfieren en la iden-- tificación de los ciclos máximos. La aparición de estos - ciclos se puede evitar por medio de la eliminación de los-- ciclos independientes conforme se van detectando. Esto se puede lograr condensando a los equipos de estos ciclos en-- seudo-equipos y se marcará a ésta con el mismo número del-- equipo que tenga el número más alto.

El flujo desde o para alguno de los equipos con número me-- nor que el mayor en el ciclo, es considerado como el flujo desde o para este seudo-equipo. Los equipos condensados - son identificados por una etiqueta, la cual indica a qué - seudo-equipo pertenecen, mientras que el flujo entre estos equipos puede ser ignorado.

Un ciclo que en la red original pasa a través de algún equi-- po condensado, ahora se considerará que pasa por un seudo-

equipo, manteniendo de esta forma la interconexión entre ciclos.

El último paso en la condensación de ciclos, es el de formar la nueva matriz adyacente que corresponda a la red condensada.

### 3.2.2.3 ORDENAMIENTO DE EQUIPOS Y DE BLOQUES DE EQUIPOS:

La simulación de conjuntos de equipos o de equipos individuales en el proceso, debe efectuarse en alguna secuencia para poder minimizar el tiempo de cálculo para la simulación de todo el proceso. El ordenamiento de los equipos en la dirección del flujo de proceso, proporcionará la información de las corrientes de entrada para los equipos o conjuntos de equipos que se están analizando.

La simulación de los equipos que forman parte de un bloque requerirá la solución simultánea de todos los equipos que lo forman y esto a su vez, se tratará por separado. Debido a que las corrientes salientes de un equipo quedan definidas después de -- que éste ha sido simulado, es indispensable determinar la secuencia de cálculo de los bloques de equipos para poder determinar el orden de cálculo de toda la red.

### 3.2.2.4 DETERMINACION DE LOS SUBCICLOS DE UN CICLO MAXIMO:

Un ciclo máximo puede involucrar más ciclos de los que se pueden identificar a simple vista. Es necesario determinar todos estos subciclos si las corrientes de corte del conjunto de corte van a ser localizadas para formular la solución de los equipos involucrados en el ciclo máximo. Se deben considerar varios métodos para la identificación de estos subciclos. -- Los métodos matriciales utilizados para la identificación de ciclos máximos, no pueden ser empleados para la identificación de-

los subciclos debido al incremento en el número de elementos diferentes de cero en las potencias altas de la matriz adyacente y además, requerirían de una alta efectividad de cómputo para poder separar los ciclos dependientes e independientes.

Se puede utilizar la estructuración en forma de árboles de todos los equipos de un ciclo máximo, esto permitirá ignorar los ciclos que se repitan. Para esto, a cada ciclo que se localiza, se le asigna un código entero único que será comparado con el de los ciclos existentes y en caso de que éste sea igual al de alguno de los ciclos previamente identificado, se ignorará.

La estructuración del árbol, se iniciará con cualquier equipo del ciclo máximo. El siguiente equipo en la estructura será el equipo marcado con el número más pequeño que recibe flujo desde el equipo seleccionado y de la misma forma, el siguiente equipo de la estructura será el equipo etiquetado con el número más pequeño que recibe flujo desde el segundo, etc. Este procedimiento se continúa hasta que alguno de los números de los equipos se repite, lo cual nos indica que se ha identificado un subciclo.

El siguiente paso, corresponde a la rotación de los números de los equipos involucrados en el subciclo con el objeto de que el más pequeño encabece la lista y, de esta forma, desarrollar el código entero tomando como base al número de equipos que intervienen en el subciclo. Si hay  $n$  unidades en el ciclo, el código entero se desarrolla por medio de la ecuación 3.2.

$$C = U_1 \cdot b^{n-1} + U_2 \cdot b^{n-2} + \dots + U_n \quad (3.2)$$

Donde:  $U_1, U_2, \dots, U_n$  = Unidades involucradas en la secuencia en el ciclo

$b$  = Base utilizada para desarrollar el código  $C$ .

En algunas ocasiones, se realiza el rastreo de la trayectoria

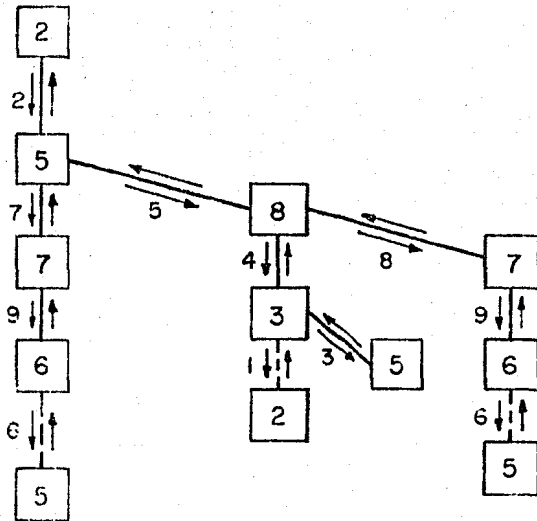
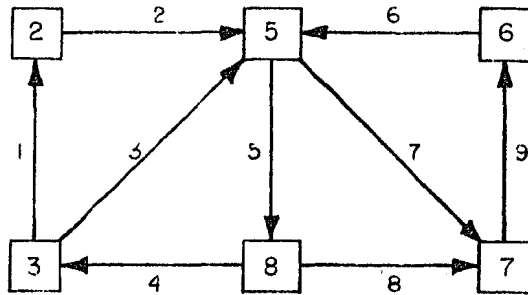
ria en sentido inverso para el equipo inicial cuando los flujos adicionales pueden ser localizados a partir de algún equipo - subsecuente. De esta forma todos los posibles ciclos tienen que ser identificados. Es posible la repetición de ciclos debido a que un equipo en el ciclo máximo que presente más de una entrada será listado repetidamente en la estructura del árbol.

Para ilustrar el procedimiento de determinación de los subciclos de un ciclo máximo se tomará como base la gráfica dirigida de la figura 3.6. Si consideramos al equipo número 2 como -- equipo inicial, la corriente 2 sale de este equipo y se dirige - al equipo 5, por lo tanto el equipo 5 y la corriente 2 son listados en la estructura. El equipo cuyo número es el más pequeño y que recibe flujo del equipo 5 es el 7 a través de la corriente - 7. De esta manera el equipo y corriente número 7 son los siguientes en la estructura. El equipo número 6 es el único que recibe flujo desde el equipo 7 por medio de la corriente 9 por lo que - el equipo 6 y la corriente 9 se colocan en la estructura de árbol después del equipo 7.

Existe flujo del equipo 6 al equipo 5 por medio de la corriente 6, sin embargo; como el equipo 5 ya se listó previamente en la estructura, se ha determinado un subciclo que involucra a los equipos 5, 7 y 6 y a las corrientes 7, 9 y 6. Como la unidad con el número más pequeño se localiza en la primera posición ya no es necesario efectuar la rotación. El código entero para el ciclo se desarrolla tomando como base 12 (número total de unidades que intervienen en la estructura y dentro de las cuales se encuentra el ciclo máximo). El código se determina como:

$$C = 5 \cdot (12)^2 + 7 \cdot (12) + 6 = 810$$

El último equipo en la estructura es el número 6 y no presenta más flujos salientes. Iniciando el rastreo de la trayectoria en sentido inverso, el equipo 7 no presenta más corrientes - salientes. El equipo 5 presenta flujo hacia el equipo 8 por medio de la corriente 5 y como consecuencia el equipo 8 y la co -



TITULO:

FIG. 3.6 RASTREO DE LAS TRAYECTORIAS DE LA ESTRUCTURA DE ARBOL PARA LA IDENTIFICACION DE UN CICLO MAXIMO.

TESIS PROFESIONAL

JOSE MARIANO RAMOS OLMOS

E.N.E.P. ZARAGOZA U.N.A.M.

corriente 5 son listados en la estructura. La corriente 4 sale -- del equipo 8 y conduce al equipo número 3 cuyo número es el menor, por consiguiente el equipo 3 y la corriente 4 son listados en la estructura.

El flujo desde el equipo 3 al equipo 2 es por medio de la corriente 1 identificándose de esta forma otro subciclo debido a que el equipo número 2 ya había sido listado en la estructura. Los equipos que forman este subciclo son 2, 5, 8 y 3, involucrando a las corrientes 2, 5, 4 y 1, el código entero del subciclo será 4275 y se determinó de:

$$C = 2 \cdot (12)^3 + 5 (12)^2 + 8 \cdot 12 + 3 = 4275$$

Desde el equipo 3, que es el último equipo de la estructura, existe flujo hacia el equipo 5 a través de la corriente 3, pero como el equipo 5 ya había sido listado en la estructura, se ha identificado un subciclo más que está formado por los equipos 5, 8 y 3 y por las corrientes 5, 4 y 3. En este caso sí es necesaria la rotación para ubicar a la unidad con número más bajo en la primera posición produciendo la secuencia 3, 5 y 8. El código para este subciclo es 500.

$$C = 3 \cdot (12)^2 + 5 (12) + 8 = 500$$

### 3.2.3 METODO DE WESTERBERG:

La modificación hecha por Westerberg (1977) al algoritmo de Forder y Hutchinson (1969), puede ser usado para determinar los subciclos de un ciclo máximo.

En este algoritmo, se define una lista L1 en donde se representan el origen y destino de la información de cada una de las corrientes. Cualquier entrada a L1 contiene el nombre de la corriente así como el origen y destino de la misma. Se define L2 como una lista que contiene información acerca de los sucesos



res inmediatos de cualquier corriente. Una entrada a L2 está -- comprendida de un nombre para la corriente, seguida por todos -- los nombres de todas las corrientes de salida de la unidad de -- proceso donde llegó la corriente original. L3 puede ser una lis- ta de los nombres de las unidades de proceso y de sus etiquetas- asociadas. Ocasionalmente, una unidad puede ser etiquetada con- un nombre de corriente en la lista L3.

En adición a esto, uno requiere una lista L5 para almace- nar todas las recirculaciones detectadas y un vector de trabajo- L4. En la figura 3.7 se ilustra el diagrama de flujo de este al goritmo. La descripción del mismo se presenta a continuación.

#### ALGORITMO:

- 1.- Construir L1
- 2.- Construir L2
- 3.- Construir L3
- 4.- Seleccionar la primer corriente  $S_1$  para L2, esta corriente- debe tener al final un sucesor no cotejado.  
Establecer la corriente considerada  $S_c$  como  $S_1$
- 5.- Colocar  $S_c$  como primer elemento de  $L_4$
- 6.- ¿Está asignado el destino de  $S_c$ ? Si lo está, vaya a 11.- - de lo contrario continúe
- 7.- Si  $S_c$  no tiene ningún sucesor no cotejado vaya a 12.-
- 8.- Etiquete el origen de  $S_c$  mediante la entrada del nombre de-  $S_c$  en L3
- 9.- Marque el primer sucesor de  $S_c$  no cotejado como cotejado. - Sustituya el sucesor por  $S_c$
- 10.- Vaya a 5.-
- 11.- Existe una recirculación entre el destino y la fuente de  $S_c$  compare la recirculación con las recirculaciones que ya han sido registradas en L5. Si la recirculación no había sido- registrada, regístrela
- 12.- Remueva  $S_c$  de L4. No etiquete en L3 la unidad que actual-- mente está etiquetada como  $S_c$ . Si L4 está lleno vaya a 15.- de lo contrario continúe

- 13.- La primera corriente de  $L4$  llega a ser la corriente considerada  $S_c$
- 14.- Vaya a 6.-
- 15.- No etiqueta  $L3$ , vaya a 4.-

EJEMPLO:

Encontrar todos los ciclos fundamentales del problema de Cavett ilustrado en la figura 3.3.

SOLUCION:

Las listas  $L1$ ,  $L2$  y  $L3$  se definen como:

LISTA L1:

Corriente	1	2	3	4	5	6	7	8
Origen	6	1	2	3	4	4	2	5
Destino	1	2	3	4	5	1	6	3

LISTA L2:

Corriente	1	2	3	4	5	6	7	8
Sucesores	2	3	4	5	8	2	1	4
		7		6				

LISTA L3: (No etiquetada aún)

Unidad	1	2	3	4	5	6
Etiqueta						

PASOS 4 y 5       $L4 = 1$

- PASOS 6 y 7 Continúan hacia el paso 8. Etiquete la unidad 6 con 1
- PASO 9 Marque la corriente 2 como cotejada (en la lista de sucesores de la corriente 1)
- PASO 5 L4 = 1, 2
- PASO 8 Etiquete la unidad 1 con 2
- PASO 9 Marque la corriente 3 como cotejada en la lista de sucesores de la corriente 2
- PASO 5 L4 = 1, 2, 3
- PASO 8 Etiquete la unidad 2 con 3
- PASO 9 Marque la corriente 4 como cotejada en la lista de sucesores de la corriente 3.
- PASO 5 L4 = 1, 2, 3, 4
- PASO 8 Etiquete la unidad 3 con 4
- PASO 9 Marque la corriente 5 como cotejada en la lista de sucesores de la corriente 4
- PASO 5 L4 = 1, 2, 3, 4, 5
- PASO 8 Etiquete la unidad 4 con 5
- PASO 9 Marque la corriente 8 como cotejada en la lista de sucesores de la corriente 5

En este punto, las listas L2 y L3 están como:

LISTA L2

Corriente	1	2	3	4	5	6	7	8
Sucesor	2*	3*	4*	5*	8*	2	1	4
				6				

LISTA L3

Unidad	1	2	3	4	5	6
Etiqueta	2	3	4	5		1

PASO 5 L4 = 1, 2, 3, 4, 5, 8

PASO 6 El destino de la corriente 8 es la unidad 3 que está etiquetada con la corriente 4

PASO 11 Existe una recirculación entre las corrientes 4, 5 y 8  
L5 = (4, 5, 8)

PASO 12 L4 = 1, 2, 3, 4, 5

PASO 13  $S_C = 5$ . No etiquete la unidad 4 en L3

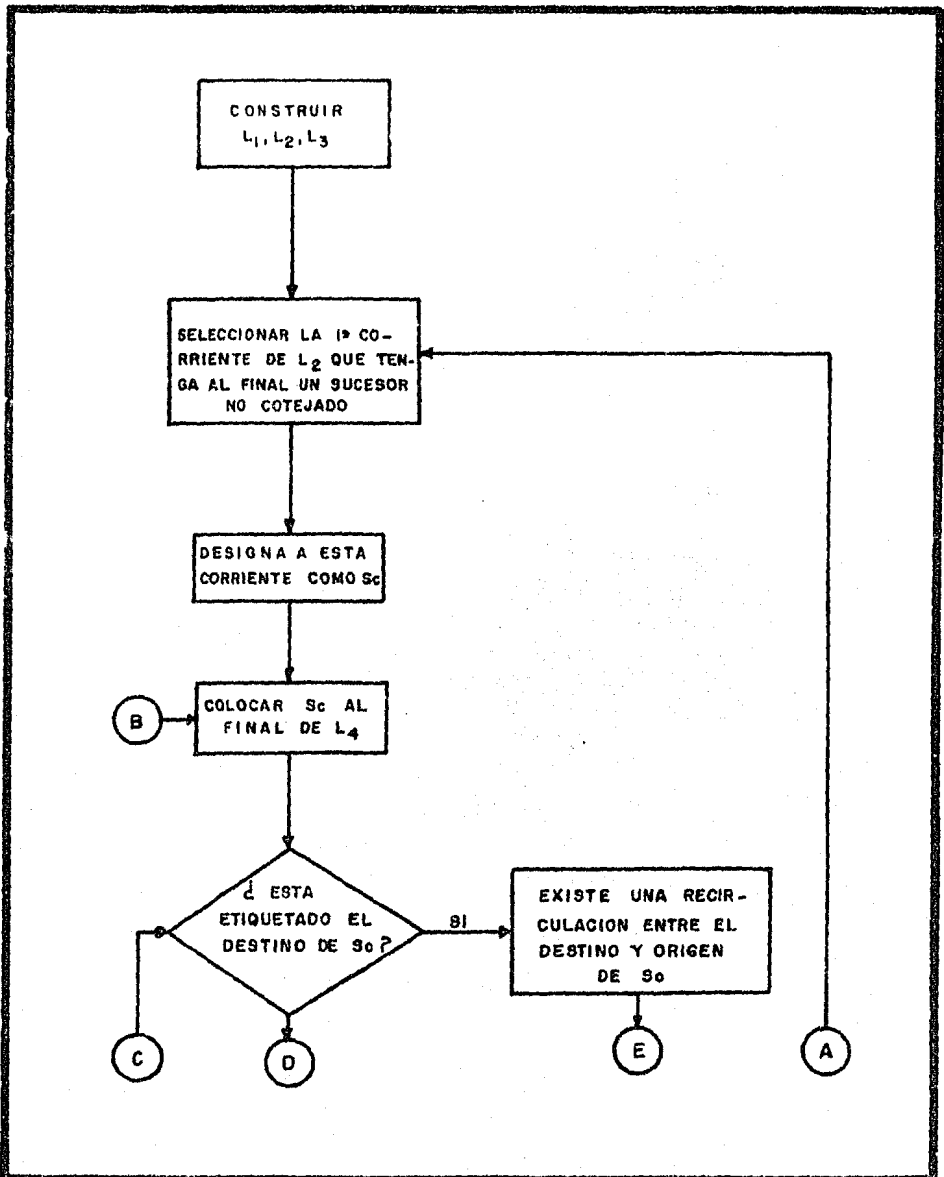
PASOS 6, 7 Vaya al paso 12.-  
Etc., Etc.

De esta manera, fueron encontradas 3 recirculaciones por el algoritmo que son:

- 1.- corrientes 4, 5, 8
- 2.- corrientes 2, 3, 4, 6
- 3.- corrientes 1, 2, 7

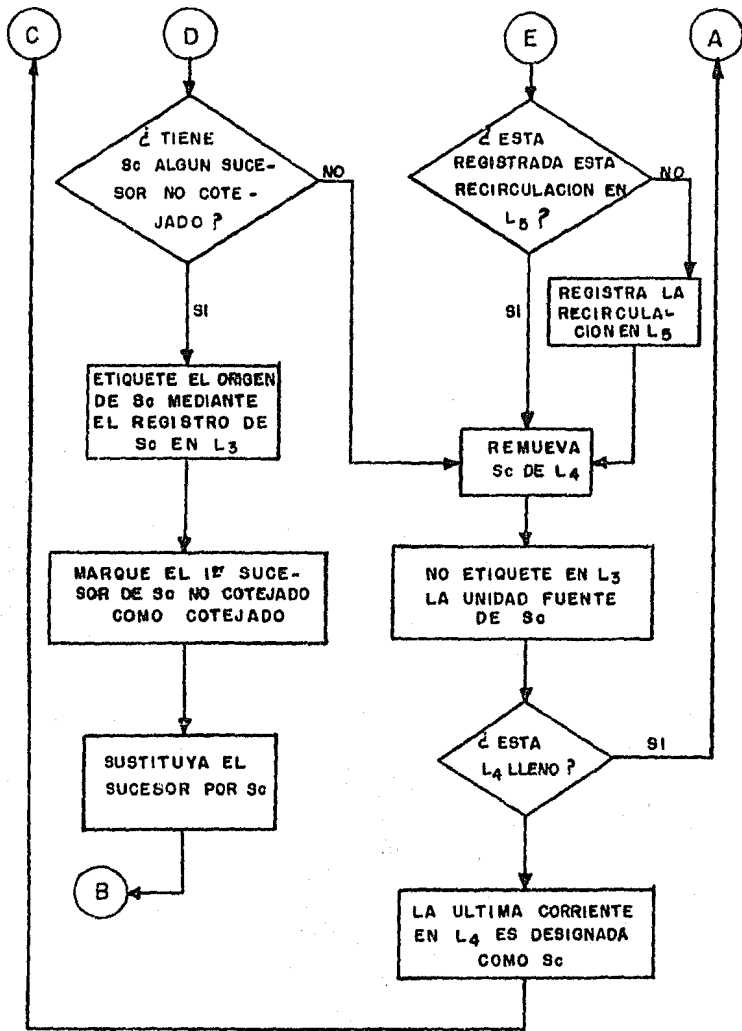
### 3.3 ROMPIMIENTO DE CICLOS:

Existen muchos algoritmos para encontrar el conjunto de corte para romper todas las recirculaciones de una gráfica dirigida de un proceso. Usualmente, el objetivo de estos algoritmos puede ser alguno de los que se enlistan a continuación:



TITULO:  
 FIG. 3.7 ALGORITMO DE WESTERBERG (1977) PARA IDENTIFICACION DE CICLOS.

TESIS PROFESIONAL  
 JOSE MARIANO RAMOS OLMOS  
 E.N.E.P. ZARAGOZA U.N.A.M.



TITULO I  
 FIG. 3.7 (CONT.) ALGORITMO DE WESTERBERG  
 (1977)

TESIS PROFESIONAL

JOSE MARIANO RAMOS OLMOS

E.N.E.P. ZARAGOZA U.N.A.M.

- 1.- Encontrar un conjunto de corte con el número mínimo de corrientes de corte (por ejemplo Barkeley y Motard (1972)).
- 2.- Encontrar un conjunto de corte con la suma total mínima de los factores de peso de las corrientes de corte (por ejemplo Pho y Lápidus (1973)).
- 3.- Encontrar un conjunto de corte con las mejores propiedades de convergencia (por ejemplo Upadhye y Grens (1975)).

En los métodos de rompimiento, el conjunto de variables independientes conocido con el nombre de conjunto de corte, se selecciona a partir del conjunto de variables inicial. Por medio de un análisis de la estructura del sistema de ecuaciones generado. Los procedimientos de rompimiento más frecuentemente utilizados, deben ser orientados hacia la obtención de un conjunto de corte con el menor tamaño posible en el sentido de un número mínimo de cortes o de un mínimo número de variables tal como lo hacen notar Kehat y Schacham (1973).

Varios autores han realizado investigaciones acerca de los métodos para rompimiento de ciclos. Ledet y Himmelblau (1970) - obtienen un conjunto de corte mínimo así como su correspondiente conjunto de variables de salida. Estos conjuntos son obtenidos por medio del reordenamiento de la matriz de incidencia que a su vez, representa la estructura de un sistema de ecuaciones. El método consta de dos fases:

- 1) Ordenar una por una las hileras de la matriz de incidencia de acuerdo con algún criterio de optimización.
- 2) Calcular las variables de salida para las ecuaciones al mismo tiempo de estar efectuando el ordenamiento de las hileras.

Con esto, se logra obtener un conjunto de corte mínimo, este método difiere de los demás desde el momento en que se requiere la especificación de un conjunto de variables de salida para su sistema de ecuaciones.

Steward (1965), Sargent y Westerberg (1964), Christiansen y Rudd (1969), Barkley y Motard (1972) y Upadhye y Grens (1972, 1975) consideran el rompimiento de ecuaciones por bloques y estos conjuntos de ecuaciones ya tienen definidas sus variables de salida.

Los algoritmos de Steward (1965) y Barkley y Motard se basan en la determinación de conjuntos de corte que contengan el número mínimo de corrientes de corte. El algoritmo de Steward se inicia con la formación de una matriz de incidencia y el orden de las ecuaciones en cada conjunto de ecuaciones se puede identificar por medio de los elementos de la matriz. Las trayectorias dentro de un bloque se enumeran durante el desarrollo de una estructura de árbol máxima tomando como base la información que proporciona la matriz incidente. La corriente de corte para cada ecuación del sistema se selecciona de tal manera que sea diferente a la variable de salida. La efectividad de cada corte se evalúa de acuerdo al tamaño de los subsistemas de ecuaciones resultantes y el conjunto mínimo de corte se obtiene por medio del análisis de todas las ecuaciones en el sistema para posibles cortes.

En el método de Barkley y Motard, la gráfica dirigida que representa a una recirculación se reduce en intervalos. Los ciclos que se repiten en la gráfica dirigida reducida se eliminan pero, se debe asignar algunas de las corrientes al conjunto de corte. Cuando ya no se puede reducir más la gráfica es necesario seleccionar una corriente heurísticamente y asignarla al conjunto de corte. El hecho de seleccionar las corrientes de corte heurísticamente puede repercutir en que el conjunto de corte obtenido no sea mínimo.

Para obtener el mejor conjunto de corte se necesita definir ya sea, la minimización del número total de variables asociadas con las corrientes de corte o, la minimización de la suma de los factores de peso de las corrientes del conjunto.

Lee y Rudd (1966) hacen uso de una matriz de ciclos que --



contiene a las corrientes involucradas en cada ciclo de la red - máxima para determinar lo que ellos llamaron "cortes óptimos".

En el momento en el que se identifica un ciclo que contenga únicamente una corriente en la matriz de ciclos, se puede romper incluyendo a esta corriente en el conjunto de corte. De esta forma todos los ciclos son rotos. En el caso de que persistan ciclos en la matriz después de haber aplicado repetidamente el algoritmo, una de las corrientes restantes deberá ser elegida y dividida en pseudo-corrientes. Este hecho trae como consecuencia que solamente se puedan usar factores de peso enteros ya que cada uno de estos factores requiere de una gran cantidad de operaciones para analizar todas las corrientes, con el objeto de determinar si éstas están contenidas en otras corrientes de la red de recirculación.

Sargent y Westerberg (1964) y Upadhye y Grens (1975) aplicaron los principios de la programación dinámica para sus métodos de selección de cortes óptimos. En el caso de Sargent y Westerberg, todas las corrientes de recirculación en una secuencia óptima de los equipos, son asignadas al conjunto de corte. La secuencia óptima de los equipos se obtiene de tal forma que la suma de los factores de peso de las corrientes de recirculación sea mínima. Se inicia con los subgrupos que están formados por un solo equipo y todos los posibles subgrupos de mayor tamaño -- son generados sucesivamente manteniendo una secuencia óptima de los equipos en cada subgrupo.

El tamaño del problema de programación dinámica es reducido mediante la eliminación de los subgrupos de una unidad de la red.

Upadhye y Grens (1975) han demostrado que puede ser encontrado un conjunto de corte con propiedades óptimas de convergencia con respecto a la substitución directa. El resultado obtenido por ellos fué denominado La Regla del Reemplazamiento. Los conjuntos de corte para un diagrama de flujo de un proceso parti

cular puede ser dividido en familias, cada miembro de la familia tiene características similares de convergencia cuando se emplea el método de substitución directa. Dado un conjunto de corte, - uno puede usar la regla del Reemplazamiento para encontrar a otro miembro de la familia. Upadhye y Greens han clasificado a las familias como:

- 1) No redundantes
- 2) Redundantes
- 3) Compuestas

Una familia no redundante es aquella en la que las corrientes que la forman, cortan una sola vez a cada uno de los ciclos de la red.

Una familia redundante es aquella en la cual las corrientes que la forman, cortan en más de una ocasión a algún ciclo de la red.

Por último, una familia compuesta es aquella que contiene corrientes redundantes y no redundantes.

Un procedimiento para encontrar un conjunto de corte no-redundante podría ser:

- 1) Encuentra cualquier conjunto de corte.
- 2) Usar la regla del reemplazamiento para encontrar otros miembros de la familia. Si una corriente aparece más de una vez en el conjunto de corte en cualquier etapa, borra todo un suceso. La familia que tenga repetida una corriente de corte es una familia redundante. El nuevo conjunto seleccionado es una nueva familia.
- 3) Repite el paso 2 hasta que la familia obtenida no contenga corrientes repetidas en su conjunto de corte.

Christiansen y Rudd (1969) proponen un método heurístico para el rompimiento en el cual, se asigna un peso a cada corriente

te, este factor de peso puede reflejar la dificultad esperada -- asociada con el rompimiento de la corriente y puede ser igual al número de los valores más importantes de las variables de corte en cada corriente.

El algoritmo se inicia con la fusión de las corrientes paralelas, eliminación de los equipos que presenten uniones simples, rompimiento de una de las corrientes de doble dirección y pasando por alto a las corrientes inelegibles para propósitos de rompimiento. En algunas ocasiones, el conjunto de corte óptimo se obtiene aplicando repetitivamente los pasos indicados anteriormente. En caso de no poder determinar el conjunto de corte por este procedimiento, se debe seleccionar un equipo indicador heurísticamente y eliminarlo de la red junto con sus corrientes conectantes.

El equipo indicador es definido como un equipo en el cual, todas sus corrientes de entrada o todas sus corrientes de salida, son elegidas para el rompimiento, una vez detectado un equipo indicador, sus corrientes son asignadas al conjunto de corte y algoritmo procede desde este punto repitiéndose todos los pasos mencionados anteriormente hasta obtener el conjunto de corte final.

El algoritmo de Pho y Lapidus (1973) es presentado por Rosen (1980) como se indica a continuación:

- a) Asignar todas las  $n$  corrientes a  $(0,1)$  variables  $X_i$  -- donde  $i = 1, n$
- b) Identificar todas las recirculaciones y construir la matriz de ciclos  $A$  donde:

$$A_{j,i} = \begin{cases} 0 & \text{Si la corriente } i \text{ no está en la} \\ & \text{recirculación } j \\ 1 & \text{Si la corriente } i \text{ está en la} \\ & \text{recirculación } j \end{cases}$$

- c) Resolver el problema de programación lineal entera.

$$\text{Min.} \quad \sum_{i=1}^n P_i X_i \quad (3.3)$$

$$\text{S.t.} \quad \sum_{i=1}^n a_{ji} X_i = 1 \quad (3.4)$$

Donde cada recirculación es rota por lo menos una vez. El problema de programación entera puede ser resuelto usando la estructuración de árboles después de que las corrientes redundantes y las recirculaciones son desechadas. Los factores de peso  $P_i$  en la función objetivo, pueden ser seleccionados para hacer que satisfagan uno de los siguientes criterios:

- 1) Menor número de recirculaciones rotas  $P_i = 1$
- 2) Menor número de recirculaciones rotas  $P_i$  es el número de corrientes rotas
- 3) Menor número de variables de corte  $P_i$  es el número de variables por corriente
- 4) Factor de peso especificado por el usuario basado en las dificultades esperadas en el rompimiento de una corriente

Por último, el algoritmo de Mahalec (1978) puede ser usado para encontrar un conjunto de corte para todas las recirculaciones en un diagrama de flujo de proceso. En la figura 3.3, se ilustra el diagrama de flujo de este algoritmo.

En este algoritmo se define a *Loops* como una lista que contiene el número de recirculaciones así como las corrientes que las forman. Se define *Itear* como una lista que contiene los números de las recirculaciones y las corrientes de corte para cada una de estas recirculaciones. Por último se define *Ijtear* como una lista que contiene las corrientes preferidas de corte.

El presente algoritmo se describe a continuación:

- 1.- Construir Loops.
- 2.- ¿Existen algunas corrientes preferidas? Si las hay, continúe, de lo contrario vaya a 6.
- 3.- Tome una corriente de Ijtear y defínala como la corriente considerada Sc.
- 4.- Encuentre las recirculaciones en Loops que contengan a Sc. Esta será una corriente de corte para esa recirculación. Coloque el número de la recirculación y Sc en Ijtear marque la recirculación en Loops como "está - siendo rota".
- 5.- ¿Está Ijtear llena? Si lo está, continúe; de lo contrario tome la siguiente corriente de Ijtear como Sc y vaya a 4.
- 6.- Escoge la primera corriente de la primera recirculación que aún no ha sido rota en Loops. Establezca la corriente considerada como Sc. La corriente de corte para esta recirculación es Sc. Coloque la corriente Sc y el número de la recirculación en Ijtear.
- 7.- ¿Está Sc contenida en alguna otra recirculación de Loops que aún no ha sido rota? Si lo está, Sc es una corriente de corte para esta recirculación; coloque el número de la recirculación y Sc en Ijtear y continúe. de lo contrario vaya a 10.
- 8.- ¿Se puede encontrar una corriente para una recirculación en Loops aún cuando no se haya roto la recirculación? Si se puede, continúe, de otra manera vaya a 10.
- 9.- Establezca la corriente considerada como Sc. La corriente de corte considerada para la recirculación en Loops, en la cual Sc fué encontrada es Sc. Coloque Sc y el número de recirculación en Ijtear y vaya a 7.
- 10.- Selecciona la primera corriente para la primera recirculación que aún no ha sido rota en Loops. Establezca la corriente considerada como Sc. La corriente de corte para la recirculación es Sc. Coloque a Sc y al número de la recirculación en Ijtear.
- 11.- Localiza las recirculaciones rotas en Loops que contengan

gan a  $S_c$  que es una corriente de corte para estas recirculaciones. Coloca a  $S_c$  y al número de la recirculación en ITFAR.

- 12.- Establece una etiqueta para denotar que este conjunto de corte es no-redundante.
- 13.- Ve a 7.

### 3.4. CALCULOS DE CONVERGENCIA:

Una vez que el conjunto de corte ha sido determinado, se procede a hacer los cálculos secuencialmente con las variables de corte supuestas y recalculadas a cada iteración para obtener la convergencia del sistema de ecuaciones.

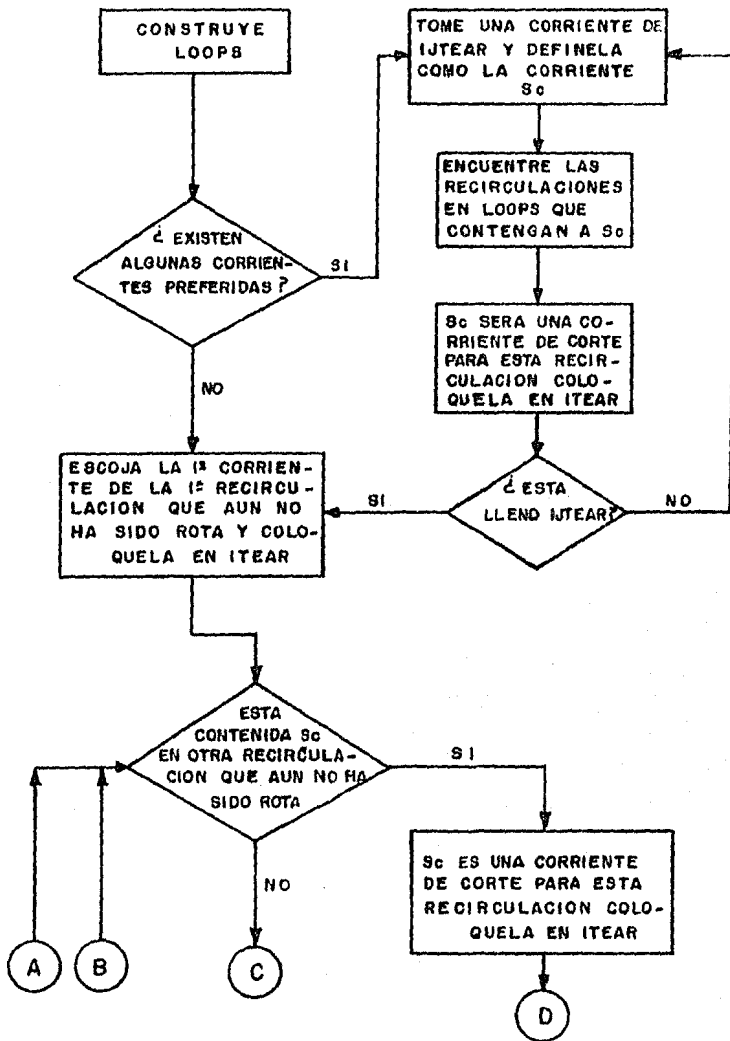
Todos los algoritmos de convergencia toman la forma:

$$x^{i+1} = x^i - t^i j^i f^i \quad (3.5)$$

Donde:  $x^i$  = Variable de rompimiento en la iteración  $i$ .  
 $f^i = x^i - r^i$  = Diferencia entre las variables de corte calculadas a cada iteración y las variables supuestas.  
 $t^i$  = Factor escalar de amortiguamiento, usualmente se le asigna el valor de 1.

Los algoritmos de convergencia difieren en la selección de la matriz  $J^i$ . Si la matriz es seleccionada como I (matriz identidad) el algoritmo será una substitución sucesiva que es una técnica lenta pero estable. El algoritmo de Newton requiere que  $J^i$  sea el inverso del Jacobiano de  $f^i$  con respecto a  $x_i$  y así sucesivamente.

Por otra parte, los métodos utilizados para resolver sistemas de ecuaciones no lineales, se pueden clasificar en dos grupos diferentes que son:



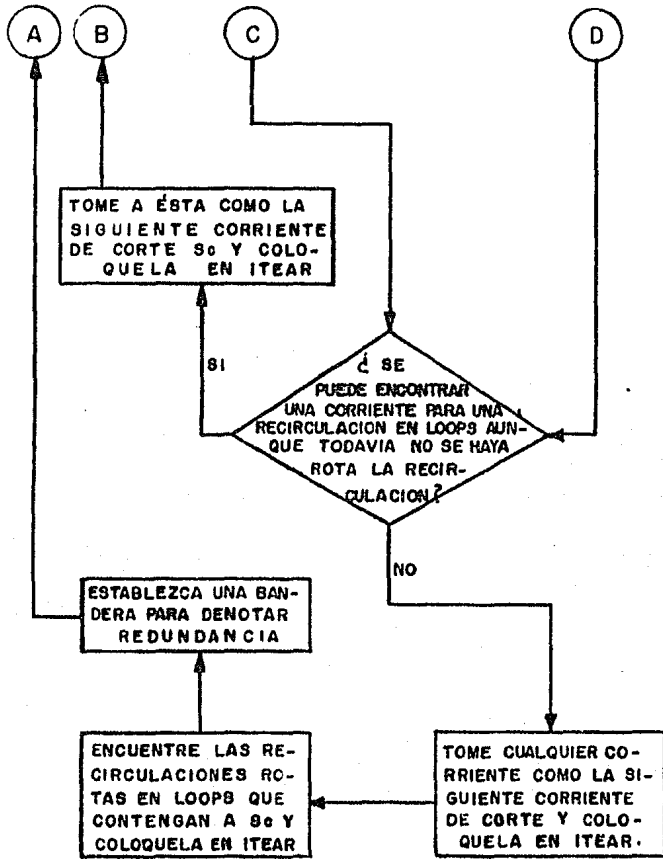
TITULO:

FIG. 3.8 ALGORITMO DE MAHALEC (1978) PARA DETECTAR UN CONJUNTO DE CORTE.

TESIS PROFESIONAL

JOSE MARIANO RAMOS OLMOS

E.N.E.P. ZARAGOZA U.N.A.M.



TITULO:

FIG. 3.8 (CONT.) ALGORITMO DE MAHALEC (1978).

TESIS PROFESIONAL

JOSE MARIANO RAMOS OLMOS

E.N.E.P ZARAGOZA U.N.A.M.



- a) Métodos de linearización directa
- b) Métodos indirectos

Los métodos de linearización directa se aplican en los casos en donde se cuente con las ecuaciones en forma explícita, -- mientras que los métodos indirectos se aplicarán a todos los casos donde se tengan las ecuaciones en forma implícita.

### 3.4.1 MÉTODOS DE LINEARIZACIÓN DIRECTA:

Estos métodos se basan en la representación del sistema de ecuaciones no lineales por medio de un sistema equivalente de -- ecuaciones lineales y como consecuencia, en la aplicación de los algoritmos de solución de sistemas de ecuaciones lineales.

Esta representación involucra una serie de aproximaciones -- y al mismo tiempo, dá origen a un proceso iterativo en el cual, -- a cada iteración las ecuaciones lineales van a llegar a ser una -- mejor aproximación de las ecuaciones no lineales. Es de suma im -- portancia la selección del sistema equivalente de ecuaciones li -- neales y de sus respectivos valores iniciales ya que de esto, de -- penderán los resultados que se obtengan.

#### 3.4.1.1 LINEARIZACIÓN:

La linearización consiste en la transformación de una ecua -- ción diferencial no lineal que describe a un sistema en una ecua -- ción diferencial lineal. Una ecuación diferencial lineal, es -- una ecuación que contiene variables de primer orden en solo uno -- de los términos de la ecuación (por ejemplo 3.6).

$$a_1 \frac{dx}{dt} + a x = \quad (3.6)$$

Para linearizar una función no lineal, únicamente se necesita expandir por medio de la serie de Taylor alrededor del punto de interés y despreciar todos los términos después de la primera derivada parcial.

Supongamos que tenemos una función no lineal  $f$  con las variables de proceso  $x_1$  y  $x_2$ . Si se expande alrededor de los valores  $\bar{x}_1$  y  $\bar{x}_2$  por medio de la serie de Taylor se obtiene:

$$f(x_1, x_2) = f(\bar{x}_1, \bar{x}_2) + \left(\frac{\partial f}{\partial x_1}\right)_{(\bar{x}_1, \bar{x}_2)} (\bar{x}_1 - \bar{x}_1) + \left(\frac{\partial f}{\partial x_2}\right)_{(\bar{x}_1, \bar{x}_2)} (\bar{x}_2 - \bar{x}_2) \quad (3.7)$$

En la figura 3.9, se ilustra gráficamente el proceso de linearización para una función de una variable. La forma generalizada para construir un método de linearización de un sistema de ecuaciones se presenta a continuación:

Supongamos que se tiene un sistema de  $n$  ecuaciones no lineales de la forma:

$$F(x) = 0 \quad (3.8)$$

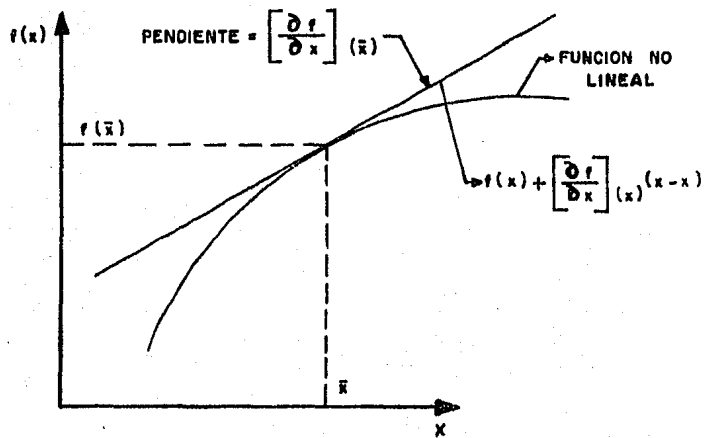
Donde:  $F$  = Vector de funciones

$x$  = Vector de variables independientes

O bien expresado como:

$$f_1(x_1, x_2, \dots, x_n) = 0 = x_1$$

$$\dots \dots \dots$$



TITULO:

FIG. 3.9 LINEARIZACION DE UNA FUNCION DE UNA SOLA VARIABLE.

TESIS PROFESIONAL

JOSE MARIANO RAMOS OLMOS

E.N.E.P ZARAGOZA U.N.A.M.

$$f_n(x_1, x_2, \dots, x_n) = 0 = x_n$$

Si expandemos a  $y_i$  por medio de series de Taylor alrededor de  $x_i$  y despreciando los términos después de la primera derivada se obtiene:

$$y_1 = f_1^i + (x_1 - x_1^i) \left\{ \frac{\partial f_1}{\partial x_1} \right\}_{x_i} + \dots + (x_n - x_n^i) \left\{ \frac{\partial f_1}{\partial x_n} \right\}_{x_i}$$

$$y_n = f_n^i + (x_1 - x_1^i) \left\{ \frac{\partial f_n}{\partial x_1} \right\}_{x_i} + \dots + (x_n - x_n^i) \left\{ \frac{\partial f_n}{\partial x_n} \right\}_{x_i}$$

Si se sustituye  $x^i$  por  $x^{i+1}$  y se supone que  $x^{c+1}$  es el siguiente valor y  $y^{i+1} = 0$ , se obtiene una fórmula de recurrencia representada en forma matricial que se conoce como método de Newton.

$$x_{N.R.}^{i+1} = x^i - (J^i)^{-1} f^i \quad (3.9)$$

Donde:  $(J^i)^{-1}$  = Inverso del Jacobiano de  $f^i$  con respecto a  $x_i$   
 $f^i = 1.0$

Aunque este método requiere de pocas iteraciones y garantiza la convergencia cuadrática cercana a la solución, ha sido usado en pocas ocasiones por la dificultad para calcular las primeras derivadas (Kehat y Shacham (1973)). Una técnica para evaluar las derivadas parciales es por medio de diferencias finitas.

Las limitaciones del método de Newton son:

- El algoritmo no converge si el estimado inicial para el vector  $x^i$  está muy lejano del vector solución.
- No converge si el Jacobiano resulta ser singular o cercano a la singularidad.
- Requiere de un método efectivo para el cálculo e inversión del Jacobiano.

De igual manera que en el caso del algoritmo de Newton, se

puede definir el método Quasi-Linear (C.L.) mediante la representación de  $F(x)$  como un sistema de ecuaciones lineales:

$$F(x) = Ax + b = 0 \quad (3.10)$$

La fórmula de recurrencia para el algoritmo Quasi-Linear será:

$$X_{C.L.}^{i+1} = - (A^i)^{-1} b^i \quad (3.11)$$

Si se substituye la ecuación 3.10 en la ecuación 3.9 se obtiene:

$$X_{N.R.}^{i+1} = X^i - (J^i)^{-1} A^i X^i + b^i \quad (3.12)$$

$$\text{Si } J^i \cong A^i$$

$$\begin{aligned} X_{N.R.}^{i+1} &= X^i - (A^i)^{-1} A^i X^i + b^i \\ &= (A^i)^{-1} b^i = X_{C.L.}^{i+1} \end{aligned} \quad (3.13)$$

La condición  $J^i \cong A^i$  implica que para obtener una convergencia cuadrática, es necesario y suficiente que los coeficientes de las ecuaciones lineales se seleccionen de tal forma que sean equivalentes a las derivadas parciales de las funciones no lineales con respecto a las variables apropiadas.

En términos más generales, si una de las ecuaciones presenta la forma:

$$f_k(X_1, X_2, \dots, X_n) = 0$$

Su forma linearizada será:

$$\sum_{i=1}^n \left( \frac{\partial f_k}{\partial X_i} \right)^P X_i = - f_k(X_1^P, X_2^P, \dots, X_n^P) + \sum_{i=1}^n \left( \frac{\partial f_k}{\partial X_i} \right)^P X_i^P \quad (3.14)$$

Donde los coeficientes del lado izquierdo de la ecuación y todos los términos del lado derecho de la ecuación pueden ser -- evaluados dada  $X_1^p$ , que es la pésima aproximación a la solución.

### 3.4.2. MÉTODOS INDIRECTOS:

#### 3.4.2.1. SUSTITUCION SUCESIVA:

Con frecuencia uno no cuenta con las funciones  $f$  en forma explícita o uno no puede desarrollar algebraicamente todas las derivadas parciales que se requieren. En la figura 3.10, se ilustra el esquema generalizado de iteración para cuando solamente las funciones  $f$  pueden ser evaluadas. Un ejemplo común es, si nosotros contáramos con una subrutina con la cual pudieramos evaluar una función  $f$  dados los valores para las variables  $X$ . La función podría tener dos formas diferentes:

- a) Puede proveer valores calculados para las variables, como ocurre con las variables de recirculación de una corriente para la aproximación secuencial modular.
- b) Que pueda proveer solamente el error, el cual será cero cuando son encontrados los valores correctos para las variables  $X$ .

El caso a) lo podemos representar como:

$X^{calc} = f(X^{supuesta}) = X^{supuesta}$  si  $X^{supuesta}$  es la solución.

El caso b) se representa como:

$e = g(X^{supuesta}) = 0$  si  $X^{supuesta}$  es la solución.

El caso a) siempre se puede transformar en el caso b) haciendo:

VARIABLES

AJUSTE DE LOS VALORES DE LAS VARIABLES.

FUNCIONES

EVALUACION DE LAS FUNCIONES

TITULO:

FIG. 3.10 METODOS INDIRECTOS

TESIS PROFESIONAL

JOSE MARIANO RAMOS OLMOS

E.N.E.P. ZARAGOZA U.N.A.M.

$$e = x^{\text{calc}} - x^{\text{supuesta}}$$

El caso a) se conoce como un "Loop de iteración explícito" mientras que el caso b) el método más comúnmente usado para obtener los nuevos valores de las variables para un Loop explícito, se denomina sustitución sucesiva. La base de este método es muy sencilla y consiste simplemente en utilizar los valores calculados del vector de variables para evaluar la función y de esta forma, obtener los nuevos valores para las variables, es decir:

$$x^{\text{(nueva)}} = x^{\text{(calculada)}} = f(x^{\text{supuesta}}) \quad (3.15)$$

Quando este método es aplicado a la solución de problemas de simulación de procesos a estado estacionario, la convergencia aunque es lenta está garantizada.

Haciendo referencia a la ecuación 3.5, el método de sustitución sucesiva toma la forma:

$$x^{i+1} = x^i - If^i$$

La cual, nos conduce a la ecuación 3.15. Dentro de los métodos de convergencia que representan a  $J$  como la matriz identidad tendríamos al método de Wegstein.

#### 3.4.2.2. METODO DE WEGSTEIN (1958):

El método de Wegstein es un algoritmo que acelera la convergencia para el método de sustitución sucesiva. Originalmente se desarrolló como una técnica para acelerar la convergencia de procesos iterativos aplicados a ecuaciones de una sola variable, no obstante, posteriormente se aplicó para resolver ecuaciones con varias variables.

La fórmula generalizada de recurrencia de este método es:



$$\bar{X}_{k+1} = q \bar{X}_k + (1-q) X_{k+1} \quad (3.16)$$

Esta fórmula de recurrencia se obtiene de la siguiente manera:

a) Dar un estimado inicial  $X_0$

$$X_1 = F(X_0)$$

b) Se defina  $\bar{X}$  como:

$$\bar{X}_1 = X_1$$

c) El siguiente punto  $X_2$  se tomará como:

$$X_2 = F(\bar{X}_1)$$

d)  $\bar{X}_2 = X_2$

e)  $X_3 = F(\bar{X}_2)$

En este punto se definen:

$$W = \frac{X_3 - X_2}{\bar{X}_2 - \bar{X}_1} \quad \text{y} \quad q = \frac{W}{W-1}$$

$W$  = Pendiente

$q$  = Parámetro de aceleración

f) Para obtener el siguiente valor  $\bar{X}_3$ , hacemos uso del parámetro de aceleración  $q$

$$\bar{X}_3 = q \bar{X}_2 + (1-q) X_3$$

El método de Wegstein es simplemente una extrapolación a lo largo de una línea recta entre los puntos  $X_k, F(\bar{X}_k)$  y  $\bar{X}_{k+1}, F(\bar{X}_{k+1})$ . La pendiente de esta línea es:

$$W = \frac{F(\bar{X}_k) - F(\bar{X}_{k-1})}{\bar{X}_k - \bar{X}_{k-1}} = \frac{X_{k+1} - X_k}{\bar{X}_k - \bar{X}_{k-1}}$$

Por consiguiente, la ecuación de una línea recta que pase a través del punto  $\bar{X}_k, F(\bar{X}_k)$  con una pendiente  $W$  será:

$$Y = W(\bar{X} - \bar{X}_k) + F(\bar{X}_k) \quad (3.17)$$

De su experiencia, Wegstein reportó que la convergencia -- siempre fué alcanzada y los valores del parámetro de aceleración

TABLA: 3.6 PROPIEDADES DEL PARAMETRO DE ACCELERACION,  $q$

<u>CASO DE ESTUDIO:</u>	<u>R A N G O</u>
Convergencia Monótona	$q \leq 0.0$
Convergencia Oscilante	$0.0 \leq q \leq 0.5$
Divergencia Oscilante	$0.5 \leq q \leq 1.0$
Divergencia Monótona	$q \geq 1.0$

permanecían en ciertos rangos, los cuales están asociados -- con el comportamiento iterativo de la variable tal como se ilustra en la tabla 3.6.

Para el caso de ecuaciones con varias variables, Wegstein indica que este método puede fallar si el método es aplicado a sistemas de más de 4 variables, debido a la interacción entre -- las variables.

La extensión de las ecuaciones del método de Wegstein para aplicarlo a la solución de ecuaciones de varias variables se describe a continuación:

Las pendientes se expresan como:

$$W = \left( \frac{\partial F}{\partial X} \right) - (I)^{-1} \left( \frac{\partial F}{\partial X} \right)$$

Donde  $F = F(x) = 0 =$  Vector de ecuaciones

$I =$  Matriz Identidad

$\left( \frac{\partial F}{\partial X} \right) =$  Matriz de derivadas parciales

Las derivadas parciales son evaluadas únicamente para los términos de la diagonal principal. Todos los demás términos son considerados como iguales a cero. El criterio de convergencia -- para este método se indica a continuación:

$$\text{Si } \left( \frac{\partial F}{\partial X} \right) > 1.0 \quad \text{Entonces } \left[ \left( \frac{\partial F}{\partial X} \right) + 1 \right] / \left[ \left( \frac{\partial F}{\partial X} \right) - 1 \right] > q > 1.0$$

La aplicación de este algoritmo es realmente sencillo y de hecho no requiere de gran cantidad de espacio de memoria. Sin -- embargo no considera la interacción entre las variables.

#### 3.4.2.3. METODO DE LOS VALORES PROPIOS DOMINANTES:

El método de los valores propios dominantes fué desarrollado por Orbach y Crowe (1971). Este método puede ser aplicado, -

en intervalos apropiados, para promover la convergencia de un proceso iterativo en particular. El algoritmo utilizado por este método se describe a continuación.

Si se tiene un vector  $Y$  de variables calculadas y un vector  $x$  de variables supuesta relacionados por la ecuación 3.18.

$$Y = \Psi(x) \quad (3.18)$$

Nuestro problema consistiría en determinar  $X_s$  de tal manera que  $Y_s = X_s$  para lo cual, se requiere de un proceso iterativo en donde la fórmula de recurrencia sería:

$$Y_k = \Psi(X_k) \quad (3.19)$$

Donde  $k$  = Número de la iteración

Este método se puede relacionar con los métodos de sustitución sucesiva y de Wegstein mediante la siguiente ecuación:

$$X_{k+1} = X_k + G(Y_k - X_k) = F(X_k) \quad (3.20)$$

Donde  $G$  = Matriz de Forzamiento

Si  $G$  es igual a la matriz Identidad  $G=I$ , la ecuación 3-20 representa al método de sustitución sucesiva. Si la matriz  $G$  es una matriz diagonal, la ecuación se convertirá en el método de Wegstein y aún más, si se selecciona a la matriz de forzamiento se define como:

$$G = \left[ I - \frac{\partial \Psi}{\partial x} \right]^{-1}$$

El método se transforma en el método de Newton.

Si la ecuación 3-20 se aproxima por medio de la serie de Taylor alrededor de un punto arbitrario  $X_1$  se obtiene la ecuación 3-21.

$$X_{k+1} = Ax_k + b \quad (3-21)$$

Donde  $b = F(X_L) - AX_L$

$$A = \left( \frac{\partial F}{\partial x} \right)_{x = x_L}$$

Siendo esta ecuación una ecuación diferencial lineal. La solución de esta ecuación toma la forma:

$$X_k - X_s = A_k (X_0 - X_s) \quad (3-22)$$

$$\delta \quad X_k = \sum_{j=1}^M C_j Z_j \lambda_j^k + X_s \quad (3-23)$$

Donde:  $X_s$  = Solución a estado estacionario

$\lambda$  = Valor propio de la matriz A

M = Número de elementos del vector X

K = Número de Iteración

Si todos los elementos de  $\lambda_j$  son diferentes se obtiene:

$$X_s = (I - A)^{-1} b$$

$$C_j = W_j^T (X_0 - X_s) \quad (W_j^T Z_j)$$

Donde:  $Z_j$  = Vector propio de  $\lambda_j$

$W_j$  = Columna propia de  $\lambda_j$

Si los valores propios de la matriz A se clasifican en orden descendente respecto a su magnitud absoluta, la condición necesaria y suficiente para que la ecuación 3-22 converja Es:

$$|\lambda_1| < 1 \quad (3-24)$$

Donde  $\lambda_1$  = Valor propio dominante

Quando k llega a ser muy grande,  $|(\lambda_j / \lambda_1)^k|_{j>1}$  decrece monótonamente y como consecuencia la ecuación 3-23 se aproxima a

una progresión geométrica transformándose la solución a:

$$x_k - x_s = C_1 Z_1 \lambda_1^k$$

Y por lo tanto la solución verdadera será obtenida de la siguiente forma:

$$x_s = x_{k-1} + \frac{(x_k - x_{k-1})}{1 - \lambda_1} \quad (3-25)$$

Mientras que el valor real de se puede obtener a partir de la ecuación 3.26

$$\Delta x_{k-1} = x_k - x_{k-1} = C_1 Z_1 (\lambda_1 - 1) \lambda_1^{k-1} \quad (3-26)$$

O bien, haciendo uso del cociente de las normas:

$$\|\Delta x_k\| / \|\Delta x_{k-1}\| = |\lambda_1| \quad (3-27)$$

#### 3.4.2.4 METODO GENERALIZADO DE LA SECANTE:

Al igual que en los métodos de linearización directa, el método de la secante se puede utilizar para resolver funciones no lineales por medio de la aproximación de una función lineal. Este método toma como base el principio del algoritmo del método de Newton, siendo las fórmulas de recurrencia las que se indican a continuación:

$$\Delta E = A \Delta x \quad (3.28)$$

$$A = \Delta E \Delta x^{-1} \quad (3.29)$$

$$x^{n+1} = x^n - A^{-1} e^n \quad (3.30)$$

Donde  $\Delta E$  = Variación del vector de errores de las funciones evaluado en los puntos  $n + 1$  y  $n$

$$\Delta E = e^{n+1} - e^n$$

$\Delta x$  = Variación del valor del vector de las variables de las funciones

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

El algoritmo del método de la secante consta de las siguientes etapas:

- 1) Evaluar  $e=g(x)$  para  $n+1$  valores de  $X$
- 2) Utilizar las ecuaciones 3-29 y 3-30 para estimar nuevos valores del vector de variables  $X$
- 3) Evaluar  $e=g(x)$  para los nuevos valores de las variables  $X$
- 4) Checar si el sistema converge. La convergencia se puede checar de dos formas:
  - a) Si los valores de  $e$  en el vector de errores son muy pequeños
  - b) Si el valor del vector  $X$  es muy pequeño para continuar
- 5) Si la convergencia no se ha alcanzado, reemplazar cualquier conjunto  $(X^i, e^i)$  utilizados en el inciso 2) por un nuevo conjunto  $(x, e)$ , volviendo a iterar desde el punto 2)
- 6) Si la convergencia ya se obtuvo, termina

#### 3.4.2.5 METODO DE BROYDEN (1965, 1969):

Dentro de los métodos que se derivan del método de Newton, el método más utilizado en Ingeniería Química ha sido el Método de Broyden. Este algoritmo se utiliza para resolver sistemas de ecuaciones no lineales y por similitud con los cálculos de las recirculaciones, se ha utilizado como método de convergencia.

El Método de Broyden (como todos los métodos Quasi-Lineales), elude el cálculo de una nueva matriz Jacobiana en cada iteración. A cambio de esto, este método parte de una aproximación del inverso de la matriz Jacobiana (por lo general será la matriz Identidad). Dicho inverso será calculado a cada iteración haciendo uso del método de la secante. A medida que la aproximación del inverso sea próxima al inverso verdadero, el Método de Broyden experimentará la convergencia cuadrática del Método de Newton.

Existe una relación muy estrecha entre el método generalizado de la secante y el método de Broyden. La fórmula generalizada de recurrencia de este método se obtiene como se indica a continuación:

Supongamos que se tiene un estimado para la matriz A. De la ecuación 3.29 que denominaremos  $A^{(1)}$  y supongamos que se evalúa  $\Delta e^{(1)}$  para un  $\Delta x^{(1)}$  especificado. Una vez determinados  $\Delta e^{(1)}$  y  $\Delta x^{(1)}$  podemos obtener una nueva matriz Identidad  $A^{(2)}$  a partir de la ecuación 3.31 y 3.32.

$$\Delta e^{(1)} = A^{(2)} \Delta x^{(1)} \quad (3.31)$$

$$A^{(2)} = A^{(1)} + u^{(1)} (v^{(1)})^T \quad (3.32)$$

Donde la matriz  $u^{(1)} (v^{(1)})^T$  es el producto exterior de los vectores  $u^{(1)}$  y  $v^{(1)}$

Substituyendo la ecuación 3.32 en la ecuación 3.31 se obtiene la ecuación 3.33.

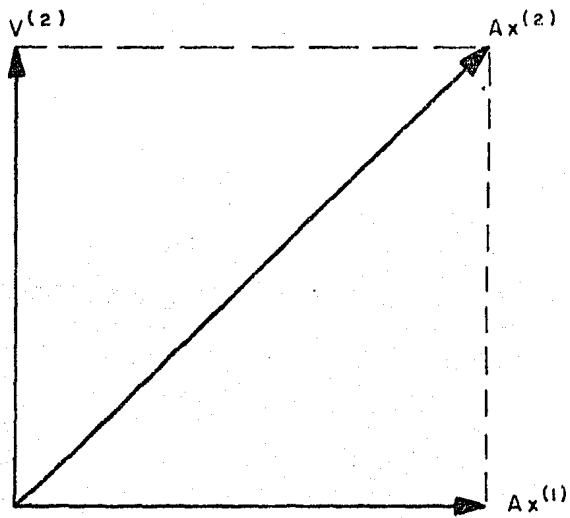
$$\Delta e^{(1)} = \left[ A^{(1)} + u^{(1)} (v^{(1)})^T \right] \Delta x^{(1)} \quad (3.33)$$

La siguiente etapa del algoritmo sería determinar la siguiente iteración o sea evaluar  $\Delta e^{(2)}$  para una  $\Delta x^{(2)}$  ( $x^{(2)}$ ) se puede obtener como resultado de resolver la ecuación (3.30).

$$x^{(2)} = x^{(1)} - (A^{(2)})^{-1} e^{(1)}$$

$$\Delta x^{(2)} = x^{(2)} - x^{(1)}$$





TITULO:

FIG. 3.11 ORTOGONALIZACION DE  $V^{(2)}$  CON  
RESPECTO A  $Ax^{(1)}$  A TRAVES  
DE  $Ax^{(2)}$

TESIS PROFESIONAL

JOSE MARIANO RAMOS OLMOS

E.N.E.P. ZARAGOZA U.N.A.M.

Para obtener  $\Delta e^{(2)}$  basta aplicar la ecuación 3.31 y 3.32.

$$\Delta e^{(2)} = A^{(3)} \Delta x^{(2)} = \left[ A^{(2)} + u^{(2)} (v^{(2)})^T \right] \Delta x^{(2)} \quad (3.34)$$

Ahora se requiere que el tercer estimado de la matriz  $A^{(3)}$  satisfaga a  $\Delta e^{(1)}$ ; esto quedará expresado como la ecuación 3.35.

$$\Delta e^{(1)} = A^{(3)} \Delta x^{(1)} = \left[ A^{(2)} + u^{(2)} (v^{(2)})^T \right] \Delta x^{(1)} \quad (3.35)$$

Si el término  $u^{(2)} (v^{(2)})^T \Delta x^{(1)}$  fuera un vector de ceros, la ecuación 3.35 se reduciría a la ecuación 3.31 con lo cual comprobamos que  $A^{(3)}$  sí satisface a  $\Delta e^{(1)}$ . Para cumplir este requisito, basta seleccionar el vector  $u^{(2)}$  ortogonal a  $\Delta x^{(1)}$ .

Para obtener la siguiente evaluación de  $\Delta e^{(3)}$  a partir de  $\Delta x^{(3)}$  se procederá como se indica:

$$\Delta e^{(3)} = A^{(4)} \Delta x^{(3)} = \left[ A^{(3)} + u^{(3)} (v^{(3)})^T \right] \Delta x^{(3)}$$

Donde:  $A^{(4)}$  necesitará satisfacer  $\Delta e^{(1)}$  y  $\Delta e^{(2)}$ ; esto es:

$$\begin{aligned} \Delta e^{(1)} &= A^{(4)} \Delta x^{(1)} \\ \Delta e^{(2)} &= A^{(4)} \Delta x^{(2)} \end{aligned}$$

$$\text{con } A^{(4)} = A^{(3)} + u^{(3)} (v^{(3)})^T$$

En este caso, el vector  $u^{(3)}$  deberá ser seleccionado de tal forma que sea ortogonal a  $\Delta x^{(1)}$  y  $\Delta x^{(2)}$ . De todo lo anteriormente expuesto, se puede deducir que la fórmula de recurrencia para este método se puede expresar como:

$$A^{(i+1)} = A^{(i)} + \left( \Delta e^{(i)} - A^{(i)} \Delta x^{(i)} \right) (v^{(i)})^T / \left[ (v^{(i)})^T \Delta x^{(i)} \right] \quad (3.37)$$

Donde  $v^{(i)}$  será ortogonalizado para  $t-1$  vectores previos a  $x^{(i)}$ .

El algoritmo propuesto se describe a continuación:

- 1) Estimar  $A^{(1)}$  (se puede suponer  $A^{(1)} = I$ )
- 2) Calcular  $\Delta x^{(1)}$  usando la ecuación 3.30  $x^{(1)} = x^{(0)} - (A^{(1)})^{-1} e^{(0)}$  donde:  $\Delta x^{(1)} = x^{(1)} - x^{(0)}$  evaluar  $\Delta e^{(1)} = e^{(1)} - e^{(0)}$
- 3) Seleccionar  $v^{(1)} = \Delta x^{(1)}$  y a partir de  $A^{(1)}$  obtener  $A^{(2)}$
- 4) Determinar  $\Delta x^{(2)}$  usando la ecuación 3.30 y evaluar  $\Delta e^{(2)}$
- 5) Seleccionar  $v^{(2)} = \Delta x^{(2)}$  ortogonal a  $\Delta x^{(1)}$ , obtener  $A^{(2)}$  para poder evaluar  $A^{(3)}$
- 6) Calcular  $\Delta x^{(3)}$  y evaluar  $\Delta e^{(3)}$
- 7) Seleccionar  $v^{(3)} = \Delta x^{(3)}$  ortogonal a  $\Delta x^{(1)}$  y  $x^{(2)}$ , estimar  $A^{(3)}$  y obtener  $A^{(4)}$
- 8) Etc.

Otra forma de expresar el Método de Broyden es haciendo uso de la fórmula de Householder.

$$\begin{aligned}
 H^{(i+1)} &= (A^{(i+1)})^{-1} = A^{(i)} + u^{(i)} (u^{(i)})^T - 1 \\
 &= (A^{(i)})^{-1} - (A^{(i)})^{-1} u^{(i)} (u^{(i)})^T (A^{(i)})^{-1} / \\
 &\quad \left\{ 1 + (v^{(i)})^T (A^{(i)})^{-1} u^{(i)} \right\} \\
 &= H^{(i)} - H^{(i)} u^{(i)} (v^{(i)})^T H^{(i)} / \left\{ 1 + (v^{(i)})^T H^{(i)} u^{(i)} \right\}
 \end{aligned}$$

La fórmula generalizada de recurrencia es la expresada por la ecuación 3.39.

$$H^{(i+1)} = H^{(i)} - (H^{(i)} \Delta e^{(i)}) (v^{(i)})^T H^{(i)} / \left\{ (v^{(i)})^T H^{(i)} \Delta e^{(i)} \right\} \quad (3.38)$$

El algoritmo propuesto se describe a continuación:

- 1) Estimar  $H^{(1)} = (A^{(1)})^{-1} = I^{-1} = I$  con los valores de  $x^{(0)}$  (valores iniciales)
- 2) Calcular el vector de errores  $e^{(0)}$  como una función de  $x^{(0)}$  si  $(e^{(0)})^T e^{(0)}$  da un valor pequeño, se termina
- 3) Estimar  $\Delta x^{(1)} = -H^{(1)} e^{(0)}$
- 4) Suponer  $x^{(i)} = x^{(i-1)} + \Delta x^{(i)}$  y evaluar  $e^{(i)}$ . Si  $(e^{(i)})^T e^{(i)}$  da un valor pequeño se termina.
- 5) Evaluar  $\Delta e^{(i)} = e^{(i)} - e^{(i-1)}$
- 6) Calcular  $\nu^{(i)} = \Delta x^{(i)}$
- 7) Calcular el denominador. Si el valor es pequeño, (por lo menos  $10^{-3}$  veces cualquier elemento de  $H^{(i)}$ ) hacer el paso 9.
- 8) Obtener  $H^{(i)}$  para calcular  $H^{(i+1)}$  con la fórmula recursiva de Broyden (3.30)
- 9) Calcular  $\Delta x^{(i+1)} = -H^{(i+1)} e^{(i)}$
- 10) Iterar nuevamente desde el paso 4 hasta cumplir con  $(e^{(i)})^T e^{(i)}$  de un valor pequeño

C A P I T U L O    I V

E S T R U C T U R A   D E L   P R O G R A M A

#### 4.1 ESTRUCTURA DEL PROGRAMA:

En este capítulo se dará una breve descripción del programa computacional por medio del cual se implementaron los métodos para identificación y rompimiento de ciclos de un diagrama de flujo de proceso.

Dicho programa está basado en el algoritmo de Westerberg para identificación de ciclos y una combinación de los algoritmos de Mahalec y Upadhye y Grens para rompimiento de ciclos.

La construcción del programa se basó en los diagramas de flujo ilustrados en las figuras 3.7 y 3.8 correspondientes a los métodos seleccionados.

Las pruebas necesarias para la depuración de errores de lógica y de sintaxis así como las corridas del programa para resolver problemas ilustrativos fueron llevadas a cabo en la máquina-computadora B-7800 del Programa Universitario de Cómputo de la Universidad Nacional Autónoma de México.

#### 4.2 DESCRIPCION DEL PROGRAMA:

El programa computacional está escrito en Fortran y consta de cuatro subrutinas para identificación y rompimiento de ciclos y de un programa principal. Las subrutinas Floops y Loopch, llevan a cabo la función de localizar las recirculaciones existentes en el sistema. Las subrutinas ATER Y ATEAR-1 seleccionan el conjunto de corte; por último, el programa principal LOOPS coordina todas las actividades mediante la lectura de datos, llamado de subrutinas e impresión de resultados.

##### 4.2.1 PROGRAMA PRINCIPAL LOOPS:

El programa principal LOOPS, es un programa escrito para -

facilitar el uso de las cuatro subrutinas.

La primera función de LOOPS es la de leer los datos de entrada entre los cuales se encuentran los siguientes vectores: - IDSTR es un vector que contiene las corrientes del sistema; LENR es un vector que contiene los números de las corrientes sucesoras de cualquiera de las corrientes y sus elementos estarán ordenados en la misma forma que los de IDSTR. ICN es un vector que contiene los sucesores de cualquier corriente y el conjunto de sucesores debe estar en el mismo orden de los elementos de IDSTR e INCPNT es un arreglo que contiene en su primera hilera los números de las unidades en donde se originan las corrientes, en su segunda hilera, los números de las unidades a donde llegan las corrientes y por último, en la tercera hilera la posición correspondiente en LENR del conjunto de corrientes origen-destino.

Al mismo tiempo que se dá lectura a los vectores antes citados, se construye el vector JP que contiene la posición del primer sucesor de cualquier corriente registrada en ICN. También se leen NOSTR que es el número de corrientes de cualquier sistema, NOSTR1 que es el número de corrientes más uno y NOTEAR que es el número de corrientes preferidas de corte. Si NOTEAR es mayor que cero, se leerá IJTEAR que es un vector que contiene el conjunto de corrientes preferidas de corte.

Una vez que se ha leído todo, LOOPS llama a la subrutina - FLOOPS para detectar las recirculaciones del sistema. Una vez detectadas éstas, llamará a ATFEAR para romper todas las recirculaciones. LOOPS termina su actividad imprimiendo los siguientes resultados: a) El número de recirculaciones encontradas y sus respectivos conjuntos de corte.

#### 4.2.2 SUBROUTINA FLOOPS:

La subrutina FLOOPS lleva a cabo la tarea de localizar todas las recirculaciones existentes en el sistema a partir del al

goritmo de Westerberg descrito en el Capítulo III. En el punto donde es localizada una recirculación FLOOPS llama a la subrutina LOOPCH para registrarla.

FLOOPS inicia con la construcción de sus vectores de trabajo: L1 es el vector INCDNT descrito en el programa principal; - L2 corresponde al vector ICN, L3 se construye a partir de INCDNT y L4 forma parte del vector de trabajo principal IWORK. L5 es un vector que contiene las corrientes identificadas y que son -- llamadas por LOOPS. En el vector ICN se etiquetarán negativamente todas las corrientes que ya han sido checadas y continuamente estas corrientes tomarán su valor positivo antes de ser retornadas a LOOPS; en adición, ICHECK es un vector de trabajo equivalente a LENR que registrará los números de los sucesores de cualquier de las corrientes.

Una vez construídos los vectores de trabajo, se inicia la identificación de las recirculaciones. En este punto, FLOOPS -- construye el vector LOOPS el cual contiene en orden el vector -- LENGTHL, el número de recirculaciones NLOOPS y las posiciones de la primera corriente de cualquier recirculación para que de esta manera, se retorne al programa principal LOOPS.

#### 4.2.3. SUBROUTINA LOOPCH:

Esta subrutina tiene como función registrar las recirculaciones identificadas por FLOOPS. Antes de efectuar esta tarea, se checa primero para ver si la recirculación recién identificada no ha sido registrada anteriormente.

Las recirculaciones detectadas por FLOOPS, son colocadas en el vector de trabajo LOP que será transferido a LOOPCH. La primera prueba que efectuará esta subrutina será la de checar si la recirculación ya fué registrada anteriormente; esta operación se lleva a cabo de la siguiente manera: Se inicia al checar si la recirculación ya fué registrada anteriormente; esta operación



se lleva a cabo de la siguiente manera: Se inicia al checar si la recirculación recién registrada en LOP tiene el mismo número de corrientes que cualquiera de las recirculaciones registradas anteriormente. Si se detecta alguna recirculación tiene el mismo número de corrientes, se procederá a checar corriente por corriente. Si la recirculación registrada en LOP no fué equivalente a ninguna recirculación registrada en LOOPS, se procederá a registrarla en LOOPS y se retornará a FLOOPS.

#### 4.2.4 SUBROUTINA ATEAR:

La subrutina ATEAR ejecuta las primeras 5 etapas del algoritmo para rompimiento de ciclos de Mahalec en la sección 3.3. - Si la respuesta a la pregunta de la etapa 2, es afirmativa y - - existen corrientes preferidas de corte, ATEAR continuará hasta la etapa 5 etapa por etapa, donde la subrutina ATEAR1 será llamada para complementar la construcción de ITEAR, vector que contiene el número de recirculaciones y sus respectivas corrientes de corte. Si la respuesta a la pregunta de la etapa 2 es negativa, ATEAR1 es llamada inmediatamente.

La pregunta de la etapa 2, es contestada mediante la revisión de los valores de NOTEAR (número de corrientes preferidas de corte, un número mayor que cero, será una respuesta afirmativa.

#### 4.2.5 SUBROUTINA ATEAR1:

La subrutina ATEAR1 complementa la selección del conjunto de corte para las recirculaciones encontradas por FLOOPS. Continúa donde ATEAR termina (Si NOTEAR es mayor que cero) o si se -- construye totalmente ITEAR. Cuando ATEAR1 termina su tarea, retorna a ATEAR para que a su vez retorne al programa principal -- LOOPS.

ATEAR1 inicia en la sexta etapa del algoritmo de Mahalec -

listado en la sección 3.3 del Capítulo III. En la etapa 6 se -- identifica la primera recirculación que no ha sido rota (una recirculación está rota si su primera corriente en el vector `LOOPS` es negativa. Para un vector `ITEAR` de `ATEAR1` parcialmente lleno, la primera recirculación con una corriente positiva es identificada por `DO-LOOP`. Si `NOTEAR` es igual a cero, la primera circulación no rota se transforma en la primera recirculación en `LOOPS`.

La etapa 7 se inicia con la identificación de la corriente de corte solicitada en la etapa 10. En la etapa 11 se localiza a las corrientes rotas en `LOOPS` que contengan a la corriente de corte seleccionada. Si no se localiza, en la etapa 12 se indica rá que el conjunto de corte es no redundante. De lo contrario, al retornar al programa principal se imprimirá un mensaje junto con los resultados que indique "si existe redundancia en el conjunto de corte".

#### 4.3 DATOS DE ENTRADA:

Los formatos de los <sup>o</sup>datos de entrada se ilustran para el caso particular del ejemplo número uno que se presenta en el capítulo V. En este ejemplo los valores presentados corresponden a:

Línea 100; se leen los valores de `NOSTR`, `NOSTR1` y `NOTEAR` respectivamente.

Línea 200; valores para `IDSTR(1)` y de `ICN`.

Línea 300-500, valores para `ICN(K)`

NOTA: Los valores para `IDSTR(2)...``IDSTR(N)` y los valores para -- `ICN(K)` son leídos desde la línea 600 hasta la línea 1900.

Líneas 2000-2500; se leerán los valores de `INCNT(I,J)`.

## CAPITULO V

### RESULTADOS

## 5.- RESULTADOS:

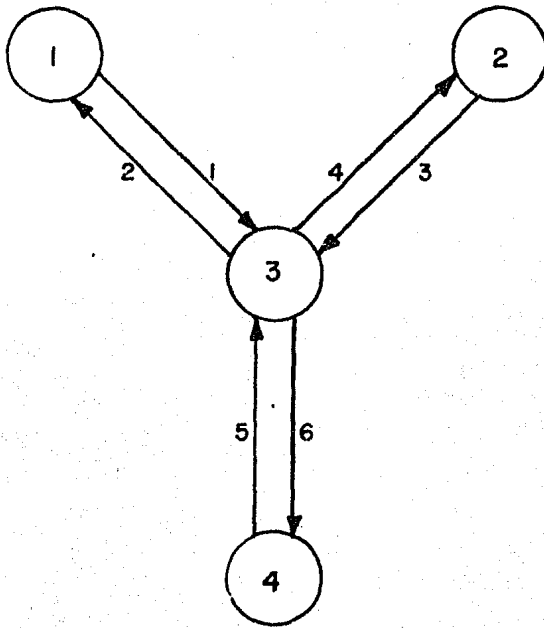
En este capítulo se presentan y analizan los resultados obtenidos al resolver los problemas ilustrativos. El objetivo y grado de complejidad de los ejemplos resueltos es variable y se detalla a continuación:

### 5.1 EJEMPLO NO. 1:

Este ejemplo se ilustra en la figura 5.21 y se planteó con el objeto de aplicar los Algoritmos de Westerberg y Mahalec a la solución de un diagrama de flujo de proceso que presenta familias de corrientes de corte no redundantes. De los resultados reportados para este ejemplo se puede observar lo siguiente:

- a) Los datos alimentados a la máquina son organizados de tal manera que el usuario pueda identificar rápidamente los siguientes elementos: El número de la unidad que dá origen a cada corriente, el número de la unidad destino de la corriente, así como todas las corrientes sucesoras a ésta.
- b) En los resultados podemos observar dos secciones que son: La correspondiente a la identificación de ciclos (recirculaciones) del diagrama de flujo y la sección correspondiente al rompimiento de los mismos.

Para cada recirculación identificada se reporta a cada una de las corrientes que la forman y al mismo tiempo, se determina la corriente de corte para la recirculación. Por último, las corrientes de corte seleccionadas se agrupan en el conjunto de corte.



TITULO:

FIG. 5.1 EJEMPLO N°1

TESIS PROFESIONAL

JOSE MARIANO RAMOS OLMOS

E.N.E.P. ZARAGOZA U.N.A.M.

## \*DATOS DE ENTRADA\*

CORRIENTE	ORIGEN	DESTINO	SUCESORES	
1	1	2	2	6
2	1	3	4	6
3	1	4	6	6
4	2	5		
5	3	6		
6	4	6		

## \*RESULTADOS\*

EXISTEN 3 RECIRCULACIONES.

RECIRCULACION NUMERO 1 CONTIENE A LAS CORRIENTES:  
 1 2  
 RECIRCULACION NUMERO 2 CONTIENE A LAS CORRIENTES:  
 4 3  
 RECIRCULACION NUMERO 3 CONTIENE A LAS CORRIENTES:  
 6 5

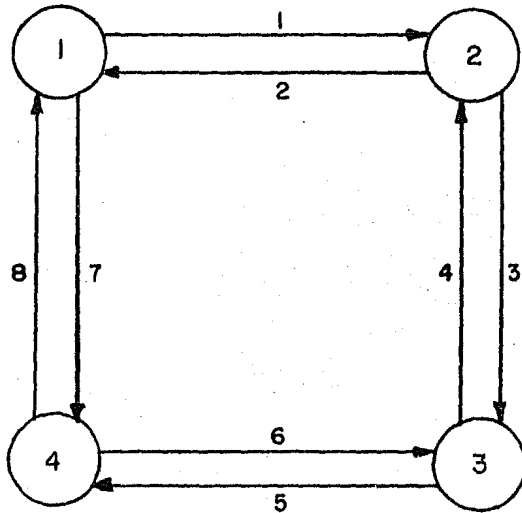
EL CONJUNTO DE CORTE ES:

RECIRCULACION	CORRIENTE DE CORTE
1	1
2	4
3	6

5.2 EJEMPLO NO. 2:

Este ejemplo se ilustra en la figura 5.1, se planteó con el objeto de aplicar los algoritmos seleccionados para la identificación y rompimiento de ciclos de un diagrama de flujo de procesos a la solución de un proceso que presente familias de corrientes de corte redundantes.

De los resultados reportados se puede observar que a diferencia del ejemplo No. 1, existe más de una corriente de corte que rompe en dos o más ocasiones a un mismo ciclo, lo cual transforma al conjunto de corrientes de corte en "Redundante" lo cual ratifica lo explicado en los capítulos III y IV de este trabajo.



TITULO:

FIG. 5.2 EJEMPLO N° 2

TESIS PROFESIONAL

JOSE MARIANO RAMOS OLMOS

E.N.E.P. ZARAGOZA U.N.A.M.



\*DATOS DE ENTRADA\*

CORRIENTE	ORIGEN	DESTINO	SUCESORES
1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8	2 3 4 5 6 7 8	2 1 4 2 6 4 6 1 3 5 3 8 5 8 7

\*RESULTADOS\*

EXISTEN 6 RECIRCULACIONES.

- RECIRCULACION NUMERO 1 CONTIENE A LAS CORRIENTES:  
1 2
- RECIRCULACION NUMERO 2 CONTIENE A LAS CORRIENTES:  
3 4
- RECIRCULACION NUMERO 3 CONTIENE A LAS CORRIENTES:  
5 6
- RECIRCULACION NUMERO 4 CONTIENE A LAS CORRIENTES:  
1 3 5 8
- RECIRCULACION NUMERO 5 CONTIENE A LAS CORRIENTES:  
2 7 6 4
- RECIRCULACION NUMERO 6 CONTIENE A LAS CORRIENTES:  
7 8

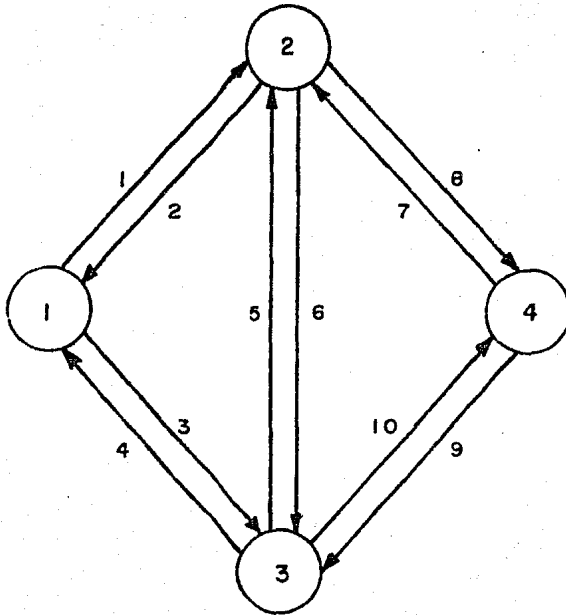
EL CONJUNTO DE CORTE ES REDUNDANTE.  
Y ES

RECIRCULACION	CORRIENTE DE CORTE
1 4 2 3 3 4 6 5	1 1 4 4 5 5 7 7

### 5.3 EJEMPLO NO. 3:

Este ejemplo se ilustra en la figura 5.3 y se planteó con el objeto de aplicar los algoritmos seleccionados a la solución de un diagrama de flujo de proceso que presente familias de corrientes de corte redundantes y además para ver el alcance del mismo; es decir, observar hasta cuántos ciclos (recirculaciones) puede identificar y romper.

De los resultados obtenidos se puede observar que se detectaron 10 recirculaciones con su respectiva familia redundante de 10 corrientes de corte.



TITULO:

FIG. 5.3 EJEMPLO Nº 3

TESIS PROFESIONAL

JOSE MARIANO RAMOS OLMOS

E.N.E.P. ZARAGOZA U.N.A.M.

**\*DATOS DE ENTRADA\***

CORRIENTE	ORIGEN	DESTINO	SUCESESORES		
1	1	2	2	6	8
2	2	3	1	3	10
3	3	1	2	4	8
4	4	2	6	5	10
5	5	3	1	6	8
6	6	4	2	7	8
7	7	1	6	5	10
8	8	2	1	3	
9	9	3	2	4	
10	10	4	6	5	

**\*RESULTADOS\***

EXISTEN 10 RECIRCULACIONES.

- RECIRCULACION NUMERO 1 CONTIENE A LAS CORRIENTES:  
1 2
- RECIRCULACION NUMERO 2 CONTIENE A LAS CORRIENTES:  
1 6 4
- RECIRCULACION NUMERO 3 CONTIENE A LAS CORRIENTES:  
1 5
- RECIRCULACION NUMERO 4 CONTIENE A LAS CORRIENTES:  
6 10 7
- RECIRCULACION NUMERO 5 CONTIENE A LAS CORRIENTES:  
10 9
- RECIRCULACION NUMERO 6 CONTIENE A LAS CORRIENTES:  
8 7
- RECIRCULACION NUMERO 7 CONTIENE A LAS CORRIENTES:  
1 8 9 4
- RECIRCULACION NUMERO 8 CONTIENE A LAS CORRIENTES:  
8 9 5
- RECIRCULACION NUMERO 9 CONTIENE A LAS CORRIENTES:  
3 4
- RECIRCULACION NUMERO 10 CONTIENE A LAS CORRIENTES:  
2 3 5

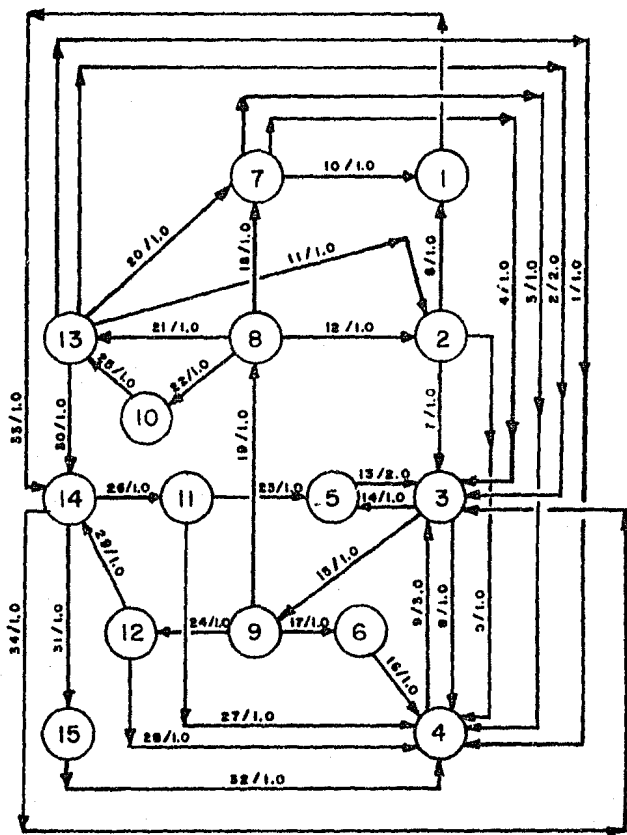
EL CONJUNTO DE CORTE ES REDUNDANTE.  
Y ES

RECIRCULACION	CORRIENTE DE CORTE
1	1
2	1
3	1
4	1
5	1
6	1
7	1
8	1
9	1
10	1

#### 5.4 EJEMPLO NO. 4:

Este ejemplo se ilustra en la figura No. 4 y se resolvió con el objeto de ver el alcance del programa: se resolvió un diagrama de flujo de proceso que contiene un número considerable de corrientes y recirculaciones.

De los resultados reportados para este ejemplo se puede observar cómo una corriente de corte (corriente 9), puede ser de vital importancia en la solución del problema, ya que el hecho de suponer los valores de las variables de la corriente 9, implica que 6 recirculaciones pueden ser resueltas inmediatamente.



TITULO:  
 FIG. 6.4 EJEMPLO N° 4

TESIS PROFESIONAL  
 JOSE MARIANO RAMOS OLMOS  
 E.N.E.P ZARAGOZA U.N.A.M.

\*DATOS DE ENTRADA\*

CORRIENTE	ORIGEN	DESTINO	SUCESORES		
1	13	4	9		
2	13	4	9		
3	7	4	9		
4	7	3	8	14	15
5	2	4	9		
6	2	1	3		
7	2	3	8	14	15
8	3	4	9		
9	3	3	8	14	15
10	1	1	3		
11	1	2	5	6	7
12	8	2	5	6	7
13	5	3	8	14	15
14	5	5	3		
15	5	9	1	17	19
16	6	4	9		
17	6	6	1		
18	8	7	3	4	10
19	9	8	1	18	21
20	3	7	6	4	10
21	8	1	1	2	11
22	8	10	2		
23	1	5	25		
24	1	5	1		
25	9	12	13	29	
26	10	13	8	2	11
27	1	1	1		
28	1	1	2		
29	1	3	9		
30	2	4	9		
31	1	14	26	31	34
32	1	14	26	31	34
33	1	15	3		
34	1	4	9		
	14	3	26	31	34
			8	14	15

\*RESULTADOS\*

EXISTEN 14 RECIRCULACIONES.

RECIRCULACION NUMERO	1	CONTIENE A LAS CORRIENTES:
9 8		
RECIRCULACION NUMERO	2	CONTIENE A LAS CORRIENTES:
14 13		
RECIRCULACION NUMERO	3	CONTIENE A LAS CORRIENTES:
9 15 17 16		
RECIRCULACION NUMERO	4	CONTIENE A LAS CORRIENTES:
9 15 19 12 5		
RECIRCULACION NUMERO	5	CONTIENE A LAS CORRIENTES:
15 19 12 6 33 26 27		
RECIRCULACION NUMERO	6	CONTIENE A LAS CORRIENTES:
9 15 19 12 6 33 31 32		
RECIRCULACION NUMERO	7	CONTIENE A LAS CORRIENTES:
15 19 12 6 33 34		
RECIRCULACION NUMERO	8	CONTIENE A LAS CORRIENTES:
15 19 12 7		
RECIRCULACION NUMERO	9	CONTIENE A LAS CORRIENTES:
9 15 19 18 3		
RECIRCULACION NUMERO	10	CONTIENE A LAS CORRIENTES:
15 19 18 4		
RECIRCULACION NUMERO	11	CONTIENE A LAS CORRIENTES:
1 9 15 19 21		
RECIRCULACION NUMERO	12	CONTIENE A LAS CORRIENTES:
1 9 15 19 22 25		
RECIRCULACION NUMERO	13	CONTIENE A LAS CORRIENTES:
9 15 24 28		
RECIRCULACION NUMERO	14	CONTIENE A LAS CORRIENTES:
15 24 29 34		

EL CONJUNTO DE CORTE ES:

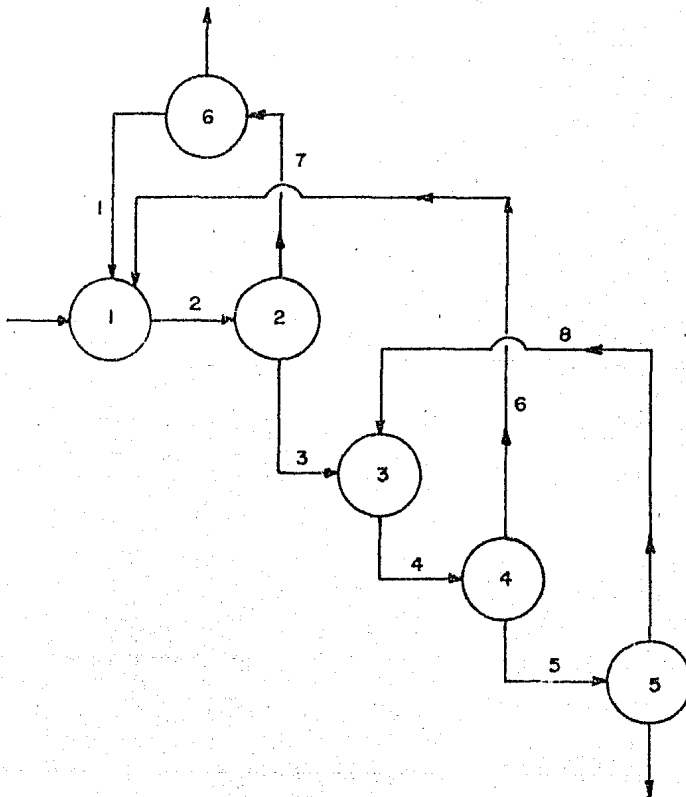
RECIRCULACION	CORRIENTE DE CORTE
-----	-----
1	9
3	9
4	9
6	9
9	9
11	9
12	9
13	9
2	9
5	9
7	9
14	6
8	6
10	6
	7
	6



5.5 EJEMPLO NO. 5:

Este ejemplo representa el problema de Cayett (1970) y se ilustra en la figura (3.3) y se resolvió con el objeto de determinar los diferentes conjuntos de corte y el orden de cálculo para cada prueba efectuada. Estos resultados se resumen en la tabla 5.1 de los cuales se desprenden las siguientes observaciones:

- a) El hecho de seleccionar una corriente preferida de corte implica que el primer ciclo que será roto, es aquel en el cual se localiza dicha corriente de corte.
- b) En el caso de que una corriente preferida de corte pertenezca a dos ciclos, el orden de cálculo para el rompimiento de los ciclos, se iniciará con el primer ciclo identificado y una vez terminado éste, se iniciará con el rompimiento del segundo ciclo del cual forma parte esta corriente.
- c) Cuando exista más de una corriente preferida de corte, el orden de cálculo se iniciará con el rompimiento del primer ciclo identificado.



TITULO:

FIG. 3.3 PROBLEMA DE CAVETT

TESIS PROFESIONAL

JOSE MARIANO RAMOS OLMOS

E.N.E.P. ZARAGOZA U.N.A.M.

**\*DATOS DE ENTRADA\***

CORRIENTE	ORIGEN	DESTINO	SUCESORES
1	6	1	2
2	1	2	3
3	2	3	4
4	3	4	5
5	4	5	6
6	5	1	2
7	2	6	1
8	3	3	4

**\*RESULTADOS\***

EXISTEN 3 RECIRCULACIONES.

RECIRCULACION NUMERO 1 CONTIENE A LAS CORRIENTES:  
 4 5 8  
 RECIRCULACION NUMERO 2 CONTIENE A LAS CORRIENTES:  
 2 3 4 6  
 RECIRCULACION NUMERO 3 CONTIENE A LAS CORRIENTES:  
 1 2 7

EL CONJUNTO DE CORTE ES:

RECIRCULACION	CORRIENTE DE CORTE
1	5
2	4
3	1

## \*DATOS DE ENTRADA\*

CORRIENTE	ORIGEN	DESTINO	SUCESORES
1	6	1	2
2	1	2	3
3	2	3	4
4	3	4	5
5	4	5	6
6	5	6	7
7	6	7	8
8	7	8	9

LAS CORRIENTES PREFERIDAS DE CORTE SON :

1

## \*RESULTADOS\*

EXISTEN 3 RECIRCULACIONES.

RECIRCULACION NUMERO 1 CONTIENE A LAS CORRIENTES:  
 4 5 8  
 RECIRCULACION NUMERO 2 CONTIENE A LAS CORRIENTES:  
 2 3 4 6  
 RECIRCULACION NUMERO 3 CONTIENE A LAS CORRIENTES:  
 1 2 7

EL CONJUNTO DE CORTE ES:

RECIRCULACION	CORRIENTE DE CORTE
3	1
1	4
2	4

**\*DATOS DE ENTRADA\***

CORRIENTE	ORIGEN	DESTINO	SUCESORES
1	6	1	2
2	1	2	3
3	2	3	4
4	3	4	5
5	4	5	6
6	5	6	7
7	6	7	8
8	7	8	9

LAS 2 CORRIENTES PREFERIDAS DE CORTE SON :

**\*RESULTADOS\***

EXISTEN 3 RECIRCULACIONES.

- RECIRCULACION NUMERO 1 CONTIENE A LAS CORRIENTES:  
4 5 8
- RECIRCULACION NUMERO 2 CONTIENE A LAS CORRIENTES:  
2 3 4 6
- RECIRCULACION NUMERO 3 CONTIENE A LAS CORRIENTES:  
1 2 7

EL CONJUNTO DE CORTE ES:

RECIRCULACION	CORRIENTE DE CORTE
2	2
3	2
1	4

**\*DATOS DE ENTRADA\***

CORRIENTE	ORIGEN	DESTINO	SUCESORES
1	6	1	2
2	1	2	3
3	2	3	4
4	3	4	5
5	4	5	6
6	5	6	7
7	6	7	8
8	7	8	

LAS CORRIENTES PREFERIDAS DE CORTE SON :  
3

**\*RESULTADOS\***

EXISTEN 3 RECIRCULACIONES.

RECIRCULACION NUMERO 1 CONTIENE A LAS CORRIENTES:

4 5 8  
RECIRCULACION NUMERO 2 CONTIENE A LAS CORRIENTES:

2 3 4 6  
RECIRCULACION NUMERO 3 CONTIENE A LAS CORRIENTES:  
1 2 7

EL CONJUNTO DE CORTE ES:

RECIRCULACION	CORRIENTE DE CORTE
2	3
1	4
3	1

**\*DATOS DE ENTRADA\***

CORRIENTE	ORIGEN	DESTINO	SUCESORES
1 2 3 4 5 6 7 8	6 1 2 3 4 5	1 2 3 4 5 6 7	2 3 4 5 6 7

LAS CORRIENTES PREFERIDAS DE CORTE SON :

4

**\*RESULTADOS\***

EXISTEN 3 RECIRCULACIONES.

RECIRCULACION NUMERO 1 CONTIENE A LAS CORRIENTES:

RECIRCULACION NUMERO 2 CONTIENE A LAS CORRIENTES:

RECIRCULACION NUMERO 3 CONTIENE A LAS CORRIENTES:

1 2 7

EL CONJUNTO DE CORTE ES:

RECIRCULACION	CORRIENTE DE CORTE
1	4
2	4
3	1

## \*DATOS DE ENTRADA\*

CORRIENTE	ORIGEN	DESTINO	SUCESORES
1	6	1	2
2	1	2	3
3	2	3	4
4	3	4	5
5	4	5	6
6	5	6	7
7	6	7	8
8	7	8	9

LAS CORRIENTES PREFERIDAS DE CORTE SON :

5

## \*RESULTADOS\*

EXISTEN 3 RECIRCULACIONES.

RECIRCULACION NUMERO 1 CONTIENE A LAS CORRIENTES:

RECIRCULACION NUMERO 2 CONTIENE A LAS CORRIENTES:

RECIRCULACION NUMERO 3 CONTIENE A LAS CORRIENTES:

1 2 7

EL CONJUNTO DE CORTE ES:

RECIRCULACION	CORRIENTE DE CORTE
1	5
2	2
3	2



**\*DATOS DE ENTRADA\***

CORRIENTE	ORIGEN	DESTINO	SUCESORES
1	6	1	2
2	1	2	3
3	2	3	4
4	3	4	5
5	4	5	6
6	5	6	7
7	6	7	8
8	7	8	9

LAS CORRIENTES PREFERIDAS DE CORTE SON :  
6

**\*RESULTADOS\***

EXISTEN 3 RECIRCULACIONES.

RECIRCULACION NUMERO 1 CONTIENE A LAS CORRIENTES:

4 5 8  
RECIRCULACION NUMERO 2 CONTIENE A LAS CORRIENTES:

2 3 4 6  
RECIRCULACION NUMERO 3 CONTIENE A LAS CORRIENTES:  
1 2 7

EL CONJUNTO DE CORTE ES:

RECIRCULACION	CORRIENTE DE CORTE
2	6
1	4
3	1

**\*DATOS DE ENTRADA\***

CORRIENTE	ORIGEN	DESTINO	SUCESORES
1 2 3 4 5 6 7 8	6 1 2 3 4 5	1 2 3 4 5 6 7	2 3 4 5 6 7

LAS CORRIENTES PREFERIDAS DE CORTE SON :  
7

**\*RESULTADOS\***

EXISTEN 3 RECIRCULACIONES.

RECIRCULACION NUMERO 1 CONTIENE A LAS CORRIENTES:

4 5 8

RECIRCULACION NUMERO 2 CONTIENE A LAS CORRIENTES:

2 3 4 6

RECIRCULACION NUMERO 3 CONTIENE A LAS CORRIENTES:

1 2 7

EL CONJUNTO DE CORTE ES:

RECIRCULACION	CORRIENTE DE CORTE
3	7
1	4
2	4

\*DATOS DE ENTRADA\*

CORRIENTE	ORIGEN	DESTINO	SUCESORES
1	6	1	2
2	1	2	3
3	2	3	4
4	3	4	5
5	4	5	6
6	4	1	8
7	2	6	1
8	5	3	4

LAS CORRIENTES PREFERIDAS DE CORTE SON :  
8

\*RESULTADOS\*

EXISTEN 3 RECIRCULACIONES.

RECIRCULACION NUMERO 1 CONTIENE A LAS CORRIENTES:  
4 5 8

RECIRCULACION NUMERO 2 CONTIENE A LAS CORRIENTES:  
2 3 4 6

RECIRCULACION NUMERO 3 CONTIENE A LAS CORRIENTES:  
1 2 7

EL CONJUNTO DE CORTE ES:

RECIRCULACION	CORRIENTE DE CORTE
1	8
2	2
3	3

**\*DATOS DE ENTRADA\***

CORRIENTE	ORIGEN	DESTINO	SUCESORES
1	6	1	2
2	1	2	3
3	2	3	4
4	3	4	5
5	4	5	6
6	5	6	7
7	6	7	8
8	7	8	9

LAS CORRIENTES PREFERIDAS DE CORTE SON :

1 2

**\*RESULTADOS\***

EXISTEN 3 RECIRCULACIONES.

RECIRCULACION NUMERO 1 CONTIENE A LAS CORRIENTES:

RECIRCULACION NUMERO 2 CONTIENE A LAS CORRIENTES:

RECIRCULACION NUMERO 3 CONTIENE A LAS CORRIENTES:

EL CONJUNTO DE CORTE ES:

RECIRCULACION	CORRIENTE DE CORTE
1	1
2	2
3	3

## \*DATOS DE ENTRADA\*

CORRIENTE	ORIGEN	DESTINO	SUCESORES
1	6	1	2
2	1	2	3
3	2	3	4
4	3	4	5
5	4	5	6
6	5	6	7
7	6	7	8
8	7	8	

LAS CORRIENTES PREFERIDAS DE CORTE SON :

1 3

## \*RESULTADOS\*

EXISTEN 3 RECIRCULACIONES.

RECIRCULACION NUMERO 1 CONTIENE A LAS CORRIENTES:

4 5 8  
RECIRCULACION NUMERO 2 CONTIENE A LAS CORRIENTES:

2 3 4 6  
RECIRCULACION NUMERO 3 CONTIENE A LAS CORRIENTES:

1 2 7

EL CONJUNTO DE CORTE ES:

RECIRCULACION	CORRIENTE DE CORTE
3	1
1	8
2	2

**\*DATOS DE ENTRADA\***

CORRIENTE	ORIGEN	DESTINO	SUCESORES
1	6	1	2
2	1	2	3
3	2	3	4
4	3	4	5
5	4	5	6
6	5	6	7
7	6	7	8
8	7	8	

LAS CORRIENTES PREFERIDAS DE CORTE SON :

1 5

**\*RESULTADOS\***

EXISTEN 3 RECIRCULACIONES.

RECIRCULACION NUMERO 1 CONTIENE A LAS CORRIENTES:

RECIRCULACION NUMERO 2 CONTIENE A LAS CORRIENTES:

RECIRCULACION NUMERO 3 CONTIENE A LAS CORRIENTES:

EL CONJUNTO DE CORTE ES:

RECIRCULACION	CORRIENTE DE CORTE
3	1
1	5
2	2

**\*DATOS DE ENTRADA\***

CORRIENTE	ORIGEN	DESTINO	SUCESORES
1	6	1	2
2	1	2	3
3	2	3	4
4	3	4	5
5	4	5	6
6	5	6	7
7	6	7	8
8	7	8	

LAS CORRIENTES PREFERIDAS DE CORTE SON :

1 6

**\*RESULTADOS\***

EXISTEN 3 RECIRCULACIONES.

RECIRCULACION NUMERO 1 CONTIENE A LAS CORRIENTES:

4 5 8

RECIRCULACION NUMERO 2 CONTIENE A LAS CORRIENTES:

2 3 4 6

RECIRCULACION NUMERO 3 CONTIENE A LAS CORRIENTES:

1 2 7

EL CONJUNTO DE CORTE ES:

RECIRCULACION	CORRIENTE DE CORTE
3	1
2	6
1	4

**\*DATOS DE ENTRADA\***

CORRIENTE	ORIGEN	DESTINO	SUCESORES
1	6	1	2
2	1	2	3
3	2	3	4
4	3	4	5
5	4	5	6
6	5	6	7
7	6	7	8
8	7	8	9

LAS CORRIENTES PREFERIDAS DE CORTE SON :  
1 7

**\*RESULTADOS\***

EXISTEN 3 RECIRCULACIONES.

RECIRCULACION NUMERO 1 CONTIENE A LAS CORRIENTES:  
4 5 8  
RECIRCULACION NUMERO 2 CONTIENE A LAS CORRIENTES:  
2 3 4 6  
RECIRCULACION NUMERO 3 CONTIENE A LAS CORRIENTES:  
1 2 7

EL CONJUNTO DE CORTE ES:

RECIRCULACION	CORRIENTE DE CORTE
3	1
3	7
1	4
2	4



\*DATOS DE ENTRADA\*

CORRIENTE	ORIGEN	DESTINO	SUCESORES
1	6	1	2
2	1	2	3
3	2	3	4
4	3	4	5
5	4	5	6
6	5	6	7
7	6	7	8
8	7	8	1
	8		2
			3
			4
			5
			6
			7
			8

LAS CORRIENTES PREFERIDAS DE CORTE SON :

1 8

\*RESULTADOS\*

EXISTEN 3 RECIRCULACIONES.

RECIRCULACION NUMERO 1 CONTIENE A LAS CORRIENTES:

4 5 8

RECIRCULACION NUMERO 2 CONTIENE A LAS CORRIENTES:

2 3 4 6

RECIRCULACION NUMERO 3 CONTIENE A LAS CORRIENTES:

1 2 7

EL CONJUNTO DE CORTE ES:

RECIRCULACION	CORRIENTE DE CORTE
3	1
2	3
1	4

TABLA 5.1 COMPARACION DEL ORDEN DE CALCULO Y DE LOS CONJUNTOS DE CORTE PARA LAS RECIRCULACIONES QUE PRESENTA EL PROBLEMA DE CAVETT CUANDO EXISTEN CORRIENTES REFERIDAS DE CORTE.

Num. DE CORRIENTE PREFERIDA DE CORTE	ORDEN DE CALCULO			CONJUNTO DE CORRIENTES DE CORTE		
	R1	R2	R3	R1	R2	R3
-	R1	R2	R3	4	4	1
1	R3	P1	R2	4	4	1
2	R2	P3	R1	4	2	2
3	R2	R1	P3	4	1	3
4	R1	R2	P3	4	4	1
5	P1	R2	P3	5	2	2
6	R2	P1	P3	4	6	1
7	R3	P1	P2	4	4	7
8	R1	R2	P3	8	2	2
1 y 2	R3	R2 R3	R1	4	2	1 y 2
1 y 3	R3	R2	R1	4	3	1
1 y 5	R3	R1	R2	5	2	1
1 y 6	R3	R2	R1	4	6	1
1 y 7	R3	R3 R1	P2	4	4	1 y 7
1 y 8	R3	R2	R1	1	3	1

CAPITULO VII

CONCLUSIONES

## 7. CONCLUSIONES:

Durante el desarrollo del presente trabajo se obtuvieron - las siguientes conclusiones:

- 1) El alcance de cualquier simulador de procesos, va a depender directamente del grado de detalle con el cual se hayan diseñado los modelos matemáticos de las unidades de proceso y del algoritmo de simulación seleccionado.
- 2) Comparando los simuladores de procesos existentes, se observa que el algoritmo de simulación más utilizado a nivel industrial es el secuencial modular.
- 3) Antes de iniciar la construcción de un nuevo simulador de procesos a estado estacionario, es indispensable -- consultar el Software computacional existente y disponible para evitar duplicación de trabajo y esfuerzo.
- 4) La eficiencia de un simulador de procesos a estado estacionario va a depender de los algoritmos seleccionados para la identificación y el rompimiento de los ciclos, y del algoritmo para efectuar los cálculos de -- convergencia.
- 5) Al comparar los simuladores de procesos de la 3a. generación con los de la 2a. y 1a. se puede observar una mejoría muy notoria respecto al alcance de los paquetes de propiedades termodinámicas, algoritmos de cálculo utilizados y la forma como se efectúa la transferencia de datos.
- 6) Es de suma importancia distinguir que la simulación de procesos es simplemente una herramienta de cálculo para el ingeniero químico, la cual nos permite analizar detalladamente un proceso de interés, pero hasta la fecha no ha podido reemplazar el criterio y juicio del -

profesionista.

- 7) El hecho de seleccionar una corriente de corte, implica que el orden de cálculo para romper los ciclos identificados, se iniciará con el rompimiento del ciclo en el que se localice la corriente seleccionada.
- 8) En el caso de que una corriente preferida de corte pertenezca a dos ciclos, el orden de cálculo para el rompimiento de estos dos ciclos se iniciará con el ciclo que se haya identificado primero y después con el rompimiento del segundo ciclo del cual forma parte esta corriente.
- 9) Cuando exista más de una corriente preferida de corte, el orden de cálculo se iniciará con el rompimiento del primer ciclo identificado.

## BIBLIOGRAFIA

- 1.- Barkley, R.W. and R.L. Motard, "Descomposition of Nets" -- Chem. Eng. J., Vol. 3, Pag. 265 (1971).
- 2.- Broyden, C.G., "A Class of Methods for Solving Nonlinear - Simultaneous Equations" Math. Comp., Vol. 19, Pag. 577 - - (1965).
- 3.- Broyden, C.G., "A new Method of Solving Nonlinear Simultaneous Equations" Comp. J., Vol. 12, Pag. 95 (1969).
- 4.- Cavett, R.H., "Application of Numerical Methods to the Convergence of Processes involving Recycle Loops" Am.Petr.- Inst. Rec., Vol. 43, No. 111, Pag. 186 (1970).
- 5.- Christiansen, J.H. and D.F. Rudd, "Structuring Desing Computations" AIChE J., Vol. 15, Pag. 94 (1969).
- 6.- Crowe, C.M.A.E. Hamielec, D.R. Woods and P.T. Shannon, - - "Chemical Plant Simulation" Prentice Hall, Englewood Cliffs New Jersey.
- 7.- Evans, L.B., D.G. Steward and C.R. Sprague, "Computer Aided Chemical Process Desing" Chem. Eng. Progr., Vol. 54, - No. 4, Pag. 39 (1968).
- 8.- Evans, L.B. and W.D. Seider, "The Requirements of one Advanced Computing System" Chem. Eng. Progr., Vol. 64, Pag. - 80 (1976).
- 9.- Flower, J.R., and B.D. Whitehead, "Computer Aided Desing: - A Survey of Flowsheeting Programas, Part. 1" Chem.Eng. - - Progr., Vol. 272, Pag. 208 (1973a)
- 10.- Flower, J.R. and B.D. Whitehead, "Computer Aided Desing: - A Survey of Flowsheeting Programs, Part. 2" Chem. Eng. - - Progr., Vol. 273, Pag. 271 (1973b).
- 11.- Forder, G.J. and H.P. Hutchinson, "The Analysis of Chemi--

- cal Plant Flowsheets" Chem. Eng. Sci., Vol. 24, Pag. 771 - (1969).
- 12.- Genna, P.L. and R.L. Motard, "Optimal Descomposition of -- Process Networks" AIChE J., Vol. 21, Pag. 656 (1975).
- 13.- Himmelblau, D.M., "Descomposition of Large Scale Systems - 1" Chem.Eng.Sci., Vol. 21, Pag. 425 (1968).
- 14.- Himmelblau, D.M. and W.D. Ledet, "Descomposition Procedure for Solving Large Scale Systems" Adv. Chem. Eng., Vol. 8,- Pag. 186 (1970).
- 15.- Hlavacek, V., "Analysis of Complex Plant-Steady State and Transient Behavior" Comp. and Chem. Eng., Vol. 1, No. 1, - Pag. 75 (1977).
- 16.- Hutchinson, H.P. and C.F. Shewchuck, "Computational Method for Multiple Distillation Towers" Trans. Inst. Chem. Eng., Vol. 52, Pag. 325 (1974).
- 17.- Jain, V.S., "Executive Programs for the Analysis and Steady State Simulation of Chemical Processes" University of - Nebraska (1974).
- 18.- Jain, V.S. and J.M. Fakman, "Identification of Process - - Flow Networks" Chem. Eng. Comp., Vol. 2, AIChE Workshop S. (1972).
- 19.- Kehat, E. and M. Schacham, "Chemical Process Simulation Program 1" Process Technol., Vol. 18 (1/2), Pag. 35 (1973a).
- 20.- Kehat, E. and M. Schacham, "Chemical Process Simulation - Program 2" Process Technol., Vol. 18 (3), Pag. 45 (1973b).
- 21.- Kehat, E. and M. Schacham, "Chemical Process Simulation -- Program 3" Process Technol., Vol. 18 (4/5), Pag. 181 - - - (1973c).



- 22.- Kessler, M.G. and M.M. Kessler, World Petrol, Vol. 29, Pag. 60 (1958).
- 23.- Lee, W. and D.F. Rudd, "On the Ordering of Recycle Calculation" AIChE J., Vol. 12, Pag. 1184 (1966).
- 24.- Mah, R.S.H., "A Constructive Algorithm for Computing the Reachability Matrix" AIChE J., Vol. 20, Pag. 1227 (1974).
- 25.- Mah, R.S.H. and M. Rafal, "Automatic Program Generation in Chemical Engineering Computation" Trans. Inst. Chem. Eng., -- Vol. 49, Pag. 407 (1971).
- 26.- Montiel Maldonado, C., "Análisis de secuencias de cálculo para Simulación de Procesos" Tesis Prof. Facultad de Química, U.N.A.M. (1981).
- 27.- Motard, R.L. M. Schacham and E.M. Rosen, "Steady State Chemical Process Simulation" AIChE J., Vol. 21, No. 3, Pag. 417 (1975).
- 28.- Motard, R.L. and A.W. Westerberg, "Exclusive tear for Flow sheets" AIChE J., Vol. 27 No. 5, Pag. 725 (1981).
- 29.- Nagiev, M.F., "Material Balances in Complex and Multistage Recycle Chemical Processes" Chem. Eng. Progr., Vol. 60, No. 9 Pag. 70 (1964).
- 30.- Naphtali, L.M., "Process Heat and Material Balances" Chem. Eng. Progr., Vol. 60, No. 9, Pag. 70 (1964).
- 31.- Norman, R.L., "A Matrix Method for Location of Cycle in a Direct Graph" AIChE J., Vol. 11, No. 3, Pag. 450 (1965).
- 32.- Orbach, O. and C.M. Crowe, "Convergence Promotion in the Simulation of Chemical Processes with Recycle the Dominant Eigenvalue Method" Can. J. Chem. Eng., Vol. 49, Pag. 509 (1971).

- 33.- Pho, T.K. and L. Lapidus, "Topics in Computer Aided Design Part I An Optimum Tearing Algorithm for Recycle Systems" - - AIChE J., Vol. 19, Pag. 1170 (1975).
- 34.- Rosen, E.M., "A Machine Computation Method for Performing Material Balances" Chem.Eng.Progr., Vol. 58, No. 10, Pag. 69 (1962).
- 35.- Rosen, E.M., "Steady State Chemical Process Simulation - - State of The Art Review" Comp. Applications to Chem.Eng. - - ACS Symposium Series No. 1124, Pag. 3 (1980).
- 36.- Rosen, E.M. and A.C. Pauls, "Computer-Aided Chemical Process Design: The FLOWTRAN System" Comp. and. Che. Eng., -- Vol. 1, No. 1, Pag. 11 (1977).
- 37.- Sargent, R.W.H. and A.W. Westerberg, "SPEED UP in Chemical Engineering Design" Trans. Inst. Chem. Eng., Vol. 42, Pag. 190 (1964).
- 38.- Sargent, R.W.H., "A review of Methods for Solving Nonlinear Algebraic Equations" Paper presented at conference on Fundations of Computer Aided Process Design, Henniker, - - H.J., July 6 (1980).
- 39.- Steward, D.V., "Partitioning and Tearing Systems of Equations" J. Soc. Int. Appl. Math. Serv. B., Vol. 2, Pag. 345 (1965).
- 40.- Uneda, T. and M. Nishio, "Comparison Between Sequential -- and Simultaneous Approaches in Process Simulation" Ind. - - Eng. Chem. Process Des. Dev., Vol. 11, Pag. 153 (1972).
- 41.- Upadhye, R.S. and E.A. Grens. "An Efficient Algorithm for Optimum Descomposition of Recycle Systems" AIChE J., Vol. 18, Pag. 533 (1972).

- 42.- Upadhue, R.S. and F.A. Grens, "Selection of Decompositions for Chemical Process Simulations" AIChE J., Vol. 21. No. 1 Pag. 136 (1975).
- 43.- Westerberg, A.W., H.P. Hutchinson, R.L. Motard and P. Winter "Process Flowsheeting", Cambridge University Press - - (1979).
- 44.- Wegstein, J.H., "Accelerating Convergence of Iterative Processes" Comp. M. Assoc. Computing Mach., Vol. 1, Pag. 9 -- (1958).