

8 200.



# UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO.

Escuela Nacional de Estudios Profesionales  
Z A R A G O Z A

metodos modulares simultaneos  
para la simulación de procesos  
quimicos de estado estacionario

t e s i s

para obtener el Título de:

Ingeniero Quimico

p r e s e n t a:

Miguel Ojeda Ramos

1 9 8 7



## **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 está 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.

# I N D I C E

	PAGINA
INDICE .....	I
LISTA DE FIGURAS .....	V
LISTA DE TABLAS .....	XI
CAPITULO 1. INTRODUCCION .....	1
CAPITULO 2. GENERALIDADES .....	13
2.1. Antecedentes .....	15
2.2. Ingeniería Química e Ingeniería de Procesos .....	18
2.2.1. Ingeniería de Procesos .....	19
2.2.2. Diseño de Procesos .....	27
2.2.2.1. Síntesis .....	31
2.2.2.2. Análisis .....	35
2.2.2.3. Optimización .....	40
2.2.3. Diseño de Procesos bajo un enfoque evolutivo .....	43
CAPITULO 3. LA SIMULACION DE PROCESOS QUIMICOS .....	47
3.1. Sistemás de simulación de procesos en estado estacionario ..	50
3.2. Orientación y enfoque de los sistemas de simulación .....	57

3.2.1.	Orientación Modular .....	59
3.2.1.1.	Enfoque Modular Secuencial .....	62
3.2.1.2.	Enfoque Modular Simultáneo .....	70
3.2.2.	Orientación Hacia las Ecuaciones .....	76
3.2.2.1.	Enfoque con Rompimiento .....	81
3.2.2.2.	Enfoque con Linearización Simultánea ....	84
3.3.	Estructura de las ecuaciones de simulación .....	87
3.4.	Formulación del Problema de Simulación .....	105
3.4.1.	Formulación I .....	111
3.4.2.	Formulación II .....	114
3.4.3.	Formulación III .....	118
3.5.	Evaluación de la matriz Jacobiana .....	119
3.5.1.	Diferenciación Analítica .....	120
3.5.2.	Perturbación Numérica Total .....	122
3.5.3.	Perturbación Diagonal .....	123
3.5.4.	Diferenciación Directa .....	125
<b>CAPITULO 4. EL CONCEPTO MODULAR SIMULTANEO EN LA SIMULACION .....</b>		<b>129</b>
4.1.	La convergencia modular simultánea .....	135
4.2.	Modelos Lineales .....	140
4.2.1.	Modelo lineal simple .....	142
4.2.2.	Modelos lineales de tipo gradiente .....	146
4.2.3.	Modelos lineales de tipo gradiente modificados ....	150
4.2.3.1.	Modelo Diferencial de fracciones de separación .....	150
4.2.3.2.	Modelo en base a funciones de residuos ..	156
4.2.3.3.	Modelo de tipo Quasi-Newton .....	163
4.3.	Modelos No Lineales .....	171
4.3.1.	Modelos Analíticos Reducidos .....	173

4.3.1.1.	Características del Modelo Analítico Reducido (MAR) .....	173
4.3.1.2.	Algoritmos para la aplicación de los MAR ..	180
4.3.1.3.	Modelos Reducidos .....	188
4.3.2.	Modelos de tipo evolutivo .....	197
4.4.	Ampliación del concepto Ciclo Interno - Ciclo Externo ....	205
4.4.1.	Variables en el Ciclo Externo .....	206
4.4.1.1.	Convergencia de todas las variables ....	207
4.4.1.2.	Convergencia de las variables de las corrientes de corte y alimentación .....	208
4.4.2.	Manejo de discontinuidades .....	211
4.4.3.	Convergencia de las variables del Ciclo Externo ...	212
4.4.4.	Cálculos Integrados .....	218
CAPITULO 5.	ESTRUCTURAS PARA LA IMPLEMENTACION DE LOS METODOS MODULARES SIMULTANEOS .....	223
5.1.	Bases para la concepción de un simulador modular .....	226
5.1.1.	Soporte computacional .....	226
5.1.2.	Interface con el usuario .....	232
5.1.3.	Organización de actividades en un simulador modular ..	236
5.1.3.1.	Etapas de una simulación .....	236
5.1.3.2.	Estructura de un simulador modular .....	251
5.2.	Orientación de las estructuras modulares simultáneas .....	256
5.2.1.	Antecedentes .....	256
5.2.2.	Estructura tipo A .....	261
5.2.3.	Estructura tipo B .....	267
5.2.3.1.	Estructura para el algoritmo lineal .....	268
5.2.3.2.	Estructura para el algoritmo no lineal ..	272
5.2.4.	Estructura tipo C .....	277

## IV

CAPITULO 6. CONCEPTO MODULAR SIMULTANEO DEL SISTEMA SMS/ZAR .....	287
6.1. Consideraciones preliminares .....	290
6.2. El simulador modular secuencial SGP/ZAR .....	293
6.2.1. Descripción General .....	293
6.2.2. Programa Ejecutivo de Control .....	299
6.3. El simulador modular simultáneo SMS/ZAR .....	313
6.3.1. Conceptos Básicos .....	313
6.3.2. Guía para su desarrollo computacional .....	324
CAPITULO 7. CONCLUSIONES .....	337
7.1. Desarrollo conceptual de la simulación .....	341
7.2. Desarrollo de los medios de implementación .....	351
APENDICE A. METODOS DE SOLUCION DE ECUACIONES NO LINEALES .....	359
A.I. Métodos de convergencia local y convergencia expandida ....	364
A.II Métodos de convergencia global .....	400
APENDICE B. METODOS DE SOLUCION DE MATRICES DISPERSAS .....	423
B.I Etapa de Reordenamiento .....	431
B.II Etapa de solución numérica .....	474
APENDICE C. MODELOS REDUCIDOS PARA OPERACIONES UNITARIAS .....	491
C.I Modelos Reducidos para operaciones unitarias simples .....	495
C.II Modelos Reducidos para operaciones unitarias complejas ....	506
NOMENCLATURA .....	533
BIBLIOGRAFIA .....	541

## LISTA DE FIGURAS

FIGURA	N O M B R E	PAGINA
2.1	Ingeniería de Sistemas de Proceso e Ingeniería Química. ....	21
2.2	Metodología de la Ingeniería de Procesos aplicada a la creación de Procesos Químicos. ....	24
2.3	El proyecto total para el diseño y construcción de una planta química. ....	28
2.4	Etapas en el desarrollo del diseño de procesos químicos. ...	30
2.5	Concepto Fundamental de la Etapa de Síntesis. ....	32
2.6	La etapa de Síntesis como un subsistema. ....	34
2.7	Concepto Fundamental de la etapa de Análisis. ....	38
2.8	La etapa de Análisis como un subsistema. ....	39
2.9	Espacio de evolución de los cálculos para el diseño de procesos. ....	46
3.1	Flujo de información en un sistema de simulación. ....	52
3.2	Estructura de bloques de un sistema de simulación. ....	54
3.3	Estructura típica de un módulo unitario. ....	56
3.4	Ejemplo de un proceso con una recirculación simple. ....	65
3.5	Ejemplo de un proceso con recirculación simple y una especificación de diseño. ....	67

## VI

3.6	Ciclos múltiples de iteración anidados del enfoque modular secuencial. ....	68
3.7	Concepto básico del enfoque modular simultáneo. ....	71
3.8	El enfoque modular simultáneo en forma de ciclo interno y ciclo externo. ....	73
3.9	Esquema de cálculo para los enfoques basados en las ecuaciones. ....	77
3.10	Flujo de información para un módulo unitario típico. ....	90
3.11	Ejemplo de proceso con una recirculación. ....	94
3.12	Ejemplo de un problema de simulación mal especificado. ....	99
3.13	Matriz de ocurrencia del problema de la Figura 3.12. ....	101
3.14	Ejemplo de un problema de simulación mal especificado. ....	103
3.15	Ejemplo de un proceso con tres corrientes de recirculación. .	107
3.16	El proceso de la Figura 3.15 con sus corrientes de recirculación en forma de corrientes de corte. ....	108
3.17	El proceso de las Figuras 3.15 y 3.16 dispuesto en la forma entrada-salida de sus corrientes de corte. ....	109
3.18	Proceso simple usado para describir las formulaciones I, II y III. ....	112
3.19	Matriz Jacobiana para la formulación I. ....	115
3.20	Matriz Jacobiana para la formulación II. ....	117
4.1	Proceso con ciclos de recirculación independientes. ....	132
4.2	Proceso con ciclos de recirculación anidados. ....	133
4.3	Descomposición de cálculo del concepto modular simultáneo. .	139
4.4	Descomposición de cálculo del enfoque modular secuencial. ..	141
4.5	Influencia de un parámetro de equipo manipulado sobre la matriz unitaria. ....	151

## VII

4.6	Esquema de cálculo para el enfoque modular simultáneo usando modelos aproximados lineales. ....	157
4.7	Esquema de cálculo para el enfoque modular simultáneo basado en la evaluación de residuos. ....	161
4.8	Descomposición de cálculo al emplear modelos reducidos en forma de ciclo interno y ciclo externo. ....	181
4.9	Esquema de cálculo para el enfoque modular simultáneo usando modelos reducidos no lineales. ....	187
4.10	Secuencia de cálculo del algoritmo modular simultáneo con dos pruebas de convergencia. ....	213
4.11	Enfoque modular simultáneo con modelos rigurosos manejados con algoritmos de ciclo interno y ciclo externo. ....	219
4.12	Representación de un simulador modular simultáneo bajo un enfoque de ciclo interno y ciclo externo total. ....	221
5.1	Diagrama de flujo de un proceso usado como ejemplo. ....	240
5.2	Diagrama de flujo de información del proceso de la Figura 5.1. ....	242
5.3	Matriz del proceso de la Figura 5.1. ....	244
5.4	Gráfica dirigida del proceso de la Figura 5.1. ....	246
5.5	Matriz adyacente de la gráfica dirigida del proceso de la figura 5.1. ....	248
5.6	Estructura general del simulador ASPEN. ....	252
5.7	Etapas en la generación del P.S.E. en el preprocesador de ASPEN. ....	254
5.8	Flujo de información en el simulador modular simultáneo SIMMOD. ....	263
5.9	Diagrama de flujo de información para la estructura A. ....	266
5.10	Relación entre los niveles de control de FLOWTRAN para el algoritmo modular simultáneo lineal. ....	269

# VIII

5.11	Relación funcional del algoritmo modular simultáneo lineal de Jirapongphan. ....	271
5.12	Relación entre los niveles de control de FLOWTRAN para el algoritmo modular simultáneo no lineal. ....	274
5.13	Relación funcional del algoritmo modular simultáneo no lineal de Jirapongphan. ....	276
5.14	Diagrama de flujo de información para la estructura C. ....	281
6.1	Relación funcional entre los programas del simulador SGP/ZAR. ....	294
6.2	Estructura del Programa Principal del simulador SGP/ZAR. ...	296
6.3	Diagrama de flujo de la subrutina SUBSET. ....	302
6.4	Diagrama de flujo de la subrutina SCAN. ....	305
6.5	Diagrama de flujo de la subrutina EQUIP. ....	306
6.6	Diagrama de flujo de la subrutina REQUIP. ....	308
6.7	Diagrama de flujo de la subrutina EQCALL. ....	309
6.8	Diagrama de flujo de la subrutina RCYCLE. ....	311
6.9	Diagrama de flujo de proceso usado como ejemplo. ....	326
A.1	Curva típica para $P(\lambda)$ usando el método Levenberg Marquardt. ....	391
A.2	Curva típica para $P(\lambda)$ usando el método híbrido de Powell. ....	392
A.3	Ejemplo del método de Homotopía-continuación clásico. ....	406
A.4	Gráfica de la función A.98. ....	407
A.5	Ruta homotópica para la ecuación A.98, empleando homotopía de Newton. ....	408
A.6	Ruta homotópica para la ecuación A.98, empleando homotopía de Punto Fijo. ....	410

## IX

A.7	Ejemplo de ruta homotópica para un problema complejo. ....	411
A.8	Diagrama de flujo de la implemetación de la homotopía termodinámica. ....	418
B.1	Estructuras organizadas de matrices dispersas elemento por elemento. ....	428
B.2	Estructuras organizadas de matrices dispersas en forma de bloques. ....	429
B.3	Justificación para elegir la forma triangular inferior al reordenar una matriz dispersa. ....	433
B.4	Matriz de ocurrencia y su gráfica asociada. ....	439
B.5	Diagrama de flujo para el algoritmo de Sargent y Westerberg. ....	441
B.6	Diagrama de flujo para el algoritmo de Tarjan. ....	443
B.7	Formas de reordenamiento para una matriz de ocurrencia típica. ....	453
B.8	Desarrollo del algoritmo P4 en el reordenamiento de una matriz dispersa. ....	458
B.9	Desarrollo de la selección de pivotes usando el método de Markowitz. ....	472
C.1	Flujo de información para el modelo reducido de un mezclador. ....	500
C.2	Flujo de información para el modelo reducido de un divisor de corriente. ....	500
C.3	Flujo de información para el modelo reducido de un calentador. ....	503

C.4	Flujo de información para el modelo reducido de un compresor. ....	503
C.5	Flujo de información para el modelo reducido de un reactor estequiométrico. ....	510
C.6	Flujo de información para el modelo reducido de un reactor con flujo tapón. ....	510
C.7	Flujo de información para el modelo reducido de una columna de absorción. ....	519
C.8	Flujo de información para el modelo reducido de una columna de destilación. ....	529

## LISTA DE TABLAS

TABLA	N O M B R E	PAGINA
4.1	Historia de las iteraciones para el problema de Cavett convergiendo todas las variables en el ciclo externo. ....	215
4.2	Historia de las iteraciones para el problema de Cavett omitiendo la convergencia de las temperaturas en las corrientes de corte. ....	217
A.1	Clasificación de los métodos de solución de ecuaciones no lineales. ....	363
C.1	Análisis de grados de libertad del modelo riguroso de una columna de absorción. ....	526
C.2	Análisis de grados de libertad del modelo reducido de una columna de absorción. ....	527

CAPITULO 1

INTRODUCCION

## CAPITULO 1. INTRODUCCION.

El ingeniero químico siempre está en la búsqueda continua de nuevos procesos que incorporen los más promisorios avances tecnológicos, incidiendo sobre los parámetros críticos de diseño y operación. Conforme crece la complejidad de un proceso, los cálculos correspondientes al balance de masa y energía no bastan y se requiere información adicional más especializada, referente a los parámetros idóneos para los equipos, tiempos de residencia en reactores, número de etapas en los equipos de separación, etc.

Para obtener información de este tipo, las simples ecuaciones descriptivas del proceso no son suficientes, siendo necesario agregar ecuaciones de equilibrio de fases, relaciones de presión-volumen-temperatura, restricciones de operación y diseño, etc.; todo lo cual dá forma a sistemas del orden de 100.000 ecuaciones, para procesos industriales típicos.

Adicionalmente, las modernas tecnologías exigen un uso eficiente de las materias primas y de las fuentes de energía utilizadas dentro de un proceso, lo que propicia la existencia de sistemas complejos de integración de energía y recirculación de materiales, dificultando la tarea de simulación y diseño.

Al abordar estos problemas se hace necesario tener en mente algunas consideraciones. Primero, deberá existir la posibilidad de manipular los parámetros y especificaciones de diseño como si fueran variables con el objeto de lograr la convergencia funcional del proceso y cumplir con su esquema de integración de energía y recirculación de materiales; en otras palabras, se busca ampliar las capacidades de una simulación para que aborde el problema de diseño, propiamente dicho. Segundo, el manejo de parámetros y ecuaciones de diseño aumenta la complejidad de la simulación y en ocasiones los resultados tienden a salirse de los márgenes de operación funcionalmente factibles, incidiendo en los costos de diseño y operación, por lo tanto, es de suma importancia que la capacidad de un simulador sea expandida para el adecuado manejo de la optimización del proceso. Tercero, la ampliación de las capacidades de un simulador, conduce también a la ampliación en el tamaño de los sistemas de ecuaciones no lineales y esto se refleja en el aumento en el número de iteraciones necesarias para obtener la convergencia, llegando a ser de más del doble de las que se necesitan para realizar una simulación simple. Cuarto, por todo lo anterior, es imprescindible la implementación de técnicas eficientes para el manejo y solución de sistemas de ecuaciones no lineales y matrices dispersas, así como innovadores métodos de partición y reordenamiento, todo ésto integrado en estructuras flexibles, que realicen un adecuado manejo del problema.

Desafortunadamente, en la mayoría de los simuladores actua-

les, tanto comerciales como académicos, estas ampliaciones son difíciles de realizar, puesto que la estructura que presentan es modular secuencial y que, generalmente, son sistemas muy eficientes para procesos con recirculaciones que no están fuertemente relacionadas, pero no son adecuados para esquemas complejos de recirculación; además, no pueden manejar los parámetros de diseño como variables de proceso y la optimización no se aborda como una extensión del simulador, es decir, se constituye como un problema adicional de simulación fuertemente restringida, duplicando el tiempo de cómputo.

Por todo lo anterior y ante la creciente complejidad de los procesos químicos, el concepto modular secuencial se ve cada vez más disminuido y se requieren instrumentos más eficientes para la simulación, diseño y optimización.

Se han propuesto varios enfoques para abrir nuevos caminos a la simulación, sin embargo, fuera de los círculos académicos, algunas de esas sendas no reciben mucho interés. La principal razón es que se tienen invertidos una gran cantidad de recursos de todo tipo en el enfoque modular secuencial y no se prevee, al menos a corto plazo, el total abandono de esta orientación.

Es importante resaltar que gran parte de los avances en materia de simulación y diseño de procesos ha tenido su origen en investigaciones y proyectos académicos, algunos de ellos financiados y posteriormente adoptados por la iniciativa privada.

Un proyecto para la creación y puesta a punto de un simulador exige el empleo de una gran cantidad de recursos de diversa índole.

Principalmente se destaca personal con conocimientos sobre las ciencias de la Ingeniería Química y con capacidad de traducir estos conocimientos en un diseño factible, tanto en aspectos técnicos como económicos.

Restringiéndonos al ámbito puramente académico, en un proyecto de este tipo participan elementos con diferentes niveles de preparación: alumnos de pregrado, graduados, de postgrado, así como técnicos, profesores, investigadores, etc.; dando forma a una fuerza creativa dentro de la institución en que se desarrolla el proyecto, lo que rendirá beneficios a corto y largo plazos, destacando la elevación del nivel académico, una mayor compenetración en una rama muy promisoría en la Ingeniería Química y la posibilidad de estructurar proyectos de investigación y desarrollo complementarios.

El proyecto para la creación de un sistema de simulación no está limitado a la obtención de un programa de computadora que produzca resultados. Debe ser visto como un sistema dinámico, susceptible de ser mejorado y con capacidad para ampliar sus capacidades, mediante la incorporación de técnicas innovadoras; sin olvidar que para su adecuado desarrollo se requiere un firme respaldo en diversos campos, destacando el Análisis Numérico, Modelado Matemático, Programación Lineal y No Lineal, Ciencias de la Computación, Termodinámica, Ingeniería de Procesos, etc.; res-

paldo que no se logra a corto plazo, sino con un mejoramiento gradual en el entorno, tanto académico como humano, en el que se presentará el proyecto.

Este trabajo pretende contribuir al enriquecimiento de ese entorno presentando las bases teóricas para el desarrollo de un simulador modular simultáneo, provisto de una estructura flexible y con capacidad para incorporar ideas innovadoras, tanto en modelos representativos de operaciones unitarias (simples y rigurosos), como en técnicas de solución de ecuaciones algebraicas no lineales, en métodos para el manejo y solución de matrices lineales dispersas y en la integración de mayores opciones termodinámicas; todo esto con el fin de abordar eficientemente los problemas de simulación simple, simulación con restricciones de diseño y una eventual ampliación al tratamiento de problemas de optimización.

Para lograr este propósito, se plantean los siguientes objetivos parciales:

-- Ubicar la posición de la Simulación de Procesos Químicos en Estado Estacionario en el contexto de la Ingeniería Química en general y en la Ingeniería de Procesos en particular.

-- Describir las orientaciones y enfoques más representativos del desarrollo de la simulación de procesos químicos en estado estacionario, destacando sus ventajas y desventajas, con

el objeto de justificar la importancia y la necesidad de desarrollar sistemas modulares simultáneos.

-- Recopilar y analizar la información existente acerca de los principales modelos de convergencia bajo el concepto modular simultáneo, destacando especialmente aquellos que muestren el actual nivel de desarrollo de este enfoque.

-- Identificar las estructuras computacionales más importantes, susceptibles de ser adoptadas bajo el enfoque modular simultáneo.

-- Elaborar planteamientos concretos acerca de las acciones a seguir para el desarrollo de un sistema de simulación, adoptando un determinado modelo de convergencia modular simultánea y bajo una adecuada estructura computacional.

-- Recopilar información sobre temas afines al objetivo central de este trabajo y que pudieran servir de apoyo en la creación del sistema de simulación, incluyendo tanto técnicas y métodos tradicionalmente empleados, como aquellos que pueden ser de importancia en el futuro próximo y que ameriten ser considerados en la planeación del proyecto creativo del simulador modular simultáneo.

A continuación se presenta un resumen de los principales temas que se abordan en cada capítulo de este trabajo, con el fin de alcanzar los objetivos planteados.

En el Capítulo 2 se describe la relación existente entre la Ingeniería Química y la Ingeniería de Procesos, destacando las etapas de Síntesis, Análisis y Optimización, dentro del Diseño de Procesos Químicos.

En el Capítulo 3 se presenta una introducción a la simulación en Estado Estacionario, incluyendo una descripción de sus principales vertientes: la orientación modular y la orientación basada en las ecuaciones (orientación no modular). También se muestra la forma en que se estructuran las ecuaciones descriptivas del proceso para formular un problema de simulación y se finaliza con algunas opciones para la evaluación del sistema de ecuaciones descriptivas, conocimientos que serán muy útiles en los siguientes capítulos.

El Capítulo 4 aborda los conceptos teóricos y la evolución de la convergencia modular simultánea, incluyendo lo más representativo en modelos lineales y no lineales, finalizando con el innovador concepto de convergencia independiente en un ciclo interno y en un ciclo externo.

El Capítulo 5 muestra las bases sobre las que se estructura un simulador modular en general, haciendolo extensivo al enfoque modular simultáneo bajo ciertas condiciones, identificando tres estructuras representativas de esta tendencia.

El Capítulo 6 indica una serie de lineamientos generales que deben considerarse al diseñar un simulador modular simultáneo. Se inicia con un breve análisis de los recursos disponibles y de la orientación que tendrá el sistema de simulación, en base a la información contenida en los Capítulos 4 y 5. A continuación se describe el simulador modular secuencial SGP/ZAR que será tomado como eje central del nuevo simulador modular simultáneo y se concluye con la recomendación de las áreas de trabajo para lograr la implementación del sistema simultáneo SMS/ZAR.

El Capítulo 7 se destina a exponer algunas conclusiones, producto de la realización de este trabajo, incluyendo algunas consideraciones de lo que se espera de los simuladores que se están diseñando en la actualidad, para que funcionen a partir de la próxima década.

Se incluyen tres apéndices, que abordan temas de apoyo, tanto para este trabajo, como para el desarrollo computacional del sistema modular simultáneo propuesto.

El Apéndice A contiene las más importantes técnicas de solución de sistemas de ecuaciones algebraicas no lineales, incluyendo los métodos actualmente en uso en simuladores secuenciales, así como técnicas desarrolladas bajo un enfoque hacia las ecuaciones, que serán de gran utilidad al ser adoptadas bajo el enfoque modular simultáneo.

El Apéndice B presenta varios temas referentes al manejo y solución de matrices lineales dispersas, del tipo que son generadas en los problemas de simulación y diseño, incluyendo técnicas propias de los simuladores actualmente en uso y otras que, a pesar de su probada eficiencia en otros campos, aún no se incorporan a la simulación de procesos químicos.

El Apéndice C contiene modelos simples de operaciones unitarias, conocidos como Modelos Reducidos, que constituyen innovadoras opciones, dignas de considerarse, especialmente por ser adaptables al concepto ciclo interno - ciclo externo, y con excelentes propiedades descriptivas y eficiente manejo computacional.

CAPITULO 2

GENERALIDADES

## CAPITULO 2. GENERALIDADES

### 2.1. ANTECEDENTES

Tal vez una de las tareas más complejas y exigentes que tiene que afrontar un ingeniero químico es el Diseño de Procesos. Es la esencia misma de su educación y el punto en donde convergen por una parte, sus conocimientos técnicos y experiencia industrial, y por otra, características propias de su ser, como son la intuición y el ingenio.

Contra lo que pudiera pensarse, el Diseño de Procesos como disciplina formal dentro de la Ingeniería Química, es de un desarrollo relativamente reciente, alentado por los cambios estructurales en la economía mundial así como en los sistemas de producción y comercialización, los cuales hicieron necesaria la revisión de las técnicas relativas a la concepción de los procesos químicos.

Cuando el objetivo de la producción era proporcionar bienes para una demanda creciente, dentro de economías de rápida expansión y en donde la energía y las materias primas estaban disponibles en cantidades abundantes, los procesos fueron diseñados en una forma totalmente empírica e intuitiva, situación que se repi-

te continuamente desde la Revolución Industrial hasta finales de la Segunda Guerra Mundial. Al finalizar la guerra se realizó una revisión de las importantes aportaciones que el afán bélico proporcionó en cuanto a producción industrial, alentándose su rápida asimilación con el objeto de mejorar los procesos ya existentes; por otra parte, existía un ímpetu innovador creado por los anhelos de reconstrucción económica, todo lo cual conformaba un clima propicio para que el empirismo disminuyera y diera paso al espíritu científico dentro del Diseño de Procesos.

Es precisamente en la década de los cincuentas, cuando se sientan las bases formales del Diseño de Procesos, logrando altos niveles de creación, creando procesos industriales muy complejos, cuyo desarrollo y puesta en marcha exigían una gran conjunción de conocimientos referentes a los más variados campos de la ciencia y la tecnología.

Es a principios de los sesentas cuando se propone el término Ingeniería de Procesos, con el objeto de agrupar los campos de la Ingeniería Química destinados a indicar cómo planear, diseñar, operar y controlar las diferentes operaciones unitarias de un proceso químico. Esta disciplina surge en el campo académico a partir de la necesidad de reordenar todo el cúmulo de información existente, así como la de fijar las pautas que habría de seguir el Diseño de Procesos, con el objeto de satisfacer los requerimientos de la expansión industrial.

Sin embargo, el impulso decisivo para el desarrollo del Diseño de Procesos fué dado a mediados de los sesentas, a través de dos sucesos, independientes entre sí. Primero, la productividad industrial muestra serios indicios de descenso en sus tasas de crecimiento; la estructura productiva existente enfrenta sus primeras limitaciones en cuanto a la disponibilidad de materias primas y energía, lo que obliga a un estudio a fondo de las eficiencias y puntos críticos de los procesos productivos, tanto existentes, como por ser creados, promoviendo la creación de diseños que contienen complejos esquemas de recirculación de materiales y de integración de energía. Segundo, anteriormente la computación, como un auxiliar en el diseño racional de procesos, era prácticamente desconocida por el ingeniero químico, sin embargo, el explosivo desarrollo de las computadoras y la creación de lenguajes de alto nivel, abren nuevas y amplias perspectivas a la creación de procesos químicos complejos y eficientes.

Ahora existen los medios para que todo el cúmulo de conocimientos generados desde los años cincuentas sea puesto a prueba, generando resultados que tuvieron un efecto multiplicador hasta constituir, en la actualidad, uno de los campos de mayor desarrollo dentro de la Ingeniería Química: el Diseño de Procesos Auxiliado por Computadora.

Los primeros pasos al respecto están representados por programas aislados, aplicables a operaciones unitarias y procesos

muy específicos, con grandes limitaciones de flexibilidad.

Actualmente, la complejidad de los procesos industriales hace imperativo el uso de grandes sistemas computacionales, capaces de manejar un proceso como un todo, pudiéndose afirmar que hoy en día ninguna gran planta química o petroquímica podría ser construida sin realizar la simulación y diseño del proceso en una computadora.

Sin embargo, la implementación de tales sistemas exige el uso de herramientas matemáticas sofisticadas, al grado de promover la creación, desarrollo y perfeccionamiento de técnicas especializadas para abordar los problemas muy particulares que se presentan al diseñar procesos químicos.

## 2.2. INGENIERIA QUIMICA E INGENIERIA DE PROCESOS.

El Diseño de Procesos Químicos ha sufrido una rápida y muy significativa evolución, tal y como fué planteado en la sección anterior. De ser una actividad netamente empírica y aislada, fué nutriéndose de los avances de los más diversos campos de la

ciencia y la técnica, hasta convertirse en un campo de investigación formal por sí mismo.

En el seno de la Ingeniería Química han madurado nuevas concepciones y surgido innovadoras tendencias que la enriquecen y la convierten en una disciplina cada vez más compleja y en donde el Diseño de Procesos adquiere una nueva dimensión.

#### 2.2.1. INGENIERIA DE PROCESOS.

A partir de 1963 se utilizó por primera vez, dentro de la Ingeniería Química, la terminología de Ingeniería de Procesos, aplicada sobre la dinámica y control de algunas operaciones unitarias, sin abordar conceptos de optimización de decisiones.

Su significado y contenido ha cambiado con el tiempo y desde la fecha en que se creó esta terminología, buscando recopilar y ordenar la información que estaba siendo generada en forma abundante, pero desarticulada, actualmente la Ingeniería de Procesos se ha constituido como un campo académico y tecnológico referente a las metodologías para la toma de decisiones en la Ingeniería Química. Tales metodologías son responsables de indicar como planear, diseñar, operar y controlar cualquier clase de operación unitaria, proceso químico y la industria química misma.

Debe notarse que esta concepción es muy amplia. Si bien es cierto, la Ingeniería Química puede partir desde el punto de vista de una operación unitaria, su objetivo final es la eficiencia en el diseño y operación de procesos; sin embargo, no debe ignorarse que estos procesos no existen en forma independiente unos de otros, por lo tanto, es razonable considerarlos como subsistemas que forman parte de otros sistemas mayores, tal y como será descrito más adelante.

Los procesos químicos están compuestos de un número relativamente pequeño de operaciones unitarias, comparado con el gran número de procesos existentes. Desde este punto de vista todos los procesos deben ser construídos razonablemente de acuerdo a los principios de diseño para operaciones unitarias, es decir, cada proceso químico puede ser considerado como un sistema de operaciones unitarias en combinación, tal y como se muestra en los ejes de  $\overline{OA}$  a  $\overline{OB}$  en la Figura 2.1.

Con el objeto de desarrollar nuevas operaciones unitarias y perfeccionar las ya existentes, es importante entender cuantitativamente los fenómenos fundamentales que son generados en ellas. De esta concepción han surgido las llamadas Ciencias de la Ingeniería Química, tales como Fenómenos de Transporte, Dinámica de Fluidos, Termodinámica, Cinética de Reacciones Químicas, y otras que han venido desarrollandose desde los años cincuentas.

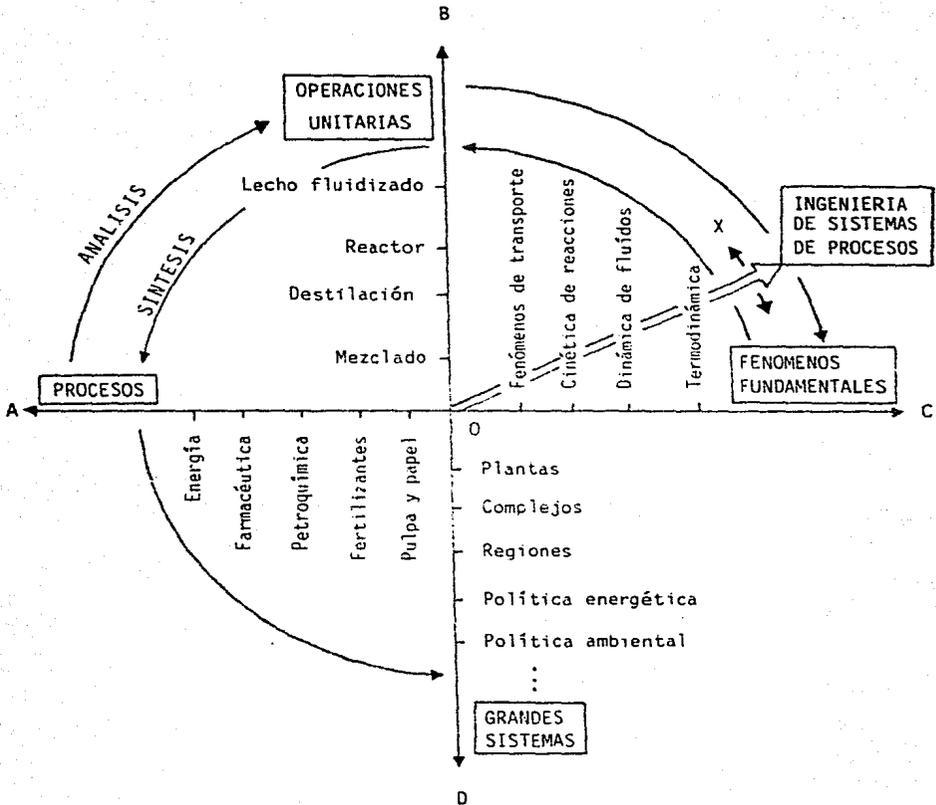


FIGURA 2.1. INGENIERIA DE SISTEMAS DE PROCESO E INGENIERIA QUIMICA.

En otras palabras, cada operación unitaria puede ser identificada como un sistema combinado de fenómenos fundamentales, tal y como es representado en los ejes de  $\overline{OB}$  a  $\overline{OC}$ , en la Figura 2.1.

Hasta aquí se han considerado tres polos fundamentales de información técnica: Fenómenos Fundamentales - Operaciones Unitarias - Procesos Químicos. Sin embargo, falta la manera de relacionar en forma coherente y funcional estos conocimientos, puesto que la información analítica de los fenómenos fundamentales puede ser útil para tomar decisiones sobre las operaciones unitarias y la información de éstas influirá sobre los procesos químicos.

Ante tal situación, la Ingeniería de Procesos deberá proporcionar metodologías destinadas a cubrir los anteriores requerimientos. Con referencia a la Figura 2.1, el eje  $\overline{OX}$ , que representa a la Ingeniería de Procesos, rota a través de todos los cuadrantes y cumple la tarea primordial de relacionar los ejes de  $\overline{OC}$  a  $\overline{OB}$  y de  $\overline{OB}$  a  $\overline{OA}$ .

Recordando el hecho de que los procesos químicos no existen independientemente de su entorno, se debe tener en cuenta la existencia de sistemas mayores, constituidos a partir de una simple empresa, creciendo hasta constituir complejos industriales y llegando a influir sobre las políticas energéticas y ambientales de toda una nación, sistemas de los cuales los procesos quí-

micos son sólo subsistemas. En la Figura 2.1, todos los procesos químicos representados en el eje  $\overline{OA}$ , deben ser responsables de los grandes sistemas contenidos en el eje  $\overline{OD}$ , es decir, la Ingeniería de Procesos deberá contribuir a planear complejos químicos, a resolver problemas ambientales y energéticos, tanto a nivel regional, como nacional y tomar decisiones de ingeniería acerca de otros problemas globales.

Ahora es posible entender que la motivación de introducir la terminología de Ingeniería de Procesos dentro de la Ingeniería Química, radica en que el diseño y operación de un proceso deberá cumplir múltiples objetivos, la mayoría de ellos fuertemente relacionados.

Anteriormente se mencionó que una de las tareas de la Ingeniería de Procesos era la de proporcionar una metodología tendiente a relacionar eficientemente Fenómenos Fundamentales - Operaciones Unitarias - Procesos Químicos, con el objeto de tomar decisiones acerca de problemas de diseño, operación y control, y para tal efecto se muestra en la Figura 2.2, una estructura general de la forma en que se aplica la Ingeniería de Procesos, y de la cual se pueden observar varios rasgos importantes.

El punto 1 se refiere a la recopilación de información analítica con respecto a varias de las Ciencias de la Ingeniería Química (Dinámica de Fluidos, Termodinámica, Fenómenos de Transporte, etc.), y a las Operaciones Unitarias (reactores, destila-

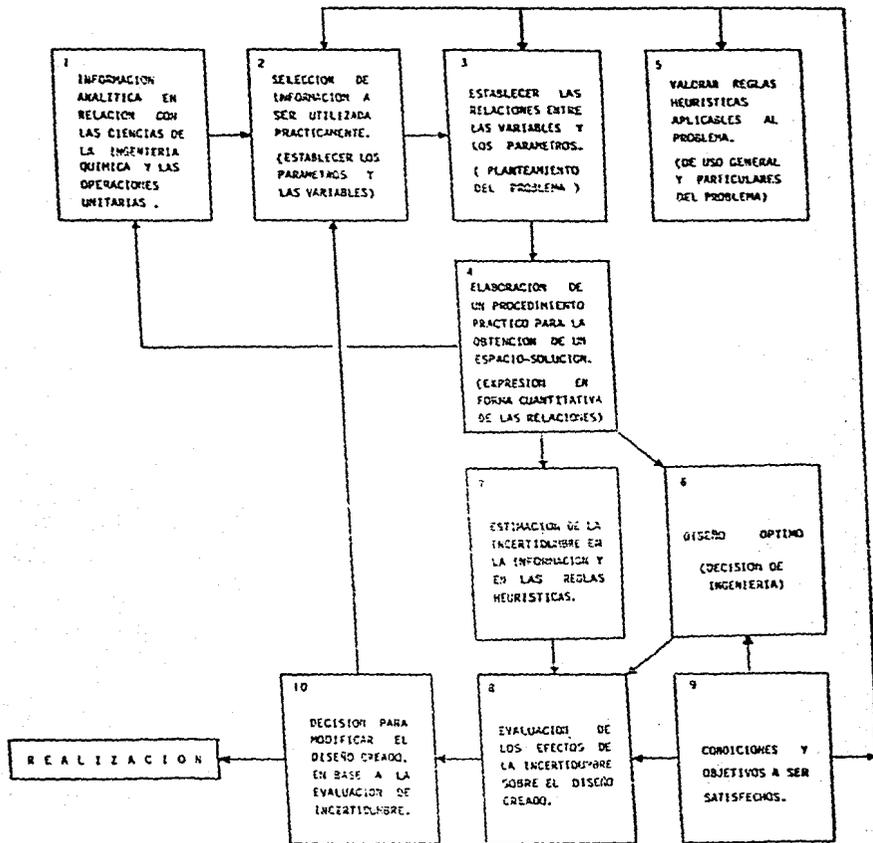


FIGURA 2.2. METODOLOGIA DE LA INGENIERIA DE PROCESOS APLICADA A LA CREACION DE PROCESOS QUIMICOS.

ción, mezclado, etc.).

En el punto 2 se realiza la selección de la información que será útil. Aquí se presenta la dificultad de precisar que cantidad de información será utilizada en forma práctica. No es costumbre emplear la totalidad de la información recopilada, puesto que existe, en la mayoría de los casos, un cierto nivel de precisión para cada condición u objetivo a ser satisfechos, además no todos los tipos de información disponibles tienen el mismo efecto sobre los objetivos y variables de decisión.

Los puntos 3, 4 y 5 se refieren esencialmente a la identificación del sistema que se va a manejar. A pesar de existir extensos trabajos acerca de como identificar cuantitativamente la relación entre variables y parámetros, aún son utilizadas las determinaciones empíricas e intuitivas, obtenidas a partir de la investigación analítica y representadas en forma de reglas heurísticas, las que son de capital importancia, principalmente en la etapa de diseño. Con estos conocimientos se elabora un procedimiento práctico que conduzca a obtener un espacio de solución, en base a la definición cuantitativa de las relaciones entre las variables de decisión y de estado, junto con los parámetros del sistema.

El punto 6, referente al Diseño de Procesos, es de suma importancia en el presente trabajo y será discutido con mayor

amplitud en las secciones siguientes.

En el punto 7 se estudia el grado de incertidumbre existente al utilizar la información disponible junto con las reglas heurísticas, para que en el punto 8 sea evaluado su efecto sobre el diseño obtenido en el punto 6. No se debe pasar por alto que la incertidumbre en la información de un sistema de proceso se expresa siempre mediante cifras estadísticas y estocásticas, pero que las decisiones de ingeniería (diseño del proceso), deben ser proporcionadas en forma determinística.

El punto 9 se refiere a las condiciones y objetivos que deberá satisfacer el sistema que se va a diseñar y por tanto participa prácticamente en todas y cada una de las etapas de la estructura de la Figura 2.2.

En el punto 10, en base a la evaluación de la incertidumbre, se deberá decidir si el proceso diseñado cumple o no con las condiciones y objetivos iniciales, a fin de promover su realización o repetir el ciclo creativo, recurriendo a la selección de mayor cantidad y calidad de información.

Se puede concluir que la creación de un proceso químico es una tarea interdisciplinaria y que debe responder a múltiples objetivos, algunos de los cuales pueden estar en conflicto, por lo tanto, se necesitan establecer con claridad las prioridades.

ponderaciones y criterios de decisión, a fin de lograr un proceso que cumpla con las necesidades existentes en una forma adecuada, minimizando la incertidumbre y elevando la eficiencia, en base a un diseño funcional y flexible.

### 2.2.2. DISEÑO DE PROCESOS.

En la sección anterior se ubicó al Diseño de Procesos dentro del marco de la Ingeniería Química en general y en forma particular dentro de la Ingeniería de Procesos. También se presentó una estructura cuyo punto central es la obtención de un diseño optimizado.

El Diseño de Procesos forma parte de las primeras etapas en el proyecto global para la realización de una planta química y para ejemplificar esto en la Figura 2.3 se muestran las principales áreas que considera el proyecto de creación para una planta química típica, desde la concepción del proceso hasta la construcción de las instalaciones.

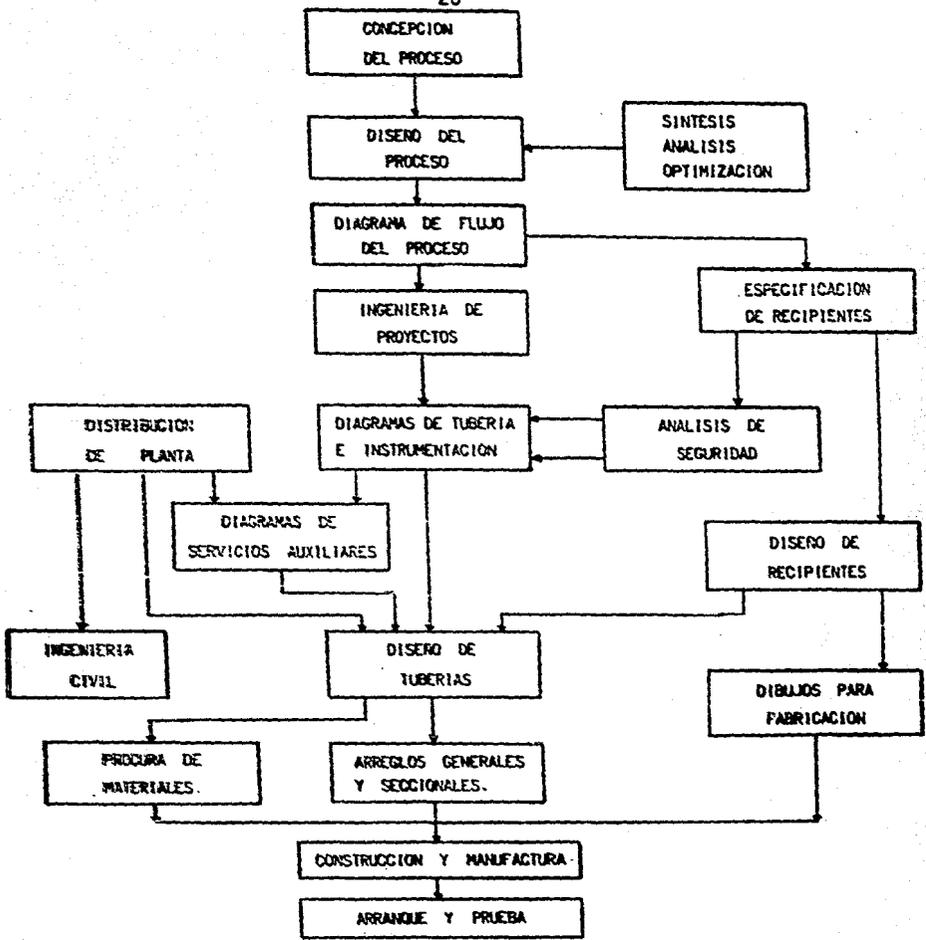


FIGURA 2.3

EL PROYECTO TOTAL PARA EL DISEÑO Y CONSTRUCCION DE UNA PLANTA QUIMICA.

El proyecto comienza con su etapa conceptual, es decir, se identifican las necesidades que cubrirá y una aproximación global a su solución, conteniendo estudios de factibilidad tanto técnica como económica.

A continuación se inicia la etapa de Ingeniería de Procesos propiamente dicha, representada esencialmente por el diseño del proceso, generando los diagramas de flujo de proceso, balances de masa y energía y especificaciones para la construcción del equipo y tuberías, integrando una estructura funcional de proceso

Con esta información se puede iniciar la etapa de Ingeniería de Proyectos, generando diagramas de tubería e instrumentación, junto con la secuencia de operaciones destinadas a la programación y control de las actividades tendientes a la erección física de la planta, incluyendo su arranque y puesta a punto.

En las Figuras 2.2 y 2.3 se ha incluido la etapa de Diseño de Procesos como un bloque más, en relación con otras etapas creativas; sin embargo, el esfuerzo de diseñar un proceso químico comprende a su vez una serie de subsistemas y algunas actividades de apoyo.

Esencialmente se consideran tres etapas: Síntesis, Análisis y Optimización, las cuales están representadas esquemáticamente en la Figura 2.4.

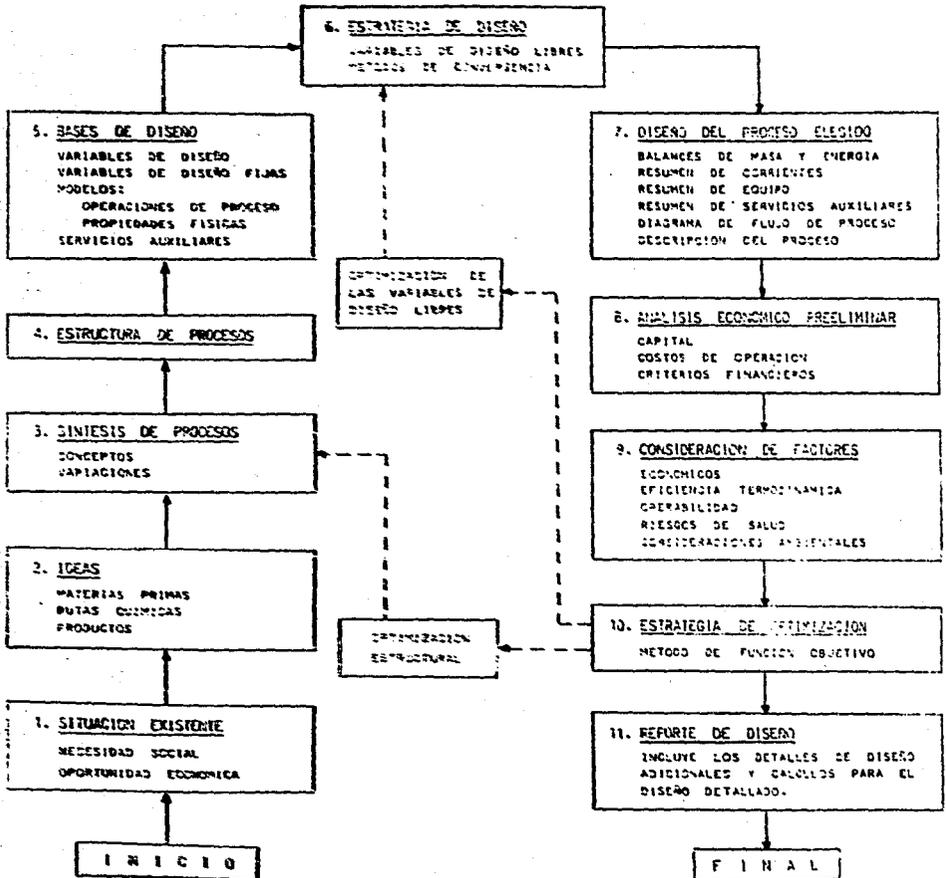


FIGURA 2.4. ETAPAS EN EL DESARROLLO DEL DISEÑO DE PROCESOS QUINICOS.

### 2.2.2.1. SINTESIS

Es la etapa inventiva en el diseño, en la cual se deciden las unidades de proceso requeridas y su interconexión, es decir, se proponen tanto la estructura como los flujos de materia y energía, necesarios para obtener los productos deseados.

A partir de una situación existente (esquina inferior izquierda, en la Figura 2.4), estando presentes tanto una necesidad social como una oportunidad económica, se conciben una serie de ideas que involucran materias primas y una secuencia de operaciones que logren su transformación en los productos deseados.

Usualmente los productos de un sistema son especificados con claridad, mientras que las materias primas pueden ser especificadas con exactitud, o bien, formar parte de un conjunto de condiciones disponibles, como se muestra en la Figura 2.5.

En cualquier caso, el diseñador deberá identificar la naturaleza y estructura del bloque desconocido, en una forma en que los objetivos iniciales sean alcanzados de la mejor manera.

En la síntesis de nuevos procesos es posible distinguir dos etapas: la elección de una ruta química de proceso, referida a las transformaciones químicas que unen materias primas con pro-

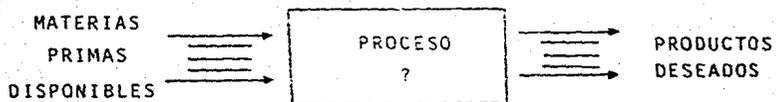
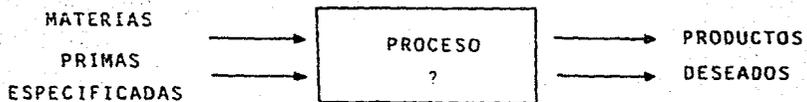


FIGURA 2.5. CONCEPTO FUNDAMENTAL DE LA ETAPA DE SINTESIS.

ductos; y el desarrollo del diagrama de flujo del proceso, que incluye la secuencias de operaciones unitarias, recirculación de materias e integración de energía. En cada fase se debe realizar una elección, dentro de un gran número de posibilidades, algunas de las cuales no están descritas en términos matemáticos. Nishida, Stephanopoulos y Westerberg (1980), describen ampliamente una forma racional de abordar estas dos etapas, en base a modelos mecanísticos y a correlaciones empíricas y heurísticas.

La síntesis de un proceso puede ser considerado desde el punto de vista de un sistema mayor, el cual incluye al sistema a ser sintetizado como un subsistema, concepto que es ilustrado en la Figura 2.6. Si las condiciones deseadas de entrada,  $X$ , y de salida,  $Y$ , son conocidas, el sistema  $S$  puede ser sintetizado usando la información analítica, tal y como se mostró en la Figura 2.2. En general, los valores de  $X$  y  $Y$  no pueden ser especificados arbitrariamente, puesto que debieron ser fijados tomando en cuenta que forman parte del sistema  $S1$ , el cual incluye al sistema  $S$  como un subsistema. Las condiciones de  $X$  y  $Y$  para diseñar el sistema  $S1$  pueden también ser restringidas por el hecho de que el mismo sistema  $S1$  puede ser un subsistema de un sistema mayor, el  $S2$ .

Desde este punto de vista habría una proyección infinita de sistemas hasta alcanzar el sistema universo, sin embargo, esto es esencialmente imposible y deberá elegirse un nivel apropiado

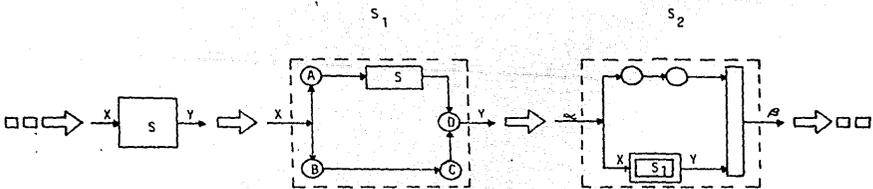


FIGURA 2.6. LA ETAPA DE SINTESIS COMO UN SUBSISTEMA.

y racional de proyección de sistemas.

Es precisamente en el punto 4 de la Figura 2.4, en donde se elige la estructura de proceso más prometedora, la cual será evaluada en la etapa de Análisis.

#### 2.2.2.2. ANALISIS

Una vez elegida cierta estructura de proceso, la siguiente etapa es la evaluación tanto de su viabilidad técnica, como económica. Con referencia a la Figura 2.4, la etapa de Análisis comprende los bloques 5, 6, 7, 8 y 9.

A partir de la estructura de proceso propuesta, se establecen sus bases de diseño las cuales consisten de la determinación de las variables de diseño, la elección de los valores de las variables de diseño fijas o dependientes y la selección de los modelos matemáticos para todas y cada una de las operaciones del proceso, para las correlaciones de propiedades termofísicas y servicios auxiliares del proceso. A continuación se determina una estrategia de diseño, la cual incluye la estimación inicial de los valores para las variables de diseño no fijadas (variables independientes, libres o manipuladas). También se

procede a elegir los métodos de cálculo y criterios de convergencia adecuados para desarrollar el balance de materia y energía.

Una vez reunida toda la información anterior se procede a transformarla a fin de producir un diseño del proceso. Esta etapa comprende varias actividades, de las cuales, la obtención del balance de materia y energía, a través de simulación en estado estacionario, es la que reviste importancia capital y será objeto de estudio en este trabajo. Otras actividades que contribuyen al diseño son la obtención de información resumida de cada corriente de proceso, información de las especificaciones de diseño de cada equipo, información acerca de los requerimientos de servicios auxiliares, presentación del diagrama de flujo de proceso, análisis termodinámicos, etc.

Hasta este nivel de desarrollo se tienen elementos suficientes para realizar un estudio económico preeliminar, considerando aspectos tales como el del capital requerido, costos de operación, criterios de plausibilidad, etc.

El siguiente paso es la confrontación de aspectos técnicos y económicos mediante la consideración de factores tales como la eficiencia termodinámica, eficiencia de operación, riesgos de salud, seguridad y ambientales, etc.

La forma en que el Análisis opera sobre la estructura del

proceso propuesta, es aplicando total o parcialmente las Bases de Diseño en la forma indicada por la Estrategia de Diseño, con el objeto de definir las condiciones de salida, que antes del Análisis eran desconocidas, tal como se muestra en la Figura 2.7.

Desde el punto de vista de la Ingeniería de Procesos, la etapa de Análisis consiste en descomponer el sistema objetivo (en este caso, la estructura de proceso propuesta), en varios subsistemas (operaciones unitarias, corrientes de proceso, ciclos de recirculación, etc.), definiendo las relaciones entre cada uno.

Como el hecho de definir las condiciones de cada subsistema, puede a su vez ser considerado como analizar cada subsistema, entonces éste puede ser susceptible de ser descompuesto en algunas fracciones menores. La Figura 2.8 ilustra la forma en que se concibe el Análisis. A fin de explicar las características del sistema S, en forma tan general como sea posible, se considera compuesto por los subsistemas A, B y C, con sus interconexiones  $\alpha_1$ ,  $\alpha_2$ ,  $\beta_1$  y  $\beta_2$ . Ahora, extrayendo el subsistema B, nuevamente es descompuesto en sus elementos a, b, c y d. De acuerdo a lo anterior, cada sistema podría ser descompuesto en sus elementos infinitesimales, con el objeto de obtener una información analítica completa del sistema, pero esto es esencialmente imposible y nuevamente debe ser elegido un nivel apropiado y racional de introspección del sistema original.



FIGURA 2.7 CONCEPTO FUNDAMENTAL DE LA ETAPA DE ANALISIS

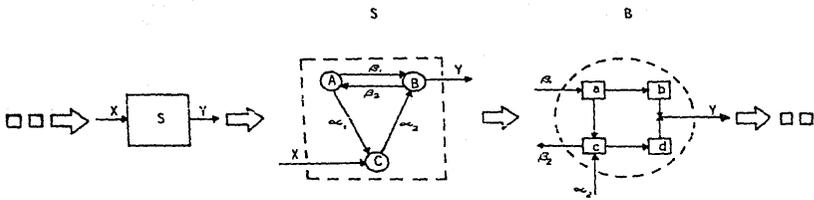


FIGURA 2.8. LA ETAPA DE ANALISIS COMO UN SUBSISTEMA.

### 2.2.2.3. OPTIMIZACION

A partir de la información obtenida en las etapas de Síntesis y Análisis, se establece una estrategia de Optimización, aplicada a una función objetivo adecuada.

La Optimización puede ser realizada tanto en la estructura como en los parámetros (condiciones de operación) del proceso.

Cuando se evalúan los datos de la etapa de Análisis, usualmente se descubre que ciertos niveles en las condiciones de operación (presiones, temperaturas, etc.), pueden influir profundamente en el dimensionamiento del equipo, en su operación, etc.; por lo tanto, se conserva la estructura propuesta en la etapa de Síntesis y únicamente se varían los valores de los parámetros de operación y diseño, ajuste que es conocido como Optimización de Parámetros, la cual influye directamente en una reconsideración de la Estrategia de Diseño, tal como se aprecia en la Figura 2.4.

En algunas ocasiones se puede decidir alterar el número y tipo de equipos, así como su interconexión, a fin de mejorar el proceso o porque el anterior arreglo resulta ser muy costoso, entonces se realiza una Optimización Estructural, la cual tiene una influencia directa en la etapa de Síntesis del proceso y por extensión también modifica la etapa de Análisis.

En la Figura 2.4 se puede apreciar que la Estrategia de Optimización consta de dos ciclos iterativos anidados, uno para la Optimización de Parámetros y otro para la Optimización de Estructura. Sin embargo, como se verá más adelante, se están desarrollando algunos métodos para optimizar un proceso en forma simultánea tanto para la estructura como para los parámetros, y aún más, se pretende lograr la convergencia del balance de materia y energía simultáneamente con la Optimización.

De modo general, las bases teóricas para el desarrollo de la Optimización de Parámetros están más avanzadas que las teorías para la Optimización de la Estructura.

Una vez obtenido el diseño óptimo del proceso, se emite un reporte conteniendo la información generada en todas y cada una de las etapas de su concepción, así como detalles adicionales del diseño de equipo, útiles para la fase de construcción.

De lo citado anteriormente, es posible entender que el desarrollo de un proceso, desde su idea original hasta su edificación exige una participación interdisciplinaria.

La descripción del proceso crece rápidamente, a partir del enunciado de objetivos, hasta reunir un vasto conjunto de datos

numéricos y estadísticos de diversa procedencia, y a pesar de que las actividades mostradas en la Figura 2.4 son esencialmente secuenciales, visto en detalle existen varias actividades paralelas e íntimamente relacionadas, lo cual exige uniformidad en la presentación de los resultados, con el objeto de que sean transferidos eficientemente de una etapa a otra.

Anteriormente, cuando no se disponía del soporte computacional adecuado, todas las etapas mostradas en la Figura 2.4 eran realizadas mediante cálculos manuales y con la aplicación de modelos muy simples, principalmente basados en reglas empíricas e intuitivas.

Actualmente, la gran mayoría de esas etapas son realizadas en forma computarizada, mediante la aplicación de programas de computadora de diversa estructura y complejidad. Por lo tanto, el punto de vista con respecto al Diseño de Procesos ha sufrido modificaciones y hoy en día se pretende desarrollar procesos con el más alto grado de optimización posible, en base al empleo justificado de los más sofisticados modelos y herramientas computacionales que se tengan disponibles.

### 2.2.3. EL DISEÑO DE PROCESOS BAJO UN ENFOQUE EVOLUTIVO.

Antes de penetrar formalmente al núcleo de este trabajo, es necesario tener un panorama general acerca del espacio en que se desenvuelve un diseñador con el objeto de dar forma a un esquema de solución a un problema dado.

Sería ingenuo pensar que un ingeniero capta un problema y lo resuelve directamente, es decir, que a partir de la formulación del problema inmediatamente se conciba su solución final.

El diseñador parte de una información mínima o muy simple acerca del proceso y paso a paso va aumentando el número de bloques funcionales así como sus interconexiones, esto significa que el diseño va evolucionando en forma cualitativa y cuantitativa. De ninguna manera se debe entender que el diseño progresa en forma desordenada o aleatoria; su evolución está sujeta a ciertas reglas específicas para cada etapa y siguiendo una tendencia previamente definida.

No es el propósito de este trabajo el análisis de estos procedimientos, ya de por sí sumamente complejo, sin embargo, se pueden considerar tres áreas de evolución principales:

Inicialmente, el diseño no contiene ningún modelo del proceso; deberá construirse partiendo de aproximaciones simples, adicionando complejidad en forma sistemática y progresiva, una vez que ha explotado todas las posibilidades del comportamiento del

modelo más simple.

La segunda evolución se refiere a la naturaleza de los cálculos que se pueden realizar con cada uno de los modelos contruidos. Los cálculos más simples comprenderán aquellos casos en que la intuición del diseñador es más predominante, en el sentido en que él tiene la posibilidad de identificar los grados de libertad para el problema, e intuye el comportamiento de los modelos. Visto de este modo, esto es similar al problema de simulación, en donde, al ser conocidas las entradas y los parámetros de diseño de una unidad, entonces el comportamiento y las salidas de su modelo también se conocerán. Una vez que la simulación funciona, se puede evolucionar a lograr el diseño, modificando el caso anterior, ahora se especifican las salidas, total o parcialmente y se emplea el mismo modelo para obtener las especificaciones del equipo y/o sus entradas, manipulando algunas variables adecuadamente. Una evolución posterior sería la introducción de algoritmos para la optimización.

La tercera evolución se caracteriza por la definición de las etapas y actividades necesarias para el desarrollo de los cálculos. El diseñador puede elegir entre controlar directamente los cálculos en mayor o menor grado, es decir, existe la posibilidad de que la etapa numérica se realice en forma automática de principio a fin con mínima intervención del usuario, o que el diseñador guíe activamente los cálculos paso a paso, seleccionando variables y criterios de convergencia en cada etapa, por ejemplo.

En la Figura 2.9 se muestra un diagrama que muestra tres ejes que guían la evolución de los cálculos en el diseño y simulación de procesos. El enfoque evolutivo implica que el diseñador se desenvuelve en el espacio constituido por estos tres ejes, a partir del origen.

El diseñador de procesos continuamente se mueve a lo largo de cada eje, buscando un adecuado equilibrio entre el tipo de modelo que representa adecuadamente su problema de acuerdo a la complejidad de los cálculos que desea realizar y el menor o mayor control que desea sobre ellos. Esta situación será continuamente expuesta durante el desarrollo de este trabajo.

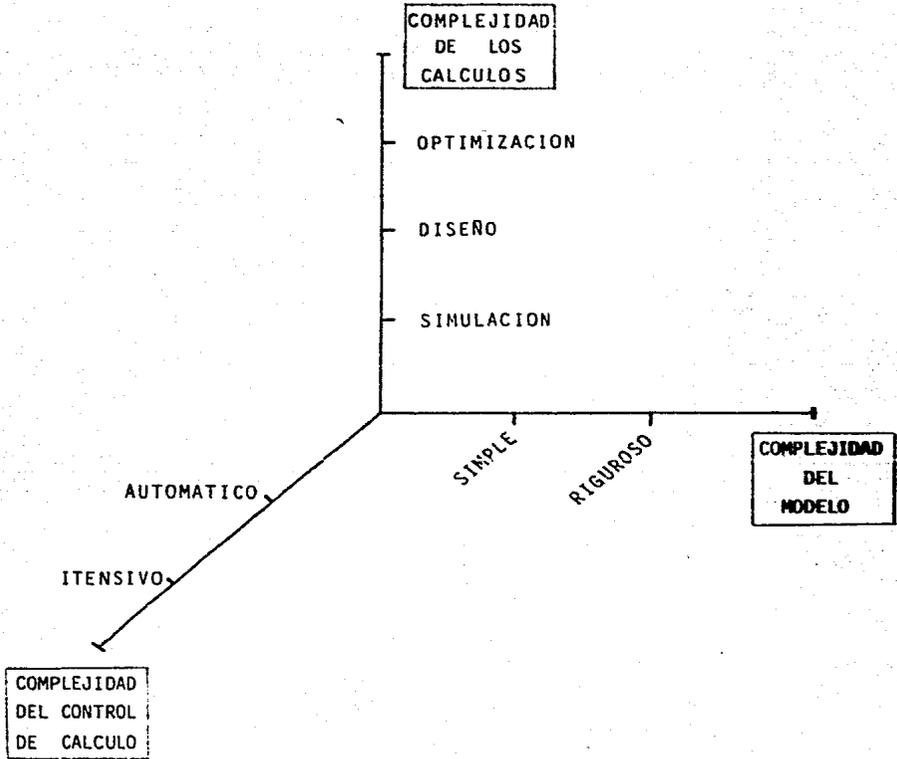


FIGURA 2.9.

ESPACIO DE EVOLUCION DE LOS  
CALCULOS EN EL DISEÑO.

## CAPITULO 3

### LA SIMULACION DE PROCESOS QUIMICOS

### CAPITULO 3. LA SIMULACION DE PROCESOS QUIMICOS.

La Simulación de Procesos es la actividad que permite la representación del comportamiento de un proceso mediante la aplicación de modelos adecuados, bajo ciertas condiciones y alcances, previamente definidos.

Fundamentalmente existen dos grandes vertientes dentro de la Simulación de Procesos: Simulación en Estado Estacionario y Simulación Dinámica.

La Simulación en Estado Estacionario está dirigida, primordialmente, al estudio del comportamiento de un proceso a régimen permanente, contemplando situaciones dentro de los límites considerados como normales en su diseño y operación. Su tarea principal es el cálculo detallado del balance de masa y energía del proceso, en base al cual se realizará el dimensionamiento del equipo y una posterior evaluación económica.

La Simulación Dinámica representa el comportamiento transiente de un proceso, considerando situaciones de arranque y paro operación bajo condiciones de emergencia, operación fuera de diseño, evaluación de la influencia de las fluctuaciones en las

corrientes de alimentación, etc., además de ser ampliamente utilizada como auxiliar en el entrenamiento del personal técnico.

Dada la complejidad de la mayoría de los procesos químicos, se hace imperativo que la simulación sea realizada con la ayuda de una computadora. Actualmente se ha superado la etapa en la que era posible emplear programas aislados o poco flexibles, para dar paso al uso de grandes sistemas computacionales, capaces de manejar un proceso en forma total, conocidos como Sistemas de Simulación de Procesos (Process Flowsheetings).

### 3.1. SISTEMAS DE SIMULACION DE PROCESOS EN ESTADO ESTACIONARIO.

Un Sistema de Simulación es un programa de computadora que, utilizando información sobre un proceso químico a un nivel de detalle de diagrama de flujo, es capaz de realizar análisis útiles en su diseño y operación. Los tipos de análisis útiles se refieren, basicamente, al balance de materia y energía, al dimensionamiento del equipo, a la estimación de costos y a la evaluación económica; aunque existen otros tipos de análisis que podrían resultar importantes en el futuro, tales como los referentes a la eficiencia energética, a la seguridad industrial,

al cálculo de la secuencia de operaciones, etc. La Figura 3.1 muestra el flujo de información dentro de un Sistema de Simulación orientado a realizar los análisis útiles.

La gran ventaja de emplear Sistemas de Simulación en el Diseño de Procesos es, ante todo, el considerable ahorro de tiempo y recursos de ingeniería, es decir, se pueden considerar mayores alternativas de diseño y el proceso creado puede ser más complejo amplio y eficiente, en términos de costo de operación, consumo de energía y materias primas, calidad del producto, etc.

Un Sistema de Simulación en Estado Estacionario, principalmente aborda la solución de dos tipos de problemas:

1.- Problema de Simulación.- En este caso, las variables asociadas a las corrientes de alimentación y los parámetros que definen el funcionamiento de cierto equipo son especificados y los perfiles de las variables dependientes y/o la información de las corrientes de salida son calculadas. Esencialmente se dirige a la evaluación del comportamiento de un proceso de diseño ya conocido.

2.- Problema de Simulación Controlada o Diseño.- Ahora, algunas variables de las corrientes de alimentación y/o algunos parámetros de equipo deberán ser ajustados con el objeto de satisfacer las especificaciones de diseño, por tanto, el número de

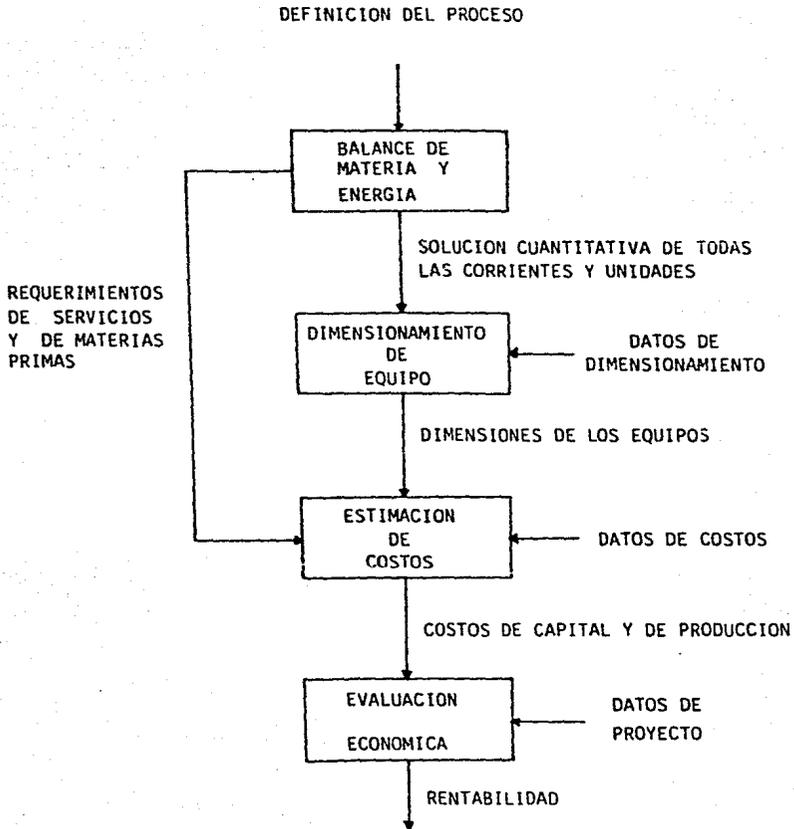


FIGURA 3.1. FLUJO DE INFORMACION EN UN SISTEMA DE SIMULACION.

variables a ser ajustadas será igual al número de especificaciones de diseño impuestas. En este caso, se obtendrá el diseño de un proceso bajo un cierto comportamiento deseable.

Para cumplir estas tareas, un Sistema de Simulación debe contar con ciertos elementos esenciales, como son: modelos, algoritmos, soporte computacional (soporte lógico ("software") del sistema y soporte físico ("hardware") de la computadora) y la interface con el usuario. En la Figura 3.2 se muestran estos elementos en una estructura de bloques.

Los modelos son el fundamento del sistema y conforman la base del análisis. Los modelos de un proceso químico usados por un Sistema de Simulación son relaciones matemáticas derivadas de las leyes de conservación, correlaciones de propiedades termodinámicas, relaciones de conexión y restricciones de diseño y control. Los modelos adquieren la forma de ecuaciones algebraicas y ecuaciones diferenciales, todas ellas descriptivas de los fenómenos que ocurren en el proceso. El más importante requerimiento del modelo es que sea apropiado a su uso en términos de rigor, descripción, nivel de detalle, precisión, validez y generalidad. Lo anterior implica que lo realmente importante en un modelo es su estructura matemática y su capacidad descriptiva, más que su carácter lineal o no lineal, puesto que la distinción cualitativa entre estos tipos de modelos aún es aplicable, pero ha perdido su importancia original, siendo la

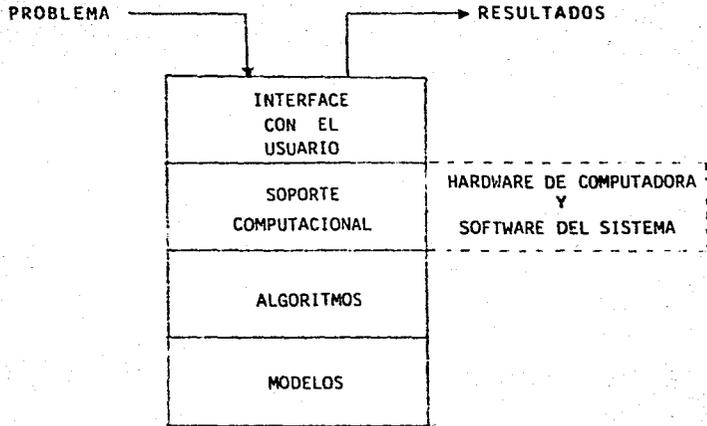


FIGURA 3.2. ESTRUCTURA DE BLOQUES DE UN SISTEMA DE SIMULACION DE PROCESOS.

diferencia en los tiempos de cómputo el criterio más relevante.

El núcleo de un Sistema de Simulación esta constituido por los modelos de las operaciones unitarias, cuya estructura general es mostrada en la Figura 3.3, y en la cual se puede observar el tipo y el flujo de la información requerida: las variables de las corrientes de entrada y los parámetros del modelo (variables requeridas para describir el funcionamiento de esa operación unitaria), variables de las corrientes de salida y las variables resultantes (tales como el requerimiento de potencia de una bomba, carga térmica en un rehervidor, etc.), y variables internas y de retención (composiciones, temperaturas por etapa en una columna de destilación, etc.)

Los algoritmos operan sobre los modelos para producir los resultados deseados. El tipo de problema matemático a ser resuelto es variable, pero siempre incluye la solución de ecuaciones algebraicas (lineales y no lineales), ecuaciones diferenciales y programación no lineal. Los algoritmos deberán ser robustos, tan generales como sea posible, eficientes en términos de ejecución y requerimientos computacionales, limpios y elegantes. Es importante hacer notar que algunos de estos requisitos pueden contraponerse, por lo tanto, algunos criterios de tipo operativo deben considerarse con el objeto de que los algoritmos exploten al máximo los recursos disponibles.

El soporte computacional incluye todos los recursos neces-

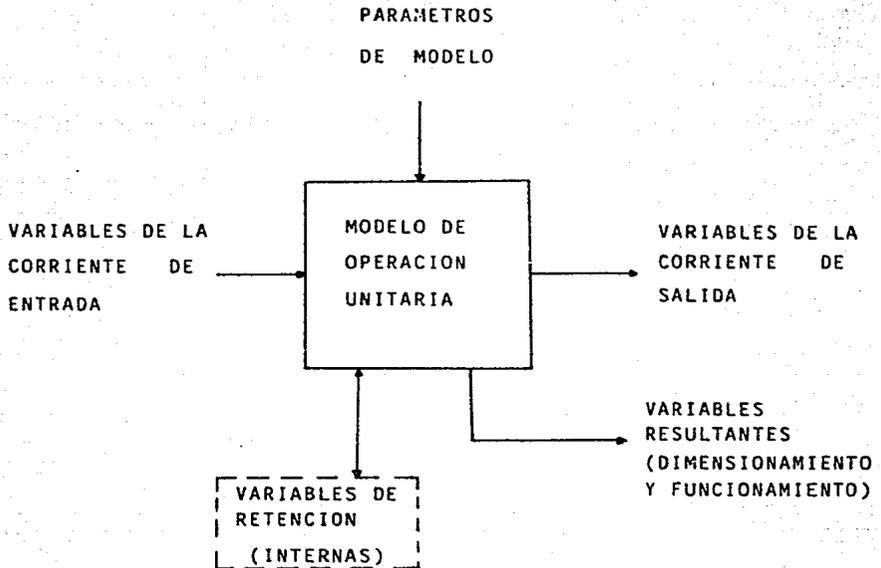


FIGURA 3.3. ESTRUCTURA TÍPICA DE UN MÓDULO UNITARIO.

rios para implementar los algoritmos en una computadora, bajo cierto sistema operativo. Se incluye dentro de esta categoría: programas, lenguajes de programación, codificación, estructura de datos, interface con el sistema de archivo, documentación y arquitectura del sistema. La bondad de un soporte computacional descansa en su facilidad para ser interpretado y en la sencillez de su operación, mantenimiento y adaptación.

La interface con el usuario incluye el protocolo de acceso al sistema, lenguaje de entrada para que el usuario describa su problema, los reportes que contienen los resultados, documentación para el uso del sistema y los protocolos para la interacción con otros programas y sistemas. Los requerimientos básicos para una interface adecuada exigen que acepte la entrada en una forma tan natural como sea posible y que los resultados sean presentados en una forma práctica y útil para el usuario.

### 3.2. ORIENTACION Y ENFOQUE DE LOS SISTEMAS DE SIMULACION.

El problema matemático más importante a ser abordado en la simulación de procesos en estado estacionario es la solución de grandes sistemas de ecuaciones. El punto de partida de la

simulación es el diagrama de bloques de los modelos de todas las operaciones unitarias que componen un proceso en donde se especifican todas las ecuaciones que deberán ser resueltas.

Prácticamente, cualquiera que sea la orientación adoptada, el número y tipo de ecuaciones que describen el proceso, esencialmente son los mismos; la diferencia reside en la forma de manejar y resolver los sistemas que estas ecuaciones conforman.

En aplicaciones típicas, el número de ecuaciones puede variar entre algunos cientos, hasta más de cien mil. Ante estos enormes sistemas de ecuaciones, es necesario explotar su particular estructura, con el objeto de que su solución se realice en una forma más eficiente.

Las dimensiones de estos sistemas de ecuaciones son mucho mayores de los que las técnicas convencionales en la solución de matrices completas pueden abordar con eficiencia, por lo tanto esto ha dado impulso al desarrollo de técnicas de solución de matrices dispersas (lineales y no lineales), que tienen la capacidad de manejar sistemas de orden de magnitud mucho mayor, como se muestra en los Apéndices A y B.

Es así como las diferentes formas de explotar la estructura del sistema, conduce a diferentes orientaciones y enfoques para resolver los problemas de simulación y diseño de procesos, dentro

de los límites impuestos por aspectos computacionales, como lo son, el tiempo de procesamiento, capacidad de almacenamiento en el núcleo central de memoria, etc.

Existen dos orientaciones principales:

1) Orientación Modular, que contiene dos enfoques:

- Modular Secuencial
- Modular Simultáneo

2) Orientación Basada en las Ecuaciones, con dos enfoques:

- Con rompimiento
- Con linearización simultánea

### 3.2.1. ORIENTACION MODULAR

En la Orientación Modular un proceso químico es representado por la unión de un conjunto de módulos, cada uno describiendo un cierto tipo de operación unitaria, de aquí que también se les designe como Módulos Unitarios. Los cálculos involucrados en la solución de los sistemas de ecuaciones no lineales que descri-

ben el proceso pueden ser comprendidos tanto a nivel de diagrama de flujo (proceso completo), como a nivel de cada módulo (operaciones unitarias).

En los cálculos a nivel de cada módulo, las ecuaciones que lo describen son resueltas con el objeto de determinar los valores de las variables en las corrientes de salida, a partir de los valores conocidos de las variables de las corrientes de entrada y los parámetros de equipo.

En los cálculos a nivel de diagrama de flujo, las ecuaciones que describen la conexión entre cada módulo y las especificaciones de diseño, son resueltas.

En general, las principales ventajas de la Orientación Modular son:

- En virtud de la gran semejanza entre el diagrama de flujo del proceso y del modelo implementado para la computadora, esta orientación es fácilmente visualizada por el ingeniero.

- Uno o más algoritmos especializados pueden ser usados para resolver las ecuaciones que describen cada módulo. Como resultado de lo anterior, los cálculos en cada módulo son eficientes y robustos. Además, cada módulo puede generar sus propios valores iniciales, necesarios para comenzar a resolver los sistemas de ecuaciones que los describen.

- El requerimiento de que cada módulo calcule los valores de las corrientes de salida, dados los valores de las corrientes de entrada y los parámetros de equipo, produce un flujo de información muy simple a nivel de cada módulo. Ante esto, los programas ejecutivos para estos sistemas son escritos con relativa facilidad.

- Los programas aislados para una cierta operación unitaria pueden ser convertidos eficientemente en módulos unitarios, para que puedan ser incorporados al sistema, lo que confiere una gran versatilidad.

- Existe la posibilidad de extraer información útil acerca de algunas secciones del proceso, aún cuando la simulación total llegase a fallar. Por otra parte, este arreglo permite la localización, diagnóstico y solución de fallas en el sistema, en una forma muy simple.

Sin embargo, existen dos grandes desventajas de la Orientación Modular.

- La lenta convergencia, a causa del excesivo manejo de las grandes matrices del proceso y de los múltiples ciclos iterativos producto de las numerosas recirculaciones que los complejos procesos actuales poseen.

- Por la razón anterior, se presentan algunos problemas en la transferencia de información entre un módulo y otro.

### 3.2.1.1. Enfoque Modular Secuencial.

En este enfoque, el programa ejecutivo llama a los módulos unitarios (descritos en forma de subrutinas), en la misma secuencia en que son presentados en el diagrama de flujo y son resueltos uno a uno, por tanto, el esquema de cálculo esta rigidamente fijado por la topología del proceso.

La solución de la mayoría de los módulos es realizada considerando los como un problema de simulación, es decir, dados los valores de las corrientes de entrada y los parámetros de equipo, las ecuaciones del modelo son resueltas para obtener los valores de las variables internas y de las variables de salida. Sin embargo, cuando una corriente de recirculación (o sea, una corriente de entrada al módulo no especificada) es encontrada, las variables de recirculación son supuestas y se inicia un ciclo iterativo, empleando algoritmos de tipo Newton o de sustituciones sucesivas, con o sin un esquema de aceleración, hasta lograr la convergencia de las corrientes de recirculación con las demás corrientes del módulo unitario. Debe tomarse en cuenta que los valores iniciales apropiados y las especificaciones (o restricciones) de las variables de recirculación deben ser proporcionados, ya sea por el usuario del sistema o mediante la aplicación de algunas reglas heurísticas.

Existe un extenso trabajo en el área de modelado de unidades

individuales de proceso, lo que ha proporcionado módulos de cálculo muy eficientes y precisos para la mayoría de las operaciones unitarias existentes. Además, la incorporación de nuevos módulos o versiones más complejas de los ya existentes, puede realizarse fácilmente, sin cambiar la estrategia de solución global.

Las ecuaciones de conexión entre módulos son manejadas en forma implícita; el programa ejecutivo transfiere los valores de salida de un módulo unitario a otros como si fueran valores de entrada, de acuerdo a la topología del proceso previamente especificada. Las especificaciones simples, tales como los datos de las corrientes de entrada y los parámetros de equipo, son fácilmente manejados, transfiriendo estos valores directamente al módulo que así lo requiera. Sin embargo, otras especificaciones, tales como las de diseño y otras que afecten directamente a las variables de las corrientes de salida, no pueden ser introducidas directamente y requieren el uso de ciclos de control.

Los problemas de diseño deben ser resueltos mediante una simulación iterativa (también conocida como simulación controlada) en la cual el proceso es simulado en una forma repetitiva hasta que las especificaciones de diseño son satisfechas, conformando lo que se conoce como ciclos iterativos de control.

A pesar de la eficiencia de los cálculos a nivel de módulo unitario, el esquema de cálculo a nivel de proceso (convergencia

de corrientes de conexión y corrientes de corte), puede ser muy ineficiente debido a la presencia de múltiples recirculaciones. Tomando como ejemplo el proceso con una recirculación simple de la Figura 3.4, se observa que el primer módulo, el mezclador, en la secuencia, será ejecutado sólo cuando se conozcan los valores de las variables de la corriente de recirculación,  $S_R$ . La forma en que un simulador modular secuencial maneja este problema es suponiendo valores para las corrientes de recirculación, que se convierten en corrientes de corte, e iterando sobre estos valores hasta obtener la convergencia entre los valores supuestos y los valores calculados, dentro de cierta tolerancia. Cada iteración sobre los valores supuestos implica la ejecución secuencial de todos los módulos unitarios que anteceden al módulo que tiene la corriente supuesta como salida, en este caso es el módulo del separador, y los valores actualizados para las variables son obtenidos al ejecutar ese módulo. En base a la diferencia entre los valores supuestos y los valores calculados (actualizados), se obtiene un valor mejorado empleando algún método adecuado de convergencia. Cuando existen múltiples corrientes de corte, cada una se hace converger en forma separada, dentro de un ciclo de convergencia, tal y como se explica en el siguiente capítulo.

El problema se complica por el hecho de que las especificaciones de diseño son tratadas como ciclos de recirculación adicionales. Al imponer una restricción de diseño al ejemplo

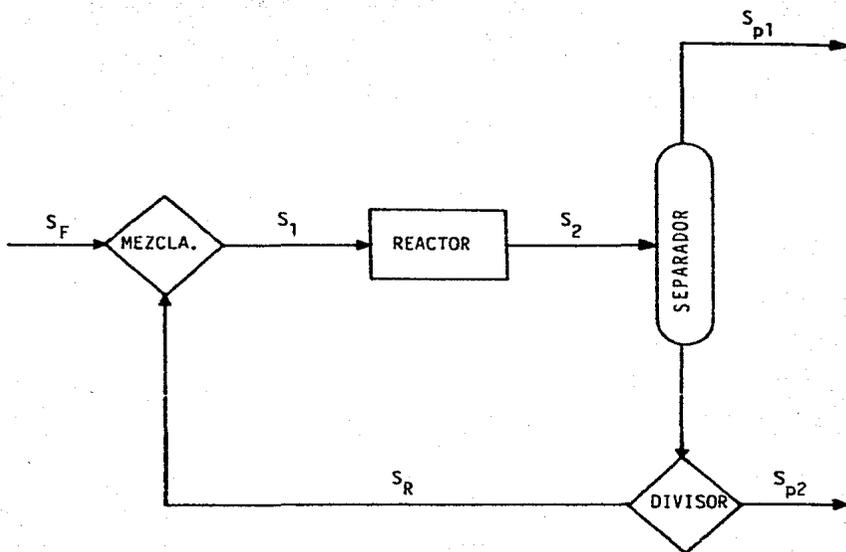


FIGURA 3.4.

EJEMPLO DE UN PROCESO CON RECIRCULACION  
SIMPLE.

de la Figura 3.4. se debe proporcionar un estimado inicial para la variable de decisión a fin de poder ejecutar los módulos dentro del ciclo de recirculación de diseño, por lo tanto, según se muestra en la Figura 3.5, ahora existirán dos ciclos iterativos, uno para la solución de los módulos y otro para las especificaciones de diseño.

Existen dos problemas que afectan la eficiencia de este enfoque: el manejo de las especificaciones de diseño y la presencia de múltiples ciclos anidados de cálculo. El primero ya ha sido abordado anteriormente. El segundo tiene que ver con la estructura jerárquica de iteración con los ciclos anidados. Los ciclos de control, con los que se logra la convergencia de las especificaciones de diseño, son los que se encuentran en la parte más externa. Inmediatamente dentro de los ciclos de control, se encuentran los ciclos necesarios para lograr la convergencia de las corrientes de recirculación (corrientes de corte). Dentro de los ciclos de recirculación están los ciclos propios de cada operación unitaria y dentro de éstos se encuentran múltiples ciclos destinados a la estimación de propiedades termofísicas, según se muestra en la Figura 3.6.

Existe aún un tercer problema, que tiene que ver con las actuales perspectivas que existen en la simulación de procesos, y se refiere a implementar la optimización del proceso como otra tarea del simulador. Al emplear el enfoque Modular Secuen-

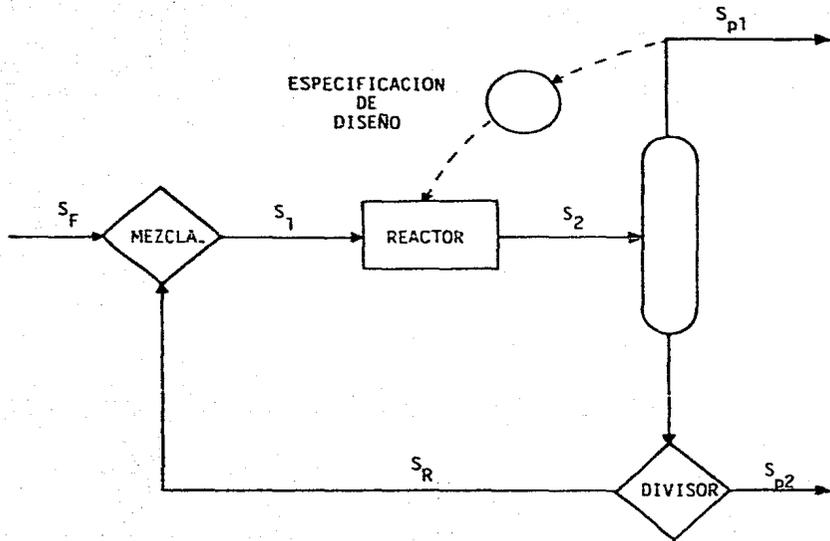


FIGURA 3.5. EJEMPLO DE UN PROCESO CON RECIRCULACION SIMPLE Y ESPECIFICACION DE DISEÑO.

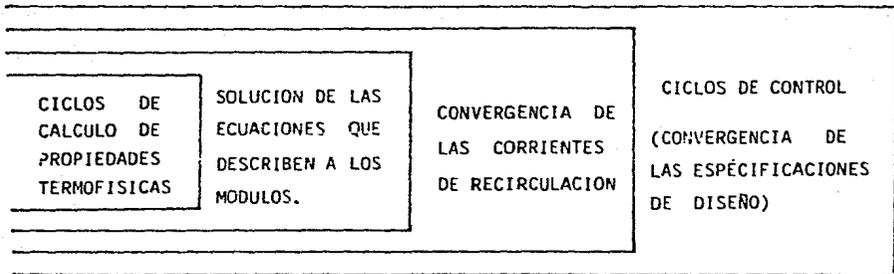


FIGURA 3.6. CICLOS MULTIPLES DE ITERACION ANIDADOS DEL ENFOQUE MODULAR SECUENCIAL.

cial, la optimización se implementa agregando otro ciclo iterativo externo, lo que acrecenta el problema de eficiencia numérica anteriormente señalado, sin olvidar el alto requerimiento computacional que se asocia con la Optimización.

Para problemas de simulación relativamente simples, con corrientes de recirculación aisladas, los ciclos de iteración anidados no representan un problema serio. Sin embargo, en virtud de los esfuerzos actuales para reutilizar las materias primas y para integrar la energía, los procesos químicos modernos presentan una estructura muy compleja de corrientes de recirculación lo que los ha convertido en prohibitivos, en cuanto a su simulación, diseño y optimización bajo el enfoque Modular Secuencial.

A pesar de las anteriores desventajas, los simuladores modulares secuenciales actualmente son los más utilizados para aplicaciones comerciales, y puesto que existen demasiados recursos invertidos en su desarrollo, al menos a corto plazo, no se prevee abandonar su uso, a pesar de sus cada vez más grandes limitaciones.

Ante estos hechos, se pretende perfeccionar este tipo de simuladores, mediante la incorporación de técnicas más eficientes en la solución de los sistemas de ecuaciones y de estrategias adecuadas para el manejo de las especificaciones de diseño.

### 3.2.1.2. Enfoque Modular Simultáneo.

Ante todo, se debe resaltar que el uso del término Modular Simultáneo para describir el enfoque al que se refiere esta sección, trata de cubrir aquellos trabajos que han sido descritos con términos como el de Doble Rompimiento ("two tier", Rosen (1962)), o simplemente como un concepto diferente de convergencia para el enfoque Modular Secuencial (Perkins (1979); Pierucci, Ranzi y Biardi (1982)).

El término Modular Simultáneo actualmente es de uso generalizado y razonablemente descriptivo del tipo de estrategia de cálculo usada; sin embargo, más adelante, cuando se describa el enfoque orientado hacia las ecuaciones con linearización simultánea, se hará una importante aclaración al respecto, puesto que existe la tendencia de conjuntar el enfoque Modular Simultáneo y a los enfoques Basados en las Ecuaciones bajo el título de enfoques con Orientación No Secuencial.

Bajo el enfoque Modular Simultáneo, la estructura modular es conservada, pero las ecuaciones de las corrientes de conexión y las especificaciones de diseño se hacen converger simultáneamente. La idea fundamental de este enfoque es mostrada en la Figura 3.7, en donde se destaca el empleo de dos tipos de modelos: simples y rigurosos.

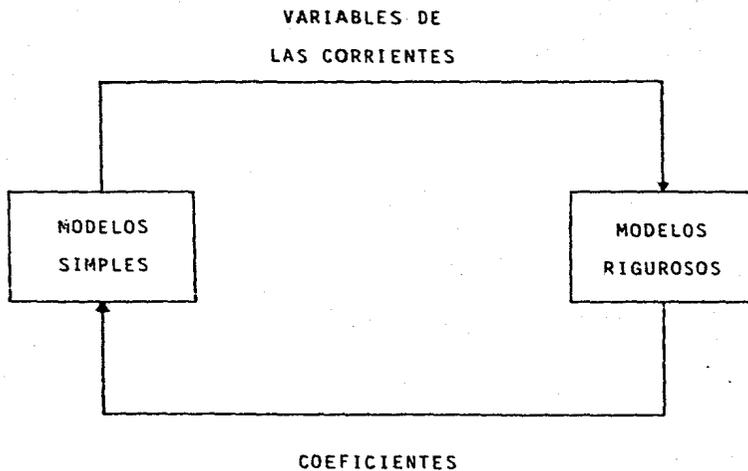


FIGURA 3.7. CONCEPTO BASICO DEL ENFOQUE MODULAR SIMULTANEO.

Los modelos rigurosos son los mismos modelos de operación unitaria empleados en los simuladores con un enfoque Modular Secuencial. Estos modelos son usados para determinar parámetros que posteriormente serán usados por los modelos simples.

Los modelos simples son modelos aproximados que requieren una cantidad mucho menor de recursos computacionales para su solución. Una vez resuelto el modelo simple se obtienen valores aproximados de todas las variables de las corrientes, con los que se verifica la convergencia; de no obtenerse, existe la posibilidad de modificar los valores de los parámetros de los modelos aproximados, o bien, llamar nuevamente a los modelos rigurosos y obtener valores más precisos. Este procedimiento continúa hasta que los cambios en los parámetros del modelo simple son lo suficientemente pequeños, para dos iteraciones sucesivas, logrando la convergencia de las variables de proceso. Este procedimiento es ilustrado en forma esquemática en la Figura 3.8, en la cual se muestra una estructura en forma de ciclo interno y ciclo externo.

Los modelos simples pueden ser de varios tipos. Los hay lineales, cuyos coeficientes pueden ser determinados mediante la perturbación numérica de los modelos rigurosos, y los hay no lineales, usualmente representados por modelos aproximados de ingeniería. Cualquiera que sea el tipo, el modelo simple genera un sistema de ecuaciones de simulación equivalente al del modelo

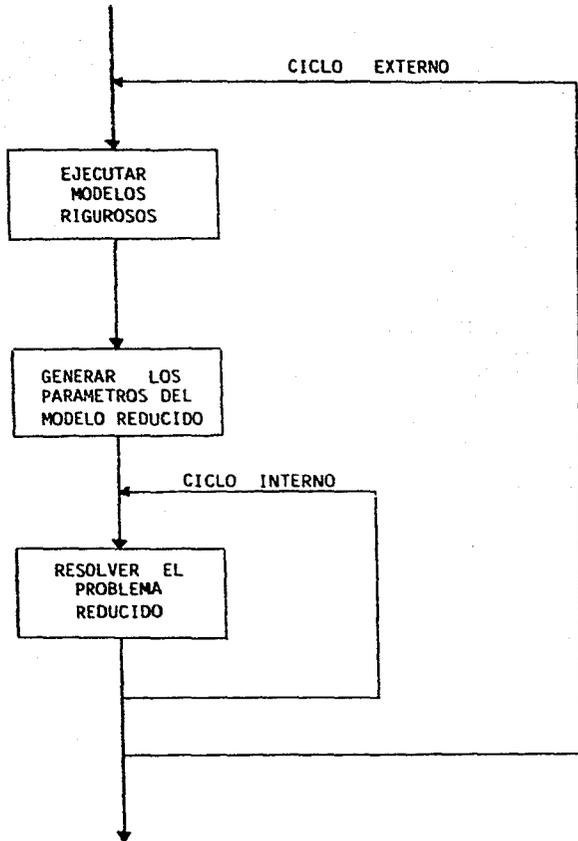


FIGURA 3.8. EL ENFOQUE MODULAR SIMULTANEO EN FORMA DE CICLO INTERNO Y CICLO EXTERNO.

riguroso, pero de menor extensión, conteniendo un número mucho menor de variables internas e incluyendo correlaciones simples para propiedades termofísicas, originando lo que se conoce como problema reducido.

Los modelos simples, en virtud del reducido número y tamaño de sus ecuaciones, comparado con las dimensiones de los modelos rigurosos, puede ser resuelto en forma muy eficiente por las técnicas desarrolladas ampliamente como parte de los simuladores orientados hacia las ecuaciones.

De lo expuesto anteriormente, el enfoque Modular Simultáneo puede ser considerado como un intento de combinar algunas de las ventajas del enfoque Modular Secuencial y las ventajas propias de un eficiente sistema de solución de ecuaciones no lineales, propio del enfoque Basado en las Ecuaciones, el cual será descrito en la siguiente sección.

Las diversas técnicas empleadas en el enfoque Modular Simultáneo basicamente varían en la forma en que la matriz Jacobiana, requerida a nivel de proceso, es generada y resuelta, y en la elección de iterar sobre todas las corrientes de conexión o sólo sobre un conjunto seleccionado de ellas.

Puesto que las especificaciones de diseño pueden ser manipuladas directamente a nivel de proceso, no hay necesidad de usar

ciclos de control para hacerlas converger, eliminando de esta manera un ciclo iterativo, con respecto al enfoque Modular Secuencial.

El enfoque Modular Simultáneo posee la gran ventaja de poder utilizar la enorme cantidad de trabajo computacional existente bajo el enfoque Modular Secuencial, tanto de programas para operaciones unitarias, como de las reglas heurísticas que ha aplicado con éxito, lo que permite obtener útiles estimados iniciales.

Además, el enfoque Modular Simultáneo es de uso familiar para el ingeniero químico, puesto que éste usualmente emplea modelos simplificados para realizar análisis preeliminares y después procede a verificar el modelo simple en base a la aplicación de modelos rigurosos.

Sin embargo, la prueba decisiva para el enfoque Modular Simultáneo se presenta al resolver las ecuaciones a nivel de proceso total, ya que se requieren procedimientos para resolver ecuaciones con excelentes propiedades de convergencia. Un problema adicional es que el tiempo y recursos computacionales asociados al cálculo del Jacobiano aproximado pueden ser muy grandes y por lo tanto se pierde la ganancia obtenida con la rápida convergencia de este enfoque. Es por eso que se pretende el empleo de métodos de aproximación muy simples, los cuales, sin embargo, pueden producir un estimado muy pobre del Jacobiano, lo

que puede acrecentar el problema de falta de precisión, exigiendo la realización de un mayor número de iteraciones.

Es un hecho que estas desventajas han evitado una amplia aceptación de este enfoque, sin embargo, con el enorme desarrollo que han tenido las técnicas de manejo y solución de sistemas de ecuaciones, esta opción se presenta con mayor atractivo, sobre todo al reconocer las limitaciones inherentes al enfoque Modular Secuencial y al poco desarrollo que, a la fecha, tienen los simuladores con una Orientación hacia las Ecuaciones.

### 3.2.2. ORIENTACION HACIA LAS ECUACIONES.

Bajo esta orientación un proceso químico es modelado mediante la reunión de todas las ecuaciones que lo describen (operaciones unitarias, corrientes de conexión y especificaciones de diseño), y son resueltas simultáneamente como un gran sistema de ecuaciones no lineales, según se muestra en la Figura 3.9.

A diferencia de la Orientación Modular, en donde existen módulos que calculan las variables de salida como una función de

SOLUCION DE LAS ECUACIONES  
QUE DESCRIBEN A LAS UNIDADES,  
ECUACIONES DE CONEXION,  
ESPECIFICACIONES DE DISEÑO,  
Y CORRELACIONES PARA LAS  
PROPIEDADES TERMOFISICAS PARA  
TODO EL PROCESO, EN FORMA  
SIMULTANEA.

FIGURA 3.9. ESQUEMA DE CALCULO PARA LOS ENFOQUES  
ORIENTADOS A LAS ECUACIONES.

las variables de entrada, los simuladores con Orientación hacia las Ecuaciones requieren de procedimientos que generen y representen las ecuaciones que describen el proceso por completo.

Matemáticamente el problema puede ser planteado como la solución de

$$F(Z_1, Z_2) = 0$$

en donde:

$Z_1$  = vector de variables de estado (dependientes)

$Z_2$  = vector de variables de decisión (independientes)

y las variables de decisión usualmente incluyen todos los parámetros de equipo y las variables de las corrientes de entrada para cada operación unitaria y las variables de estado incluyen todas las variables internas y de las corrientes de salida.

Alternativamente, en un simulador orientado hacia las ecuaciones, el problema de simulación y diseño puede ser formulado como un problema de optimización:

Maximizar o minimizar  $P(Z_1, Z_2)$

con  $F(Z_1, Z_2) = 0$

agregando, además, restricciones adicionales impuestas por el planteamiento de la Optimización. La restricción de igualdad

$F(Z_1, Z_2) = 0$  es el mismo sistema de ecuaciones que se ha manejado anteriormente, pero en lugar de especificar arbitrariamente todas las variables de decisión, éstas son seleccionadas a fin de maximizar o minimizar alguna función objetivo  $P(Z_1, Z_2)$ .

Según Westerberg (1980), el problema de diseño es formulado en una forma más natural como un problema de optimización. Sin embargo, una gran limitación para formular problemas de simulación y diseño como problemas de optimización, ha sido la falta de algoritmos apropiados de programación no lineal que pudieran manejar con eficiencia grandes cantidades de restricciones de igualdad.

Comparada con la Orientación Modular, la Orientación hacia las Ecuaciones es más flexible en cuanto a que especificaciones diferentes generarán diferentes sistemas de ecuaciones no lineales y éstos serán potencialmente más susceptibles de ser resueltos eficientemente, puesto que evitan los serios problemas causados por los ciclos iterativos anidados.

En cuanto al problema de flujo de información, propio de la Orientación Modular, en la Orientación hacia las Ecuaciones es prácticamente inexistente, puesto que al no haber módulos, toda la información es manejada exclusivamente a nivel de proceso en forma de ecuaciones y variables.

Los métodos orientados hacia las ecuaciones han sido usados ampliamente en modelos de operaciones unitarias individuales y en la solución de problemas de simulación y diseño muy especializados, en los cuales son incluidos conjuntos homogéneos de equipos, tales como columnas de destilación, intercambiadores de calor, etc.

A pesar de que los métodos orientados a las ecuaciones prometen una rápida convergencia y una formulación más natural del problema de diseño como un problema de optimización, aún no han sido aplicados en forma eficiente en los simuladores comerciales por presentar varias desventajas. Primero, esta orientación es muy rígida en cuanto a la elección a-priori del sistema de ecuaciones que se deben resolver, siendo muy frecuente la aparición de inconsistencias termodinámicas, lo que impide la solución del sistema elegido, o bien, conducir a soluciones físicamente inexistentes. Segundo, para su operación eficiente requiere de valores iniciales razonables, los cuales son prácticamente imposibles de obtener sin realizar un recorrido secuencial a través del proceso. Por ejemplo, un problema común es la dificultad de conocer a-priori la condición de equilibrio en una corriente de proceso, o sea, predecir la existencia de una sola fase o de un equilibrio líquido-vapor. El recorrido secuencial de las unidades dentro del ciclo que incluye a la corriente en cuestión, permite conocer la situación en base a los valores locales de las variables de estado. Tal posibilidad es pro-

hibitiva en un método orientado hacia las ecuaciones, en virtud del esquema fijo de ecuaciones que debe resolver simultáneamente.

Tercero, estos métodos no obtienen provecho del gran avance existente, a nivel industrial y académico, en el desarrollo de eficientes modelos de operaciones unitarias basados en la Orientación Modular, en especial bajo el enfoque Modular Secuencial.

Cuarto, al existir un error en la operación, no se pueden extraer resultados parciales y es muy complicado diagnosticar la falla y corregirla.

Perkins (1983), presenta una revisión muy completa de los Sistemas de Simulación Orientados hacia las Ecuaciones existentes a la fecha a nivel comercial y académico, haciendo resaltar sus ventajas y puntos débiles, tratando de identificar las principales afeas para un desarrollo futuro.

#### 3.2.2.1. Enfoque Basado en las Ecuaciones con Rompimiento.

Bajo este enfoque, son supuestos los valores iniciales para un número suficiente de variables como para permitir el cálculo directo de las variables restantes. Las ecuaciones pueden

ser resueltas para los nuevos valores de las variables supuestas, en base a ciertos procedimientos de sustituciones sucesivas, tomando en cuenta que en las ecuaciones que contengan las variables supuestas, éstas estén en forma explícita. Si éste no fuera el caso, los residuales en las ecuaciones que contienen las variables supuestas (llamadas "ecuaciones supuestas", aunque en realidad sean ecuaciones propias del proceso), pueden ser usadas para aplicar algún otro procedimiento de evaluación de raíces.

De cualquier modo, las ventajas que promete este enfoque es que un sistema de tipo ecuación-variable extremadamente grande es subsecuentemente reducido a un subsistema mucho menor del tipo ecuación supuesta-variable supuesta y que es más adecuado para ser resuelto simultáneamente.

El punto clave en este enfoque es elegir una estrategia de solución adecuada, mediante modelos eficientes en cuanto al flujo de información, que permitan a las variables supuestas converger rápidamente hacia la solución. Lo anterior significa que el usuario debe decidir a que variables se les asignará un valor supuesto, que ecuaciones deberán ser resueltas con que variables (esquema de solución) y en que secuencia se deberán resolver (orden de precedencia). Aún más, puesto que la mayoría de los sistemas que describen un proceso químico están bajo restricciones, es necesario designar ciertas variables como variables de diseño.

Algunas estrategias proponen encontrar el número mínimo de variables supuestas, necesario a fin de reducir al mínimo el tamaño del subsistema y por lo tanto, también reducir el número de variables que requieren iteración.

Otras estrategias involucran consideraciones de sensibilidad. El problema de sensibilidad implica que, una premisa básica del enfoque con rompimiento, la que indica que un problema de baja dimensión es más fácil de resolver que uno de alta dimensión, no necesariamente es correcta. Lin y Mah (1978), han enfatizado que debido a las largas cadenas de cálculo típicamente existentes entre los valores supuestos y los residuales en las ecuaciones supuestas, puede presentarse un problema de sensibilidad que puede causar algunas divergencias, aún para valores iniciales muy cercanos a la solución.

Comparado con la Orientación Modular, el Enfoque Basado en las Ecuaciones con Rompimiento, es típicamente más rápido y capaz de abordar problemas de simulación y diseño más complejos, sin embargo, a causa de los problemas de eficiencia encontrados, una estrategia confiable de solución aún no ha sido formulada, por lo que todavía existe cierta desconfianza en su aplicación, fuera de los círculos académicos.

El trabajo de Stadtherr y Wood (1982), ha indicado que este enfoque puede reducir el número de ecuaciones que deben ser re-

sueltas simultáneamente, en un factor de cuatro. Esto significa que, para aplicaciones típicas, el sistema reducido de ecuaciones no lineales es aún muy grande para ser resuelto simultáneamente, sin el auxilio de técnicas muy especializadas en el manejo de matrices dispersas.

### 3.2.2.2. Enfoque Basado en las Ecuaciones con Linearización Simultánea.

Este enfoque es conocido también como Enfoque Quasi-lineal, e implica la linearización simultánea de todas las ecuaciones y la iteración sobre todas las variables, usando métodos del tipo Newton-Raphson, quasi-Newton o alguna combinación de ambos.

Este enfoque presenta varias ventajas importantes. Puesto que las variables son iteradas simultáneamente, no se tiene el problema de los ciclos iterativos anidados, inherente a la Orientación Modular, ni los problemas de sensibilidad y orden de precedencia del enfoque sobre las ecuaciones con rompimiento; además, las ecuaciones pueden ser resueltas en cualquier orden que se desee, lo que evita el problema de flujo de información de los

enfoques modulares. Adicionalmente, pueden emplearse técnicas de convergencia de primer orden, que son altamente precisas pero lentas, o bien, emplear métodos de segundo orden, que son más rápidos pero menos precisos.

En el enfoque con linearización simultánea de las ecuaciones se pueden adoptar las ventajas de los demás enfoques anteriormente descritos, con la posibilidad de implementar estructuras de recirculación muy complejas y resolver adecuadamente el problema de diseño.

No se duda del gran futuro que tiene este enfoque, sin embargo, actualmente aún tropieza con varios obstáculos que impiden su uso práctico. Ante todo, se presenta la dificultad de resolver el gran sistema de ecuaciones que genera, tomando en cuenta que los problemas prácticos de simulación y diseño exigen la solución de varias decenas de miles de ecuaciones, el requerimiento computacional que esto implica es muy alto; además, las técnicas conocidas para la solución de matrices dispersas aún no son capaces de resolver eficientemente sistemas de estas dimensiones, en forma simultánea.

Es muy frecuente que al emplear el enfoque con linearización simultánea se obtengan soluciones matemáticamente correctas, pero físicamente inexistentes, en virtud de que las ecuaciones descriptivas del proceso no esten correctamente formuladas y puesto

que este tipo de simuladores las manejan como parte de un gran sistema de ecuaciones, es muy difícil detectar el origen de la formulación errónea. De lo anteriormente descrito, se entiende la necesidad de que estos simuladores cuenten con un generador de ecuaciones de proceso realmente eficaz y confiable.

Finalmente, es de suma importancia hacer una aclaración que evite confusiones en la terminología.

El término Modular Simultáneo, puede prestarse a varias interpretaciones. Estrictamente, la denominación Modular Simultánea, comprende el enfoque descrito en la sección 3.2.1.2., y que es ampliado en el Capítulo 4 de este trabajo, referente a la estructura en la que se emplean tanto modelos rigurosos como modelos simples para hacer converger las variables de proceso y las especificaciones de diseño, en forma simultánea, hacia la solución del sistema.

Sin embargo, la confusión nace ante el hecho de que otras estructuras, tanto de optimización (en la cual, la simulación puede considerarse como una Solución Factible), como de la Orientación hacia las Ecuaciones (tanto del enfoque con partición, como del enfoque con linealización simultánea), han pro-

puesto retener el concepto modular unicamente con el objeto de generar las ecuaciones del proceso (más no para ser resueltas en los módulos), y posteriormente esas ecuaciones son reunidas y presentadas al sistema que las va a solucionar.

Erróneamente este último procedimiento podría ser considerado como Modular Simultáneo, en virtud de que las ecuaciones se generan en los módulos y todo el conjunto de ecuaciones descriptivas es resuelto simultáneamente, aunque afuera de ellos.

### 3.3. ESTRUCTURA DE LAS ECUACIONES DE SIMULACION.

Las ecuaciones descriptivas de un proceso químico sobre las cuales se realiza su simulación y diseño, en términos generales,

comprende estas categorías:

A) Ecuaciones de conservación.— Ecuaciones que describen la conservación de masa, energía y momento, por ejemplo: las ecuaciones estequiométricas de una reacción química, ecuaciones de conexión (al unirse dos o más flujos de masa y energía), etc.

B) Ecuaciones constitutivas.— Describen el funcionamiento de las unidades que están siendo simuladas, por ejemplo: la fracción de un flujo recirculado desde un punto en particular, la extensión de una reacción química, el funcionamiento de una columna de destilación, etc.

C) Ecuaciones restrictivas.— Describen las restricciones particulares impuestas sobre el sistema que está siendo modelado, por ejemplo: la composición o flujo de una determinada corriente; la pureza del producto, la conversión y la temperatura de un reactor, etc.

Estas ecuaciones están compuestas por uno o varios tipos de variables, pertenecientes a alguna de estas categorías:

A) Variables de entrada.— Asociadas con las corrientes de entrada al proceso en general y a cada equipo en particular, por

ejemplo: presión, temperatura, entalpía, composición, etc.

B) Parámetros de equipo.- Son variables de entrada que tienen la particularidad de estar asociadas únicamente con un determinado equipo y no con el total de las corrientes de proceso, por ejemplo: número de etapas en una columna de destilación, relación de reflujo, etc.

C) Variables de salida.- Asociadas con las corrientes de salida al proceso en general y a cada equipo en particular, por ejemplo: presión, temperatura, entalpía, composición, etc.

D) Variables internas.- Son variables de salida que son generadas como parte de los cálculos realizados a nivel de cada equipo y que se eliminan después de que el cálculo en la operación unitaria es concluido, por ejemplo: perfiles internos de una columna de destilación, cargas de calor en el rehedidor y condensador, etc.

E) Variables de retención.- Son parte de las variables generadas en los cálculos de una operación unitaria y que son reunidas en un vector para volver a utilizarse posteriormente.

En forma esquemática, el flujo de información para un módulo típico está representado en la Figura 3.10, en donde:

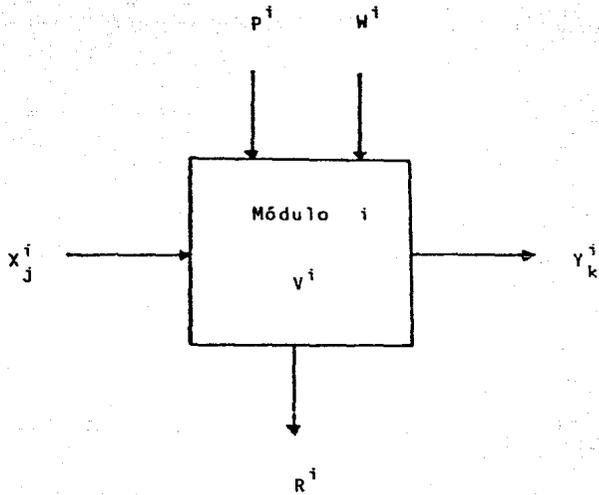


FIGURA 3.10. FLUJO DE INFORMACION PARA UN MODULO UNITARIO TÍPICO.

$X_j^i$  = vector para la  $j$ -ésima entrada.

$P^i$  = vector de parámetros de equipo fijos.

$W^i$  = vector de parámetros de equipo libres.

$V^i$  = vector de variables internas.

$R^i$  = vector de variables de retención.

$Y_k^i$  = vector para la  $k$ -ésima salida.

Estas variables se pueden agrupar así:

$$g(X, Y, V, P) = 0 \quad (3.3)$$

sistema que es conocido como ecuaciones descriptivas del proceso.

El número total de grados de libertad en este sistema de ecuaciones es el número de parámetros de equipo más el número de variables asociadas con las corrientes de alimentación al proceso representadas por  $X_f$  y que en realidad son un subconjunto de  $X$ . Los valores para estos parámetros y variables deben ser especificados anticipadamente con el objeto de resolver las ecuaciones descriptivas del proceso para las variables restantes.

Con frecuencia es deseable imponer ciertas restricciones sobre las variables del proceso, las cuales deberán ser resueltas junto con las demás ecuaciones del proceso. Estas ecuaciones se conocen como especificaciones de diseño y se agrupan así:

$$h(X, Y, V, P) = 0 \quad (3.4)$$

Para cada especificación de diseño, uno de los parámetros de equipo,  $P^i$ , o una variable de la corriente de alimentación,  $X_f$ , que de otra forma deberían ser especificados, ahora serán calculados junto con las demás variables del proceso, con el objeto de satisfacer los sistemas de ecuaciones 3.3 y 3.4. Estos parámetros de equipo se conocen como variables manipuladas o variables de decisión y se representan con  $W$ , por lo tanto

$$W(P, X_f) \quad (3.5)$$

Se deduce que el complemento de  $W$ , deberá ser especificado.

Es precisamente en este punto en donde el enfoque Modular Secuencial presenta una gran desventaja, puesto que sus programas ejecutivos y sus subrutinas son escritos en tal forma que un módulo unitario no puede generar resultados cuando en una corriente de entrada aparece un parámetro de equipo no especificado.

Por simplicidad, las ecuaciones descriptivas del proceso y las especificaciones de diseño (sistemas de ecuaciones 3.3 y 3.4, respectivamente), pueden ser representados por  $W$  en lugar de  $P$ , entendiéndose que todos los parámetros y variables en  $P$  y  $X$  no incluidos en  $W$ , tienen valores especificados, por lo tanto:

$$G(W, X, Y, V) = 0 \quad (3.6)$$

$$H(W, X, Y, V) = 0 \quad (3.7)$$

$$R(X_f, P) \geq 0 \quad (3.8)$$

y en donde la desigualdad  $R$  representa el vector de los límites de las variables de decisión. Estos valores límite definen la región de operabilidad del proceso, por lo tanto, para un problema de simulación bien definido, la solución deberá satisfacer estas especificaciones.

En el desarrollo de este trabajo, el término Simulación de Procesos será aplicado a la solución del Sistema General de Ecuaciones de Simulación, representado por los sistemas de ecuaciones 3.6, 3.7 y 3.8.

Para algunos procesos se tienen perfectamente determinados los valores para todas las variables en las corrientes de entrada  $y$ , por lo tanto, las variables en las corrientes de salida pueden ser calculadas al resolver el Sistema General de Ecuaciones de Simulación mediante un recorrido simple a través del proceso.

Sin embargo, en la gran mayoría de los procesos conocidos, algunas de sus corrientes son recirculaciones, es decir, corrientes que unen una salida con una entrada, corriente arriba en el proceso, tal y como es mostrado en la Figura 3.11.

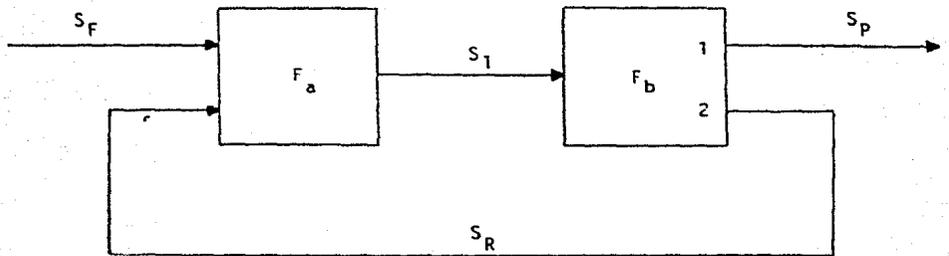


FIGURA 3.11. EJEMPLO DE PROCESO CON UNA RECIRCULACION SIMPLE.

Si el flujo en la corriente de recirculación,  $S_R$ , fué conocido, entonces, el valor de todas las corrientes restantes sería calculado en un sólo recorrido a través del sistema.

Al simular el comportamiento de este proceso se establece un conjunto de ecuaciones simultáneas de este tipo

$$S_1 = F_a (S_F, S_R) \quad (3.9)$$

$$S_p = F_{b1} (S_1) \quad (3.10)$$

$$S_R = F_{b2} (S_1) = F_{b2} (F_a (S_F, S_R)) \quad (3.11)$$

en donde,  $S_F$  es conocida,  $S_p$  es una variable explícita y  $S_R$  y  $S_1$  son variables implícitas.

Ahora bien, el comportamiento de un proceso químico siempre debe ajustarse a ciertos criterios de operabilidad, un ejemplo de los cuales podría ser que la temperatura de cierta corriente alcance los 200 °F, o que la concentración molar de un reactivo se conserve en 0.001, o que el 98% de un producto sea removido de cierta corriente haciéndolo pasar por una columna de destilación. Como se mencionó anteriormente, estos criterios se conocen como especificaciones de diseño y usualmente son estructuradas en forma de funciones que son satisfechas cuando alcanzan un valor de cero, en este caso:

$$T_{S_1} - 200 = 0 \quad (3.12)$$

$$C_{\text{reac.}} - 0.001 = 0 \quad (3.13)$$

$$F_{\text{sup}} C_{10} / (F_{\text{sup}10} + F_{\text{inf}10}) - 0.98 = 0 \quad (3.14)$$

Los simuladores de procesos poseen varias unidades que pueden aceptar especificaciones de diseño cuyo propósito es variar los parámetros clave de una o más unidades a fin de cumplir con ciertas especificaciones de diseño, fijadas por el usuario.

De lo anteriormente expuesto, se resume que los problemas básicos al simular un proceso son: primero, la obtención los valores adecuados para las variables implícitas de las corrientes de recirculación; y segundo, ajustar los parámetros de cada unidad para satisfacer las especificaciones de diseño.

La solución de estos problemas se logra empleando métodos aplicables a sistemas de ecuaciones simultáneas tanto directos como iterativos, en los cuales algunas de las variables son de carácter implícito, tal y como se muestra en los Apéndices A y B.

La mayoría de estos métodos se inician con valores supuestos para las variables implícitas, que no necesariamente son los valores correctos, pero que al menos están próximos a ellos para que a continuación se aplique un procedimiento adecuado para obtener un nuevo conjunto de valores que se acerquen más a la solución del sistema. Al repetir este procedimiento varias ve-

se puede obtener un conjunto solución que se aproxime al valor real tanto como sea requerido. Cuando las iteraciones tienen éxito, en cuanto a que producen valores más cercanos a la solución, se dice que el método converge. Si, por el contrario, las iteraciones producen valores erróneos, oscilantes o que se alejan de la solución, se dice que el método diverge.

La mayoría de los métodos de convergencia estructura las ecuaciones de simulación de modo que una variable implícita viene a ser una función de sí misma, por ejemplo, en la ecuación 3.11,  $S_R$  es función de  $S_R$ .

Por otro lado, es usualmente aceptado que cuanto más flexible es un sistema de simulación, existen más oportunidades de realizar una representación errónea del problema que se requiere resolver, en donde las especificaciones elegidas por el usuario pueden ser inconsistentes, o no definir el proceso completamente.

Existen tres clases de especificación errónea de un problema de simulación, las cuales, en orden creciente de la dificultad en su diagnóstico, son:

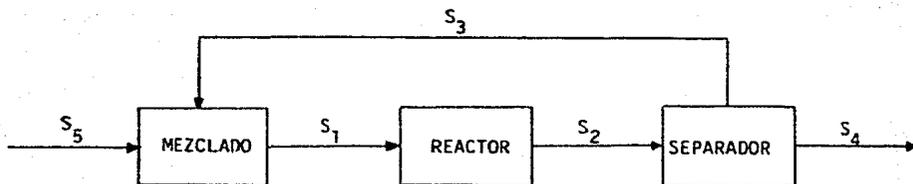
- 1) Proponer un número equivocado de especificaciones, de modo que el problema esté mal especificado, ya sea por exceso o por defecto.

2) Proporcionar el número correcto de especificaciones, pero eligiendo un conjunto que conduzca hacia un sistema de ecuaciones singular, por ejemplo, el que las especificaciones elegidas no sean independientes.

3) Proporcionar el número correcto de especificaciones, eligiendo un conjunto independiente, pero seleccionando valores para las variables especificadas que sean físicamente inexistentes.

Los problemas del tipo 1 son fácilmente detectables, puesto que sólo es necesario contar el número de ecuaciones y el número de variables desconocidas para saber si el sistema está exactamente definido o si existe una sobre o subdeterminación.

En la Figura 3.12 se muestra un proceso simple para ilustrar los problemas del tipo 2. El sistema produce C a partir de los reactivos A y B. El usuario ha decidido no especificar ni el flujo de A en la corriente de alimentación, ni la conversión en el reactor, especificaciones que podrían ser muy útiles para la simulación, ya sea secuencial o simultánea; en su lugar, se han especificado los flujos de B y C a la salida (corriente  $S_4$ ). La razón por la cual el problema está mal especificado es que la cantidad de B y C en la corriente  $S_4$  está relacionada con la estequiometría de la reacción, por lo tanto, al especificar ambas cantidades no se proporcionan dos orígenes diferentes e independientes de información. Los problemas



Componente clave: B

<b>Especificaciones:</b>	Alimentación: flujo de B; flujo de C=0
	Separador: 95% de A en corriente $S_3$ 40% de B en corriente $S_3$ Nada de C en corriente $S_3$
	Producto: flujo de B en corriente $S_4$ flujo de C en corriente $S_4$

FIGURA 3.12. EJEMPLO DE UN PROBLEMA DE SIMULACION MAL ESPECIFICADO.

de este tipo están caracterizados por una matriz Jacobiana singular, tanto en forma estructural, como funcional. En la Figura 3.13 se muestra la matriz de ocurrencia del proceso de la Figura 3.12, en la cual:

$a_i$  = flujo de A en la corriente  $i$

$b_i$  = flujo de B en la corriente  $i$

$c_i$  = flujo de C en la corriente  $i$

$v$  = conversión en el reactor

y en donde el origen de las ecuaciones 1 y 2 es el mezclador; de las ecuaciones 3 a 5 es el reactor y de las ecuaciones 6 a 10 es el separador.

Una breve inspección a la Figura 3.13 hace resaltar que las ecuaciones 2, 4, 5, 7 y 10 sólo dependen de cuatro incógnitas ( $b_1$ ,  $b_2$ ,  $b_3$ ,  $x$ ). Sin embargo, las cinco variables de  $a_1$  a  $a_5$  sólo se presentan en cuatro ecuaciones (1, 3, 6, 9).

La singularidad estructural de una matriz se puede determinar al encontrar la transversal máxima, esto es, el mayor número de elementos diferentes de cero que es posible colocar sobre la diagonal principal, mediante la permutación de filas y columnas. Si esta cantidad es menor que el número de ecuaciones o variables, la matriz es singular. En el Apéndice B se muestran varios algoritmos creados para abordar este problema.

	$a_1$	$a_2$	$a_3$	$a_4$	$a_5$	$b_1$	$b_2$	$b_3$	$c_2$	$v$
1	1		1		1					
2						1		1		
3	1	1				1				1
4						1	1			1
5						1			1	1
6		1		1						
7							1			
8									1	
9		1	1							
10							1	1		

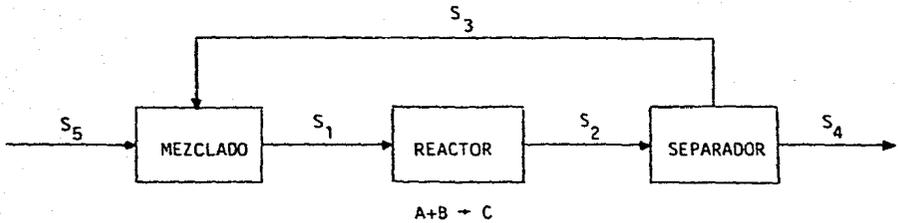
FIGURA 3.13. MATRIZ DE OCURRENCIA DEL PROBLEMA DE SIMULACION DE LA FIGURA 3.12.

La singularidad funcional es producida cuando se eligen valores numéricos para las especificaciones los cuales causan relaciones de dependencia entre las ecuaciones, o crean elementos de la matriz Jacobiana con valor de cero y que habían sido considerados como diferentes de cero en el análisis estructural.

En la Figura 3.14 se replantea el proceso de la figura 3.12; ahora es un problema de simulación en donde las corrientes de alimentación y los parámetros de equipo están completamente especificados, por lo tanto, estructuralmente el problema es no singular. Sin embargo, cuando la fracción de separación de A en el separador se convierte en uno, entonces sobreviene una definición errónea del problema. Este tipo de situaciones sólo pueden ser detectadas al evaluar numéricamente la matriz Jacobiana y definir si la singularidad es realmente funcional (es decir, que sea independiente del punto en que el Jacobiano es evaluado), en cuyo caso el problema está mal especificado, o que se trate de una singularidad numérica (es decir, que sea dependiente del punto de evaluación del Jacobiano), en cuyo caso, una elección diferente del punto de cálculo, la removerá.

Los problemas erróneamente definidos del tipo 3 son imposibles de detectar a-priori. Pueden tener varios orígenes, siendo los más frecuentes la violación de los balances de materia globales, por ejemplo:

- 1) Especificar un número mayor de componentes saliendo de



Especificaciones:	Alimentación:	flujo de A y B. C=0
	Reactor:	conversión de B
	Separador:	100% de A en corriente S <sub>3</sub>
		40% de B en corriente S <sub>3</sub>
		Nada de C en corriente S <sub>3</sub>

FIGURA 3.14. EJEMPLO DE UN PROBLEMA DE SIMULACION MAL ESPECIFICADO.

un proceso de separación física, de los que fueron introducidos.

2) Especificando una cantidad mayor de productos de los que son posibles de obtener de acuerdo a la alimentación al proceso y la estequiometría de las reacciones.

Este tipo de problemas puede causar una falla en el procedimiento de solución, particularmente si fueron impuestos ciertos límites y condiciones sobre las iteraciones con el objeto de mantener realidad física; o puede causar convergencia hacia soluciones físicamente inexistentes.

Se debe tener especial atención al efectuar un diagnóstico de fallas de un sistema, ya que puede ser difícil distinguir entre fallas debidas al planteamiento erróneo del problema, falla debida a que el método numérico empleado no es lo suficientemente poderoso para el problema que se está manejando y fallas por no proporcionar los estimados iniciales adecuados para iniciar la simulación.

Locke y Westerberg (1982), proponen una aproximación evolutiva para realizar la definición de las ecuaciones y variables de simulación, en la que sugieren introducir las especificaciones del proceso una a una y evaluando los posibles cambios que implican, lo que permite cambiar totalmente estas especificaciones o modificarlas parcialmente para lograr una correcta definición.

### 3.4. FORMULACION DEL PROBLEMA DE SIMULACION.

Es práctica común que antes de iniciar la simulación de un proceso, éste deba ser sometido a un procedimiento mediante el cual se divida en bloques irreducibles de unidades, es decir, el proceso es descompuesto con el objeto de facilitar su representación y solución. Estos procedimientos son descritos en el Apéndice B, relacionados con la solución de Matrices Dispersas.

El reordenamiento del proceso original va dirigido principalmente a las corrientes de recirculación. Usualmente las corrientes de alimentación y de productos son conocidas como las entradas y salidas de un proceso; sin embargo, al promover la convergencia en la simulación, las corrientes de alimentación son sólo cantidades "conocidas y constantes", mientras que las cantidades "variables y desconocidas", están representadas en las corrientes de los ciclos de recirculación.

Cuando los ciclos son cortados, ciertas corrientes clave son divididas y el orden de cálculo de cada bloque es ajustado de modo que la corriente de recirculación inicial se convierta en dos nuevas corrientes, una de entrada y otra de salida. Cuando se realiza el rompimiento adecuado de un proceso, este no debe contener ninguna recirculación y únicamente los valores de entrada desconocidos para las corrientes cortadas deberán ser determinados.

En la Figura 3.15 se presenta un proceso que contiene tres corrientes de recirculación,  $R_1$ ,  $R_2$  y  $R_3$ ; mientras que en la Figura 3.16, ya se ha realizado el rompimiento de estas corrientes, y finalmente, en la Figura 3.17 se presenta el proceso con las corrientes de recirculación dispuestas en una estructura del tipo entrada/salida.

Como parte del desarrollo secuencial se ha promovido la búsqueda de estrategias óptimas de corte. Los primeros intentos al respecto se hicieron aplicando la teoría de gráficas, cuyo desarrollo es similar al ordenamiento de ecuaciones, en el cual ciertos nodos (módulos) deben ser partidos y ordenados, mientras que el conjunto irreducible restante debe ser resuelto en forma iterativa; estas técnicas son descritas como parte del Apéndice B. La descomposición y reordenamiento a nivel de módulo de proceso son relativamente más fáciles que a nivel de ecuaciones, en virtud de que el conjunto de salida (valores calculados por el módulo) están ya fijados por la estructura del proceso.

Pho y Lapidus (1973) y Rosen (1980), representan la tarea de elegir un conjunto de corte como un problema de minimización:

$$\text{Min} \quad \sum_{j=1}^n p_j \cdot x_j \quad (3.15)$$

$$\text{sujeto a} \quad \sum a_{ij} \cdot x_j \geq 1 \quad (3.16)$$

$$x_j = 0.1 \quad j = 1, n$$

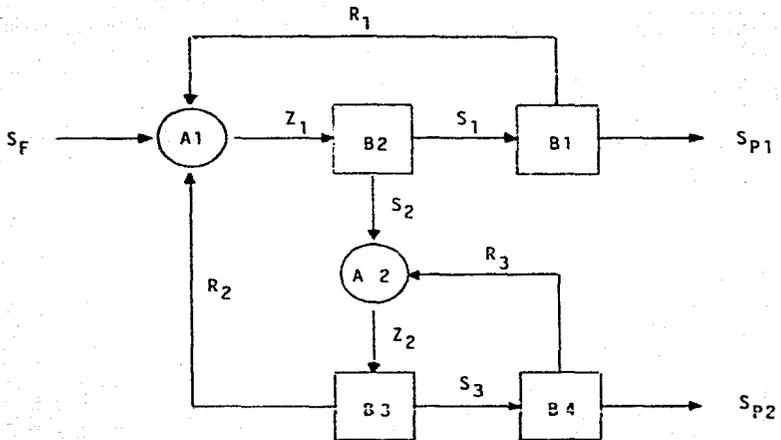


FIGURA 3.15. EJEMPLO DE UN PROCESO CON TRES CORRIENTES DE RECIRCULACION.

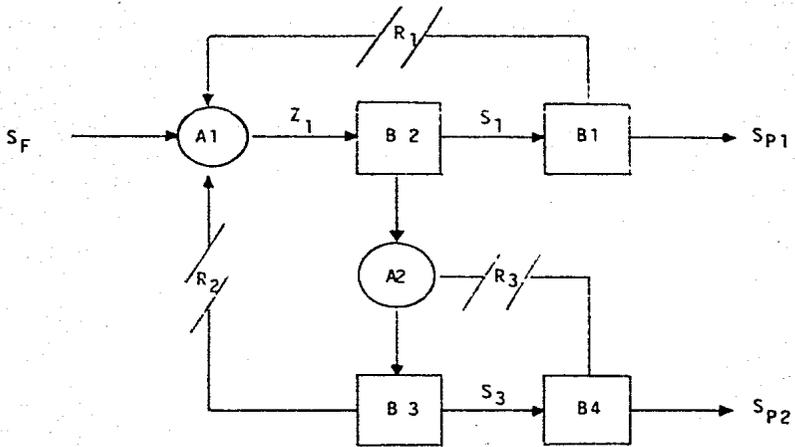


FIGURA 3.16.

EL PROCESO DE LA FIGURA 3.15 CON  
SUS CORRIENTES DE RECIRCULACION EN  
FORMA DE CORRIENTES DE CORTE.

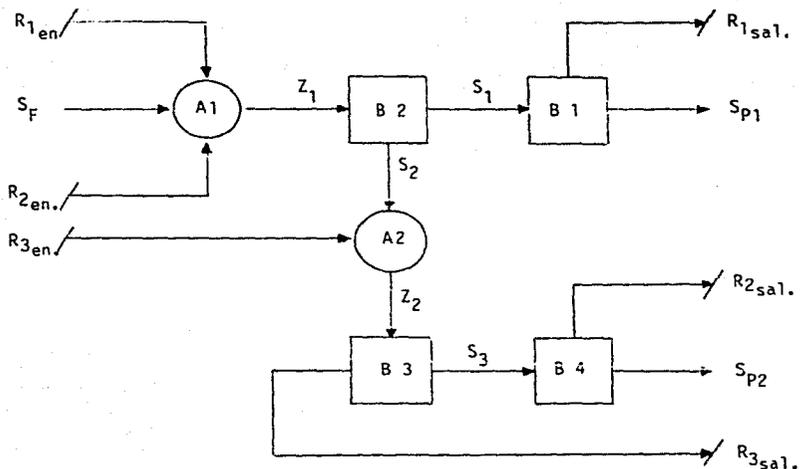


FIGURA 3.17.

EL PROCESO DE LAS FIGURAS 3.15 Y 3.16 DISPUESTO EN LA FORMA ENTRADA-SALIDA DE LAS CORRIENTES DE CORTE.

en donde, los valores para  $x_j$  corresponden a si la corriente  $j$  es de corte ( $x_j = 1$ ), o no ( $x_j = 0$ ). Los coeficientes  $a_{ij}$  usualmente son ordenados como una matriz de ciclo, en donde un valor  $a_{ij} = 1$  indica que la corriente  $j$  aparece en el ciclo  $i$ , de no ser así,  $a_{ij} = 0$ . Finalmente, los elementos  $P_j$  son factores de ponderación definidos por el usuario y que operan sobre las corrientes de corte. El objetivo es encontrar un conjunto de corte con el factor de ponderación mínimo y que rompa cada ciclo al menos una vez. A fin de obtener un conjunto de corte óptimo, se han propuesto algunos criterios para elegir los valores de  $P_j$ :

1) Cortar el mínimo de corrientes, con  $P_j = 1$  (Barkeley y Motard (1972)).

2) Cortar el mínimo de variables en las corrientes, siendo  $P_j =$  número de variables en la corriente  $j$ .

3) Cortar el mayor número de ciclos posible sólo una vez (cortes no redundantes). (Upadhye y Grens (1975)).

4) Elegir un conjunto de corte óptimo obteniendo una medida de los eigenvalores para la convergencia (Genna y Motard (1975))

Sin embargo, el concepto de conjunto de corte óptimo aún es difícil de definir aplicado a un enfoque modular simultáneo, el cual tiene una tendencia al manejo más eficiente de las ecuacio-

nes. Adicionalmente, ninguna de las anteriores estrategias aborda explícitamente el problema de las especificaciones de diseño, las cuales no pueden ser introducidas directamente como un parámetro de entrada al proceso.

Ante esta situación se hace necesario plantear el problema de simulación y diseño en una forma diferente.

Existen varias formulaciones del problema bajo el enfoque Modular Simultáneo, sin embargo, de manera general pueden ser consideradas dentro de tres categorías, tal y como lo proponen Chen (1982) y Chen y Stadtherr (1985). La descripción de cada categoría será realizada con respecto al sencillo proceso mostrado en la Figura 3.18, el cual contiene cuatro unidades: un mezclador, un reactor, un separador y un divisor. Se debe tomar en cuenta que el divisor contiene un parámetro de equipo,  $w$ , que debe ser ajustado para cumplir con la especificación de diseño  $h$ , impuesta a la salida del separador.

#### 3.4.1. FORMULACION 1.

En esta formulación todas las corrientes de conexión son divididas y cada una es considerada como dos corrientes separadas,

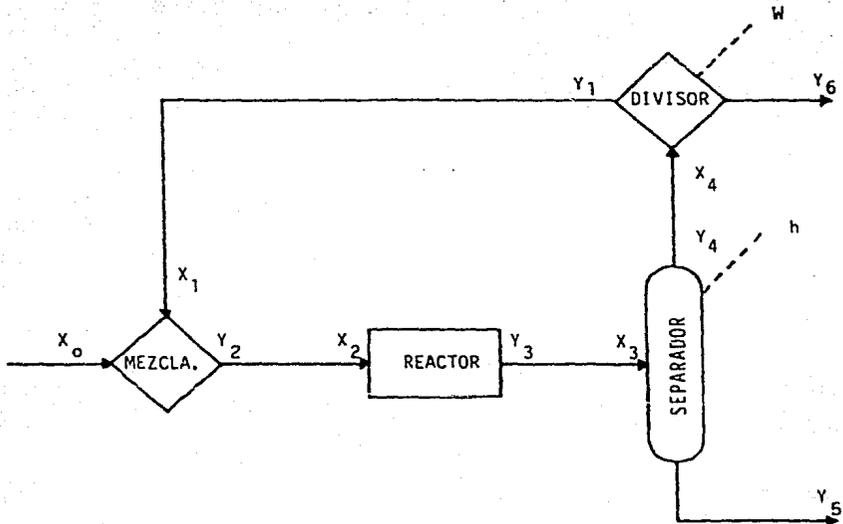


FIGURA 3.18. PROCESO SIMPLE USADO PARA DESCRIBIR LAS FORMULACIONES I, II Y III.

una parte como corriente de salida de un equipo y la otra parte como corriente de entrada al siguiente equipo.

En base a la orientación modular, las ecuaciones de simulación consisten de las ecuaciones del modelo unitario, ecuaciones de las corrientes de conexión y especificaciones de diseño:

Ecuaciones del modelo unitario:

$$y_2 = g_2(x_1) \quad (3.17 \text{ a})$$

$$y_3 = g_3(x_2) \quad (3.17 \text{ b})$$

$$y_4 = g_4(x_3) \quad (3.17 \text{ c})$$

$$y_1 = g_1(x_4, w) \quad (3.17 \text{ d})$$

Ecuaciones de las corrientes de conexión:

$$x_2 = y_2 \quad (3.17 \text{ e})$$

$$x_3 = y_3 \quad (3.17 \text{ f})$$

$$x_4 = y_4 \quad (3.17 \text{ g})$$

$$x_1 = y_1 \quad (3.17 \text{ h})$$

Especificación de diseño:

$$h^s = h(x_3) \quad (3.17 \text{ i})$$

en donde  $h$  es una variable de salida que deberá adquirir un valor de  $h^S$ . Al emplear esta formulación un problema general de simulación o diseño puede ser representado por  $2(c+2)n_c + n_w$  ecuaciones no lineales, en donde  $c$ ,  $n_c$  y  $n_w$  representan el número de especies químicas, corrientes de conexión y especificaciones de diseño, respectivamente. Debe notarse que la expresión  $(c+2)$  significa que cada corriente puede ser caracterizada mediante  $c$  flujos por componente, la entalpía y la presión

Cuando las ecuaciones y variables son reordenadas apropiadamente, la matriz Jacobiana requerida para resolver todas las ecuaciones del conjunto 3.17, presenta la forma mostrada en la Figura 3.19. Las submatrices A y B, de esa figura, son matrices de derivadas parciales de las variables de salida de los módulos con respecto a las variables de entrada.

### 3.4.2. FORMULACION II.

En la Formulación I cada corriente de conexión es tratada como si fueran dos corrientes separadas, lo que produce un sistema de ecuaciones demasiado extenso. En la Formulación II, cada corriente de conexión es considerada como una corriente de

	$y_2$	$x_2$	$y_3$	$x_3$	$y_4$	$x_4$	$y_1$	$x_1$	$w$
3.17 a		1							$-A_{21}$
3.17 e	-1	1							
3.17 b			$-A_{32}$	1					
3.17 f			-1	1					
3.17 c					$-A_{43}$	1			
3.17 g					-1	1			
3.17 d							$-A_{14}$	1	$-A_{1w}$
3.17 h							-1	1	
3.17 i									$-B_3$

FIGURA 3.19 MATRIZ JACOBIANA PARA LA FORMULACION I

entrada y las ecuaciones del modelo unitario son sustituidas dentro de éstas. Con respecto al proceso de la Figura 3.18, el sistema de ecuaciones que lo representan, de acuerdo a esta formulación es:

Ecuaciones de las corrientes de conexión:

$$x_2 = g_2 (x_1) \quad (3.18 \text{ a})$$

$$x_3 = g_3 (x_2) \quad (3.18 \text{ b})$$

$$x_4 = g_4 (x_3) \quad (3.18 \text{ c})$$

$$x_1 = g_1 (x_4, w) \quad (3.18 \text{ d})$$

Especificación de diseño:

$$h^s = h (x_3) \quad (3.18 \text{ e})$$

Empleando esta formulación, un problema de simulación y diseño puede ser representado por  $(c+2)n_c + n_w$  ecuaciones no lineales; por lo tanto, el número de ecuaciones a nivel de proceso es reducido en un 50%, comparado con la Formulación I. La matriz Jacobiana de todas las ecuaciones del conjunto 3.18, reordenadas convenientemente, se muestra en la Figura 3.20. Debe notarse que las submatrices A y B, que deben ser calculadas, son las mismas de la Formulación I.

	$x_2$	$x_3$	$x_4$	$x_1$	$w$
3.18 a	1	•		$-A_{21}$	
3.18 b	$-A_{32}$	1			
3.18 c		$-A_{43}$	1		
3.18 d			$-A_{14}$	1	$-A_{1w}$
3.18 e		$-B_3$			

FIGURA 3.20

MATRIZ JACOBIANA PARA LA FORMULACION II

## 3.4.3. FORMULACION III.

Para aplicaciones típicas, el número de ecuaciones generadas al emplear la Formulación II, aún puede ser mayor de 10,000 y tales sistemas son muy grandes para ser resueltos mediante técnicas convencionales para matrices completas.

Con el objeto de reducir aún más el número de ecuaciones no lineales que deberán ser resueltas simultáneamente, un subconjunto apropiado de las corrientes de conexión del proceso deben ser divididas, en lugar de la totalidad de ellas, como sucede en la Formulación I. Las corrientes no divididas pueden ser eliminadas usando las ecuaciones correspondientes de las corrientes de conexión. Para el proceso de la Figura 3.18, la corriente  $x_1$  puede ser dividida y las corrientes  $x_2$ ,  $x_3$  y  $x_4$  pueden ser eliminadas usando las ecuaciones 3.18 a, 3.18 b y 3.18 c, generando este sistema de ecuaciones:

Ecuaciones de las corrientes de conexión:

$$x_1 = g_1 (g_4 (g_3 [g_2 (x_1)]), w) = \bar{g}_1 (x_1, w) \quad (3.19 a)$$

Especificación de diseño:

$$h = g_3 [g_2 (x_1)] = \bar{h} (x_1) = h^s \quad (3.19 b)$$

Usando esta formulación, un problema de simulación y diseño puede ser representado por  $(c+2)n_d + n_w$  ecuaciones no lineales en donde  $n_d$  es el número de corrientes divididas. Puesto que en la mayoría de las aplicaciones típicas  $n_d$  es un número relativamente pequeño, todas las ecuaciones del conjunto 3.19 pueden ser resueltas mediante las técnicas conocidas para matrices completas, tomando en cuenta que, usualmente, la matriz Jacobiana para esta formulación es de ese tipo.

Es importante destacar que en las tres formulaciones, un problema de simulación y diseño es descrito por un sistema de ecuaciones no lineales de la forma  $f(x) = 0$ , en donde la evaluación de  $f(x)$  requiere un sólo recorrido a través de todos los módulos unitarios.

### 3.5. EVALUACION DE LA MATRIZ JACOBIANA.

Es un hecho indiscutible que la eficiencia de la Orientación Modular depende en buena parte de la forma en que la matriz Jacobiana es evaluada.

Si los cálculos involucrados son excesivos, el enfoque Modular Simultáneo podría no ser competitivo con el enfoque Modular Secuencial. Además, si es utilizada una aproximación muy pobre del Jacobiano, el enfoque Modular Simultáneo puede requerir demasiadas iteraciones para resolver el problema y nuevamente resultará ineficiente.

Cualquiera que sea el tipo de aproximación, siempre existirá la necesidad de determinar las derivadas parciales, o una aproximación a ellas, que resuelvan las ecuaciones a nivel de proceso. Estas derivadas parciales están representadas por las variables de conexión de las corrientes de salida y algunas variables en el vector de retención, variadas con respecto a las variables de conexión de las corrientes de entrada y a los parámetros de equipo libres. A continuación se presentan algunas de las opciones existentes para calcular o aproximar estas derivadas parciales.

### 3.5.1. DIFERENCIACION ANALITICA.

Dada la complejidad de los módulos unitarios, a primera vista parece poco práctico determinar las relaciones lineales entrada-salida en forma analítica. Sin embargo, si todas las

ecuaciones que describen un módulo son resueltas mediante linealización simultánea, es posible determinar estas relaciones analíticamente.

Recordando que el módulo unitario calcula variables internas, variables de retención y variables en las corrientes de salida, a partir de valores conocidos de las variables de las corrientes de entrada y de los parámetros de equipo, las ecuaciones que describen al módulo pueden ser expresadas así:

$$G(Z_1, Z_2) = 0 \quad (3.20)$$

en donde el vector  $Z_1$  incluye a las variables internas, a las variables de retención y a las variables de las corrientes de salida, mientras que el vector  $Z_2$  incluye a las variables de las corrientes de entrada y los parámetros de equipo. Además, puesto que el número de ecuaciones en el sistema 3.20 es el mismo que el número de variables en  $Z_1$ , usando la relación de Euler:

$$\begin{bmatrix} \frac{\partial G}{\partial Z_1} \\ \frac{\partial G}{\partial Z_2} \end{bmatrix} G = - \begin{bmatrix} \frac{\partial G}{\partial Z_1} \\ \frac{\partial G}{\partial Z_2} \end{bmatrix}^{-1} Z_2 \cdot \begin{bmatrix} \frac{\partial G}{\partial Z_2} \\ \frac{\partial G}{\partial Z_1} \end{bmatrix} Z_1 \quad (3.21)$$

en donde, todas las derivadas parciales pueden ser determinadas sin resolver repetidamente el sistema de ecuaciones 3.20 para  $Z_2$ .

Tal vez esta sea la técnica general más apropiada para evaluar las derivadas parciales, sin embargo, en ocasiones no es

recomendable su uso en virtud de consumir demasiado tiempo de cómputo y porque la mayoría de los módulos no resuelven ecuaciones no lineales mediante linearización simultánea.

### 3.5.2. PERTURBACION NUMERICA TOTAL.

Otra técnica simple para evaluar derivadas es calculándolas mediante una aproximación diferencial, mediante esta relación:

$$\frac{\partial y_k}{\partial x_j} = \frac{y_k' - y_k}{x_j' - x_j} \quad (3.22)$$

Esto requiere un cálculo, a nivel de módulo, después de que cada variable de entrada es perturbada en una pequeña cantidad,  $x_j'$ . Esta técnica es apropiada para aplicarse en las Formulaciones I y II, existiendo  $(c+2)n_{c_i} + n_{w_i}$  variables de entrada desconocidas para el módulo  $i$ . Por lo tanto, para evaluar la matriz Jacobiana, los cálculos en el módulo  $i$  deben ser ejecutados  $(c+2)n_{c_i} + n_{w_i} + 1$  veces.

Lo anterior indica que las variables de las corrientes de entrada deberán ser perturbadas una a una, es decir:

$$\frac{\partial y_k}{\partial x_j} = \left[ \begin{array}{c} \Delta y_k \\ \Delta x_j \end{array} \right]_{x_m \neq j} \quad ; \quad m = 1, \dots \quad (3.23)$$

lo que implica que la matriz resultante será una matriz completa.

### 3.5.3. PERTURBACION DIAGONAL.

La perturbación numérica total requiere muchos cálculos a nivel de módulo para el cálculo de las derivadas parciales.

Mahalec, Kluzik y Evans (1979), proponen el uso de valores aproximados para estas derivadas, con el objeto de reducir el número de cálculos. Ellos proponen una relación linearizada entre las corrientes de entrada ( $x$ ) y de salida ( $y$ ):

$$y = Ax + b \quad (3.24)$$

por tanto,

$$A_{kj} = \frac{\partial y_k}{\partial x_j} \quad (3.25)$$

en donde,  $y_k$  es el vector de la  $k$ -ésima salida y  $x_j$  es el vector de la  $j$ -ésima entrada, siendo  $A_{kj}$  una matriz diagonal.

Se perturban todos los elementos del vector  $x_j$  al mismo tiempo en una cantidad  $\Delta x_j$  y a continuación se determina el cambio ocurrido en el vector  $y_k$ , representado por  $\Delta y_k$ , y se obtienen los elementos  $mm$  de  $D_{kj}$ .

$$\begin{bmatrix} D \\ k j \end{bmatrix}_{mm} = \frac{\begin{bmatrix} \Delta y_k \end{bmatrix}_m}{\begin{bmatrix} \Delta x_j \end{bmatrix}_m} \quad (3.26)$$

siendo  $D_{kj}$  una aproximación diagonal a la matriz  $A_{kj}$ .

Debe notarse que en este procedimiento se supone que los  $m$ -ésimos elementos de la corriente de salida sólo son afectados por los  $m$ -ésimos elementos de la corriente de entrada.

Mientras que esta suposición puede ser razonable para flujos de componentes y presiones de una corriente, el desprestigiar las interacciones entre estas variables y la entalpía de los flujos, puede conducir, en ciertos casos, en una aproximación muy pobre.

En general, se requieren  $n_{c_i} + n_{w_i} + 1$  cálculos del módulo  $i$  para evaluar la matriz Jacobiana mediante este procedimiento, el cual es apropiado para ser aplicado a las Formulaciones I y III, descritas anteriormente.

### 3.5.4. DIFERENCIACION DIRECTA.

Anteriormente se indicó que al emplear la Formulación III un proceso químico es representado por un sistema de ecuaciones no lineales relativamente pequeño, cuya matriz Jacobiana usualmente contiene muy pocos elementos con valor cero. Según demostró Perkins (1979), el método más directo para calcular el Jacobiano de esta formulación es mediante aproximación diferencial, sin embargo, esta técnica puede resultar menos eficiente que las técnicas de perturbación (tanto diagonal, como total), usadas en las Formulaciones I y II, por estas razones:

Primero, cuando el proceso a ser simulado es muy extenso y complicado, el número de corrientes de corte se verá altamente incrementado y por lo tanto el número de cálculos en los módulos aumentará también. Al emplear técnicas de perturbación para calcular el Jacobiano, el número de cálculos en cada módulo es proporcional al número de corrientes de conexión que entran a él y puesto que el número de estas corrientes viene siendo el mismo aún cuando el proceso sea muy grande y complejo, los métodos de perturbación pueden resultar más ventajosos.

Segundo, cuando las técnicas de perturbación se usan para evaluar la matriz Jacobiana, todos los cálculos para un determinado módulo son realizados consecutivamente. Esto significa que es relativamente fácil retener en forma temporal las varia-

bles internas de un módulo en particular y utilizar estos valores para cálculos subsecuentes de éste, con lo que es posible reducir el tiempo de cómputo.

Tercero, puesto que el desarrollo de los sistemas Modulares Simultáneos continúa, es muy posible que las derivadas de ciertos módulos, con un alto consumo de procesamiento, sean modeladas para evaluarlas analíticamente. Esta opción puede ser fácilmente implementada cuando se usan las técnicas de perturbación.

En base a las anteriores consideraciones, Chen (1982), propone que cuando se use la Formulación III, la matriz Jacobiana sea calculada usando métodos de perturbación, usualmente asociados con las Formulaciones I y II. Esto es posible porque la única diferencia entre las Formulaciones I y II, con respecto a la III, es que en esta última, algunas ecuaciones de la Formulación II son usadas para eliminar algunas variables.

Matemáticamente, las ecuaciones en la Formulación II pueden ser divididas así:

$$f(S,T) = 0 \quad (3.27 \text{ a})$$

$$g(S,T) = 0 \quad (3.27 \text{ b})$$

El sistema 3.27 a representa las ecuaciones de las corrientes de conexión no divididas, correspondientes a las ecuaciones

ciones de 3.18 a hasta 3.18 c, de la segunda formulación, y el sistema 3.27 b representa las ecuaciones de las corrientes de conexión divididas y las especificaciones de diseño, correspondientes a las ecuaciones 3.18 d y 3.18 e.

El vector  $T$  representa las variables de las corrientes divididas y las variables libres, mientras que el vector  $S$  representa las variables de las corrientes no divididas. Para obtener la Formulación III, el sistema 3.27 a se usa para determinar  $S$  como una función de  $T$  a fin de reducir las ecuaciones 3.27 a y b, a este sistema:

$$g(S,T) = g[ S(T), T ] = \bar{g}(T) \quad (3.28)$$

correspondiente a la ecuación 3.19 a de la Formulación III.

Usando la relación de Euler, es posible obtener el Jacobiano para la Formulación III, a partir del Jacobiano para la Formulación II, mediante:

$$\left[ \frac{d \bar{g}}{d T} \right]_f = \left[ \frac{\partial g}{\partial T} \right]_S - \left[ \left[ \frac{\partial g}{\partial S} \right]_T \right]^{-1} \cdot \left[ \frac{\partial f}{\partial S} \right]_T \cdot \left[ \frac{\partial f}{\partial T} \right]_S \quad (3.29)$$

En conclusión, con respecto al método de evaluación del Jacobiano a emplearse en la Formulación III, se tiene que la diferenciación directa puede ser aplicada, aunque con las consideraciones anteriormente citadas; siendo preferible aplicar alguna de las técnicas de perturbación con ayuda de la ecuación 3.29.

## CAPITULO 4

EL CONCEPTO MODULAR SIMULTANEO EN LA  
SIMULACION DE PROCESOS QUIMICOS .

## CAPITULO 4. EL CONCEPTO MODULAR SIMULTANEO EN LA SIMULACION DE PROCESOS.

En el capítulo anterior fueron presentados los cuatro principales enfoques bajo los cuales se ha desarrollado la simulación de procesos químicos en estado estacionario.

Para un proceso cuyos ciclos de recirculación son independientes, como el mostrado en la Figura 4.1, el enfoque Modular Secuencial, con algún esquema de aceleración de convergencia del tipo Wegstein (1958), ha demostrado ser satisfactorio.

Sin embargo, para procesos complejos, con ciclos de recirculación anidados y con severas restricciones de diseño, como el mostrado en la Figura 4.2, el enfoque Modular Secuencial presenta un comportamiento de convergencia muy pobre, en virtud de las fuertes interacciones entre los cálculos inherentes a los ciclos anidados.

Alternativamente, los enfoques orientados hacia las ecuaciones presentan la ventaja de poder manejar los ciclos de cálculo en forma simultánea, sin que su eficiencia se vea disminuída por las fuertes interacciones, sin embargo, para aplicaciones típicas,

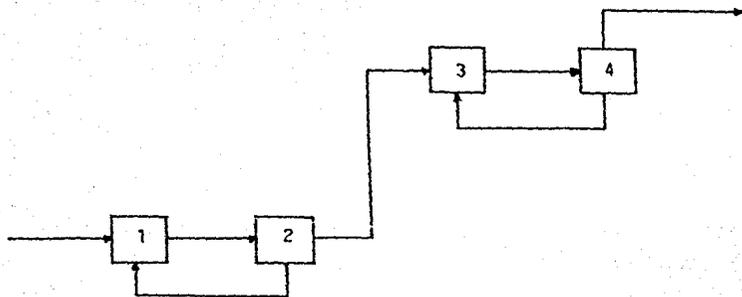


FIGURA 4.1. PROCESO CON CICLOS DE RECIRCULACION INDEPENDIENTES.

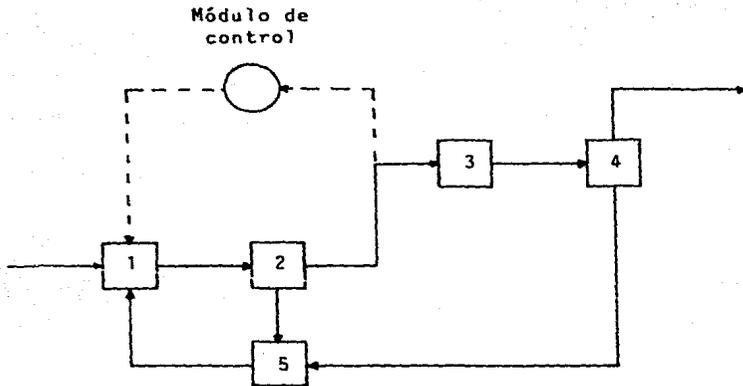


FIGURA 4.2. PROCESO CON CICLOS DE RECIRCULACION ANIDADOS.

estos enfoques requieren una gran cantidad de memoria de computadora, además, no es posible emplear los programas ya existentes, los cuales están provistos de eficientes y robustos modelos para el manejo de módulos unitarios.

Ante esta situación se presentan dos opciones a fin de mejorar la eficiencia de los cálculos:

1) Desarrollar totalmente nuevos simuladores, capaces de resolver grandes sistemas de ecuaciones no lineales y de hacer converger en forma simultánea tanto las especificaciones de diseño como los balances de masa y energía.

2) Convertir los actuales simuladores modulares secuenciales en simuladores con un manejo más eficiente de las ecuaciones.

La primera alternativa parece ser la solución para un eficiente manejo de los problemas de simulación, diseño y optimización; sin embargo, aún se encuentra en etapa de investigación y presenta la gran desventaja de no sacar provecho del gran volumen de programas referentes al desarrollo modular. Por lo tanto, en la actualidad, la segunda opción resulta más atractiva, puesto que requiere de una menor cantidad de recursos para ser puesta a punto y poder brindar los beneficios de un verdadero simulador con orientación al manejo más eficiente de las ecuaciones.

#### 4.1. LA CONVERGENCIA MODULAR SIMULTANEA.

En la sección 3.2.1.2. se indicó que el término Modular Simultáneo abarca varias tendencias que parten de la idea de mejorar el esquema de convergencia del enfoque Modular Secuencial. Algunas de estas tendencias se manifiestan unicamente como un punto de vista "diferente" de convergencia, como en el caso de Perkins (1979), quien sugiere conservar la estructura secuencial y usar el algoritmo de Broyden (1965) para lograr la convergencia de las corrientes de recirculación seleccionadas (como parte del procedimiento secuencial), en forma simultánea con los parámetros de equipo no especificados o libres.

Sin embargo, el presente trabajo se ocupará de los esquemas de convergencia que hacen al enfoque Modular Simultáneo conceptualmente innovador dentro de la Simulación de Procesos en Estado Estacionario.

Antes de continuar, es pertinente realizar una observación de tipo interpretativo. En los siguientes capítulos la palabra modelo será adoptada para indicar todo un planteamiento de la convergencia del proceso y no deberá confundirse con la designación de métodos y técnicas numéricas. Existen bien conocidas técnicas de aplicación general para la solución de siste-

mas de ecuaciones algebraicas, ya sean lineales o no lineales; sin embargo, cuando se habla de un modelo de convergencia lineal, por ejemplo, en realidad se está haciendo alusión a un entorno general que incluye un particular método de representación modular simultánea (intimamente ligado a ciertos modelos de operaciones unitarias), un determinado procedimiento para el manejo de las ecuaciones de simulación, formas de obtener los parámetros del modelo y generar valores más próximos a la solución, tipo de control de iteraciones y criterios de terminación, etc., y no únicamente a las técnicas de solución numérica aplicables al modelo; de hecho, tal y como se muestra más adelante, cada modelo de convergencia es susceptible de emplear más de una técnica de solución numérica y aún los modelos denominados no lineales tienen una importante etapa de manejo lineal dentro de su procedimiento de solución.

Otro punto digno de destacar es la diferencia entre los modelos representativos de operaciones unitarias y los modelos de convergencia. El modelo unitario reúne una serie de técnicas y procedimientos con el objeto de lograr cierto nivel de representatividad acerca del comportamiento real de una operación unitaria, sin embargo, debe quedar claro que los modelos unitarios constituyen un grupo independiente a los modelos de convergencia, aunque ambos se desarrollen en forma paralela. Un modelo de convergencia puede adoptar uno o más tipos de modelos

unitarios para representar una misma operación unitaria y se puede dar el caso que bajo ciertas condiciones un esquema de convergencia sea más eficiente empleando determinados modelos unitarios y al cambiar esas condiciones sea necesario recurrir a otros modelos diferentes con el objeto de conservar su nivel de eficiencia.

En el caso particular del enfoque Modular Simultáneo, la idea fundamental sugiere que el modelo de convergencia emplee una o más técnicas de solución numérica, dentro de un algoritmo que vincule eficientemente dos clases de modelos unitarios, rigurosos y aproximados (simples), con el objeto de dar solución a todas las corrientes de proceso (y no únicamente a las corrientes de recirculación seleccionadas, como en el caso Modular Secuencial), satisfaciendo las restricciones de diseño en forma simultánea.

Basicamente, los modelos rigurosos son los mismos que se emplean en el enfoque Modular Secuencial, por lo tanto, lo realmente innovador radica en la forma de concebir y resolver el mo-

delo aproximado y en el procedimiento para relacionarlo con el modelo riguroso.

A partir de las ecuaciones descriptivas del proceso, presentadas como ecuaciones de simulación, el modelo riguroso proporciona valores que serán usados como parámetros para el modelo simple, el cual genera ecuaciones aproximadas que serán resueltas en forma simultánea.

Visto de este modo, el concepto de convergencia modular simultánea considera dos niveles de descomposición de las variables y de los cálculos de un proceso a nivel de módulo unitario.

En un nivel se resuelven las ecuaciones descriptivas de cada módulo unitario en forma rigurosa para las variables internas y en el segundo nivel se resuelven las ecuaciones del modelo simple para cada módulo unitario, en forma simultánea con las restricciones de diseño.

Esta descomposición, mostrada en la Figura 4.3, considera dos ciclos de cálculo anidados, en donde el ciclo externo ejecuta los cálculos en forma rigurosa y obtiene los coeficientes aplicables al modelo aproximado, mientras que en el ciclo interno se hacen converger las ecuaciones descriptivas aproximadas del proceso para las variables de entrada y salida.

Converger las expresiones del modelo aproximado y las restricciones de diseño simultáneamente, para las variables de entrada y salida.

- Resolver las ecuaciones descriptivas, en forma rigurosa, de cada módulo unitario para las variables de salida y las variables internas.
- Resolver para obtener los coeficientes del modelo aproximado.

FIGURA 4.3. DESCOMPOSICION DE CALCULO DEL CONCEPTO MODULAR SIMULTANEO.

En comparación, en la Figura 4.4, se puede observar que en la convergencia Modular Simultánea se ha extraído el ciclo interno de cálculo del enfoque Modular Secuencial, el cual representa a los modelos rigurosos y que son los que más tiempo consumen en su ejecución y los sitúa en el ciclo externo, el cual es ejecutado con menor frecuencia.

En su actual estado de desarrollo, el enfoque Modular Simultáneo presenta dos grandes vertientes en cuanto al tipo de modelos aproximados que emplea: modelos lineales y modelos no lineales.

En ambos casos se pretende obtener más información del proceso y su estructura a fin de poder aplicar técnicas de convergencia más eficientes, en lugar de buscar únicamente los conjuntos de corte óptimos y la secuencia de cálculo adecuada.

#### 4.2. MODELOS LINEALES

Estas estrategias están basadas en técnicas de simulación orientadas hacia las ecuaciones, en forma especial a la aplicación del método Newton-Raphson, ya sea sólo o acoplado con méto-

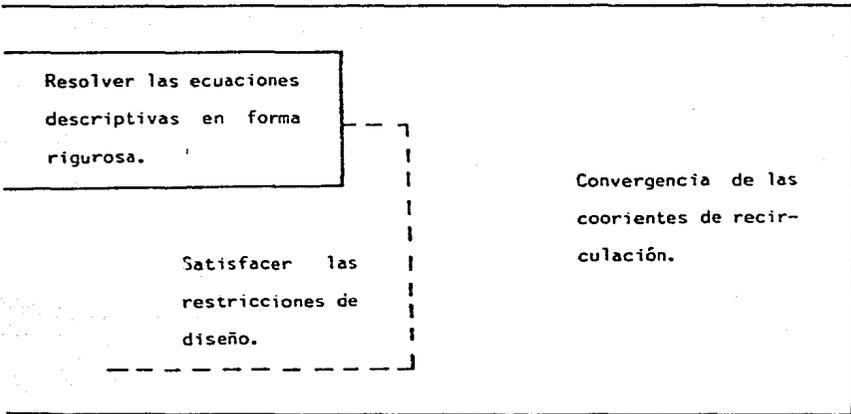


FIGURA 4.4. DESCOMPOSICION DE CALCULO DEL ENFOQUE MODULAR SECUENCIAL.

dos más robustos (Powell, Homotopía-continuación, etc.), para la solución de los sistemas de ecuaciones, algunos de los cuales son descritos en el Apéndice A.

#### 4.2.1. MODELO LINEAL SIMPLE.

En su forma más general, las ecuaciones lineales usadas para modelar cada unidad pueden ser expresadas así:

$$y = Ax + b \quad (4.1)$$

en donde  $x$  representa el flujo por componente a la entrada,  $y$  representa el flujo por componente a la salida,  $A$  es una matriz de coeficientes lineales y  $b$  es un vector de residuos.

Rosen (1962), propuso por vez primera el uso de la Doble Partición ("two tier") como parte del concepto Modular Simultáneo al sugerir el uso de dos clases de modelos:

1) Modelo Riguroso.- Describe una unidad mediante un conjunto de ecuaciones no lineales exactas, auxiliandose de las ecuaciones de conexión.

2) Modelo lineal.— Aproxima las ecuaciones descriptivas no lineales como funciones lineales en términos de los flujos de cada componente en las corrientes de entrada y salida.

Rosen utilizó el modelo lineal de Fracciones de Separación, propuesto por Vela (1961), con el objeto de calcular la matriz de coeficientes lineales, A, la cual es una matriz diagonal con:

$$a_{kj} = \frac{y_k}{x_j} \quad , \quad \text{para } j = k \quad (4.2)$$

$$a_{kj} = \frac{y_k}{x_j} = 0 \quad , \quad \text{para } j \neq k \quad (4.3)$$

siendo, en ambos casos,  $b = 0$ .

El algoritmo propuesto por Rosen puede ser descrito así:

1) Suponer valores para todas las corrientes del proceso, lo que implica formar una matriz de fracciones de separación para cada uno de los equipos involucrados en el proceso.

2) Combinar las matrices de fracciones de separación junto con las corrientes de conexión, conformando una matriz de proceso de estructura lineal.

3) Resolver el sistema de ecuaciones lineales para los flu-

jos por componente, para cada corriente en el proceso.

4) Con los resultados del paso 3, aplicar los modelos rigurosos y obtener valores a la salida de cada equipo.

5) Desarrollar una relación lineal entre las corrientes de entrada y salida de cada equipo.

6) Formar un sistema de ecuaciones lineales del proceso.

7) Resolver el sistema lineal, a fin de obtener nuevos valores para todas y cada una de las corrientes del proceso.

8) Realizar la prueba de convergencia; si ésta es satisfactoria, detener el procedimiento. Si no se logra la convergencia, sustituir la solución lineal de las variables de cada corriente (obtenidas en el paso 7) y regresar al paso 4, para aplicar nuevamente los modelos rigurosos.

Obviamente, el éxito de este procedimiento depende de que tan bien los modelos lineales representen a los modelos rigurosos y en los métodos de solución empleados en los pasos 5 y 7.

La aplicación de este algoritmo a la simulación de diversos procesos ha mostrado casos de aproximación adecuada, sin embargo, en muchos casos exhibe un comportamiento inestable, lo que se debe en gran parte a la mala aproximación local del sistema de ecuaciones lineales, puesto que este comportamiento es de espe-

rarse al emplear modelos de fracciones de separación en procesos iterativos. Además, el despreciar las interacciones entre componentes, puesto que los elementos fuera de la diagonal principal de la matriz de un modelo lineal son tomados como cero, debería reducir una influencia negativa sobre el comportamiento de la convergencia. En comparación, el método de Wegstein (1958), aplicado ampliamente en el enfoque Modular Secuencial, también desprecia las interacciones entre los componentes, pero emplea una aproximación de tipo secante sobre el último punto de iteración y por tanto exhibe mejores propiedades de convergencia.

A pesar de lo anterior, el método de Rosen posee varias ventajas. En virtud de la estructura diagonal del modelo, es posible resolver las ecuaciones de simulación componente por componente, lo cual reduce en buena medida los requerimientos de almacenaje en memoria. Una ventaja adicional es que los modelos lineales no necesitan ser escritos por anticipado, ya que éstos son generados mediante cálculos numéricos simples.

La inestabilidad del procedimiento anterior ha conducido a promover el desarrollo de modelos lineales susceptibles de ser resueltos analíticamente. Umeda (1972), empleó los modelos lineales desarrollados por Nishimura (1967), demostrando que los modelos lineales derivados analíticamente tienen mucho mejores propiedades de convergencia. Los modelos analíticos tienen

la ventaja de no requerir el cálculo de derivadas, además de que estos modelos son únicos para cada tipo de equipo y son funciones de las variables de las corrientes de entrada y de los parámetros de ese equipo en particular, lo que permite incorporar una mayor cantidad de información.

Desafortunadamente, la principal desventaja de un algoritmo que emplea modelos lineales analíticos es precisamente la inexistencia de modelos adecuados para muchos de los equipos usualmente empleados. Por lo tanto, cada vez que un nuevo módulo unitario es agregado al sistema de simulación, una considerable cantidad de tiempo se debe emplear en encontrar, si ésto es posible, el modelo analítico lineal apropiado. Esto resta flexibilidad al sistema y, desde el punto de vista del usuario, es considerado como no aplicable a condiciones variables en la simulación.

#### 4.2.2. MODELOS LINEALES DE TIPO GRADIENTE.

Ravicz y Norman (1964) y Naphtali (1964), propusieron por vez primera un modelo lineal de tipo gradiente, el cual tiene su fundamento en el método Newton-Raphson para resolver ecuaciones no lineales. La estructura parte de la ecuación 4.1, pero

ahora cada elemento  $a_{kj}$  de la matriz de coeficientes, es la derivada parcial de la  $k$ -ésima variable de salida,  $y_k$ , con respecto a la  $j$ -ésima variable de entrada,  $x_j$ , por lo tanto:

$$a_{kj} = \frac{\partial y_k}{\partial x_j} \quad (4.4)$$

Este modelo exhibió una adecuada convergencia, a pesar de requerir expresiones analíticas para calcular los coeficientes  $A$  y  $b$ ; sin embargo, al aplicarse a procesos complejos, su eficiencia se vió disminuída drásticamente. Tal vez, la mayor contribución de este modelo haya sido la de dar una nueva orientación al uso de modelos lineales, mediante el uso de gradientes.

Una variación del anterior procedimiento consiste en evaluar numéricamente los elementos de la matriz de coeficientes, mediante diferencias finitas, empleando:

$$a_{kj} = \frac{\partial y_k}{\partial x_j} = \frac{\hat{y}_k - y_k}{\hat{x}_j - x_j} \quad (4.5)$$

Esta aproximación requiere perturbar la variable de entrada,  $x_j$ , a un valor  $\hat{x}_j$  y múltiples ejecuciones de las ecuaciones descriptivas rigurosas. Si sólo un elemento de la variable de entrada es perturbada cada vez al calcular los coeficientes

lineales, como en

$$\frac{\partial y_k}{\partial x_j} = \left[ \frac{\Delta y_k}{\Delta x_j} \right]_{x_k \neq j} \quad (4.6)$$

entonces, la matriz de coeficientes lineales, será una matriz llena.

En este caso, el conjunto de ecuaciones rigurosas debe ser ejecutado tantas veces como sea el número de variables de entrada a esa unidad.

Las matrices completas (llenas) se presentan al describir unidades de proceso con fuertes interacciones entre componentes, por ejemplo, un reactor químico, y es adecuado emplear la perturbación numérica total.

Existen, sin embargo, unidades de proceso con una pobre o nula interacción entre componentes (mezcladores, intercambiadores de calor, unidades de destilación, unidades de operación en flash, bombas, etc.), para las cuales es satisfactorio aproximar la matriz de coeficientes en forma de una matriz diagonal, utilizando perturbación diagonal sobre todas las variables de entrada al mismo tiempo, y por lo tanto las ecuaciones rigurosas serán ejecutadas sólo una vez.

Los elementos de la matriz de coeficientes lineales evaluados mediante perturbación diagonal se pueden expresar así:

$$\frac{\partial y_k}{\partial x_j} = \left[ \frac{\Delta y_k}{\Delta x_j} \right]_{x_j=k} \quad (4.7)$$

en donde:

$$\Delta x_j = (1 + e) x_j \quad (4.8)$$

siendo e un factor de perturbación.

De la experiencia obtenida en el desarrollo de los procedimientos anteriormente descritos, se ha hecho notar que un algoritmo que utilice modelos lineales aproximados, deberá tener estas características:

A) Los coeficientes de los modelos unitarios aproximados deberán ser evaluados numéricamente, mientras no esten disponibles modelos analíticos adecuados a cada equipo. Esto permitirá al usuario agregar módulos unitarios rigurosos, mientras que el programa ejecutivo desarrollará los modelos aproximados, ya sean lineales o de algún otro tipo.

B) La evaluación numérica de los modelos aproximados no deberá requerir cálculos excesivos, en comparación con los modelos rigurosos.

C) El sistema deberá ser capaz de aceptar la adición de las especificaciones de diseño requeridas por el proceso.

D) El procedimiento deberá exhibir un comportamiento estable en la convergencia.

#### 4.2.3. MODELOS LINEALES DE TIPO GRADIENTE MODIFICADOS.

##### 4.2.3.1. Modelo diferencial de fracciones de separación.

Mahalec, Kluzik y Evans (1979), propusieron una variación al modelo de Rosen(1962), a la cual denominaron Modelo Diferencial de Fracciones de separación, cuya relación linearizada es:

$$y_k - A_{kj} \cdot \Delta x_j = 0 \quad (4.9)$$

y que mostró ser equivalente al algoritmo de Wegstein (1958), para un proceso con un ciclo de recirculación simple, y que en sistemas con múltiples recirculaciones, muestra una mejor convergencia que el algoritmo citado.

Dentro de este procedimiento se puede considerar el caso para el cual algunos de los parámetros de equipo se convierten en variables de entrada; por ejemplo, al manipular la relación de reflujo en un módulo de destilación a fin de alcanzar cierta especificación, corriente abajo en el proceso.

Puesto que los parámetros de equipo manipulados,  $w_j$ , pueden ser cualquier parámetro unitario, la única forma de incluirlos es con la evaluación numérica de la columna de la matriz correspondiente a éstos. En otras palabras, la matriz unitaria,  $A$ , ya no tendrá una estructura diagonal simple, como la mostrada en la Figura 4.5 a, adoptando ahora una estructura diagonal con

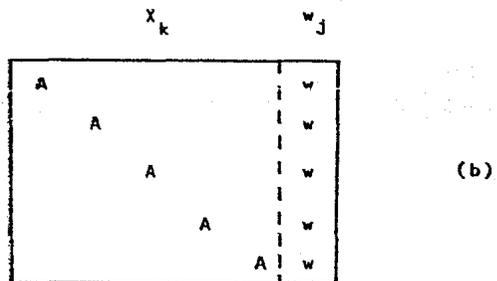
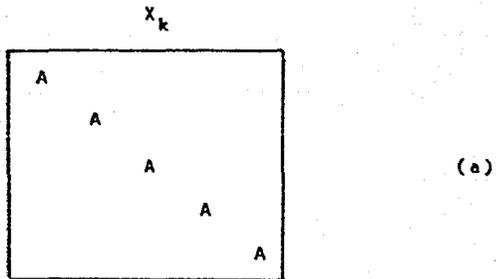


FIGURA 4.5. INFLUENCIA DE UN PARAMETRO DE EQUIPO MANIPULADO SOBRE LA MATRIZ UNITARIA.

banda, según se puede observar en la Figura 4.5 b. Por lo tanto, si existen  $w$  parámetros de equipo manipulados, se requerirán  $w + 2$  evaluaciones del módulo unitario, a fin de generar la matriz unitaria  $A$ .

Otra valiosa aportación de este procedimiento radica en la posibilidad de manejar eficientemente las especificaciones de diseño. En el desarrollo propuesto por Mahalec y sus colaboradores, los balances unitarios han sido intencionalmente escritos en tal forma que las variables de salida son calculadas en forma explícita. Esto permite imponer especificaciones de diseño sobre las variables de cierta corriente.

Por ejemplo, se podría desear que el flujo de un cierto componente,  $t$ , en la corriente  $y_k$  sea alcanzado mediante la manipulación del factor de conversión,  $K$ , en un reactor, lo que se puede expresar así:

$$f(x_t, K) = x_t - x_t^{(0)} = 0 \quad (4.10)$$

en donde,  $K$  es el parámetro manipulado,  $x_t$  es la variable de prueba la cual adquiere el valor deseado y  $x_t^{(0)}$  es el valor de  $x_t$  en la iteración inmediata anterior.

La ecuación 4.10 es agregada al sistema de ecuaciones, por lo tanto, el número de variables del sistema aumenta en una, ya que el factor de conversión,  $K$ , se convierte en una variable,

y esta ecuación específica que durante las iteraciones:

A) El lado derecho de la ecuación sea calculado así:

$$f = x_t - \overset{(o)}{x}_t \quad (4.11)$$

B) La relación entre el parámetro manipulado, en este caso  $K$ , y la variable de prueba,  $x_t$ , es implícitamente definida a través de la estructura del proceso.

Esta última observación es particularmente significativa, puesto que las ecuaciones linealizadas representan la totalidad de la estructura del sistema, sólo es necesario evaluar la influencia del parámetro manipulado sobre sus propias corrientes unitarias de salida y esta influencia se toma en cuenta durante la solución del sistema de ecuaciones lineales, sin necesidad de establecer una relación explícita entre las variables manipuladas y las variables de prueba. Esto representa una ventaja significativa sobre el enfoque Modular Secuencial, en donde una relación explícita debe ser evaluada a fin de hacer converger las especificaciones de diseño.

Debe notarse que el lado derecho de las ecuaciones 4.10 y 4.11, es evaluado como una diferencia entre el valor deseado y el valor de la última iteración de la función de diseño, mientras que el lado izquierdo de esas ecuaciones sólo contiene las deri-

vadas parciales de la ecuación de diseño con respecto a las variables de prueba.

El algoritmo Diferencial de Fracciones de Separación es del tipo Newton, con la matriz de proceso evaluada por perturbación diagonal, calculando el tamaño de paso en cada iteración a partir del sistema de ecuaciones linealizadas, cuya convergencia se promueve de la siguiente manera:

- 1.- Suponer valores iniciales para las variables de proceso.
- 2.- Probar la convergencia. Si no se logra, continuar en el paso 3. Si se logra la convergencia, pasar a 12.
- 3.- Resolver el sistema de ecuaciones linealizadas:

$$C \cdot \Delta x = -f \quad (4.12)$$

en donde,  $C$  es la matriz aproximada del sistema.

- 4.- Calcular la norma Euclideana de  $f(x + \Delta x)$ ; si esta desigualdad se cumple

$$\|f(x + \Delta x)\| < \|f(x)\| \quad (4.13)$$

continuar en el paso 11; si no se cumple, pasar a 5.

- 5.- Hacer:

$$f_0 = \|f(x)\| \quad (4.14 a)$$

$$f_1 = \|f(x + \Delta x)\| \quad (4.14 b)$$

$$f_2 = f(x + 0.5 \Delta x) \quad (4.14 \text{ c})$$

6.- Ajustar una función cuadrática  $f(\Delta x)$ , mediante  $f_0$ ,  $f_1$  y  $f_2$ .

7.- Ajustar el parámetro  $S$  de modo que la función cuadrática adquiera un valor extremo, por ejemplo:

$$\frac{\partial g(x + S \Delta x)}{\partial S} = 0 \quad (4.15)$$

8.- Si esta desigualdad se cumple

$$\|f(x + S \Delta x)\| > \|f(x)\| \quad (4.16)$$

pasar a 10, en caso contrario proseguir en 9.

9.- Hacer:

$$x = x + S \Delta x \quad (4.17)$$

y pasar a 2.

10.- Abandonar la última iteración obtenida del sistema linealizado. Realizar una sustitución directa mediante un recorrido secuencial a través del proceso. Concluido el recorrido pasar a 2.

11.- Hacer:

$$x = x + \Delta x \quad (4.18)$$

y pasar a 2.

## 12.- Detener el proceso iterativo.

El algoritmo promueve la reducción de la norma de los errores en cada iteración. Cuando esto no es posible, un recorrido secuencial a través del proceso es usado con el objeto de desplazar el punto de iteración erróneo.

En forma general, al utilizar modelos aproximados lineales, ya sean simples o de tipo gradiente, se sigue un procedimiento similar al mostrado en la Figura 4.6.

### 4.2.3.2. Modelo en base a funciones de residuos.

Esta es una variación del modelo lineal de tipo gradiente y fué propuesta por Jirapongphan (1980), en términos de resolver las relaciones de entrada y salida a cada módulo mediante el método de Newton.

Para este propósito, la variable calculada por el procedimiento riguroso será escrita como una función no lineal de las variables de entrada:

$$y = f(x) \quad (4.19)$$

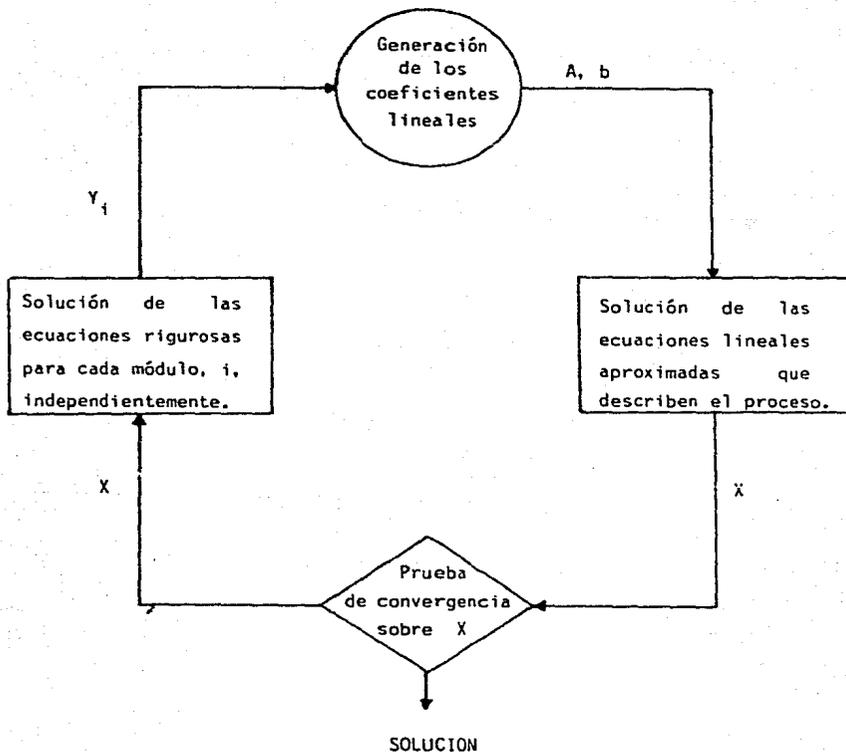


FIGURA 4. 6. ESQUEMA DE CALCULO PARA EL ENFOQUE MODULAR SIMULTANEO USANDO MODELOS APROXIMADOS LINEALES.

y la relación entre entradas y salidas puede ser expresada en términos de esta función de residuos:

$$q(y,x) = y - f(x) \quad (4.20)$$

y al encontrarse su solución, el valor de la función de residuo será cero, o un valor muy cercano a él.

El problema de simulación es entonces equivalente a encontrar los ceros de las funciones de residuos para cada unidad del proceso, satisfaciendo las ecuaciones de conexión y las restricciones de diseño en cada corriente.

Si,  $Q$ , representa el conjunto de todas las funciones de residuos junto con las ecuaciones de conexión y si  $U$  representa el conjunto de todas las variables de entrada y salida de cada unidad en el proceso, las cuales son aludidas por las ecuaciones de conexión, entonces el problema de simulación implica resolver:

$$Q(U) = 0 \quad (4.21)$$

$$H(U) = 0 \quad (4.22)$$

siendo  $H$  el vector de restricciones de diseño.

La solución de las ecuaciones 4.21 y 4.22 por el método Newton-Raphson requiere la matriz Jacobiana de  $Q(U)$  y  $H(U)$  en cada iteración.

La matriz Jacobiana de las funciones de residuo para la unidad  $i$ ,  $q_i(y_i, x_i)$ , es:

$$\nabla q_i^T = \begin{bmatrix} \frac{\partial q_i}{\partial y_i} \\ \frac{\partial q_i}{\partial x_i} \end{bmatrix}^T = \begin{bmatrix} I \\ - \frac{\partial f_i}{\partial x_i} \end{bmatrix}^T \quad (4.23)$$

El término  $\partial f_i / \partial x_i$  puede ser calculado numéricamente mediante la perturbación de las variables de entrada y reejecutando las ecuaciones rigurosas como:

$$\frac{\partial f_i}{\partial x_i} = \frac{\Delta y_i}{\Delta x_i} \quad (4.24)$$

Sin embargo, la ecuación 4.24 es igual a la ecuación 4.5, la cual es la expresión para la matriz coeficiente del modelo lineal de tipo gradiente.

De lo anterior se puede concluir que la matriz Jacobiana para  $Q(U)$  es equivalente a la matriz lineal de proceso, por lo que el algoritmo Modular Simultáneo con un modelo lineal de tipo gradiente es equivalente a resolver las funciones de residuos entre las entradas y salidas mediante el método Newton.

El esquema de cálculo para este procedimiento se ilustra en la Figura 4.7, y el algoritmo propuesto por Jirapongphan es:

### 1.- Inicialización.

- 1.1.- Ajustar los valores de las corrientes de recirculación a cero y suponer los valores para los parámetros de equipo manipulados y las variables de decisión.
- 1.2.- Realizar un recorrido secuencial sobre las ecuaciones rigurosas del proceso.
- 1.3.- Ajustar los valores de las corrientes de recirculación.
- 1.4.- Usar los valores obtenidos para las corrientes de recirculación como valores iniciales para las corrientes de entrada,  $x$ , para calcular la submatriz Jacobiana de la función de residuos para cada unidad  $i$ ,  $\nabla q_i^T$  secuencialmente a través del proceso, mediante perturbación numérica (ver paso 3).
- 1.5.- Incluir las variables de salida y los parámetros de equipo de cada módulo, excepto de los mezcladores, dentro del vector de variables apropiado, con el objeto de inicializar.
- 1.6.- Ajustar los valores de las funciones de residuos a cero excepto para los módulos de mezclado, en donde estos valores serán diferentes entre los valores calculados

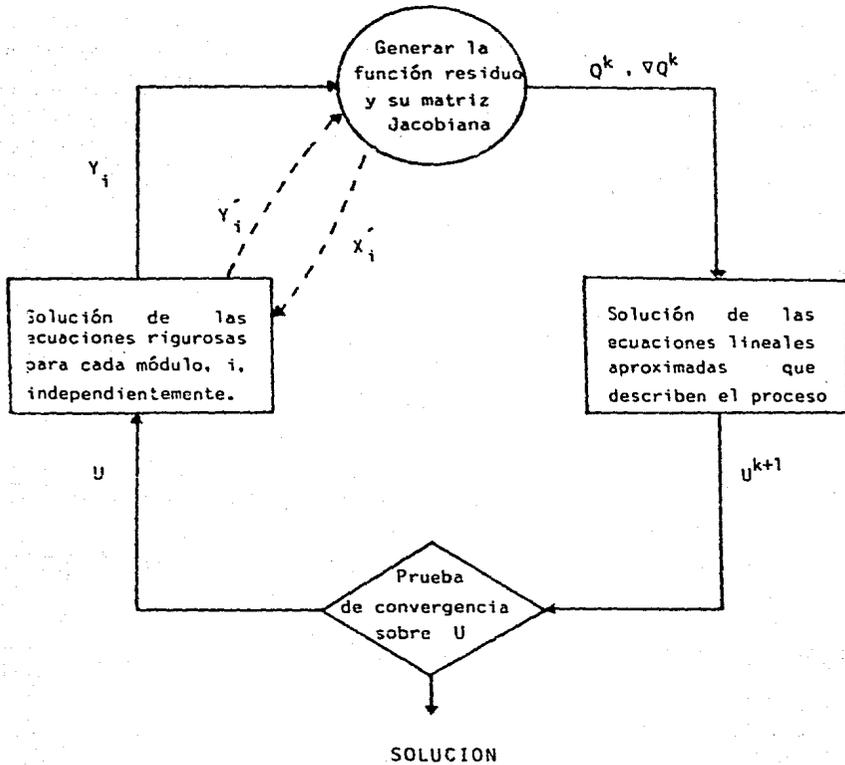


FIGURA 4.7. ESQUEMA DE CALCULO PARA EL ENFOQUE MODULAR SIMULTANEO BASADO EN LA EVALUACION DE LOS RESIDUOS.

de sus corrientes de salida y los valores iniciales para esas corrientes.

## 2.- Cálculos lineales.

2.1.- Calcular  $H$  y  $\nabla H$ .

2.2.- Obtener un nuevo estimado para  $U$ .

2.3.- Probar la convergencia para  $U$ . Si se cumple, suspender los cálculos, de lo contrario pasar a 3.

## 3.- Ajuste de variables y evaluación de funciones de residuos.

3.1.- Incluir los nuevos estimados de las corrientes de entrada y de los parámetros de equipo en cada módulo.

3.2.- Ejecutar las ecuaciones rigurosas en cada módulo.

3.3.- Evaluar la función de residuos,  $q(y, x)$ .

## 4.- Cálculo de la matriz Jacobiana.

4.1.- Para cada módulo en el proceso:

4.1.1.- Ejecutar las ecuaciones rigurosas del módulo con  $x_j = (1 + e) x_j$ , en donde  $e$  es el factor de perturbación numérica.

4.1.2.- Calcular los elementos de la matriz Jacobiana con respecto a  $x_j$  :

$$\left[ \begin{array}{c} \Delta f_k \\ \Delta x_j \end{array} \right]_{x_{m \neq j}} = \frac{f_k(x_j^0) - f_m(x_j)}{x_j^0 - x_j} \quad (4.25)$$

en donde,  $x_j^0$ , es el valor de la variable en la iteración previa.

4.2.- Incluir las matrices Jacobianas unitarias dentro de la matriz Jacobiana del proceso, la cual será una matriz dispersa del tipo triangular con borde.

4.3.- Regresar al paso 2.1. e iniciar una nueva iteración.

#### 4.2.3.3. Modelo de tipo quasi-Newton.

El algoritmo con perturbación diagonal (sección 4.2.3.1), mostró que la convergencia de los balances en el proceso es posible con el uso de una matriz Jacobiana aproximada del sistema y, puesto que cada iteración mediante perturbación diagonal requiere al menos dos evaluaciones del proceso, la rapidez de convergencia

puede ser incrementada en el mismo factor si una matriz Jacobiana aproximada fuera calculada a través de una simple evaluación del proceso.

Mahalec, Kluzik y Evans (1979), propusieron un algoritmo de tipo quasi-Newton, aplicando el procedimiento de Broyden (1965), para la solución de las ecuaciones no lineales. Este algoritmo explota el hecho de que la matriz lineal de proceso es en esencia una matriz Jacobiana que presenta una estructura dispersa del tipo triangular con banda, como se muestra en la figura 4.5, por lo tanto, este método aproxima los coeficientes lineales de la matriz, sin realizar perturbación numérica, lo cual reduce el número de ejecuciones de los módulos rigurosos en cada iteración.

Se debe notar que el algoritmo de tipo quasi-Newton puede aproximar tanto una matriz Jacobiana, como su inversa. Puesto que la matriz del sistema es dispersa, se elige trabajar con la aproximación del Jacobiano, en virtud de que la inversa de una matriz dispersa, no es dispersa.

En cada paso de la iteración, el siguiente conjunto de ecuaciones deberá ser resuelto:

$$C \cdot \Delta Z = -f(Z) \quad (4.26)$$

en donde:

C = matriz Jacobiana aproximada del sistema.

$\Delta Z$  = incremento en las variables del sistema.

$f(Z)$  = función de errores.

La matriz Jacobiana aproximada es calculada en cada iteración, de acuerdo a la fórmula:

$$C = C^0 + \frac{[f - (1 - t^0) f^0] \Delta Z^0{}^T}{t^0 \cdot \Delta Z^0{}^T \cdot \Delta Z^0} \quad (4.27)$$

en donde, los valores obtenidos de la iteración inmediata anterior son designados con el subíndice,  $^0$ .

El cálculo de  $C$ , tal y como fué definido, no es disperso y por lo tanto, no puede ser aplicado directamente al algoritmo Modular Simultáneo. Schubert (1970), propuso un método mediante el cual la matriz  $C$  conserva la dispersión original a través de los cálculos, de acuerdo a este procedimiento:

1.- Calcular la matriz  $C$ , pero imponiendo un cero en cada entrada  $j$  de  $\Delta Z^0$ , para cada elemento constante conocido,  $j$ , de la matriz Jacobiana.

2.- Hacer cero todas las entradas en la fila de la matriz obtenida por la ecuación 4.27, que corresponda a las constantes de la matriz Jacobiana original. Esto implica que sólo entradas diferentes de cero y no constantes serán calculadas.

3.- Agregar la fila de la matriz dispersa calculada, a la

fila correspondiente en la matriz Jacobiana original.

Mahalec y sus colaboradores probaron este procedimiento aplicándolo al problema de Cavett (1963), comparándolo con su algoritmo de tipo Newton y con el algoritmo correspondiente al enfoque Modular Secuencial. En los tres casos se probó con varias opciones en la forma de inicializar la matriz Jacobiana (perturbación total, perturbación diagonal, una combinación de las dos anteriores y a partir de matrices identidad para algunas o todas las submatrices unitarias).

Se concluyó que el algoritmo de tipo quasi-Newton es más rápido en convergencia, que el mejor algoritmo Modular Secuencial disponible en ese tiempo; además, como era de esperarse, entre mejores fueron los valores iniciales, la convergencia se logró en forma más rápida.

De estas experiencias se obtuvo una estrategia con el objeto de inicializar el algoritmo Modular Simultáneo:

- 1.- Suponer valores iniciales para el conjunto de corte no redundante, obtenido por el procedimiento Modular Secuencial.

- 2.- Desarrollar una o más iteraciones mediante sustitución directa en el algoritmo Modular Secuencial, lo que proporcionará los valores iniciales para todas las corrientes del sistema.

3.- Inicializar la matriz Jacobiana mediante perturbación diagonal para todos los equipos, excepto para los reactores, cuales serán evaluados mediante perturbación numérica total.

4.- Se continúan las iteraciones usando el algoritmo quasi-Newton para matrices dispersas, hasta lograr la convergencia.

El uso de modelos lineales, a pesar de sus limitaciones, es muy importante, sobre todo en las actuales perspectivas de incluir la Optimización como otra tarea de los simuladores, ya que la solución detallada al final del proceso de cálculo se obtiene precisamente mediante modelos lineales.

El tipo más completo de modelos aproximados lineales, al menos los que más refinamientos han tenido a la fecha, son los modelos de tipo gradiente. La idea general que motiva el uso de estos modelos es que las derivadas de las variables de salida con respecto a las variables de entrada, predichas en cada bloque por el modelo lineal, son las mismas que las predichas por los modelos rigurosos.

Usualmente los elementos de la matriz de coeficientes necesitan ser evaluados mediante perturbación numérica, sin embargo, se presentan dos desventajas importantes:

Primero, el modelo no tiene ningún significado físico y en puntos diferentes a los puntos de base, el balance de materia y energía sobre una unidad, puede no ser satisfecho del todo; además, la extrapolación hacia puntos diferentes a los puntos de base, usualmente es muy pobre, lo que origina una convergencia muy lenta del ciclo externo de cálculo.

La segunda desventaja se refiere a su alto consumo de tiempo de cómputo, necesario para calcular las derivadas numéricas durante cada iteración sobre el ciclo externo.

La eficiencia de este tipo de modelos se puede incrementar sustituyendo algunas de las ecuaciones de tipo gradiente, por otras ecuaciones lineales que tengan significado físico dentro de la unidad modelada. Por ejemplo, las ecuaciones del balance de materia total son globales, por lo tanto, la introducción de éstas dentro del modelo lineal, garantizan que todos los balances de materia para cada unidad, serán satisfechos en cualquier punto. Esta inclusión añade un punto de vista de ingeniería, a lo que de otra manera sería un modelo del tipo "caja negra".

Tomando como ejemplo un modelo para operaciones de flash a dos fases, especificando la presión,  $P$ , y la carga térmica,  $Q$ , siendo las variables de estado los flujos por componente en la corriente de entrada,  $F$ ; la temperatura y presión,  $T_f$  y  $P_f$ , en

la corriente de entrada y la presión y carga térmica de operación,  $P_s$  y  $Q_s$ , que son especificadas. Las variables de salida son aquellas asociadas con el líquido y vapor de las corrientes de salida: flujo por componente L y V, temperaturas  $T_L$  y  $T_V$ , y presiones  $P_L$  y  $P_V$ . El modelo lineal de tipo gradiente es de la forma:

$$y = Ax + b \quad (4.28)$$

lo que significa:

$$\begin{bmatrix} L \\ P_L \\ T_L \\ V \\ P_V \\ T_V \end{bmatrix} = [A] \begin{bmatrix} F \\ P_f \\ T_f \\ P_s \\ Q_s \end{bmatrix} + b \quad (4.29)$$

Para un sistema de  $n$  componentes, la matriz de coeficientes  $A$  contiene  $(2n + 4)(n + 4)$  derivadas parciales, que necesitan ser calculadas numéricamente.

Este modelo puede ser mejorado al sustituir algunas de las ecuaciones de tipo gradiente por ecuaciones lineales rigurosas; por ejemplo, las ecuaciones del balance global de materia por componente pueden ser incluidas en el modelo lineal simple, junto con las relaciones que especifican que ambas corrientes de salida

tienen la misma presión y temperatura, obteniéndose :

$$\begin{bmatrix} L \\ T_L \end{bmatrix} = [A'] \begin{bmatrix} F \\ P_f \\ T_f \\ P_s \\ Q_s \end{bmatrix} + b \quad (4.30)$$

$$\begin{aligned} V &= F - L \\ P_L &= P_s \\ P_V &= P_L \\ T_V &= T_L \end{aligned} \quad (4.31)$$

A pesar de que este modelo es más simple y preciso que el modelo de tipo gradiente original, la nueva matriz de coeficientes,  $A'$ , sólo contiene  $(n + 1) \cdot (n + 4)$  elementos.

Para la simulación Modular Simultánea, que mantiene la entalpía de cada corriente en el ciclo interno de cálculo, el balance global de energía puede ser incluido también como parte del modelo lineal, en esta forma:

$$H_f - H_L - H_V - Q_s = 0 \quad (4.32)$$

en donde,  $H$  representa el flujo total de entalpía para una determinada corriente de proceso.

#### 4.3. MODELOS NO LINEALES.

Del anterior desarrollo, mediante una aproximación de las ecuaciones descriptivas de un módulo mediante relaciones lineales entre las variables de entrada y salida, se pueden resaltar dos propiedades fundamentales del enfoque Modular Simultáneo.

La primera, que en realidad fué la motivación principal para su primera implementación, es que transforma la dificultad de manejar ecuaciones no lineales, en un cierto número de subsistemas con ecuaciones lineales, los cuales pueden ser resueltos uno a uno, en una forma más eficiente.

La segunda, es la reducción en el número de ecuaciones y variables que deben ser resueltas simultáneamente, puesto que las variables internas de los módulos han sido eliminadas de las ecuaciones lineales aproximadas del proceso, y es así como el número de variables a ser evaluadas simultáneamente es reducido al conjunto de variables de entrada y salida.

Sin embargo, las ecuaciones rigurosas que describen el proceso pueden ser altamente no lineales, por lo que una aproximación lineal será muy pobre, cuando la solución de las ecuaciones aproximadas se encuentra lejos del punto en donde los coeficientes lineales fueron generados.

Por esta razón es deseable incluir un cierto grado de no linealidad en las ecuaciones aproximadas, con el objeto de incrementar el rango de extrapolación de estas ecuaciones.

Si bien es cierto, al incluir componentes no lineales en las ecuaciones aproximadas se incrementa su confiabilidad, también su solución resulta más difícil, en comparación con las ecuaciones lineales. A pesar de esto, el efecto global al usar aproximaciones no lineales aumenta la eficiencia del enfoque Modular Simultáneo, en base a la premisa que indica que el número de iteraciones se verá disminuido en virtud del amplio rango de cobertura de las ecuaciones aproximadas no lineales y porque existe una reducción en el número de ejecuciones de los modelos rigurosos, los cuales tienen alto consumo de tiempo de cómputo.

Adicionalmente, puesto que aún el más eficiente de los métodos lineales requiere de algunas iteraciones sobre el problema de simulación no lineal, cualquier ganancia en la eficiencia de cómputo sólo podrá lograrse si se dispone de mayores datos que no se limiten únicamente a la información de tipo gradiente del sistema. Por lo tanto, no es extraño que los más recientes desarrollos de la simulación modular simultánea, esten enfocados a la solución de los balances de masa y energía mediante aproximaciones de tipo no lineal.

Para enfatizar la propiedad de reducir el número de varia-

bles y ecuaciones del enfoque Modular Simultáneo, el término Modelo Reducido será empleado para designar a las ecuaciones descriptivas aproximadas de carácter no lineal.

#### 4.3.1. MODELOS ANALITICOS REDUCIDOS.

Jirapongphan (1980), propuso un algoritmo en base a un modelo reducido no lineal, el cual contiene coeficientes no lineales que pueden ser calculados analíticamente, sin utilizar perturbación numérica, de aquí que se le conozca como Modelo Analítico Reducido (MAR).

##### 4.3.1.1. Características del Modelo Analítico Reducido.

Los coeficientes del MAR son generalizados a un coeficiente  $\alpha$  y no son limitados a las matrices lineales A y b.

La forma funcional del MAR para el módulo i, es:

$$Q_i(Z_1, Z_2, V_R; \alpha) = 0 \quad (4.33)$$

y la expresión para evaluar el coeficiente del MAR, es:

$$\alpha = Q_i (Z_1, Z_2, V_R) \quad (4.34)$$

y agrupando variables:

$$\bar{Q}_i (U, \alpha^k) = 0 \quad (4.35)$$

$$\alpha = (U) \quad (4.36)$$

en donde:

$Z_1$  = vector de variables de salida (corrientes de salida, variables internas y de retención).

$Z_2$  = vector de variables de entrada (corrientes de entrada y parámetros de equipo).

$U$  = vector de variables agrupadas ( $Z_1, Z_2, V$ ).

$V$  = vector de variables internas.

$V_R$  = vector de variables internas del MAR.

$\alpha$  = vector de coeficientes del MAR.

$k$  = contador de iteraciones.

El modelo reducido es aplicado en los cálculos del ciclo interno mostrado en la Figura 4.8.

Al aplicar el MAR se puede lograr una reducción en el número de ecuaciones y variables en un factor de  $4n_c + 2$ , el cual es muy significativo cuando el número de componentes,  $n_c$ , es

muy grande. Una reducción mayor puede ser alcanzada cuando el módulo unitario incluye cálculos en etapas múltiples, en donde las variables internas en las ecuaciones descriptivas rigurosas incluyen composiciones, flujos y perfiles de presión y temperatura de la unidad.

Otra ventaja importante al utilizar el Modelo Analítico Reducido es que sus expresiones son independientes de las correlaciones para propiedades termofísicas, las cuales han sido tomadas en cuenta, en forma implícita y compacta, en las expresiones de los coeficientes del MAR, de aquí que las expresiones globales del modelo reducido (incluyendo su matriz Jacobiana) continúan siendo las mismas, independientemente de las correlaciones empleadas. Por otro lado, las derivadas de las ecuaciones del modelo riguroso (incluyendo su matriz Jacobiana), dependen de las correlaciones para propiedades termofísicas empleadas y, en especial, las ecuaciones rigurosas referentes a los coeficientes de fugacidad y entalpía, pueden ser drásticamente diferentes.

Treviño-Lozano (1985), trabajando en el desarrollo de nuevos modelos analíticos reducidos y en base al análisis numérico, presentó diez características deseables de estos modelos, agrupadas en tres categorías:

A) Requerimientos básicos de un modelo reducido.

1) El número de ecuaciones en un modelo reducido deberá ser mucho menor que el número de ecuaciones del modelo riguroso.

2) Las ecuaciones del modelo reducido deberán tener estabilidad en su convergencia.

3) El número de ecuaciones y variables del modelo reducido de cada unidad, debe conservar los grados de libertad del modelo riguroso.

4) Cada modelo reducido debe incluir un conjunto de coeficientes, cuyas expresiones sean únicas para cada tipo de unidad; expresiones que sean analíticas y que incluyan variables de entrada y salida, variables internas y parámetros de equipo.

5) Las ecuaciones del modelo reducido deben generar la misma solución que el modelo riguroso, en el punto en donde sus coeficientes fueron generados.

6) Las ecuaciones del modelo reducido y la matriz Jacobiana asociada a ellas, debe ser evaluada analíticamente, sin requerir otros cálculos auxiliares, tales como propiedades termodinámicas, y sin recurrir a cálculos iterativos.

Los requerimientos 1, 2 y 6, garantizan que los cálculos ne-

cesarios para converger el ciclo interno, deben ser relativamente simples, por lo tanto, algunos métodos orientados hacia las ecuaciones podrán ser usados para resolver el modelo reducido.

Puesto que se desea un sistema de ecuaciones de simulación más simple que el generado por el modelo riguroso, los cálculos en el ciclo interno pueden ser simplificados si las ecuaciones reducidas y sus derivadas son calculadas analíticamente.

El requerimiento 3 asegura que el problema definido en el ciclo interno es equivalente al problema original en términos de las variables que son calculadas al resolver el problema de simulación. El requerimiento 4 se refiere a la adecuada selección de las variables que deben aparecer en el modelo reducido; selección que es particularmente importante al manejar especificaciones de diseño, en donde algunos parámetros del modelo, usualmente especificados, ahora pueden convertirse en variables.

El requerimiento 5 garantiza que el procedimiento modular simultáneo converge hacia la solución de las ecuaciones de simulación rigurosas.

Se debe notar que los anteriores requerimientos no imponen ninguna restricción a los modelos actualmente en uso, es decir, estas características aplican tanto a los modelos aproximados lineales, como a los modelos reducidos no lineales.

**B) Requerimientos para los modelos reducidos no lineales.**

7) Las ecuaciones de un modelo reducido no lineal, no deben ser relaciones empíricas, sino funciones ampliamente representativas del tipo de unidad que esta siendo modelada.

8) Los balances de materia y energía de cada unidad deberán ser satisfechos totalmente, aún para condiciones alejadas del punto de partida básico.

9) Las expresiones para los coeficientes del modelo reducido deberán ser analíticas y susceptibles de incluir cualquier variable del modelo riguroso.

Los requerimientos 7 y 8 indican que los modelos reducidos deben hacer uso del conocimiento que se tenga del proceso físico que tiene lugar en cada operación unitaria, lo que amplía la posibilidad de incluir adecuadas aproximaciones de ingeniería de las ecuaciones rigurosas, con lo cual se logra la extrapolación de las ecuaciones reducidas, aún para situaciones alejadas del punto en donde los coeficientes reducidos fueron generados.

El requerimiento 9 intenta abandonar el uso de los modelos de tipo gradiente, que requieren el cálculo de derivadas numéricas, cuya obtención exige demasiado tiempo de cálculo y deben ser evitados, en lo posible.

C) Requerimiento de coincidencia en la solución.

10) La matriz Jacobiana de entrada-salida predicha por un modelo reducido debe ser la misma que la predicha por el modelo riguroso, para una misma unidad.

Los modelos de tipo gradiente siempre satisfacen este requerimiento porque los coeficientes del modelo son las derivadas de las variables de salida con respecto a las variables de entrada y la aplicación de este enunciado a otros tipos de modelos podría requerir el cálculo de derivadas numéricas a partir de los modelos rigurosos, a fin de introducir los factores de corrección adecuados en los modelos reducidos. Este procedimiento, sin embargo, introduce cálculos adicionales, contradiciendo lo establecido en el requerimiento 9.

La solución a este problema es desarrollar modelos reducidos no lineales que contengan suficiente significado físico como para dar una adecuada aproximación a la matriz Jacobiana de entrada-salida, usando parámetros que puedan ser evaluados analíticamente. Tales modelos pueden generar una solución subóptima, que usualmente es cercana a la solución óptima verdadera y si se requiere una mayor precisión, se pueden emplear modelos lineales en las últimas iteraciones del ciclo externo, a fin de alcanzar la solución real, usando la solución subóptima

obtenida con los modelos no lineales como un punto de partida cuando los modelos lineales son introducidos.

Jirapongphan (1980), hace una extensión de los Modelos Analíticos Reducidos, destinada a incluir la optimización como una tarea más del simulador Modular Simultáneo, logrando una implementación bastante innovadora y efectiva. Puesto que la Optimización de Procesos queda fuera de los alcances de este trabajo, se recomienda consultar la obra original de este autor para mayor información acerca de su algoritmo de optimización con Modelos Analíticos Reducidos.

#### 4.3.1.2. Algoritmo para la aplicación de los M A R.

El algoritmo para la aplicación del Modelo Analítico Reducido (MAR), está estructurado en el contexto de un enfoque con un ciclo interno y un ciclo externo de cálculo, los cuales son mostrados en la Figura 4.8.

Visto de esta manera, el algoritmo del MAR contiene cuatro partes principales:

Obtener los nuevos estimados de las  $x^k$  con respecto a las ecuaciones no lineales aproximadas, que describen el proceso.

Resolver las ecuaciones rigurosas que describen el proceso, para obtener los coeficientes del modelo reducido de cada unidad en forma independiente.

FIGURA 4. 8. DESCOMPOSICION DE CALCULO AL EMPLEAR MODELOS REDUCIDOS EN FORMA DE CICLO INTERNO Y CICLO EXTERNO.

A) Ejecución de las ecuaciones rigurosas y obtención de los coeficientes del modelo reducido,  $\alpha$ . Por lo tanto, estos son los cálculos del ciclo externo y los coeficientes  $\alpha$  serán las variables de ese ciclo.

B) Prueba de convergencia de los coeficientes del modelo reducido para el proceso completo.

C) Obtención de nuevos estimados para las variables  $x^m$ , contenidas en el vector  $U$ , a partir de las ecuaciones no lineales del modelo reducido. Estos son los cálculos del ciclo interno y las variables en  $U$ , son las variables de ese ciclo.

D) Prueba de convergencia para los nuevos estimados de  $U$ .

El algoritmo propuesto por Jirapongphan (1980), es:

### 1.- Inicialización.

1.1.- Ajustar los valores de las corrientes de recirculación a cero y suponer los valores de los parámetros de equipo que sean variables de decisión.

1.2.- Realizar un recorrido secuencial sobre las ecuaciones rigurosas del proceso.

- 1.3.- Ajustar los valores de las corrientes de recirculación.
  - 1.4.- Usar los valores obtenidos para las corrientes de recirculación como valores iniciales para las corrientes de entrada para cada unidad posterior a las corrientes de recirculación.
  - 1.5.- Ejecutar las ecuaciones rigurosas de cada unidad, situando las variables de salida y parámetros de equipo dentro del vector de variables del ciclo interno.
  - 1.6.- Evaluar los parámetros del modelo reducido de esa unidad y situarlos en los vectores de variables del ciclo externo.
  - 1.7.- Regresar al paso 1.5 para los cálculos de cada unidad de acuerdo a la secuencia de cálculo de cada ciclo de recirculación.
  - 1.8.- Cuando los parámetros del modelo reducido para todas las unidades en el proceso ya han sido obtenidos, continuar con los cálculos del ciclo interno, en el paso 2.
- 
- 2.- Cálculos del ciclo interno.
    - 2.1.- Recuperar la matriz Hessiana de la iteración del ciclo externo anterior y situar el contador de iteraciones

del ciclo interno en cero, es decir,  $m = 0$ .

2.2.- Calcular:  $\bar{Q}(U^m)$ ,  $\nabla \bar{Q}(U^m)$ ,  $H(U^m)$  y  $\nabla H(U^m)$ .

2.3.- Hacer:  $m = m + 1$ .

2.4.- Obtener nuevos estimados para cada  $x^m$  a partir del MAR

2.5.- Probar la convergencia de los valores de  $U$ ; si se cumple, ir al paso 3; de lo contrario, pasar a 2.7.

2.6.- Si el número máximo de iteraciones del ciclo interno,  $m$ , no ha excedido su valor límite, digamos 50, retornar a 2.2.

2.7.- Si se ha rebasado el valor máximo de  $m$ , pasar a 3.

3.- Prueba de convergencia para las variables del ciclo interno.

3.1.- Calcular la norma del vector de error,  $E$ , en donde cada elemento es definido así:

$$E_j = \frac{x_j - x_j^0}{x_j} ; \quad x_j \neq 0 \quad (4.37)$$

en donde  $x_j^0$ , es el valor de la variable del ciclo interno en la iteración previa.

3.2.- Probar si  $\|E\| \leq \epsilon$ , en donde  $\epsilon$  es una cierta to-

tolerancia, digamos  $10^{-4}$ ; si se cumple, detener el proceso de cálculo.

3.3.- Si no se cumple, continuar con los cálculos del ciclo externo en el paso 4.

4.- Cálculos en el ciclo externo.

4.1.- Para cada unidad en la secuencia de los ciclos de recirculación:

4.1.1.- Transferir los valores de las corrientes de entrada a la primera unidad en el ciclo de recirculación (corrientes de corte) del vector de variables del ciclo interno.

4.1.2.- Transferir las variables manipuladas y de decisión del vector de  $x$  a los parámetros de equipo de las unidades.

4.1.3.- Ejecutar las ecuaciones rigurosas de cada unidad.

4.1.4.- Evaluar el coeficiente del modelo reducido.

4.1.5.- Transferir las cantidades calculadas y las variables de las corrientes de productos del proceso, de la unidad que esta siendo evaluada al vector de  $x$ , como un estimado inicial para

los cálculos siguientes del ciclo interno.

4.1.6.- Si esta fué la última unidad en la secuencia de ciclos de recirculación, continuar con la prueba de convergencia para las variables del ciclo externo en el paso 5.

5.- Prueba de convergencia para las variables del ciclo externo.

5.1.- Calcular la norma del vector de error,  $E$ , en donde cada elemento es definido como:

$$E_j = \frac{\alpha_j - \alpha_j^0}{\alpha_j} ; \quad \alpha_j = 0 \quad (4.38)$$

en donde  $\alpha_j^0$  es el valor de las variables del ciclo externo en la iteración previa.

5.2.- Probar si  $\|E\| \leq \xi$ , en donde  $\xi$  es una cierta tolerancia, digamos  $10^{-4}$ ; si se cumple, detener el proceso de cálculo.

5.3.- Si no se cumple, continuar con los cálculos del ciclo interno en el paso 2.

El esquema de cálculo para el algoritmo de Simulación Modular Simultánea No Lineal usando modelos reducidos, se presenta en la Figura 4.9.

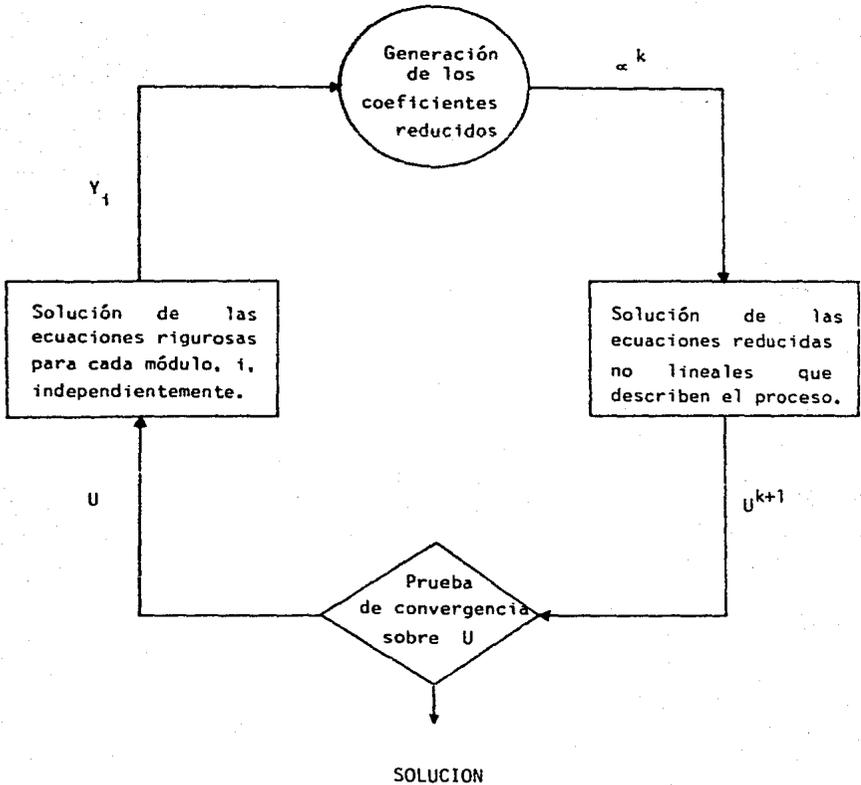


FIGURA 4.9. ESQUEMA DE CALCULO PARA EL ENFOQUE MODULAR SIMULTANEO USANDO MODELOS REDUCIDOS NO LINEALES.

#### 4.3.1.3. Modelos Reducidos para Operaciones Unitarias.

Jirapongphan (1980), desarrolló varios modelos reducidos para las unidades empleadas con más frecuencia en la simulación de procesos. Treviño-Lozano (1985), realizó algunos refinamientos a estos modelos y propuso otros que los complementan. En el Apéndice C de este trabajo se presenta una breve descripción de los Modelos Analíticos Reducidos de mayor aplicación.

Los Modelos Analíticos Reducidos están desarrollados sobre aproximaciones de ingeniería razonables, aplicadas a las ecuaciones que describen las unidades, incluyendo términos de propiedades termodinámicas, tales como las que se aplican al cálculo de la entalpía y de las constantes de equilibrio líquido-vapor de cada corriente. Para ilustrar la forma en que se obtienen los modelos reducidos a partir de los modelos rigurosos, a continuación se presenta un ejemplo, citado por Jirapongphan.

Considerando la operación de una unidad de "flash" con una corriente de alimentación con  $n_c$  componentes, una corriente de salida de vapor y una corriente de salida de líquido.

El modelo riguroso para esta unidad está formado por estas ecuaciones:

$$B H_v - (1 - B) H_l - H_f - Q / \sum f_n = 0 \quad (4.39 a)$$

$$\beta \sum f_n - \sum v_n = 0 \quad (4.39 \text{ b})$$

$$v_n / \sum v_n - K_n \cdot l_n / \sum l_n = 0 \quad (4.39 \text{ c})$$

$$l_n + v_n - f_n = 0 \quad (4.39 \text{ d})$$

$$K_n - \phi_n^l / \phi_n^v = 0 \quad (4.39 \text{ e})$$

$$\phi_n^v - \phi_n^v(v, T, P) = 0 \quad (4.39 \text{ f})$$

$$\phi_n^l - \phi_n^l(l, T, P) = 0 \quad (4.39 \text{ g})$$

$$H_v - H_v(v, T, P) = 0 \quad (4.39 \text{ h})$$

$$H_l - H_l(v, T, P) = 0 \quad (4.39 \text{ i})$$

$$H_f - H_f(f, T, P) = 0 \quad (4.39 \text{ j})$$

$$T_v - T = 0 \quad (4.39 \text{ k})$$

$$T_l - T = 0 \quad (4.39 \text{ l})$$

$$P_v - P = 0 \quad (4.39 \text{ m})$$

$$P_l - P = 0 \quad (4.39 \text{ n})$$

en donde:

$f_n$  = flujo molar del componente  $n$  en la alimentación.

$v_n$  = flujo molar del componente  $n$  en el vapor.

$l_n$  = flujo molar del componente  $n$  en el líquido.

$K_n$  = relación de equilibrio

$H_v, H_l, H_f$  = entalpía de las corrientes de vapor, líquido y alimentación, respectivamente.

$\phi_n^v$  = coeficiente de fugacidad en la fase vapor.

$\phi_n^l$  = coeficiente de fugacidad en la fase líquida.

$\beta$  = fracción de vapor en la alimentación.

En este modelo existen  $5n_c + 9$  ecuaciones y  $6n_c + 13$  variables, por lo tanto, los grados de libertad son  $n + 4$ , los cuales pueden ser satisfechos al especificar  $f_n, T_f, P_f, T$  y  $P$ ; siendo las variables de estado ( $f_n, T_f, P_f$ ); las variables de salida ( $v_n, T_v, P_v$ ) y ( $l_n, T_l, P_l$ ); los parámetros de equipo son  $T$  y  $P$ ;  $Q$  y  $\beta$  son cantidades calculadas y las variables internas son  $H_v, H_l, H_f, K_n, \phi_n^v, \phi_n^l$ .

El desarrollo del Modelo Analítico Reducido para la unidad de "flash" parte de estas definiciones:

1) Equilibrio de fase:

$$y_n - K_n \cdot x_n = 0 \quad n = 1, \dots, n_c \quad (4.40)$$

2) Balance de materia por componente:

$$x_n^L + y_n^V - z_n^F = 0 \quad (4.41)$$

3) Balance total de materia:

$$L + V - F = 0 \quad (4.42)$$

4) Balance de entalpía:

$$\bar{H}_n L + \bar{H}_n V - \bar{H}_n F - Q = 0 \quad (4.43)$$

5) Ecuación constitutiva:

$$\sum x_n - \sum y_n = 0 \quad (4.44)$$

La fracción de vapor y el parámetro de volatilidad relativa se expresan así:

$$B = V/F \quad (4.45)$$

$$\alpha_n = K_n/K_b \quad (4.46)$$

en donde  $K_b$  se define según Boston y Britt (1978):

$$\ln(K_b) = \sum \omega_n \cdot \ln(K_n) \quad (4.47)$$

$$\omega_n = t_n / \sum t_n \quad (4.48)$$

$$t_n = y_n / (1 - B - \beta K_n) \quad (4.49)$$

El efecto de la temperatura y de la presión sobre  $K_b$  puede ser modelado así:

$$\ln(K_b P) - A - B/T = 0 \quad (4.50)$$

en donde A y B son los parámetros del modelo.

Se define una entalpía promedio sobre una base másica,  $m_n$ :

$$H = \frac{\bar{H}}{\sum (m_n \cdot z_n)} \quad (4.51)$$

$$= (a + bT)$$

Se definen las variables del flujo por componente:

$$l_n = x_n \cdot L \quad (4.52)$$

$$v_n = y_n \cdot V \quad (4.53)$$

$$f_n = z_n \cdot F \quad (4.54)$$

sustituyendo la ecuación 4.46 en la ecuación 4.40, para  $K_n$  y las ecuaciones 4.52 y 4.53 para  $l_n$  y  $v_n$ , se forma:

$$v_n/V - \alpha_n \cdot K_b \cdot l_n/L = 0 \quad (4.55)$$

A partir del balance total de materia, ecuación 4.42 y de la definición para  $\beta$  en la ecuación 4.45, se puede expresar  $F/V$  y  $F/L$  así:

$$F/V = 1/B \quad (4.56)$$

$$F/L = 1/(1-B) \quad (4.57)$$

Multiplicando la ecuación 4.55 por  $F$ , sustituyendo términos en las ecuaciones 4.56 y 4.57 y reordenando, se tiene:

$$v_n(1 - \beta) - \alpha_n \cdot K_b \cdot l_n \cdot \beta = 0 \quad (4.58)$$

La ecuación 4.45 puede ser reescrita en términos de los flujos por componente, así:

$$\beta \sum f_n - \sum v_n = 0 \quad (4.59)$$

y los balances de materia y entalpía por componente son:

$$l_n + v_n - f_n = 0 \quad (4.60)$$

$$(a+bT)_F \sum m_n f_n + Q - (a+bT)_L \sum m_n l_n - (a+bT)_V \sum m_n v_n = 0 \quad (4.61)$$

Adicionalmente, se tienen las ecuaciones de conservación:

$$T_v - T = 0 \quad (4.62)$$

$$T_l - T = 0 \quad (4.63)$$

$$P_v - P = 0 \quad (4.64)$$

$$P_l - P = 0 \quad (4.65)$$

Las ecuaciones 4.50, 4.58, 4.59, 4.60, 4.61, 4.62, 4.63, 4.64 y 4.65 son las que constituyen el modelo reducido para la unidad de "flash". Este sistema contiene  $2n_c + 7$  ecuaciones.

ciones con  $3n + 11$  variables ( $f_n, v_n, l_n, T_f, P_f, T_v, P_v, T_1, P_1, T, P, Q, K_b$  y  $B$ ); con  $n + 8$  coeficientes del modelo reducido ( $f_n, A, B, (a,b)_f, (a,b)_v$  y  $(a,b)_1$ ) y en donde ( $f_n, T_f, P_f, T, P$ ) son las variables de entrada; ( $v_n, T_v, P_v, l_n, T_1, P_1, Q$  y  $B$ ) son las variables de salida y  $K_b$  es la variable interna del modelo reducido. Es importante destacar que los grados de libertad en el modelo reducido son  $n + 4$ , los cuales son coincidentes con los del modelo riguroso.

El modelo reducido es evaluado analíticamente, tomando en cuenta que a partir de la definición de  $K_b$  en la ecuación 4.50 el parámetro  $B$  se puede obtener así:

$$B = \left[ \frac{\partial (\ln K_b)}{\partial (1/T)} \right]_{x,y,P} \quad (4.66)$$

en donde la derivada es calculada numéricamente calculando las constantes de equilibrio,  $K_b'$ , a una determinada perturbación de la temperatura,  $T'$ , manteniendo la composición y la presión constantes.  $K_b'$  a la temperatura  $T'$  es calculada usando la ecuación 4.47 y la expresión numérica para la ecuación 4.47 en esta forma:

$$B = \frac{\ln(K_b') - \ln(K_b)}{(1/T') - (1/T)} \quad (4.67)$$

El coeficiente A puede ser calculado directamente usando la ecuación 4.50. Las expresiones para calcular los coeficientes de entalpía, (a,b), son calculados según el procedimiento mostrado en la sección C.2 del Apéndice C.

La reducción en el número de ecuaciones y variables en un factor de  $4n_c + 2$  es muy significativo cuando el número de componentes,  $n_c$ , es muy grande. Se puede alcanzar una mayor reducción cuando la unidad modelada contiene cálculos para etapas múltiples, en donde las variables internas de las ecuaciones descriptivas rigurosas incluyen las composiciones, flujos, temperaturas y presiones.

Otra ventaja de este modelo reducido es que sus expresiones son independientes de las correlaciones para propiedades termofísicas, las cuales han sido tomadas en cuenta, en forma implícita y compacta, en las expresiones para los coeficientes del MAR.

Treviño-Lozano (1985), hizo una revisión de los Modelos Analíticos Reducidos propuestos por Jirapongphan, introduciendo algunas mejoras, tal como el cálculo de la entalpía de cada corriente mediante funciones de ajuste, del tipo:

$$H = H^{id} + A + B(T - T_{ref}) \quad (4.68)$$

en donde,  $H$  es la entalpía molar de la corriente;  $H^{id}$  es la entalpía de gas ideal;  $T$  es la temperatura de la corriente y  $A$  y  $B$  son parámetros del modelo reducido obtenidos del cálculo riguroso de la entalpía a cierta temperatura de referencia,  $T_{ref}$

La introducción del término de entalpía de gas ideal expresa una dependencia de la entalpía sobre la composición en los balances de energía, dentro del ciclo interno de cálculo.

Treviño-Lozano desarrolló modelos reducidos propios, incluyendo algunos destinados a unidades complejas de separación y para varios tipos de reactores. Estos modelos son descritos brevemente en el Apéndice C de este trabajo.

Tanto los modelos reducidos propuestos por Jirapongphan como por Treviño-Lozano cumplen con los diez requerimientos enunciados en la sección 4.3.1.1.

#### 4.3.2. MODELOS DE TIPO EVOLUTIVO.

Pierucci, Ranzi y Biardi (1982), presentaron un trabajo en base a lo que denominan Modelos Evolutivos, los cuales transforman el problema de promover la convergencia de un proceso cíclico, en la solución de un sistema reducido de ecuaciones algebraicas no lineales, producto de la sucesiva descomposición del sistema original.

Estos autores definen su método como un promotor de convergencia para un simulador modular secuencial, mediante la aplicación de dos clases de modelos: modelos rigurosos, representados por los modelos propios del enfoque Modular Secuencial, y modelos simplificados, referidos como modelos evolutivos, los cuales son establecidos sobre bases físicas y conformados por parámetros ajustables cuya determinación es realizada en cada iteración, en base a los modelos rigurosos.

El procedimiento de solución consiste de dos fases interrelacionadas. La primera se refiere a un recorrido secuencial empleando los modelos rigurosos, con el objeto de evaluar los parámetros internos del modelo evolutivo. La segunda fase consiste de la solución del sistema de ecuaciones obtenido al reunir todos los modelos evolutivos de las unidades pertenecientes a la red cíclica.

El esquema de cálculo se aplica a la red de ciclo máximo del proceso, de acuerdo con Steward (1965), y la presencia de más de un ciclo máximo, conduce a la aplicación repetida del procedimiento a cada uno.

Tomando en cuenta que los valores de las variables en las corrientes de alimentación y los parámetros de diseño para cada unidad son conocidos, de acuerdo al enfoque modular secuencial, el conjunto de modelos rigurosos obtendrá los valores de las variables internas y de las variables en las corrientes de salida, mediante la solución del proceso de acuerdo a una cierta secuencia, la cual implica la identificación de un conjunto adecuado de corrientes de corte.

Cada corriente de proceso es caracterizada por un conjunto de variables, las cuales definen su flujo por componentes, su temperatura y su presión. Usualmente, las presiones son determinadas en cada punto como un dato fijo, siendo los flujos y la temperatura las principales incógnitas dentro del balance de materia y energía.

Sin embargo, los modelos de tipo evolutivo únicamente promueven la convergencia sobre los flujos de materia por componente, por lo tanto, su aplicación está limitada al balance de materia de procesos muy simples.

Se definen como "corrientes internas" aquellas que relacionan dos unidades dentro del ciclo máximo en estudio, mientras que las "corrientes externas" son aquellas que conectan una de estas unidades con el resto del proceso, es decir, con unidades que no forman parte del ciclo máximo que está siendo simulado.

El conjunto de flujos molares de los componentes  $c$  en las corrientes internas  $j$ , son:

$$\bar{x} = \begin{bmatrix} x_j, & c_n \end{bmatrix} \quad \begin{array}{l} j = 1, \dots, n_i \\ n_i = 1, \dots, n_c \end{array} \quad (4.69)$$

en donde,  $n_i$  es el número de corrientes internas y  $n_c$  es el número de componentes, además:

$$\bar{x}^F = \begin{bmatrix} x^F \\ x_{j,n} \end{bmatrix} \quad \begin{array}{l} j = 1, \dots, n_j \\ n = 1, \dots, n_c \end{array} \quad (4.70)$$

siendo  $n_j$  el número de corrientes de entrada a una unidad.

El modelo riguroso de la unidad  $i$ ,  $m_i$ , se compone básicamente de un vector de parámetros de equipo,  $P_i$ , un vector de variables internas,  $V_i$ , además de los flujos por componente a la salida de esa unidad,  $x_k = x_{k,n}$ , y los flujos por componente en las corrientes de alimentación a la unidad  $x_f = x_{f,n}$ .

Por lo tanto, al proporcionar valores para  $x_f$  y  $P_i$ , entonces el sistema de  $m^i$  podrá ser resuelto para las incógnitas  $x^F$  y  $V_i$  y la simulación secuencial del ciclo máximo se iniciará al asumir valores para estas incógnitas.

Las unidades del ciclo son resueltas iterativamente al definirse un conjunto adecuado de corrientes de corte:

$$T = [T_n] \quad n = 1, \dots, n_c \quad (4.71)$$

lo que representa los flujos por componente asociado a las corrientes de corte, además

$$\sigma = \{ m_1, \dots, m_j, \dots, m^{nu} \} \quad (4.72)$$

es la secuencia de solución para las  $nu$  unidades del ciclo.

Una iteración comienza al asignar valores al conjunto  $T$ , lo que permite una solución secuencial de las unidades de acuerdo a  $\sigma$ . Cuando toda la secuencia  $\sigma$  es completada (fin de la iteración), entonces un valor calculado es asignado a cada flujo  $x$  de las corrientes internas, y en particular a aquellas que pertenecen al conjunto  $T$ , por lo tanto, al asumir que los valores calculados para las variables son  $F(T)$ , entonces:

$$T - F(T) = 0 \quad (4.73)$$

lo que se cumple cuando  $T$  es la solución del problema.

Es precisamente en la etapa de actualización de las variables de iteración en  $T$ , en donde se define la economía y eficiencia de todo el procedimiento de cálculo.

La opción propuesta por Pierucci y sus colaboradores se basa en el hecho de que cada modelo riguroso,  $m^i$ , puede ser representado por un modelo aproximado de tipo evolutivo,  $ev^i$ , de acuerdo a esta definición:

$$\left. \begin{aligned}
 x_{k,1} &= \omega_{k,1}^i (x_f, \pi_i, \delta_i) \\
 x_{k,2} &= \omega_{k,2}^i (x_f, \pi_i, \delta_i) \\
 &\vdots \\
 &\vdots \\
 &\vdots \\
 x_{k,n_c} &= \omega_{k,n_c}^i (x_f, \pi_i, \delta_i) \quad (k=1, \dots, n_{k_i}) \\
 C^i(x_k, x_f, \pi_k, \delta_k) &= 0
 \end{aligned} \right\} ev^i \quad (4.74)$$

y en forma compacta:

$$ev^i \left\{ \begin{aligned}
 x_k &= \omega_k^i \\
 C^i &= 0
 \end{aligned} \right. \quad (4.75)$$

siendo:

$$\omega_k^i = \{ \omega_{k,n}^i \} \quad (4.76)$$

$$C^i = (C_L^i) \quad (4.77)$$

en donde,  $n=1, \dots, n_c$ ,  $k=1, \dots, n_{k_i}$  y  $L=1, \dots, n_{u_i}$ ; siendo,  $n_c$ ,  $n_{k_i}$  y  $n_{u_i}$ , el número de componentes, el número de corrientes de salida del módulo  $i$ , y el número de variables desconocidas dentro del modelo evolutivo  $i$ , respectivamente.

En forma adicional,

$$\pi_i = ( \pi_{i,L} ) \quad (4.78)$$

$$\delta_i = ( \delta_{i,e} ) \quad (4.79)$$

$\pi_{i,L}$  son las variables internas desconocidas, las cuales pueden ser de cualquier naturaleza, usualmente coincidentes con alguna o todas las variables internas,  $V_i$ , del modelo riguroso original,  $m_i$ . Los parámetros de diseño evolutivos,  $\delta_i$ , son los que dan el aspecto evolutivo al modelo  $ev_i$ , siendo parámetros adaptables con el objeto de que los modelos evolutivos representen el comportamiento de los modelos rigurosos.

En resumen, el modelo evolutivo consta de  $(n_c \cdot n_{k_i} + n_{u_i})$  ecuaciones en las incógnitas  $x_k$  y  $\pi_k$ . Al proporcionar los valores para  $x_f$  y  $\delta_i$ , el sistema puede ser resuelto para obtener los valores de las incógnitas.

El modelo evolutivo  $ev^i$  depende tanto del tipo de unidad

como del modelo riguroso,  $m^i$  los cuales tienen que ver con la precisión deseada para calcular el valor de las variables en las corrientes de corte.

Tres características son importantes en los modelos evolutivos. Primero, la estructura de estos modelos mantiene las variables de salida,  $x_k$ , en forma explícita en las variables restantes del modelo. Segundo, no existe ninguna limitación a-priori en la estructura de las funciones  $\omega_{k,n}^i$  y  $C_L^i$ . Tercero, no se requieren técnicas especiales de descomposición ni de solución del sistema de ecuaciones evolutivas, por lo tanto el sistema es visto en términos de la solución de un sistema no lineal mediante algoritmos ya existentes y de probada eficiencia.

Un procedimiento simplificado para promover la convergencia de la simulación de procesos al emplear modelos evolutivos, es:

- 1.- Asignar valores a los conjuntos de  $x$  y  $P_i$ , suponiendo valores iniciales para  $T$ .
- 2.- Resolver la secuencia  $\alpha(m^1, \dots, m^{nu})$  y calcular los parámetros  $\delta_i$  de cada modelo evolutivo.
- 3.- Definir una medida de error al final de cada cálculo.
- 4.- Si el error es menor que el de un valor aceptable, el cálculo es concluido; de otra manera, actualizar el valor de  $T$ .

mediante la solución del siguiente sistema lineal:

$$\begin{aligned} T - F(T) &= 0 \\ C^i &= 0 \end{aligned} \quad (4.80)$$

siendo las incógnitas  $T$  y  $\pi_i$ ; los valores de  $F(T)$  son obtenidos por la secuencia  $(\omega_k^1, \omega_k^2, \dots, \omega_k^K, \dots, \omega_k^{nu})$ . Regresar al punto 2 y repetir el procedimiento.

Pierucci, Ranzi y Biardi, presentaron una serie de modelos evolutivos muy simples, referidos a operaciones unitarias de uso común en la simulación de procesos (mezcladores, separadores, reactores, intercambiadores de calor, etc.) haciendo notar que sólo son una muestra de la gran variedad de modelos de este tipo que pueden ser generados, debido a que son muy generales y con una estructura flexible, lo que permite diferentes posibilidades de descripción, y aún más, extender su uso a problemas de diseño.

Sin embargo, tal y como los autores lo reconocen, hace falta mayor investigación en el desarrollo de modelos más sofisticados y precisos, siguiendo las pautas por ellos marcadas, en especial para modelos evolutivos que incluyan cálculos sobre propiedades termofísicas a fin de converger sobre la temperatura y obtener el balance de energía. De cualquier manera, es necesario aplicar estos modelos sobre procesos químicos reales y poder así

evaluar sus limitaciones y definir con precisión los alcances en su aplicación.

#### 4.4. AMPLIACION DEL CONCEPTO CICLO INTERNO - CICLO EXTERNO.

En el desarrollo de este capítulo se ha hecho énfasis en la actual orientación de los simuladores modulares simultáneos para ser considerados como dos ciclos de convergencia anidados: un ciclo interno y un ciclo externo.

El ciclo interno está representado por los modelos simples, tanto lineales como no lineales, descritos anteriormente; mientras que el ciclo externo se representa por los modelos rigurosos, propios del enfoque Modular Secuencial.

A continuación se abordarán brevemente algunos conceptos referentes al manejo de información a nivel del ciclo externo y su repercusión en el comportamiento del ciclo interno.

#### 4.4.1. VARIABLES EN EL CICLO EXTERNO.

Boston y Britt (1978), interpretan su algoritmo Interno-Externo ("inside-out") de cálculo de operaciones de separación, como un método que utiliza parámetros de un modelo simple como variables de iteración, en lugar de las variables propias del proceso. La ventaja de este cambio de variables de iteración es que los parámetros del modelo reducido son menos dependientes de las condiciones de proceso, que las variables originales; por esta razón se hacen converger los parámetros del modelo reducido en el ciclo externo, definiendo las tolerancias adecuadas para este ciclo y obteniendo nuevos estimados en términos de estos parámetros. Puesto que los valores de las variables de proceso predichas en el ciclo interno sólo dependen de los valores de los parámetros del modelo reducido, la convergencia de estos parámetros automáticamente asegura la convergencia de las variables originales.

Sin embargo, cuando se afronta la convergencia global de un proceso es más simple visualizar las variables de proceso, que los parámetros del modelo. Un esquema de convergencia en términos de variables de proceso es más fácil de implementar que un esquema de convergencia en términos de parámetros. En forma adicional, las tolerancias en términos de parámetros de modelos reducidos son muy difíciles de definir cuando varios módulos unitarios diferentes están involucrados.

Existen dos opciones para elegir las variables que deberán ser convergidas en el ciclo externo. La primera opción es la de considerar la totalidad de las variables presentes en el ciclo externo. La segunda opción es considerar sólo las variables asociadas con las corrientes de alimentación y las corrientes de corte junto con las variables internas de cada unidad.

#### 4.4.1.1. Convergencia de todas las variables de las corrientes.

La idea de mantener todas las variables en el ciclo externo es la de tener un conjunto totalmente independiente de puntos de base para calcular los parámetros del modelo reducido de cada módulo. Las variables de cada corriente y de cada unidad son trasladadas directamente del ciclo interno hacia todas las corrientes y variables en el proceso. Entonces cada módulo riguroso es ejecutado independientemente de los demás, en base a los valores de entrada para cada corriente que fueron proporcionados por la última iteración del ciclo interno. Esto puede ser interpretado como una estrategia en donde todas las corrientes en el proceso son cortadas y convergidas simultáneamente en el ciclo externo.

Existen varias ventajas al utilizar esta aproximación, siendo la más obvia el que el algoritmo se torne completamente inde-

pendiente de la estructura del proceso. Una vez inicializado el algoritmo, no se requiere ninguna información acerca de la topología del proceso (corrientes de corte, secuencia de cálculo, estructura de recirculación, etc.) para continuar los cálculos, por lo tanto, el cálculo de los bloques rigurosos en el ciclo externo puede ser realizado en cualquier orden. Otra ventaja es la que se refiere al concepto de cálculos modulares simultáneos integrados, el cual será abordado con mayor amplitud más adelante. Los cálculos integrados pueden resultar muy importantes para incrementar la eficiencia del ciclo externo al buscar la simplificación de los cálculos dentro de los bloques rigurosos, los cuales deben ser ejecutados independientemente para evitar la propagación de errores de un módulo a otro.

La principal desventaja al hacer converger todas las variables de las corrientes en el ciclo externo es que los cálculos de entalpía y equilibrio de fases necesitan ser realizados sobre cada corriente de proceso cada vez que una nueva iteración del ciclo externo comienza, lo que representa un consumo excesivo de tiempo de cómputo.

#### 4.4.1.2. Convergencia de las variables en las corrientes de corte y en las corrientes de alimentación.

El número de variables que se hacen converger en el ciclo

externo pueden ser sustancialmente reducidas cuando sólo se manejan las corrientes de alimentación y las corrientes de corte.

En esta aproximación, sólo las variables de las corrientes de alimentación y de corte son trasladadas del ciclo interno hacia el ciclo externo. Una vez que estas corrientes son re-inicializadas con variables actualizadas, los bloques de cálculo rigurosos son ejecutados uno tras otro siguiendo una secuencia de cálculo apropiada. Por lo tanto, un recorrido modular secuencial es realizado al comienzo de cada iteración del ciclo externo, no sólo para calcular nuevos parámetros del modelo reducido, sino también para calcular los nuevos estimados para las variables en las demás corrientes que no son las de alimentación ni las de corte.

La anterior estrategia puede ser interpretada como un algoritmo de convergencia muy sofisticado para las corrientes de corte en un simulador modular secuencial. El algoritmo requiere un conjunto de corte y una adecuada secuencia de cálculo. La elección de las corrientes de corte es de efecto decisivo sobre el número de iteraciones del ciclo externo necesarias para converger el proceso.

La principal ventaja de esta aproximación se refiere a la gran reducción en el número de variables manipuladas en el ciclo externo; por lo tanto, menos corrientes necesitan ser evaluadas

Es bien conocida y aceptada la necesidad de los recorridos secuenciales con el objeto de inicializar el algoritmo, sin embargo, otra situación en donde estos recorridos son importantes para la simulación modular simultánea, es en el manejo de discontinuidades. Por ejemplo, en una operación de tipo "flash", durante el proceso de convergencia del ciclo interno, la fracción de vapor en el tanque puede tener un valor de cero o uno, convirtiendo la salida de dos fases, a una sola. Si durante la siguiente iteración del ciclo interno la fracción de vapor no se mueve otra vez hacia la región de dos fases, el ciclo interno convergerá a una solución que contiene una sola fase. Si esta es la solución real del problema, el algoritmo continuará sin contratiempos. Sin embargo, si la solución real no es una sola fase a la salida del tanque, entonces el algoritmo modular simultáneo fallará, puesto que las ecuaciones reducidas generadas después del siguiente ciclo externo serían ecuaciones de una sola fase, las cuales no permitirán la posibilidad de tener las dos fases en el separador.

Una posible solución a este problema es ejecutar uno o varios recorridos secuenciales del proceso antes de que el cambio de fase sea aceptado en el ciclo externo. Esto deberá conducir los cálculos en la dirección correcta (tanto para la región de una fase o para la región de dos fases), antes de que el siguiente problema reducido sea formulado.

en su entalpía y equilibrio de fases, durante la transición del ciclo interno al ciclo externo; según reporta Treviño-Lozano (1985), la reducción es de aproximadamente del 10% del número total de corrientes.

Esto puede resultar en iteraciones del ciclo externo más breves, a pesar de que esta ventaja puede ser fácilmente superada por los ahorros potenciales alcanzados por la integración de los cálculos. Otra ventaja potencial relativa al pequeño número de variables manipuladas, es la posibilidad de emplear métodos numéricos más sofisticados para hacer converger el ciclo externo sin incrementar los requerimientos computacionales.

#### 4.4.2. MANEJO DE DISCONTINUIDADES.

Anteriormente se indicó que ningún recorrido modular secuencial del proceso es estrictamente requerido por el algoritmo modular simultáneo si todas las variables de las corrientes de iteración son conservadas en el ciclo externo. Sin embargo, los recorridos modulares secuenciales pueden ser usados para trabajar como orientadores o auxiliares de la convergencia, aún si ellos no fueran necesarios, en cada iteración del ciclo externo.

Lo anterior implica una regla heurística tendiente a auxiliar en la solución de discontinuidades, en base a la capacidad de los módulos en el simulador modular secuencial para detectarlas. Por esta razón, es el ciclo externo (con modelos de cálculo rigurosos) el que conduce al algoritmo hacia la solución correcta. La introducción de una discontinuidad en el ciclo interno podría crear los mismos problemas observados en los simuladores orientados hacia las ecuaciones, puesto que esta parte del método modular simultáneo es resuelta en forma global.

#### 4.4.3. CONVERGENCIA DE LAS VARIABLES DEL CICLO EXTERNO.

El procedimiento computacional mostrado en la Figura 4.10 presenta un procedimiento general de cálculo para el ciclo externo, destacando que la prueba de convergencia es aplicada después de que el ciclo externo se ha hecho converger.

Al promover la convergencia del proceso, cada variable manipulada en el ciclo externo debe converger dentro de una cierta tolerancia. Por esta razón, parece ser más apropiado checar la convergencia de cada variable del ciclo externo por separado, en lugar de verificar únicamente la norma de los residuos, además

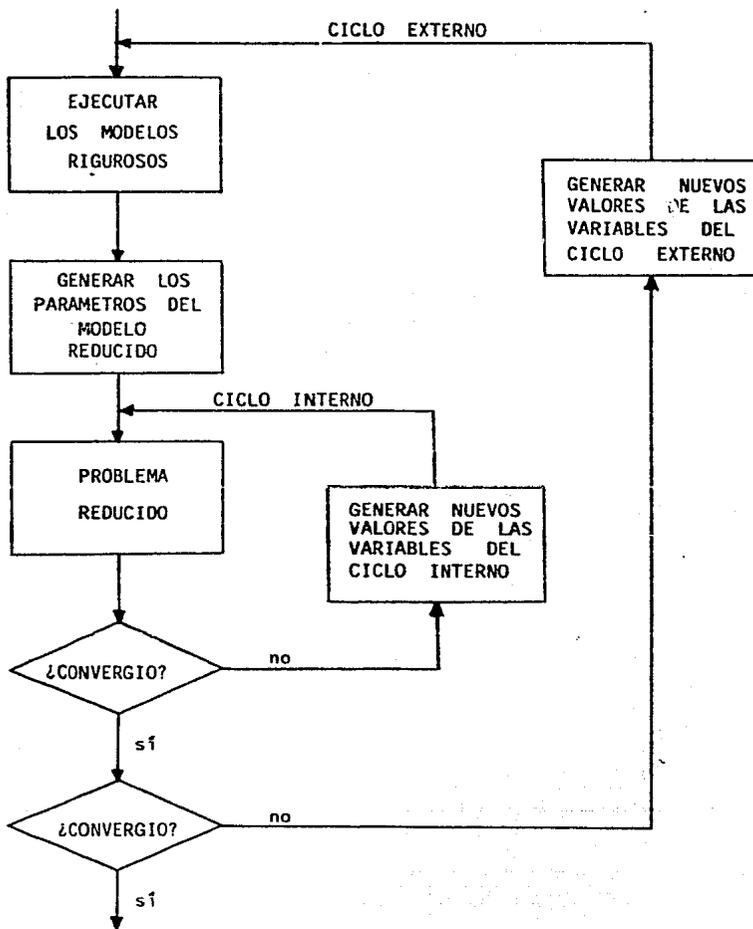


FIGURA 4.10.

SECUENCIA DE CALCULO DEL ALGORITMO MODULAR  
SIMULTANEO CON DOS PRUEBAS DE CONVERGENCIA.

si todas las variables deben converger dentro de la misma tolerancia, ésta debe ser definida en términos relativos, o aplicada a valores escalados de cada variable, lo cual es necesario con el objeto de tomar en consideración la gran diferencia en magnitud entre las variables de proceso.

En la forma en que el método ha sido definido, la convergencia es verificada antes de que los modelos rigurosos sean ejecutados en el ciclo externo, para la siguiente iteración. Se debe notar que los modelos rigurosos deben ser ejecutados una última vez después de que el ciclo externo ha convergido. Este último recorrido riguroso permite el cálculo de las variables de proceso que no aparecen en el ciclo interno (variables internas de cada unidad, variables complejas de corriente, etc.); adicionalmente, este recorrido secuencial incrementa la precisión y eficiencia del método.

Treviño-Lozano presenta un problema que ilustra la utilidad de los recorridos secuenciales periódicos, incluyendo el recorrido propuesto al final. El problema de Cavett (1963), se usó para determinar la utilidad de los recorridos secuenciales; una descripción más amplia de este problema típico, se presenta en el Capítulo 6 de este trabajo.

En la Tabla 4.1 se muestra la historia de las iteraciones para los cálculos modulares simultáneos de este problema. El

NUMERO DE ITERACIONES

CICLO	EXTERNO	CICLO	INTERNO
	1		5
	2		2
	3		2
	4		2
	5		2
	6		2
	7		1
	8		1
	9		1
	10		1
	11		1
			20

TABLA 4.1. HISTORIA DE LAS ITERACIONES PARA EL PROBLEMA DE CAVETT, HACIENDO CONVERGER TODAS LAS VARIABLES EN EL CICLO EXTERNO.

ciclo externo tiene una tolerancia de convergencia de  $10^{-3}$  en las variables escaladas, usando sustitución directa en la solución; mientras que el ciclo interno tiene una tolerancia de  $10^{-4}$ , alcanzando la convergencia del proceso en 11 iteraciones del ciclo externo y 20 iteraciones del ciclo interno.

Al analizar los cálculos en detalle, es posible observar que después de la tercera iteración del ciclo externo, las únicas variables del ciclo externo no convergidas son las temperaturas de las corrientes de corte, las cuales oscilan de una a otra iteración del ciclo externo. Sin embargo, es sencillo de entender que si los flujos de las corrientes de corte son convergidas, los valores correctos de las temperaturas de las corrientes de corte serán obtenidos simplemente al realizar un recorrido secuencial del proceso. Por lo tanto, las iteraciones del ciclo externo pueden ser detenidas después de que los flujos han sido convergidos, despreciando la no convergencia de las temperaturas.

En la Tabla 4.2 se presenta la historia de las iteraciones para el mismo problema, pero con la temperatura de las corrientes de corte escaladas para que sean dos ordenes de magnitud más pequeñas que las demás variables, con el objeto de que las variables calculadas siempre satisfagan la tolerancia del ciclo externo. En este caso el proceso es convergido en 4 iteraciones del ciclo externo y un total de 11 iteraciones para el ciclo interno.

## NUMERO DE ITERACIONES

CICLO	EXTERNO	CICLO	INTERNO
1			5
2			2
3			2
4			2
			<hr/>
			11

TABLA 4.2. HISTORIA DE LAS ITERACIONES PARA EL PROBLEMA DE CAVETT, OMITIENDO LA CONVERGENCIA DE LAS TEMPERATURAS DE LAS CORRIENTES DE CORTE.

#### 4.4.4. CALCULOS INTEGRADOS.

Anteriormente se mencionó que el concepto modular simultáneo puede ser visto como una generalización, a nivel de proceso, del algoritmo propuesto por Boston (1978), implicando el uso de dos tipos de modelos a dos niveles diferentes de convergencia.

Desde este punto de vista, el conocimiento de la operación de la unidad que está siendo modelada puede ser explotado para elaborar procedimientos eficientes de convergencia para cada ciclo, específicos de esa unidad.

Para un proceso conteniendo operaciones unitarias modeladas de acuerdo al enfoque de ciclo interno-ciclo externo, un simulador modular simultáneo puede incluir un algoritmo de doble partición a nivel de proceso y otro algoritmo de doble partición más especializado para lograr la convergencia dentro de los módulos; visto de esta manera, es posible hablar de ciclos internos con modelos simples y ciclos externos con modelos rigurosos para cada módulo, dentro de las iteraciones del ciclo externo usado para la convergencia global del proceso. Este esquema de cálculo es mostrado en la Figura 4.11.

Todo lo anterior sugiere integrar los algoritmos de ciclo interno-ciclo externo de cada módulo a la estructura modular simultánea global del simulador. Treviño-Lozano (1985), sugie-

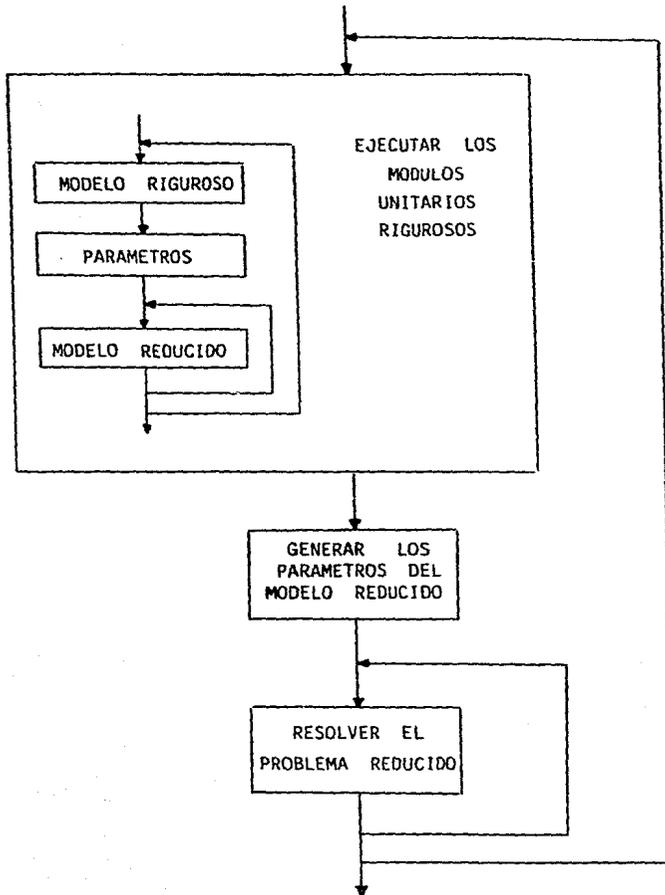


FIGURA 4.11.

ENFOQUE MODULAR SIMULTANEO CON MODELOS RIGUROSOS  
 MANEJADOS CON ALGORITMOS DEL TIPO CICLO INTERNO  
 Y CICLO EXTERNO.

re dos grados de integración:

1.- Simulador Modular Simultáneo Integrado.- En este caso se realizan una o dos iteraciones en el ciclo externo a nivel de módulo. Los parámetros del modelo reducido son calculados a partir de la solución no convergida en el módulo.

2.- Simulador con un enfoque ciclo interno-ciclo externo total.- En este caso los cálculos del bloque riguroso son destinados únicamente a la obtención de la primera generación de parámetros en el algoritmo local de ciclo interno-ciclo externo; por lo tanto los ciclos internos de cada bloque nunca se hacen converger. En otras palabras, los bloques de cálculo rigurosos vienen a ser únicamente generadores de parámetros de los modelos simples para el ciclo interno global. El esquema de cálculo para este tipo de estructura se muestra en la Figura 4.12.

La aproximación Modular Simultánea Integrada puede ser vista como una modificación del concepto modular simultáneo típico, en donde ciertos cálculos rigurosos de los bloques en el ciclo externo tienen una tolerancia muy flexible.

Sin embargo, el concepto Ciclo interno-Ciclo externo Total es una idea innovadora dentro de la simulación de procesos, puesto que tanto los módulos como el proceso total se hacen converger

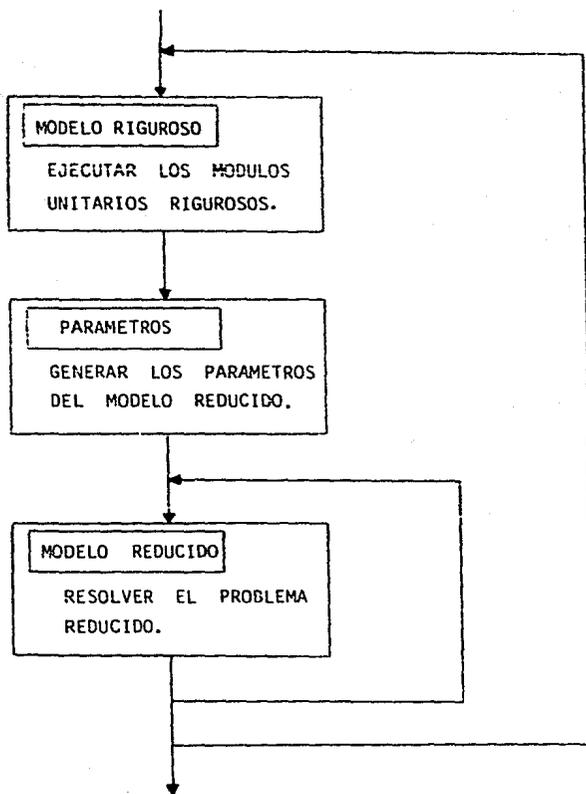


FIGURA 4.12. REPRESENTACION DE UN SIMULADOR MODULAR SIMULTANEO BAJO UN ENFOQUE DE CICLO INTERNO - CICLO EXTERNO TOTAL.

simultáneamente, con el ahorro de cálculos rigurosos en los bloques. Las principales ventajas de la integración son:

1.- Los cálculos rigurosos de cada bloque no son necesarios en el ciclo externo a nivel de proceso.

2.- Los modelos reducidos ya desarrollados con algoritmos de ciclo interno-ciclo externo para la simulación de unidades de separación complejas, también son explotados a nivel de proceso (modelos tales como columnas de destilación complejas, destilación a tres fases, extractores, etc.)

3.- Los parámetros del modelo reducido generados automáticamente por el algoritmo de cada módulo, también son usados a nivel de proceso total.

Una desventaja de los cálculos integrados es que se puede necesitar una mayor cantidad de iteraciones del ciclo interno para hacer converger el proceso, en comparación con las implementaciones típicas de este enfoque. Sin embargo, Treviño-Lozano resolvió varios procesos bajo este nuevo enfoque y comprobó que aún así, este procedimiento es más ventajoso que el enfoque modular secuencial y altamente competitivo con otras implementaciones del concepto modular simultáneo.

## C A P I T U L O     5

ESTRUCTURAS PARA LA IMPLEMENTACION DE  
LOS METODOS MODULARES SIMULTANEOS .

## CAPITULO 5. ESTRUCTURAS PARA LA IMPLEMENTACION DE LOS METODOS MODULARES SIMULTANEOS.

Una vez conocido el desarrollo que han tenido los modelos y algoritmos empleados bajo el enfoque modular simultáneo se hace necesario saber la forma en que pueden ser implementados con el objeto de integrar un sistema de simulación de procesos.

En el capítulo 3 fueron presentados los cuatro pilares en los que descansa la estructura general de un simulador de procesos de carácter modular. Primeramente son considerados los modelos tanto de operaciones unitarias como de apoyo, y los algoritmos que operan sobre estos modelos con el objeto de que rindan los resultados que de ellos se esperan, todo esto implementado a través de un soporte computacional en forma de soporte lógico (software) del sistema y de soporte físico (hardware) de la computadora, contando con el auxilio de una adecuada interface de comunicación entre el sistema y el usuario.

En el capítulo 4, al hacer la presentación de los modelos y algoritmos modulares simultáneos, se reconoció que su grado de desarrollo se incrementa día a día y que, sin embargo, la creación de estructuras computacionales adecuadas para la implementación práctica de estas técnicas no progresa al mismo ritmo. Precisamente un obstáculo en el avance del enfoque modular simultá-

neo ha sido la carencia de una estructura flexible, poderosa y confiable que permita evaluar las capacidades reales de los modelos y algoritmos, y valorar abiertamente sus ventajas y desventajas.

En este capítulo se realizará una breve descripción de algunas estructuras creadas para la implementación de los métodos modulares simultáneos. Se inicia describiendo la forma en que se concibe un sistema de simulación en base a un soporte lógico y una interface con el usuario; se muestran las etapas por las que atraviesa la creación de un sistema modular y la forma en que se organizan sus elementos y actividades, con el objeto de rendir los resultados deseados. Finalmente se muestran las tres estructuras más representativas de las tendencias marcadas por el actual grado de desarrollo de este enfoque.

## 5.1. BASES PARA LA CONCEPCION DE UN SIMULADOR MODULAR.

### 5.1.1. Soporte Computacional.

El soporte computacional incluye todo aquello que se requiere para implementar los modelos y algoritmos en una computadora en particular y bajo un cierto sistema operativo. En virtud de que los costos del soporte físico (hardware) van decreciendo,

los costos del soporte lógico (software) tienden a ser los dominantes en el costo global del desarrollo de un sistema de simulación. En las siguientes secciones se abordarán algunos aspectos referentes al soporte lógico, puesto que el estudio del soporte físico queda fuera de los objetivos de este trabajo.

Al desarrollar el soporte lógico de un sistema de simulación se tienen en cuenta aspectos tales como el programa y arquitectura del sistema, la estructura de datos, interface entre el sistema y los archivos, lenguajes de programación, documentación del sistema, etc..

Como se mencionó anteriormente, el desarrollo de soporte lógico para sistemas de simulación ha recibido muy poca atención por parte de los ingenieros de diseño auxiliado por computadora, quienes están más orientados al desarrollo de buenos modelos y algoritmos, ya que el problema de programación es tratado como un detalle de rutina, que de alguna manera será resuelto una vez que se presente. Afortunadamente, durante la década pasada se generaron aproximaciones sistemáticas que están transformando la producción de soporte lógico, de ser un simple arte, a convertirse en un campo tecnológico formal, motivado en gran parte por la falta de estructuras adecuadas para poner realmente a prueba los innovadores modelos y algoritmos, tanto existentes como por ser construídos.

Existen cinco etapas en el ciclo de vida de cualquier pro-

grama de computadora: definición, diseño, implementación, prueba y entrega. En el caso de un sistema de simulación de tipo comercial (ASPEN, PROCESS, etc.) el tiempo entre la definición y la entrega es de aproximadamente 7 años; para programas académicos de simulación el tiempo es menor, aunque puede variar grandemente, dependiendo de los alcances que se pretendan, partiendo de programas que se generan durante un ciclo escolar, hasta sistemas más avanzados, que incluso pueden ser usados comercialmente. De cualquier manera, las cinco etapas se realizan una tras otra.

#### a) Definición.

Aquí se determinan las especificaciones funcionales y los criterios de diseño, es decir, se delimitan las funciones y capacidades del sistema, aunque todavía no se defina cómo se lograrán. Se elige el tipo de modelos que serán incluidos, los tipos de análisis que se realizarán, las características de la interface entre la computadora y el sistema operativo, etcétera. Es muy importante tomar en cuenta si las capacidades a ser implementadas en el nuevo sistema ya han sido probadas en sistemas ya existentes, en sistemas prototipo o en sistemas similares para otras aplicaciones; en esta etapa de desarrollo es vital el conocimiento de sistemas ya existentes y evaluar sus ventajas para agregarlas, en lo posible, al nuevo simulador, es decir, no es necesario reinventar conceptos ya conocidos. Este hecho es de capital importancia en el desarrollo del presente trabajo, tal y cómo se verá en el Capítulo 6. Es necesario no perder de

vista los objetivos finales del proyecto, tomando en cuenta la evolución en las necesidades de los usuarios, siendo muy común, sobre todo en sistemas prototipo y académicos, que se incluyan algunas capacidades no requeridas expresamente por los usuarios, con el objeto de explorar en cuanto a nuevas aplicaciones del simulador.

#### b) Diseño

En esta etapa se debe especificar en detalle la forma en que se lograrán los requerimientos funcionales definidos previamente; es un proceso de síntesis en el cual se van a considerar y evaluar varias alternativas. Una metodología para el diseño de sistemas, conocida como "diseño estructurado", ha sido desarrollada por varios grupos de ingenieros de programación, destacando Yourdon y Constantine (1975), Myers (1975), Dicho-ver (1977) y Peters y Tripp (1977). El diseño estructurado es un procedimiento sistemático para organizar un gran sistema en varios elementos más simples, mediante la definición de los requerimientos de cada subsistema, mostrando su flujo de información y estructurando la interface entre ellos. El diseño estructurado incluye una notación y terminología propia, así como técnicas de análisis para evaluar alternativas, elaborar normas para lograr diseños adecuados a las necesidades y un procedimiento mediante el cual se da forma a un diseño que es sistemáticamente revisado, probado y expresado en detalle. Toda esta metodología tiene como objetivo hacer la codificación, depuración

y modificación más fácil y rápida, mediante una reducción en la complejidad del proceso de diseño de grandes programas.

c) Implementación.

La implementación es el procedimiento de traducir el diseño del sistema en un código de computación para que se pueda aplicar un lenguaje de programación adecuado. Actualmente, el lenguaje de programación más utilizado en aplicaciones de ingeniería es el FORTRAN, debido a que se encuentra disponible en casi todas las computadoras y porque este es el lenguaje de alto nivel que reúne las ventajas de facilitar el manejo matemático y de estar firmemente arraigado entre los profesionistas relacionados con el diseño de procesos químicos. Desafortunadamente el FORTRAN no es un lenguaje estructurado, y en cierta forma no es el más adecuado para implementar complicadas estructuras de datos y secuencias de operaciones; por lo tanto es recomendable que estas dos áreas del programa se mantengan aisladas, tanto del resto del programa como entre sí, puesto que es precisamente en estas secciones en donde pueden surgir discrepancias entre las versiones de FORTRAN existentes, y tal vez sea necesario realizar algunos cambios, sin afectar la totalidad del programa. Si bien es cierto que algunos lenguajes, como el PL/1, PASCAL y FORTRAN 77, se presentan como lenguajes mejorados y adecuados para la implementación de sistemas de simulación, FORTRAN aún prevalecerá por bastante tiempo, en virtud de su amplia acepta-

ción y a que existe una gran cantidad de desarrollo computacional basado en este lenguaje, que no puede ser fácilmente traducido a otro lenguaje. En la implementación se debe elegir un determinado nivel de rigor en los estándares del estilo de programación y documentación, con el objeto de que el programa pueda ser aplicado a varios sistemas operativos y entendible por los usuarios.

d) Prueba.

Esta es una de las etapas más costosas en tiempo y recursos, puesto que los métodos empleados parecen primitivos, en comparación a los usados en otras etapas. Se requiere un alto grado de confianza, el programa debe estar libre de errores de programación y debe existir mucha reserva en cuanto a la validez de los resultados de las pruebas. Cada programa y cada subsistema es probado en forma independiente, mediante corridas sobre problemas típicos de los cuales se tiene amplia información y se pide a algunos usuarios, cuidadosamente elegidos, que lo utilicen y reporten sus resultados. En muchas ocasiones la prueba se realiza sobre problemas reales, incluyendo situaciones en las que otros sistemas han fallado, en un ambiente de prueba estrictamente controlado.

e) Entrega.

A partir de su entrega, un programa es usado en forma rutinaria por el usuario, con una mínima intervención del equipo que lo desarrolló. Se debe preparar una estrategia con el objeto de recabar información del usuario y poder mantener y reforzar el sistema, así como para organizar los medios de entrenamiento y consulta de los operadores.

Los recursos económicos requeridos en el desarrollo de un sistema de simulación son muy altos, y es común que exista cierta renuencia a ser proporcionados por las autoridades encargadas de hacerlo. Una solución típica, muy utilizada en la industria (Proctor, 1980), es comenzar con un sistema a pequeña escala, que forma parte de un sistema mayor, obtener algunos resultados y en base a estos obtener más fondos para extender las capacidades del sistema. Desafortunadamente, un sistema de simulación es incapaz de producir resultados tangibles en corto plazo, al menos no antes de alcanzar las últimas etapas de su implementación. Evans (1981), hace una analogía entre el procedimiento de desarrollo de un soporte lógico mediante una aproximación sistemática y la forma en que se construye una casa. Existe un largo periodo de tiempo durante el cual los ingenieros preparan los planos de construcción y un periodo lento para la preparación del terreno. Una vez realizado lo anterior, existe un periodo muy rápido en el cual la estructura crece y finalmente existe

otro periodo muy largo para la terminación, decoración y corrección de errores de edificación.

#### 5.1.2. Interface con el Usuario.

La interface con el usuario es la ventana a través de la cual el usuario visualiza el sistema de simulación. Incluye los medios para que el operador describa su problema, de acuerdo a un lenguaje o conjunto de claves específicos de cada sistema; y genera los reportes de salida, en los que el sistema presenta sus resultados.

Son requerimientos ineludibles de una adecuada interface: que sea poderosa, es decir, que tenga una amplia capacidad descriptiva; que sea útil en cuanto a que represente un adecuado medio de comunicación con el sistema, y que los resultados generados sean los que en realidad desean los usuarios, tanto en forma como en contenido; debe ser flexible y fácil de usar, que sea clara en los requerimientos de información y que al mismo tiempo proporcione una adecuada lista de opciones que el sistema pueda aceptar sin necesidad de mecanismos complicados, por lo tanto, debe estar a un nivel de generalización de acuerdo al tipo de usuario que la empleará con mayor frecuencia. Se debe entender que algunos de estos requerimientos están en conflicto unos con otros, por lo tanto, se debe decidir, en orden de prio-

riedad, que capacidades y características serán las dominantes.

En la mayoría de los simuladores, aún en los más avanzados, existe poca flexibilidad en los reportes de resultados intermedios y finales, puesto que la información final generada está en relación directa con la información proporcionada al inicio y de las opciones elegidas al momento de definir el proceso. Existen reportes de resultados estándar, conteniendo valores de las variables de proceso para cada corriente (balances de masa y energía), así como datos de funcionamiento de los equipos involucrados, datos de costos del proceso, etcétera. Sin embargo, actualmente se han incorporado opciones tales como las de datos y gráficas de análisis de sensibilidad, en los cuales uno o varios parámetros del proceso son variados a través de algunas corridas y su efecto sobre el proceso es graficado, para su comparación y análisis.

La introducción inicial de información es tal vez la que más opciones ofrece. Los simuladores más simples admiten los datos de acuerdo a formatos fijos, previamente establecidos, y la información es manejada en una forma muy rígida y específica, tanto en el tipo como en la secuencia de operaciones que deben realizarse.

En simuladores más sofisticados, la interface con el usuario es del tipo conversacional, es decir, se utiliza un formato libre para introducir los datos y esta información es solicitada mediante secuencias organizadas en palabras, preguntas y párrafos, de acuerdo a ciertas claves simples o según la posición de las

preguntas. Además, el sistema proporciona un cierto número de opciones y valores, considerados como estándar, para inicializar las variables, que opera cuando el usuario no tiene elementos para una acertada elección a-priori. Existen proposiciones del tipo INSERT, mediante las cuales el usuario puede introducir información que considere útil y que no está siendo solicitada expresamente por el sistema.

Actualmente existen interfaces prototipo que incluyen la descripción de un proceso en forma gráfica, especialmente en lo referente a su topología y correlación gráfica, para la cual no existen ecuaciones empíricas, como lo describe Rosen (1980).

Otra innovación consiste en introducir la información mediante Sistemas Expertos o Sistemas Basados en el Conocimiento los cuales se apoyan en estudios sobre Inteligencia Artificial, y cuya misión es la de guiar al usuario en la correcta descripción de su problema y a la mejor elección de las múltiples opciones que ofrece un simulador, tal y como lo describen Bañares-Alcántara, Westerberg y Rychener (1985) con los sistemas PROSPECTOR y CONPHYDE.

### 5.1.3. Organización de las Actividades en un Simulador Modular.

Con el objeto de realizar análisis útiles sobre un determinado proceso, un simulador lleva a cabo una amplia variedad de actividades, algunas de las cuales requieren una intervención directa por parte del operador; otras, en cambio, son desarrolladas ya sea por instrucciones propias del programa de simulación, o bien, cumpliendo protocolos que son parte del sistema operativo de la computadora, sin la intervención del usuario.

#### 5.1.3.1. Etapas de una simulación.

Los simuladores modulares son construidos en base a alguno de estos tipos de estructura de cálculo: estructura fija y estructura variable.

La estructura fija se caracteriza porque el programa ejecutivo es exactamente el mismo, sin importar el tipo de proceso que se esté simulando, requiriendo que todos los módulos unitarios sean cargados al sistema, independientemente si son necesarios o no para el problema dado. La principal ventaja de esta estructura es que requiere poco tiempo de cómputo. Sin embargo, su desventaja más sobresaliente son la falta de flexibilidad para abordar varios problemas, sus limitaciones al abordar procesos complejos y pocas posibilidades de expansión e incorporación de

de módulos unitarios, amplias capacidades termodinámicas y mayores opciones de entrada y salida.

En una estructura variable se forma un programa ejecutivo diferente, aplicable a cada caso en particular. Si el sistema está escrito directamente en FORTRAN, por regla general se realizan llamados a los módulos unitarios y de servicio necesarios, además, existe la posibilidad de emplear un Lenguaje Orientado al Problema (POL, por sus siglas en inglés), el cual puede ser interpretado directamente en un código ejecutable, o bien, hacer que el POL genere una serie de procedimientos que serán compilados, unidos y ejecutados. Un sistema de simulación que utilice un POL y que éste a su vez genere instrucciones en FORTRAN, tiene varias ventajas, destacando ante todo el hecho de que sólo serán llamados los módulos necesarios, además, el usuario puede agregar otras subrutinas con facilidad y el volumen de memoria puede ser predefinido con exactitud. A pesar de que se requiere un esfuerzo adicional para traducir, compilar y usar estos procedimientos, éste es retribuido al obtener una mayor flexibilidad y modularidad del sistema.

Cualquiera que sea la estructura empleada, es posible distinguir cuatro fases en la operación de un simulador: Fase de entrada, fase de preprocesamiento, fase de cálculo y fase de salida.

## A) FASE INICIAL

Esta es la interface más crítica entre el usuario y el sistema. En esta etapa se realizan tres actividades generales: introducción de información, verificación de datos y almacenamiento de los mismos.

La información requerida por el sistema se resume así:

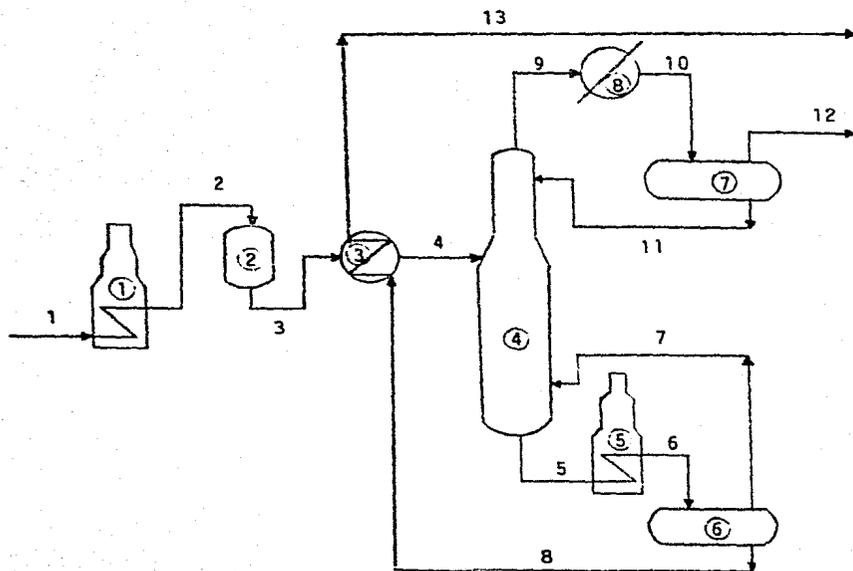
- a) Topología del proceso. Se indica la disposición e interconexión de los principales equipos junto con sus corrientes significativas, obteniendo el número mínimo de especies químicas que son representativas de los flujos del proceso.
- b) Corrientes de alimentación al proceso. Se introducen los valores de las variables significativas en las corrientes de alimentación al proceso, incluyendo sus propiedades termofísicas.
- c) Parámetros de equipo. Se introducen los parámetros de equipo que rigen el comportamiento de las unidades del proceso.
- d) Criterios de Convergencia. Se eligen los criterios y opciones que regirán la convergencia de los ciclos iterativos, métodos de solución de ecuaciones, técnicas de aceleración y orientación de valores hacia la solución, precisión deseada, tolerancia permitida, número máximo de iteraciones, etc.
- e) Orden de cálculo. Esta información no es requerida por todos los sistemas a la entrada, puesto que algunos realizan el análisis del proceso en la etapa de preprocesamiento, identificando recirculaciones, corrientes de corte y orden de cálculo para cada

ciclo y para el proceso en general. Los simuladores más simples no realizan estas tareas y esta información debe ser proporcionada por el usuario.

f) Parámetros de costo. Cuando el simulador realiza evaluación económica del proceso, se deben proporcionar los parámetros e índices de diseño, de funcionamiento, etc.

g) Opciones de cálculo de propiedades termofísicas. Se eligen los métodos para la estimación de propiedades termofísicas, ya sea como parte del conjunto de opciones que el simulador ofrece, o bien, agregándolas en forma de subrutinas compatibles con el sistema de simulación. También se identifican los datos y parámetros necesarios, extrayéndolos del banco de datos del simulador o proporcionados por el usuario. La elección de los métodos está en función del tiempo de cálculo disponible y de la precisión deseada, por lo tanto no se deben elegir a la ligera, tomando en cuenta que un alto porcentaje del tiempo de cómputo en una simulación se destina a estas estimaciones.

La información referente a la elección de opciones, indicación de límites (convergencia, recirculaciones, etc.), etc., se introduce por petición expresa del sistema; sin embargo, la información descriptiva del proceso debe sufrir una adaptación a partir del diagrama de flujo del proceso hasta obtener un problema aceptable por el sistema de simulación. En la Figura 5.1 se muestra un diagrama de flujo de un proceso que será usado como



① CALENTADOR

② REACTOR

③ INTERCAMBIADOR

④ COLUMNA DE DESTILACION

⑤ CALENTADOR

⑥ DIVISOR

⑦ DIVISOR

⑧ CONDENSADOR

FIGURA 5.1

DIAGRAMA DE FLUJO DE PROCESO  
USADO COMO EJEMPLO.

ejemplo. Este tipo de diagrama es de dominio de todo ingeniero químico, sin embargo, debe ser transformado a un diagrama de flujo de información, como el mostrado en la Figura 5.2, en el cual se sustituye el nombre del equipo, por el nombre del módulo unitario equivalente, y es así como la interconexión entre módulos representa tanto flujo de información como flujo de materia y energía. Una forma de transferir la topología del proceso en una estructura aceptable por un computador es construir la matriz del proceso. En la Figura 5.3 se muestra la matriz del proceso de ejemplo. En esta matriz, para cada unidad existe una lista de números de identificación de sus corrientes, siendo positivo para las corrientes de entrada y negativo para las de salida.

Más adelante se hará mención de otro tipo de transformaciones de la información, con el objeto de tener mayor eficiencia en su manejo computacional.

Es importante señalar que la transformación del diagrama de flujo de proceso (Figura 5.1), a un diagrama de flujo de información (Figura 5.2), no se realiza en correspondencia de uno a uno, puesto que dependiendo de los objetivos de la simulación, se deberán considerar un mayor número de subrutinas que el número de equipos; por ejemplo, se puede dar el caso de que una unidad sea descrita por varios módulos unitarios acoplados, o por lo contrario, se puede simplificar el proceso con representaciones muy simples, por ejemplo, se puede usar un módulo de "flasheo" a una presión y temperatura dadas puede usarse para simular una unidad

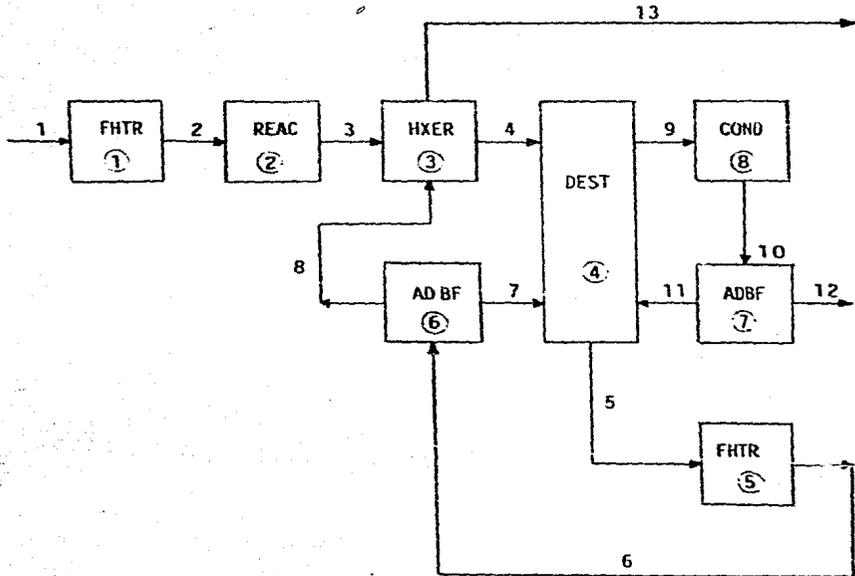


FIGURA 5.2 DIAGRAMA DE FLUJO DE INFORMACION DEL PROCESO DE LA FIGURA 5.1.

de compresión con uno o varios enfriamientos entre etapas.

## B) FASE DE PREPROCESAMIENTO.

Los límites entre la fase de preprocesamiento y la fase de cálculo puede parecer un poco difusa. En la fase de preprocesamiento esencialmente se realiza una adaptación final de la información, dándole una estructura adecuada para que la fase de cálculo se realice eficientemente.

Inicialmente se realiza una precompilación introduciendo los coeficientes para estimación de propiedades termofísicas; también se introducen los procedimientos especiales que no contiene el simulador y que son proporcionados por el usuario.

Una vez contenida toda la información se procede a compilar y a unir los diferentes procedimientos para dar forma al programa de simulación ejecutable.

En esta fase se realizan una serie de tareas vitales para la buena marcha de la simulación, especialmente en su etapa de cálculo. Estas tareas se refieren al análisis del proceso en cuanto a la identificación de recirculaciones, corrientes de corte y creación de una secuencia de cálculo apropiada.

Con referencia a la Figura 5.1, se tiene que la composición y temperatura de la corriente 3 pueden calcularse fácilmente si

UNIDAD	CORRIENTES		ASOCIADAS		
1	1	-2			
2	2	-3			
3	3	8	-4	-13	
4	4	7	11	-9	-5
5	5	-6			
6	6	-8	-7		
7	10	-11	-12		
8	9	-10			

FIGURA 5.3

MATRIZ DE PROCESO DE LA FIGURA 5.1

todos los datos de corrientes de alimentación y variables de diseño de los equipos 1 y 2, son conocidos; pero, la temperatura de la corriente 4 no se puede calcular porque no existe información de la corriente 8. El mismo caso se repite para las corrientes 11 y 7. El subsistema integrado por los equipos 3, 4, 5, 6, 7 y 8, que están conectados por la corriente de recirculación, es llamado el Sistema de Recirculación ó Ciclo Máximo y es resuelto mediante un esquema iterativo.

El procedimiento de identificar ciclos máximos y ordenar los equipos no incluidos en éstos, es lo que se conoce como Partición

Las corrientes que son elegidas para adquirir valores supuestos al inicio del esquema iterativo y servir como prueba de convergencia de un ciclo máximo, se conocen como corrientes de corte y se eligen de acuerdo a un procedimiento de corte, conocido también como descomposición. En el Apéndice B se amplían los conceptos de corte y descomposición.

Con el propósito de realizar la partición y descomposición, un diagrama de flujo de información se representa como una gráfica dirigida, en donde el equipo se representa por nodos (vértices) y las corrientes por líneas con una cierta dirección, como se puede observar en la Figura 5.4, que representa la gráfica dirigida del proceso de la Figura 5.2. La gráfica dirigida puede ser introducida al sistema en forma de una matriz adyacente la cual es una matriz cuadrada de tipo Booleano, en donde el número de filas y columnas es igual al número de nodos en la gráfica dirigida. Si existe una conexión del nodo  $i$  al nodo

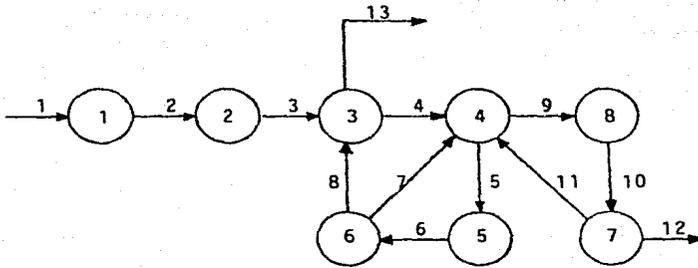


FIGURA 5.4. GRAFICA DIRIGIDA DEL PROCESO DE LA FIGURA 5.1.

j, entonces el elemento  $ij$  de la matriz adyacente es 1, en caso contrario será 0. En la figura 5.5 se muestra la matriz adyacente de la gráfica dirigida de la Figura 5.4.

Estos procedimientos se realizan una sola vez durante la simulación y por lo tanto aquí se debe transferir la mayor cantidad de información y decisiones lógicas que sea posible, puesto que del programa aquí compilado dependerá el éxito de los cálculos.

### C) FASE DE CALCULO.

Una vez que se tiene el programa de simulación ejecutable (previamente compilado y unido), el simulador está listo para realizar operaciones que generen resultados. Esta etapa se caracteriza por la aplicación repetitiva del programa ejecutivo de simulación (módulos unitarios, módulos de servicio, archivos, bases de datos, etc.). Precisamente la mayoría de los temas que se abordan en este trabajo están dirigidos a la implementación eficiente de la etapa de cálculo, por lo tanto aquí únicamente se abordarán aspectos de índole general.

Todo sistema de simulación incluye una dotación de módulos unitarios y paquetes de apoyo y servicio. En el caso de los módulos unitarios son bien conocidos y para la mayoría de ellos existen eficientes rutinas de cálculo riguroso, sin embargo, en la actualidad no se evalúa la sofisticación de un simulador por

UNIDAD	1	2	3	4	5	6	7	8
1		1						
2			1					
3				1				
4					1			1
5						1		
6			1	1				
7				1				
8							1	

FIGURA 5.5

MATRIZ ADYACENTE DE LA GRAFICA  
DIRIGIDA DE LA FIGURA 5.4.

el número de módulos que maneja, sino más bien en cuanto a su poder de representación y en la eficiencia de los cálculos realizados en ellos.

Se da como un hecho que los módulos unitarios más simples (mezcladores, divisores, determinadores de fase, bombas, etc.), son altamente eficientes y representativos, para compensar en parte el modesto poder de representación y velocidad de ejecución de módulos unitarios más complejos (destilación, fraccionamiento, reactores, etc.).

Sin embargo, aún el más completo módulo unitario será inútil si no se cuenta con herramientas adecuadas para resolverlo, es decir, que se debe contar con módulos de apoyo, tales como los paquetes termodinámicos, de solución de ecuaciones, etc.

Afortunadamente, la mayoría de los esfuerzos por mejorar los sistemas de simulación, tanto en desarrollo de modelos, como de algoritmos y estructuras, están destinados esencialmente a dar mayor eficiencia a la etapa de cálculo.

#### D) FASE DE SALIDA.

Este aspecto de la simulación es, tal vez, la que menos flexibilidad proporciona, puesto que básicamente proporciona información en etapas intermedias de cálculo e información final

de la simulación, incluyendo el desplegado de información acerca del estado del sistema de cómputo.

El formato de salida generalmente incluye una tabla en que se lista cada corriente indicando su composición, flujo, presión, temperatura, estado físico, y tal vez otro tipo de información como densidad, viscosidad, etc. Adicionalmente se presentan las especificaciones calculadas para los equipos (caídas de presión, dimensiones, número de etapas, requerimientos de servicios auxiliares, etc.).

También se incluyen los indicadores de tiempo de cómputo, recursos computacionales utilizados, volumen de memoria, número de iteraciones sobre el sistema, etc., junto con mensajes de localización y diagnóstico de fallas.

De este modo, el usuario básicamente se limita a indicar la frecuencia de las impresiones y la cantidad y tipo de la información que necesita.

### 5.3.1.2. Estructura de un Simulador Modular.

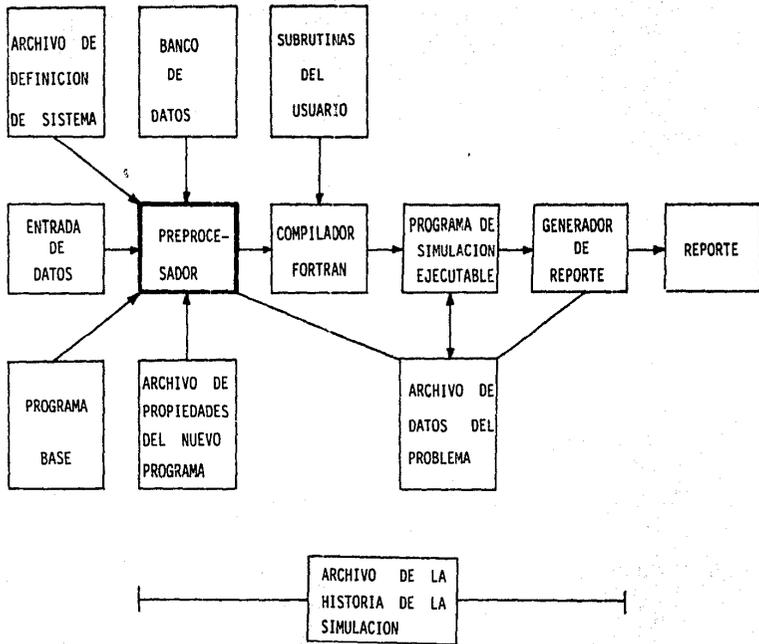
Hasta aquí se ha hecho referencia a un simulador de tipo modular, sin hacer alusión al carácter secuencial o simultáneo, puesto que estas etapas son comunes para ambos casos.

Es precisamente en la estructura del programa principal de simulación, específicamente en la subrutina de llamado de los módulos, en donde radica la diferencia entre un simulador modular secuencial y uno modular simultáneo.

Existen varias configuraciones para los simuladores modulares secuenciales, dependiendo de su grado de sofisticación, sin embargo, es posible alcanzar cierto grado de generalización en base a la descripción de un simulador de propósito general de arquitectura avanzada, como lo es ASPEN (1980, 1984), cuyo arreglo se muestra en la Figura 5.6.

El primer paso consiste en captar la información descriptiva del proceso, de acuerdo a los protocolos propios de este simulador. ASPEN utiliza un preprocesador, el cual interpreta los datos de entrada, los agrega a un programa base y los enriquece adicionando información proveniente de sus bancos de datos, archivos para definición de procesos y subrutinas especiales, todo lo cual conforma un programa mejorado y de particular aplicación al proceso que se va a simular. Un compilador FORTRAN se encarga de revisar el nuevo programa, agregando algunas subruti-

FIGURA 5.6. ESTRUCTURA GENERAL DEL SIMULADOR ASPEN.



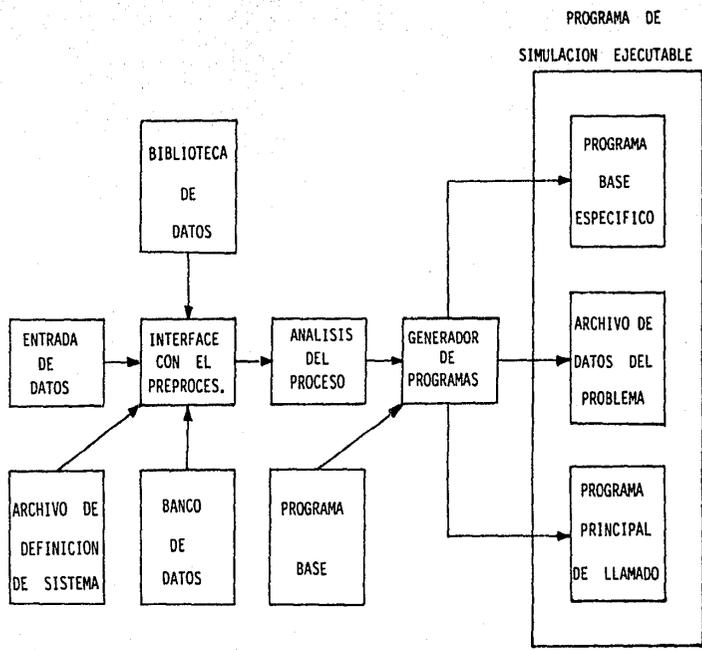
nas proporcionadas, tanto por el usuario, como por el sistema ASPEN, creando un Programa de Simulación Ejecutable (PSE).

El Programa Ejecutable es el núcleo del sistema de simulación. Mediante la ejecución repetida y controlada de este programa se van generando los resultados deseados, los cuales se van archivando para elaborar el reporte final. Durante el transcurso de la simulación, desde la introducción de datos hasta la generación del reporte final, se realiza una historia de los cálculos y de los mensajes de diagnóstico emitidos, formando un archivo especial cuyo contenido es reportado al final de la corrida con el objeto de entenderla y perfeccionarla.

Se debe resaltar un hecho importante con respecto a la creación del PSE. Es deseable que el sistema de simulación genere su propio programa principal de control, es decir, que a partir de la incorporación de los datos de definición del proceso al programa base, se realicen una serie de modificaciones y ajustes sucesivos para elaborar el nuevo programa hasta obtener finalmente el PSE.

En la Figura 5.7 se muestra en mayor detalle la estructura del preprocesador de ASPEN, con el objeto de ejemplificar la creación del PSE. Definido el proceso y elegidas las opciones de operación, el preprocesador ordena la información y realiza un análisis, obteniendo la estructura de las recirculaciones, las corrientes de corte y la secuencia de cálculo. El Generador de Programa agrega toda esta información al programa base y

FIGURA 5.7. ETAPAS EN LA GENERACION DEL P.S.E. EN EL PREPROCESADOR DE ASPEN.



crea un Programa de Simulación Ejecutable específico para ese proceso, compuesto de tres elementos principales: un programa base particular (que contiene los módulos unitarios y de servicio que requiere ese proceso, en particular), un programa principal de llamado de módulos y subrutinas y un sistema de manejo de archivos y bancos de datos.

El programa principal de llamado opera como un director de orquesta, el cual llama a los módulos conforme los va necesitando, les transfiere información y llama a las subrutinas que los resolverán, y una vez resueltos, envía estos datos a un archivo, retornando el control de la simulación al programa base particular, y así sucesivamente hasta resolver todos los módulos.

En el caso particular del simulador ASPEN, el PSE tiene una ventaja muy importante: mantiene en dos niveles separados el llamado de cada uno de los módulos, y la solución de éstos.

Esto es muy valioso cuando se trata de implementar un sistema modular simultáneo a partir de un simulador modular secuencial ya existente, puesto que el programa de llamado creado en el Generador de Programa no tiene que ser directamente modificado para lograr la conversión modular simultánea.

En el simulador ASPEN, existe una subrutina llamada SEQMON la cual se encarga de controlar los cálculos en la versión modular secuencial, y que no es modificada en absoluto cuando se aplica el preprocesador para crear el programa base particular; por lo tanto, si la subrutina SEQMON es estructuralmente modi-

ficada con el objeto de que realice las funciones de un programa de control simultáneo, tal y como se describe en la sección 5.2.3, se tendrá una versión modular simultánea, sin alterar el resto del PSE (estructura de datos, módulos unitarios, modelos y subrutinas de solución, etc.).

Más adelante se podrá captar, en toda su dimensión, la importancia de estas características, sobre todo cuando se trata de implementar el concepto modular simultáneo sobre estructuras modulares secuenciales más rígidas.

## 5.2. ORIENTACION DE LAS ESTRUCTURAS MODULARES SIMULTANEAS.

### 5.2.1. Antecedentes.

El grado de madurez que han alcanzado los modelos y algoritmos modulares simultáneos y el cúmulo de información que se ha venido generando desde los años sesentas, hacen imperativa la creación de estructuras lo suficientemente flexibles y poderosas como para poder probar adecuadamente esos algoritmos y compararlos sobre bases equivalentes, apreciar sus ventajas y desventajas y estar en posición de sentar las pautas para un desarrollo futuro hacia las direcciones más promisorias, tratando de cubrir las necesidades presentes y futuras.

La creación de simuladores de procesos eficientes requiere una conjunción de recursos materiales, de tiempo y humanos muy grande. La prueba más contundente de esta afirmación es el hecho de que a nivel comercial, en la actualidad, no existe todavía ninguna implementación realmente eficiente del enfoque Modular Simultáneo. Si bien es cierto, se conocen cuando menos dos trabajos avanzados al respecto, representados por Jirapongphan (1981), apoyado en el simulador FLOWTRAN y Treviño-Lozano (1985), basado en ASPEN PLUS, los cuales están aún en su etapa de prueba, presentando todavía algunos contratiempos, como para hacerlos competitivos con los simuladores modulares secuenciales, de probada eficacia, aunque con crecientes limitaciones. De hecho, un simulador modular simultáneo total, aún no existe a nivel comercial, sino que aún se ensaya con híbridos, en mayor o menor grado, de un modular secuencial.

A nivel académico existen varios simuladores modulares simultáneos con una participación secuencial mínima, sin embargo se trata de sistemas muy simples y destinados exclusivamente a probar una determinada estructura en particular, lo cual los convierte en simuladores muy rígidos y limitados, con pocas posibilidades de apertura a la inclusión de algunas técnicas avanzadas en el manejo de ecuaciones. Un ejemplo de este tipo de sistemas es SIMMOD, creado por Chen (1982).

Aún cuando en la actualidad no están claramente definidas las sendas por las que se desarrollarán las estructuras modulares

simultáneas, de manera general se pueden identificar dos posiciones opuestas. Una implica el diseño y puesta a punto de un simulador totalmente nuevo, desarrollado específicamente para realizar cálculos con la arquitectura que impone un método simultáneo en particular. Otra posición pretende utilizar la estructura modular secuencial como eje central del nuevo simulador, con una mayor o menor modificación del programa principal de control secuencial, agregando una serie de subrutinas que orienten el control de los cálculos en forma simultánea.

Antes de continuar, es importante resaltar algunos hechos. Existe mucho adelanto en el modelado de operaciones unitarias bajo el enfoque modular secuencial, de hecho, se han creado eficientes modelos para la representación rigurosa de las principales operaciones unitarias, por lo tanto, cuando se habla de desarrollar un simulador modular simultáneo total o parcialmente independiente del enfoque modular secuencial, se refiere única y exclusivamente a la estructura del simulador, más no significa abandonar los modelos rigurosos del desarrollo secuencial, los cuales son el producto de más de 25 años de investigación al respecto. Los modelos rigurosos conocidos son sustancialmente eficientes y cumplen adecuadamente el propósito para el que fueron creados.

Tal como se reconoció en el Capítulo 2, lo que se pretende con el enfoque modular simultáneo esencialmente es acelerar la convergencia y agilizar el manejo de restricciones de diseño, ampliando su capacidad para incluir otro tipo de tareas, como lo

son la simulación de procesos complejos y la optimización, sin tener que sacrificar la precisión que ya se tiene actualmente bajo el enfoque modular secuencial.

Una vez aclarado lo anterior, es posible reconocer que las ventajas de diseñar un nuevo simulador son, ante todo, proporcionar facilidad en su empleo y asegurar cálculos eficientes, puesto que toda su estructura estará orientada a implementar adecuadamente cierto método modular simultáneo en particular; sin embargo, los recursos que se necesitan para desarrollar un sistema de simulación totalmente nuevo son demasiado altos y se prescindiría de las ventajas que ciertas áreas de la estructura secuencial tienen; por lo tanto, esta opción no presenta mucho atractivo, fuera de los círculos académicos y, aún dentro de éstos, su uso está dirigido principalmente a probar modelos y estructuras muy simples, más que a conformar un simulador de propósito general.

En el caso contrario, emplear rígidamente la estructura modular secuencial permite una implementación del enfoque simultáneo relativamente simple y con un gasto moderado de recursos; sin embargo, el sistema resultante será poco flexible, puesto que al utilizar la estructura secuencial como eje central del nuevo simulador. Los cálculos serán guiados por módulos de control (que en realidad son módulos para el manejo de especificaciones de diseño) y se debe recordar que el uso excesivo de estos módulos constituye una de las desventajas, en cuanto a la eficiencia de cálculo, de los simuladores secuenciales, especialmente cuando

se abordan problemas de diseño. En virtud de lo anterior, gran parte de la eficiencia de los cálculos que se pretende con un sistema simultáneo, se pierde. Se ha planteado como posible solución a este problema el que sean proporcionadas varias opciones con el objeto de que el usuario controle la dirección de los cálculos, sin embargo, en la mayoría de los casos no es posible conocer a-priori que opción es la más adecuada para cada problema en particular, por lo tanto, el hecho de definir totalmente la estrategia de solución puede ser una tarea aún más difícil que alcanzar su solución, además, el esquema de solución para un determinado problema es únicamente aplicable a él y difícilmente se haría extensivo a otros casos.

Ante este panorama surge una tercera posición, que viene a conciliar algunos aspectos de las dos anteriores.

Esta posición propone el uso de un simulador modular secuencial ya existente y desarrollar e implementar una interface eficiente que lo vincule a un sistema simultáneo, el cual estará orientado al manejo eficiente de las ecuaciones. Para lograr este objetivo se deben realizar modificaciones mínimas necesarias en ciertas áreas del programa principal de control del sistema original, para que la interface realice un flujo de información eficiente entre el sistema secuencial y el sistema simultáneo. La parte simultánea esencialmente funcionará como un acelerador de convergencia, promoviendo la eficiencia de los cálculos mediante la incorporación de las más avanzadas técnicas de solución de ecuaciones lineales y no lineales, propias de los enfoques ba-

sados en las ecuaciones, técnicas que no pueden ser incorporadas directamente a los sistemas secuenciales en virtud de características propias de su estructura y jerarquización de los cálculos, características que fueron discutidas ampliamente en los capítulos 3 y 4. La desventaja de esta posición es que son pocos los simuladores modulares secuenciales ya existentes en los que se pueden hacer estas modificaciones del programa de control; en la mayoría de ellos, realizar estos cambios implica rediseñar una buena parte del programa de simulación, especialmente en aquellos simuladores que no separan las tareas de llamar a los módulos y resolverlos, en dos niveles independientes.

A continuación se presenta un breve desarrollo de cada una de las posiciones anteriormente aludidas. Se debe aclarar que en forma arbitraria se han designado como estructuras A, B y C.

#### 5.2.2. ESTRUCTURA TIPO A.

Esta estructura representa el diseño de un simulador modular simultáneo totalmente nuevo, utilizando aspectos secuenciales únicamente en cuanto a la captación de datos, análisis y generación de ecuaciones descriptivas del proceso y obtención de valores iniciales para las variables de iteración. También hace uso de los modelos rigurosos para operaciones unitarias, que for-

man parte del desarrollo secuencial, sin embargo, su solución es obtenida en una forma totalmente orientada a las ecuaciones, bajo un enfoque modular simultáneo.

Un ejemplo representativo de esta estructura es el simulador SIMMOD, creado por Chen (1982), y que fué diseñado específicamente para probar su particular concepción del enfoque modular simultáneo y demostrar la posibilidad de incluir técnicas poderosas en la solución de las ecuaciones de simulación y optimización

Este es un simulador muy simple, si es comparado con la compleja estructura de ASPEN; sin embargo, presenta varias etapas que son equivalentes, según se puede apreciar en el diagrama de flujo de información, mostrado en la Figura 5.8.

El programa principal hace el llamado de la subrutina ENTPM encargada de captar la definición del proceso, opciones de cálculo, etc. A continuación llama a la subrutina PTEAR que realiza el análisis del proceso, generando las corrientes de corte y una secuencia de cálculo. Existe la subrutina SAVEPM, la cual archiva los resultados de toda la corrida y los manda imprimir. Con toda esta información, el programa principal llama a la subrutina SIMMOD, la cual se encarga de dirigir los cálculos. La subrutina SIMMOD primeramente llama a cada módulo, con el objeto de checar datos; si nota alguna anomalía grave envía mensajes de diagnóstico y si se trata de un error fatal, da por terminada la corrida; en caso de no encontrar errores, inicia la etapa de cálculo de la simulación. En la Figura 5.9

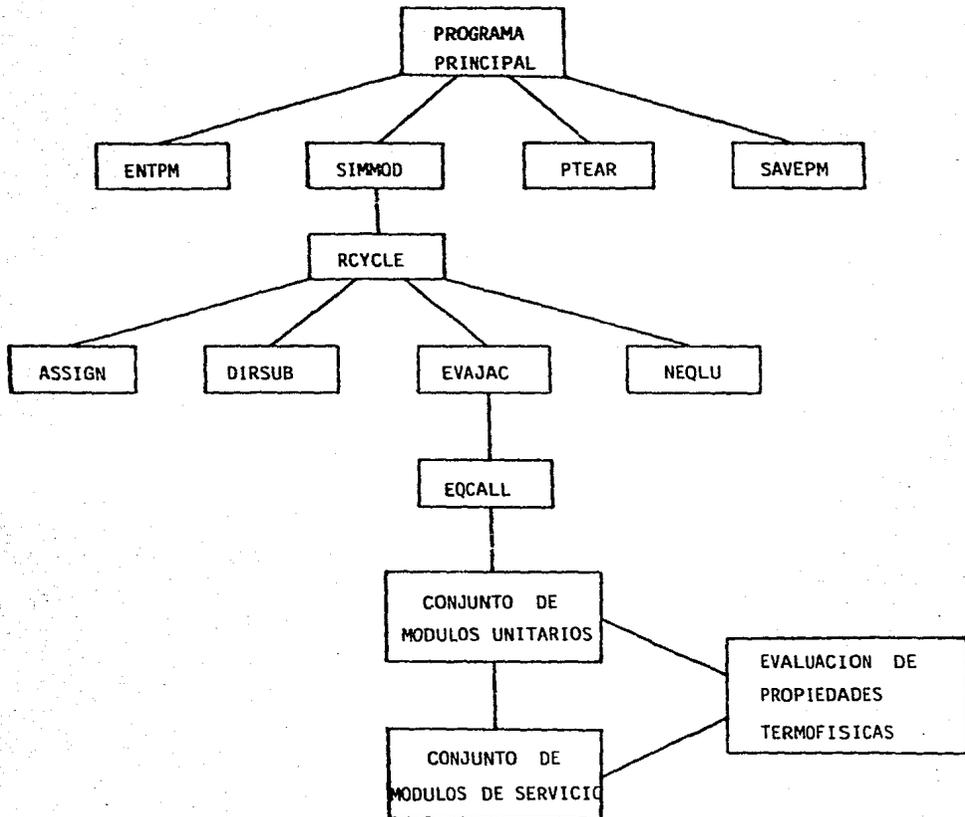


FIGURA 5.8. FLUJO DE INFORMACION EN EL SIMULADOR MODULAR SIMULTANEO SIMMOD.

se muestra la estructura de la subrutina SIMMOD, la cual orienta los cálculos bajo un enfoque modular simultáneo total.

Cuando la subrutina SIMMOD ha sido llamada, ya fueron realizadas tareas tales como definición y análisis del proceso, así como la introducción de valores iniciales en las variables de iteración. También se han realizado un determinado número de recorridos modulares secuenciales a través del proceso, con el objeto de fijar bases de cálculo razonables. En estos recorridos no se incluyen las especificaciones de diseño y no se requiere demasiada precisión en los cálculos, puesto que no se busca la convergencia del proceso, sino únicamente encontrar valores para ser utilizados como punto de partida.

Una vez concluidos los recorridos secuenciales, SIMMOD llama a la subrutina RCYCLE, que analiza cada bloque irreducible, siguiendo la secuencia de cálculo previamente definida. Si el bloque irreducible sólo contiene una unidad y no existen especificaciones de diseño, entonces se resuelve el módulo, almacenando sus ecuaciones y variables; en caso de existir varias unidades y especificaciones de diseño, RCYCLE llama a la subrutina ASSIGN que identifica y numera consecutivamente todas y cada una de las variables y ecuaciones del bloque, retornando el control a RCYCLE que llama a la subrutina DIRSUB con el objeto de realizar uno o varios recorridos utilizando sustitución directa dentro del bloque irreducible que se está resolviendo, sin incluir todavía las especificaciones de diseño. Esto conduce a obtener valores iniciales más precisos para las variables de las corrientes

de corte. Una vez obtenidos estos valores, se agregan las especificaciones de diseño con el objeto de completar el conjunto de las ecuaciones de simulación, regresando el control a RCYCLE, quién llama a la subrutina NEQLU que realiza las iteraciones modulares simultáneas, resolviendo las ecuaciones no lineales con la ayuda de la subrutina EVAJAC, que da forma al Jacobiano del bloque irreducible y lo resuelve aplicando diferenciaciones; se escalan las variables para que adquieran órdenes de magnitud semejantes y se puedan hacer converger bajo la misma tolerancia. A continuación se aplica el o los métodos de solución de ecuaciones algebraicas no lineales, para los conjuntos de ecuaciones de conexión y especificaciones de diseño, en forma simultánea. Se verifica la convergencia, si no es adecuada su velocidad, con los valores calculados se estructura un nuevo Jacobiano y se aplica diferenciación, reanudando el ciclo de cálculo. Si no se logra la convergencia, pero la velocidad de avance es adecuada, entonces se redefine el problema aplicando un esquema apropiado de generación de nuevos valores y se aplica el método de solución de ecuaciones. Cuando se logra la convergencia de ese bloque irreducible, se desescalan las variables, se realiza un recorrido secuencial del bloque y se llama al módulo de identificación de fase de las corrientes, con lo que se obtiene la solución total del ciclo. A continuación se verifica si existe otro bloque irreducible, de ser así, se repite el procedimiento hasta aquí descrito, según se muestra en la Figura 5.9.

Se debe destacar que esta estructura no realiza recorridos

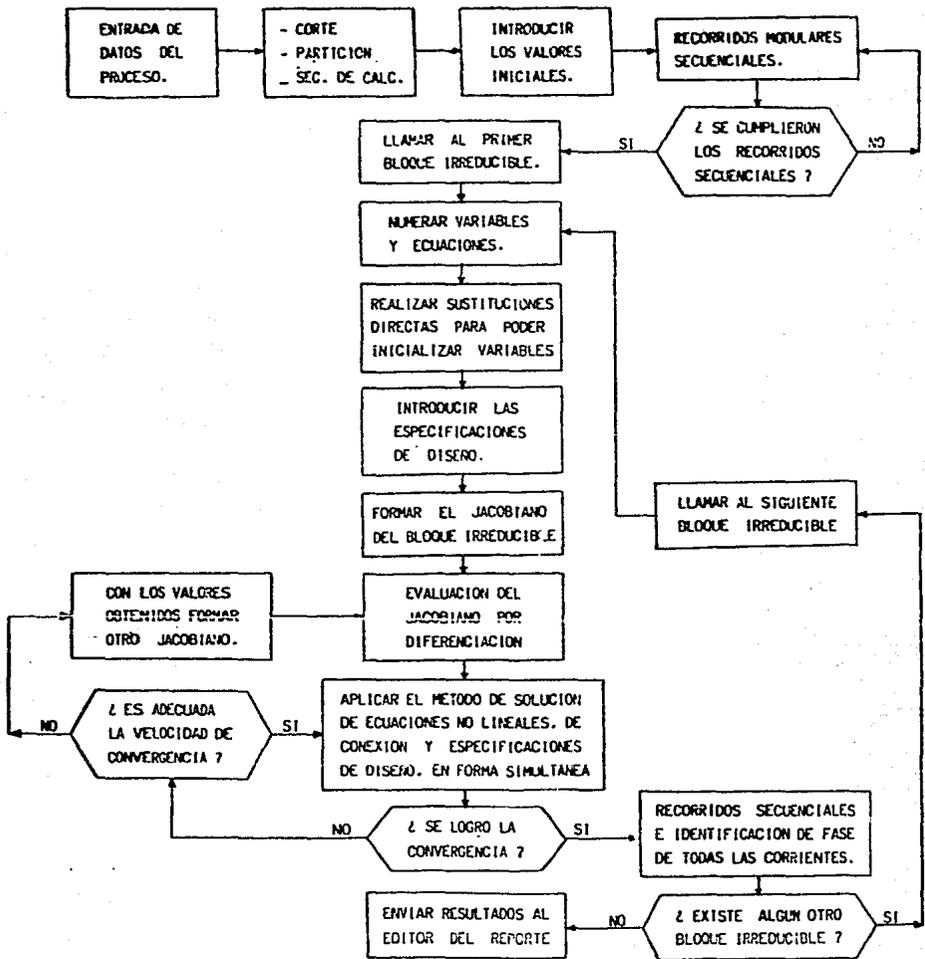


FIGURA 5.9. DIAGRAMA DE FLUJO DE INFORMACION PARA LA ESTRUCTURA A.

secuenciales periódicos con el objeto de reorientar los cálculos en la dirección correcta hacia una solución, sino que recurre exclusivamente a técnicas matemáticas, como alguno de los tipos de diferenciación indicados en el Capítulo 3, a fin de hacer avanzar los cálculos.

Esta estructura no presenta modelos simples específicos para cada unidad, sino que estos son desarrollados mediante un esquema de perturbación de los modelos rigurosos, en forma implícita dentro de las subrutinas NEQLU y EVAJAC. Para más detalles acerca del método Modular Simultáneo que utiliza este simulador, consultar los Capítulos 3 y 4, puesto que aquí únicamente se hace referencia a la estructura del programa de simulación.

### 5.2.3. ESTRUCTURA TIPO B.

Este tipo de estructura no requiere la creación de un simulador totalmente nuevo, puesto que su implementación se realiza tomando como base el programa de un simulador modular secuencial ya existente, al cual se le agregan un conjunto de módulos de control auxiliares, destinados a dirigir los cálculos conforme a un algoritmo modular simultáneo, sin modificar la estructura del programa principal secuencial.

Un ejemplo de esta estructura es la implementación realizada por Jirapongphan (1980) sobre el simulador modular secuencial FLOWTRAN. Para mayores detalles acerca de los programas de FLOWTRAN, se recomienda consultar los trabajos de Seader, Seider y Pauls (1977) y Rosen y Pauls (1977).

Jirapongphan desarrolló dos algoritmos de tipo modular simultáneo: uno lineal de tipo gradiente, y otro no lineal con modelos reducidos, los cuales fueron descritos ampliamente en el Capítulo 4. En ambos casos prevalece la idea central en cuanto a la estructura de su implementación: agregar bloques de control auxiliares al programa principal, con el objeto de guiar los cálculos en forma simultánea.

#### 5.2.3.1. Estructura para el algoritmo lineal.

Las rutinas de control de preprocesador de FLOWTRAN deben ejercer dos niveles de control, según los requerimientos de este algoritmo. El primer nivel se destina al control de las iteraciones (tanto para inicializarlas, como para realizar una búsqueda lineal o resolver la matriz Jacobiana). El segundo nivel, es para conducir las perturbaciones numéricas y reejecución de cada módulo unitario riguroso y calcular los elementos de la submatriz Jacobiana para esa unidad, con respecto a cada perturbación. La relación entre estos niveles de control, con los módulos unitarios y las rutinas de solución, se muestran en la Figura 5.10.

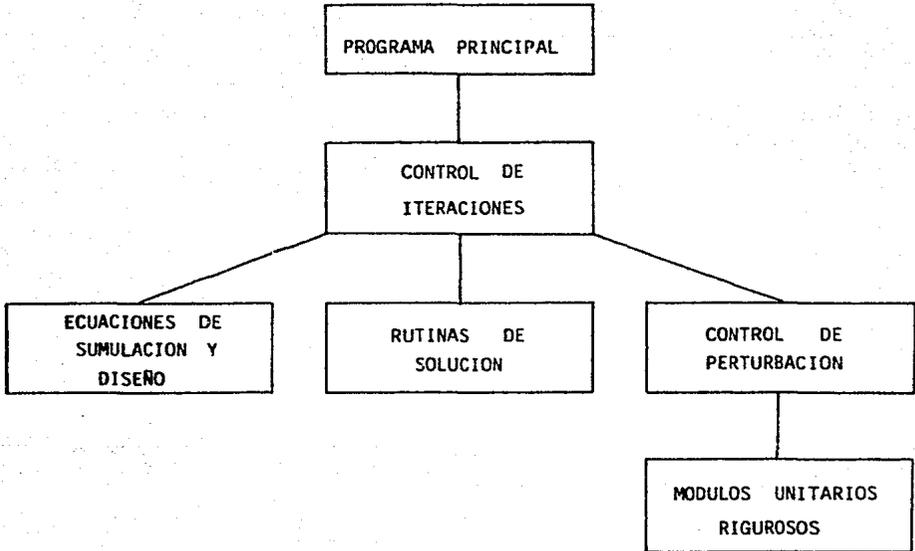


FIGURA 5.10. RELACION ENTRE LOS NIVELES DE CONTROL DE FLOWTRAN PARA EL ALGORITMO MODULAR SIMULTANEO LINEAL.

El control de iteraciones transfiere la ejecución al módulo unitario que inicia la secuencia de cálculo de un determinado ciclo cuando no se ha satisfecho la convergencia, indicando si la iteración es una búsqueda lineal (sólo sobre los residuos), o si se trata de una iteración completa (sobre los residuos y sobre la matriz Jacobiana).

El control de la perturbación se realiza a través de un módulo de control multivariable, el cual transfiere la ejecución hacia un determinado módulo unitario para realizar su perturbación, tal y como fué desarrollado por Kluzik (1979). Sin embargo, esta implementación requiere una cantidad relativamente grande de parámetros de equipo y variables de las corrientes, con el fin de llevar a cabo la perturbación.

Jirapongphan desarrolló un método diferente, con el objeto de reducir el número de parámetros y variables necesarios, proponiendo el uso de un módulo lineal de interface, el cual es específico para cada módulo unitario, y un módulo de transferencia de control hacia el módulo unitario que se va a perturbar.

Puesto que cada módulo de interface es escrito específicamente para cada tipo de unidad, la relación entre las variables de entrada y salida durante las perturbaciones se conoce explícitamente. La relación funcional entre el módulo de interface (INTER), el módulo de transferencia (TRANS), los módulos unitarios rigurosos y el módulo de control de iteraciones (llamado FINAL), se muestra en la Figura 5.11.

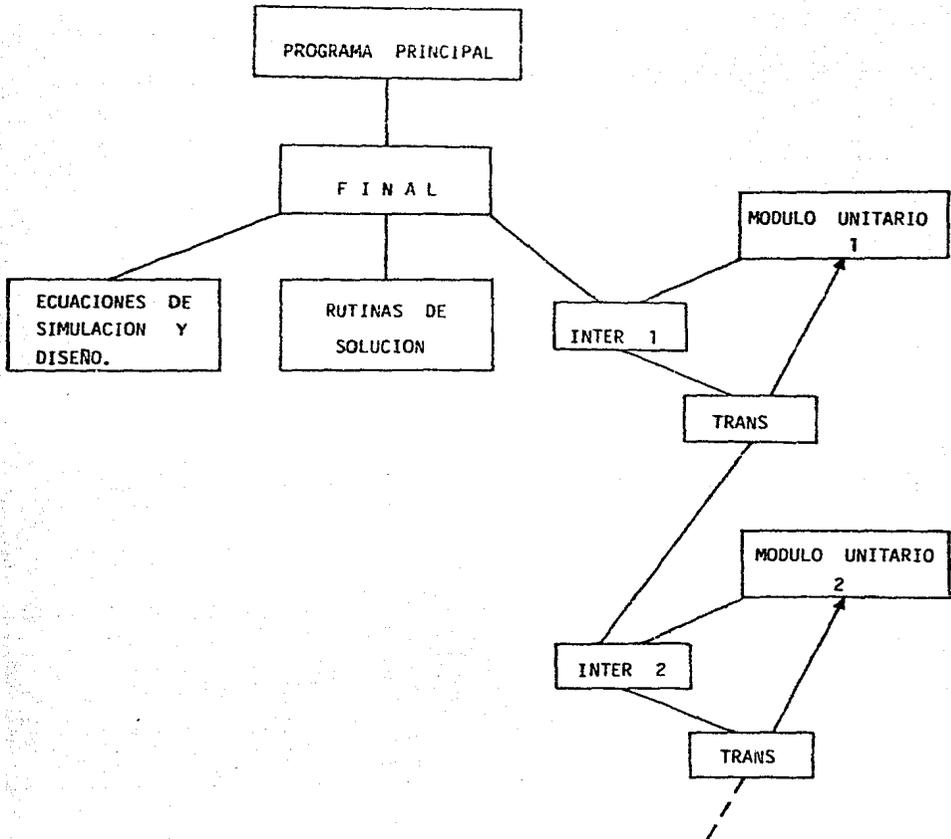


FIGURA 5.11 RELACION FUNCIONAL DEL ALGORITMO MODULAR SIMULTANEO LINEAL DE JIRAPONGPHAN.

El módulo de interface decide, con respecto a cada variable y parámetro de la unidad que representa, primeramente que corriente de entrada será perturbada y segundo, de la corriente elegida, que variable será la que se perturbará. Una vez decidido esto, se transfiere el control al módulo TRANS para que llame al módulo unitario. Si la perturbación ya ha sido terminada, el módulo TRANS pasa el control al siguiente módulo INTER para preparar las perturbaciones del siguiente módulo unitario, hasta completar el ciclo.

Se debe reconocer que el escribir módulos INTER específicos para cada módulo unitario es una tarea ardua; el esfuerzo es recompensado ampliamente, puesto que esta labor se realiza una sola vez y conduce a significativos ahorros en tiempo de cómputo.

#### 5.2.3.2. Estructura para el Algoritmo No Lineal.

El algoritmo no lineal fué desarrollado bajo un enfoque de ciclo interno-ciclo externo, como se describió en el Capítulo 4.

El ciclo externo está representado por la ejecución de los módulos unitarios rigurosos, con el objeto de generar los parámetros de los modelos reducidos, mientras que el ciclo interno resuelve las ecuaciones que constituyen el problema reducido.

Los módulos de control del simulador FLOWTRAN deben controlar las iteraciones tanto del ciclo interno como del ciclo ex-

terno; guiar la generación de los coeficientes del modelo reducido para cada unidad y resolver el problema reducido, tal y como se muestra en la Figura 5.12. Nuevamente se requieren dos niveles de control: uno para el control de las iteraciones del proceso completo y otro para controlar las ejecuciones de cada módulo unitario.

Jirapongphan propone una estructura basada en la adición de módulos de control auxiliares, tal y como lo hizo con el algoritmo lineal. El control de las iteraciones sobre el proceso se realiza con un módulo que transfiere la ejecución al módulo unitario que encabeza el orden de cálculo, cuando la convergencia aún no se logra, ya sea en los ciclos de cálculo interno y externo. Se utiliza un índice para indicar el tipo de iteración; en el caso de las iteraciones sobre el ciclo interno, se emplea otro índice para mostrar si se trata de una iteración sobre la matriz Jacobiana completa o sólo una búsqueda lineal.

El segundo módulo de control, aplicable a cada módulo unitario, es implementado como una combinación de módulo, exclusivo de cada tipo de módulo unitario, y un módulo de transferencia de control aplicable tanto al módulo unitario riguroso, como a los demás módulos de interface.

El módulo de interface emplea los valores de las variables de entrada y salida, parámetros de equipo y variables del vector de retención del módulo unitario que representa y ejecuta tanto las rutinas de cálculo de los coeficientes del modelo reducido,

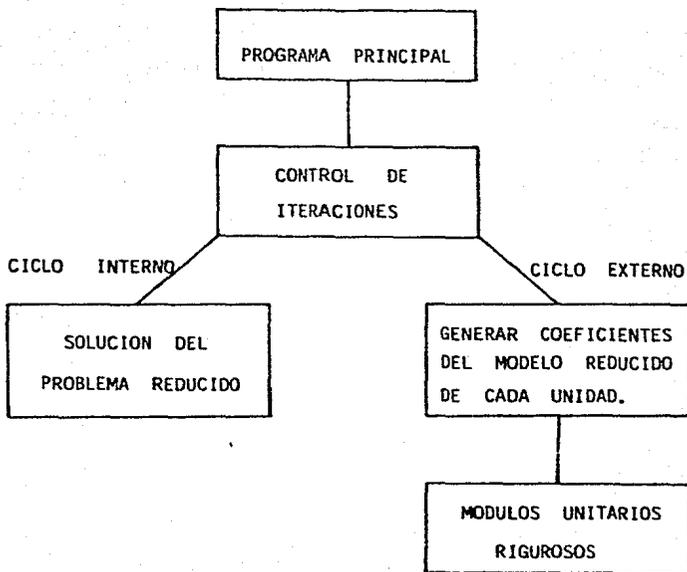


FIGURA 5.12. RELACION ENTRE LOS NIVELES DE CONTROL DE FLOWTRAN PARA EL ALGORITMO MODULAR SIMULTANEO NO LINEAL.

como las rutinas para calcular el valor funcional y la submatriz Jacobiana referentes a las expresiones del modelo reducido para una unidad en particular. La relación funcional entre el módulo de control de iteraciones (llamado LAST), el módulo interface (INTER #), el módulo de transferencia (TRANS), las rutinas del modelo reducido y los módulos unitarios rigurosos, se muestra en la Figura 5.13.

En los apéndices del trabajo de Jirapongphan (1980), se muestra en detalle tanto los requerimientos del sistema, como la estructura de datos de las dos estructuras aquí mostradas.

El hecho de agregar módulos de control auxiliares a un simulador secuencial existente, con el objeto de crear un simulador modular simultáneo, es más económico en recursos, comparado con el gasto de crear un simulador totalmente nuevo; sin embargo, se debe reconocer que esta estructura es muy rígida y susceptible de fallas de definición, puesto que el usuario debe definir los nuevos módulos de control específicos para cada operación unitaria y emplear una serie de enunciados para lograr su aceptación dentro del sistema. Además, se debe recordar que una de las mayores desventajas del enfoque modular secuencial es precisamente el uso de módulos de control, los cuales consumen excesivo tiempo de cómputo, y si a esto se agregan ciclos de control auxiliares, es muy posible que el tiempo de cómputo se eleve. La justificación para el uso de módulos de control auxiliares es la reducción en el número de parámetros y variables que debe manejar

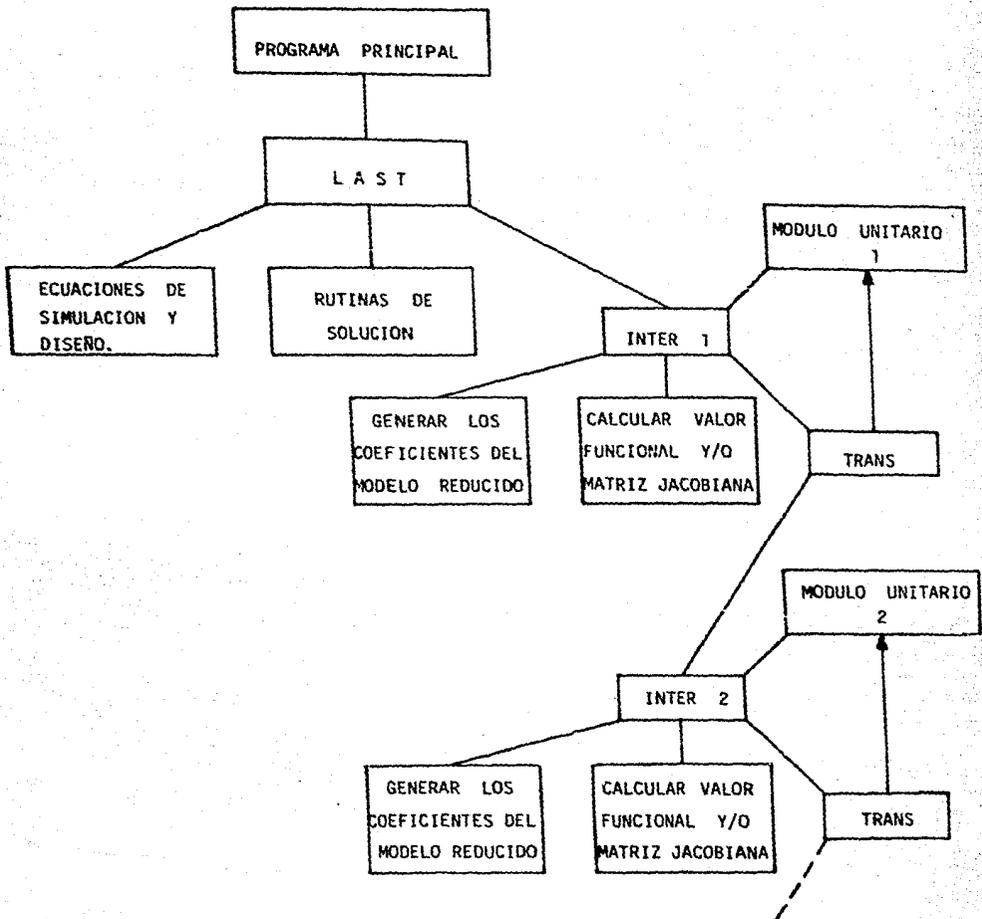


FIGURA 5.13 RELACION FUNCIONAL DEL ALGORITMO MODULAR SIMULTANEO NO LINEAL DE JIRAPONGPHAN.

el programa principal de control, lo cual deberá reducir el número de iteraciones necesarias para lograr la convergencia; sin embargo, es recomendable evaluar si la reducción en el número de iteraciones es significativo, y si en realidad cada una de esas iteraciones requerirá menor tiempo de cómputo, tomando en cuenta que los modelos reducidos por sí mismos son de un consumo de recursos bajo.

Este tipo de estructura modular simultánea es especialmente adecuada para su implementación sobre simuladores modulares secuenciales con una estructura de cálculo cerrada, es decir, para aquellos simuladores que manejan en un mismo nivel el llamado de los módulos y su solución, y cuyas respectivas subrutinas no son independientes y por lo tanto la orientación de los cálculos en forma simultánea se debe realizar con rutinas externas, tal y como fué planteado anteriormente.

#### 5.2.4. ESTRUCTURA TIPO C.

Esta estructura viene a conciliar varios aspectos de los dos enfoques anteriormente descritos. En la estructura tipo A, todo el programa de simulación está orientado a los cálculos modulares simultáneos. En cuanto a la estructura tipo B, no se modifica en forma directa el programa de control secuencial, sino

que utiliza un sistema de subrutinas (módulos), fuera del programa principal, para guiar los cálculos simultáneos.

Treviño-Lozano (1985), propone la implementación de una nueva estructura, en base al simulador modular secuencial ASPEN PLUS, mediante una modificación directa sobre la subrutina SEQMON la cual controla los cálculos durante la simulación. Esta modificación se realizó bajo el enfoque ciclo interno-ciclo externo.

Se sabe que el programa de control de un simulador modular secuencial realiza estas cuatro funciones principales:

a) Selecciona bloques irreducibles, corrientes de corte y diseña una secuencia de cálculo apropiada.

b) Inicializa corrientes de corte y corrientes de entrada.

c) Decide el orden de llamado de los módulos, de acuerdo a la secuencia de cálculo y tipo de ciclos de recirculación.

d) Verificar la convergencia de cada ciclo.

Treviño-Lozano plantea nuevas funciones asignadas al programa de control, de acuerdo a su concepción de un simulador modular simultáneo, resumiéndolas en:

a) Selecciona bloques irreducibles, corrientes de corte y diseña una secuencia de cálculo apropiada.

b) Inicializa corrientes de corte y corrientes de entrada.

c) Desarrolla recorridos secuenciales para obtener bases de cálculo razonables.

d) Dirige el cálculo en los bloques rigurosos (ciclo externo) para generar los parámetros de los modelos reducidos (ciclo interno).

e) Controla la generación de información requerida para hacer converger el ciclo interno (Jacobiano de las ecuaciones reducidas, gradiente de la función objetivo, residuales, etc.).

f) Controla la solución del problema reducido.

g) Verifica la convergencia del ciclo externo; si no se logra, transfiere las variables necesarias desde el ciclo interno hacia el ciclo externo.

h) Repite los pasos d, e, f y g, hasta lograr la convergencia del ciclo externo.

Al comparar las listas de las funciones propias de un programa de control para un simulador modular secuencial y para un modular simultáneo, parece que deben realizarse extensas modificaciones, sin embargo, esto no es del todo cierto, puesto que varias etapas son equivalentes.

En el nuevo simulador los módulos realizan otras funciones, aparte de la solución de las variables de salida a partir de las variables de entrada, y que se pueden agrupar en:

a) Generar los parámetros del modelo reducido.

b) Generar la matriz Jacobiana de las ecuaciones reducidas.

c) Evaluar los residuales de las ecuaciones reducidas.

En la sección 5.1.3., se resaltó la importancia de que el programa principal maneje en dos niveles separados el llamado de cada módulo y su solución, lo cual facilita la implementación de la estructura modular simultánea. En este caso, los ajustes y modificaciones se hacen directamente en la subrutina que controla los cálculos de la simulación con el objeto de que realice las nuevas funciones del programa principal de control y se pueda tener una versión modular simultánea a partir de un simulador modular secuencial. En la Figura 5.14 se muestra el flujo de información en la subrutina de control de cálculos (SEQMON), una vez implementada la estructura tipo C en ASPEN (ver Figuras 5.6 y 5.7).

Visto de este modo, esta subrutina decide que bloques deben ser ejecutados y en que orden, así como las opciones que operan sobre los modelos reducidos (lineales o no lineales); opciones para la convergencia de las variables de las corrientes (únicamente las corrientes de corte, o todas las corrientes del proceso, o ciertas corrientes seleccionadas, etc.); y el método de convergencia para los cálculos del ciclo externo (sustitución directa, con o sin aceleración, etc.). La convergencia del ciclo interno es monitoreada directamente por el método de solución que se eligió para el problema reducido.

Con referencia a la Figura 5.14, se puede notar que la introducción de datos y el análisis del proceso (para generar la partición en bloques irreducibles, corrientes de corte y secuen-

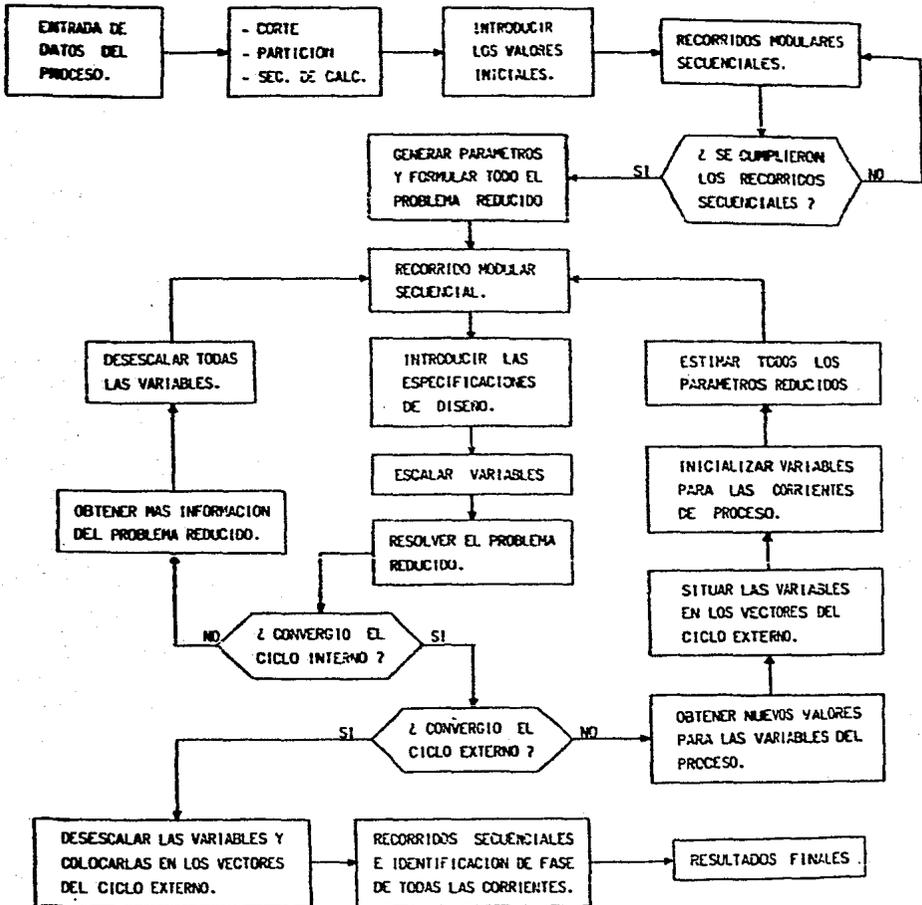


FIGURA 5.14. DIAGRAMA DE FLUJO DE INFORMACION PARA LA ESTRUCTURA C.

cia de cálculo), son equivalentes a las de un simulador modular secuencial.

La etapa de inicialización puede ser vista a dos niveles: primero, para evaluar la condición termodinámica de las corrientes de entrada, y segundo, para obtener valores iniciales tanto para las corrientes de corte, como para las variables internas de cada módulo. El primer caso se resuelve mediante un recorrido secuencial simple, sin incluir las restricciones de diseño, y con especial atención a los módulos de identificación de fase.

El segundo caso requiere elegir una determinada opción para resolver los modelos rigurosos (ciclo externo) y realizar un determinado número de recorridos secuenciales para obtener puntos de partida razonables para los cálculos simultáneos; sin embargo, estos recorridos no son tan rigurosos como en un simulador secuencial, puesto que no se pretende obtener la convergencia de cada ciclo de recirculación, basta con obtener valores aproximados. Esta estrategia equivale a usar el simulador modular secuencial para iniciar la convergencia del proceso a partir de un punto alejado de la solución, y una vez obtenido un acercamiento, cambiar al cálculo modular simultáneo y obtener la solución con el grado de precisión deseado.

El ciclo interno (problema reducido), comienza una vez que los parámetros reducidos han sido generados, usualmente obtenidos durante el último recorrido secuencial, y se reúnen los elementos para formular el problema reducido. El problema reducido conduce a la solución simultánea del sistema de ecuaciones reducidas del proceso, en el caso de la simulación. En la optimización

el problema reducido es un caso altamente restringido, en donde las ecuaciones reducidas del proceso son tratadas como restricciones de igualdad. En ambos casos, la idea es resolver el problema reducido en forma global, por lo tanto, los vectores y matrices que caracterizan el problema deben incluir todas las ecuaciones reducidas del proceso y no sólo las ecuaciones reducidas de un módulo individual.

Las especificaciones de diseño pueden tener cualquier forma, elegida por el usuario, y es recomendable manejarlas en forma de subrutinas separadas de las demás, puesto que la estructura de sus ecuaciones no se conoce a-priori y este arreglo puede brindar mayor versatilidad al simulador.

La última parte del problema reducido que necesita ser caracterizada es la función objetivo para optimización. Esta función puede ser manejada en la misma forma que una especificación de diseño. Visto de este modo, el problema de simulación puede considerarse como un caso especial del problema general de optimización. Una vez caracterizado el problema reducido, se procede a escalar sus variables para hacerlas converger bajo la misma tolerancia.

Al aplicar el método de solución elegido, se evalúa la convergencia del ciclo interno; si esta no se logra, se procede a desescalar las variables y los valores obtenidos en la última iteración se utilizan para realizar un recorrido secuencial con el objeto de obtener información más precisa acerca del problema reducido y repetir el procedimiento de cálculo descrito.

Una vez que el ciclo interno ha convergido se tienen nuevos valores de las variables de proceso; por lo tanto, la convergencia del ciclo externo debe ser evaluada. Una forma de buscar esta convergencia es calcular la diferencia entre el valor de las variables del ciclo interno calculadas durante dos iteraciones consecutivas del ciclo externo. Una tolerancia se puede definir en términos de la diferencia máxima entre las variables escaladas o en términos de la norma de las diferencias de todas las variables escaladas.

Si la convergencia no se alcanza, es preciso calcular nuevos estimados de las variables del ciclo externo, ya sea por sustitución directa o aplicando el método de Wegstein (1958). Estos nuevos valores deben ser colocados en los vectores de las corrientes y de los parámetros de equipo de los modelos rigurosos originales (sección modular secuencial) y las corrientes de proceso necesitan ser reinicializadas para obtener toda la información que se necesita en la solución de los métodos rigurosos. A partir de la ejecución de los modelos rigurosos se obtienen nuevos parámetros reducidos para reiniciar otra iteración del ciclo interno. Ahora bien, si la convergencia del ciclo externo se logra, entonces cada valor obtenido es colocado en los respectivos vectores de las corrientes y de parámetros de equipo y las variables adquieren su orden de magnitud original. Se realiza un último recorrido secuencial, incluyendo módulos de identificación de fase para todas y cada una de las corrientes de proceso, obteniéndose así la definición de todo el sistema.

Una característica muy importante de este tipo de estructura es que realiza recorridos secuenciales periódicos, con el objeto de reorientar los cálculos en la dirección correcta hacia la convergencia. Al igual que la estructura de tipo B, aquí se adopta el uso de modelos reducidos específicos para cada operación unitaria, sin recurrir a la diferenciación directa sobre los modelos rigurosos.

No se debe pasar por alto que la totalidad del anterior desarrollo ha sido implementado sobre la subrutina que controla los cálculos, sin recurrir al uso de subrutinas especiales que dirijan los cálculos, es decir, la modificación fué realizada directamente sobre la subrutina SEQMON, de ASPEN.

C A P I T U L O      6

EL CONCEPTO MODULAR SIMULTANEO  
DEL SISTEMA SMS/ZAR .

## CAPITULO 6

## EL CONCEPTO MODULAR SIMULTANEO DEL SISTEMA SMS/ZAR.

Con la información contenida en los capítulos anteriores, se tiene un amplio panorama acerca de las principales tendencias que exhibe el desarrollo del enfoque modular simultáneo dentro de la simulación de procesos, tanto en la creación de modelos de operaciones unitarias, que reúnan en forma funcional características de sencillez y poder descriptivo, como en el perfeccionamiento de algoritmos de solución y estructuras para el manejo de información que se beneficien de las innovadoras y poderosas técnicas de solución de ecuaciones no lineales y matrices dispersas, en forma armónica con los crecientes avances en materia de manejo computacional.

En este capítulo se propone la implementación de un simulador modular simultáneo a partir de un simulador modular secuencial ya existente. Primeramente se justifica esta decisión mediante un breve análisis de los fines que se persiguen y de los recursos con que se cuenta para realizar este trabajo. Acto seguido, se describen las características generales del simulador modular secuencial SGP/ZAR, sobre el cual se propone la conversión modular simultánea. Finalmente, se describe el entorno operativo del simulador modular simultáneo SMS/ZAR, identificando

el tipo de estructura susceptible de ser adaptada y las áreas de trabajo más significativas para lograr su implementación.

### 6.1. CONSIDERACIONES PRELIMINARES.

En el Capítulo 5 se reconoció la existencia de dos grandes tendencias en la concepción de un simulador modular simultáneo: una identificada con la creación de un simulador modular simultáneo totalmente nuevo; la otra, representando la adaptación de un determinado esquema modular simultáneo en base a un simulador modular secuencial ya existente y de probada funcionalidad.

Se hizo especial énfasis en el hecho de que la creación de un simulador modular simultáneo totalmente nuevo y orientado a implementar eficientes técnicas de solución de ecuaciones no lineales, implícitamente está dejando a un lado la robusta estructura y la excelente precisión del enfoque secuencial, en busca de mayor eficiencia computacional y la posibilidad de adoptar técnicas que incorporen parámetros de diseño y funciones objetivo, en una forma más manejable.

Otro aspecto, digno de tomarse en cuenta, muestra que la creación de un simulador modular simultáneo totalmente nuevo impone demasiada rigidez en cuanto a estructura, puesto que los intentos por crear simuladores de este tipo se destinan a imple-

mentar y probar técnicas simultáneas muy específicas y esta estructura no es aplicable a evaluar otras técnicas. Esto es muy importante si se toma en cuenta que el desarrollo modular simultáneo está en etapa de expansión, aún sin llegar a consolidarse firmemente, por lo tanto, es necesario crear estructuras flexibles y que permitan probar varias técnicas y sus variantes. El diseño de un simulador modular simultáneo totalmente nuevo exige el empleo de valiosos recursos, que es preferible canalizar hacia la obtención de una estructura eficiente bajo este enfoque, o cuando menos, a definir rutas de investigación más promisorias y fructíferas.

Ante esta situación, existe la opción generalizada para crear una base de simulación (generalmente con participación modular secuencial) y a partir de ahí generar una estructura modular simultánea y evaluar su comportamiento.

No se descarta del todo la idea de crear sistemas simultáneos totalmente nuevos, puesto que una vez reconocida una ruta promisoriosa, será necesario diseñar simuladores de este tipo para explotar sus beneficios, sin embargo, mientras no se definan estas sendas de desarrollo, esta opción aún no goza de amplia aceptación, aún dentro del campo académico, puesto que ante la opción de crear simuladores simultáneos totalmente nuevos, se prefiere diseñar simuladores orientados a las ecuaciones y fortalecer así una base de investigación y desarrollo realmente competitiva con el enfoque modular simultáneo.

Aunado a lo anterior, se debe tener en cuenta que la creación de un simulador totalmente nuevo implica que no se obtendrán resultados prácticos en el corto plazo y esto es de suma importancia cuando los recursos para su desarrollo son limitados y proporcionados de acuerdo a la evaluación de resultados, tal y como sucede en la mayoría de las instituciones, tanto educativas como comerciales. Por el contrario, implementar un sistema simultáneo sobre un simulador secuencial ya existente, permite generar resultados en el corto plazo e irlos refinando en etapas sucesivas, hasta obtener resultados más satisfactorios.

Otro tipo de consideraciones se refiere específicamente a los recursos reales, disponibles en la institución en donde se desarrollará el proyecto. En el Capítulo 5 se reconoció la gran cantidad de recursos que exige la implementación de un simulador y, restringiéndonos al ámbito puramente académico, en el Capítulo 1 fueron presentadas varias aplicaciones de la simulación en este nivel.

Por lo anteriormente expuesto, se considera que la mejor opción es la adaptación de una estructura modular simultánea sobre un simulador modular secuencial de probada funcionalidad.

Para tal efecto se encuentra disponible un simulador modular secuencial de propósito general, el SGP/ZAR (1983), el cual es totalmente funcional y con una estructura poco sofisticada y susceptible de aceptar alguna de las estructuras simultáneas conocidas en la actualidad.

## 6.2. EL SIMULADOR MODULAR SECUENCIAL SGP/ZAR.

### 6.2.1. Descripción General.

El simulador SGP/ZAR está realizado en el lenguaje de alto nivel FORTRAN IV y en su actual estado de desarrollo cuenta con 75 subrutinas y más de 12,000 líneas de instrucciones, con la posibilidad de ser expandido, agregando los módulos unitarios y de servicio que el usuario juzgue convenientes, además de contar con otras posibilidades que serán abordadas más adelante.

En general, el sistema posee dos grandes componentes: los módulos ejecutivos y los módulos de soporte, que a su vez incluye los módulos unitarios y los módulos de servicio. En la Figura 6.1 se muestra la estructura global del simulador SGP/ZAR y la relación de los módulos con el programa principal.

El simulador cuenta con 20 módulos ejecutivos, 29 módulos unitarios y 5 módulos de paquete termodinámico. En forma adicional existe un banco de datos que contiene los parámetros y constantes empleados en las correlaciones para el cálculo de propiedades termofísicas, con información para 275 componentes (especies químicas) considerados como estándar y datos de fracciones de petróleo, existiendo la posibilidad de aceptar hasta 100 componentes especificados por el usuario y que no están incluidos en el banco de datos.

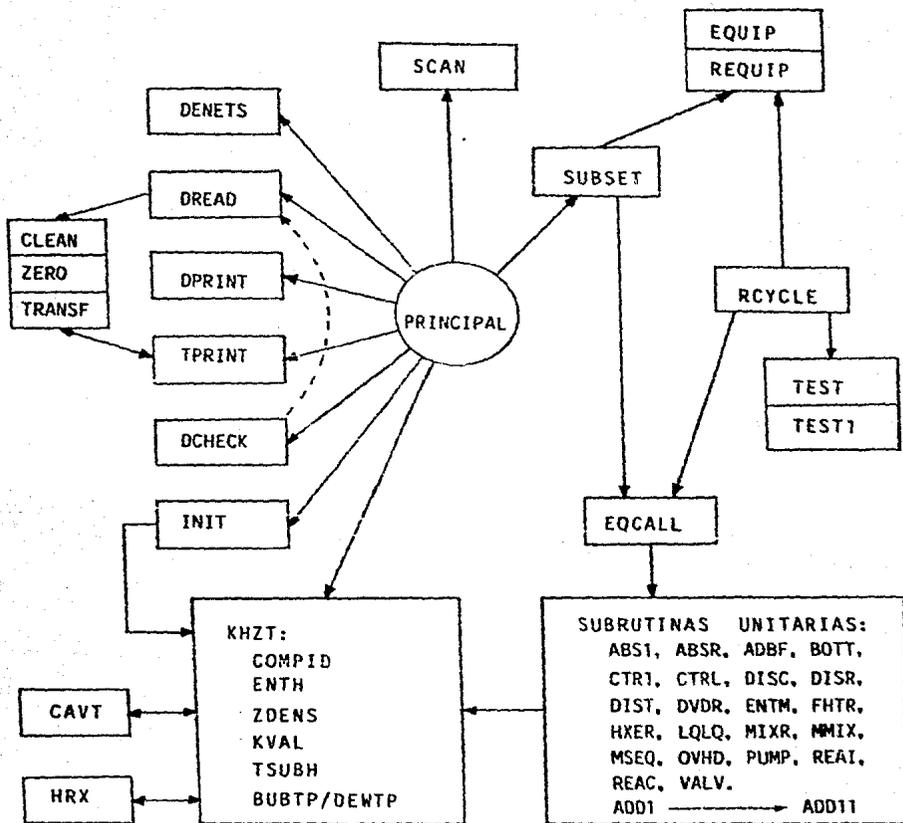


FIGURA 6.1

DISTRIBUCION DE LOS PROGRAMAS DEL  
SIMULADOR SGP/ZAR.

### A) Módulos Ejecutivos.

Forman la base del sistema de simulación, con importantes asignaciones, entre las que destacan: la captura e incorporación de los datos de entrada al proceso; realizar el análisis del proceso para obtener la identificación de recirculaciones, corrientes de corte y obtención de una secuencia de cálculo; incorpora al sistema los parámetros y correlaciones para la evaluación de propiedades termofísicas e inicializa las variables del proceso; verifica la consistencia de la información y controla el llamado de los módulos unitarios; organiza el cálculo de recirculaciones y promueve su convergencia; dirige las operaciones de entrada, salida y transferencia de información, etc.

En la Figura 6.2 se presenta la estructura del programa principal del sistema SGP/ZAR y la secuencia de llamado de los principales módulos ejecutivos. Para una descripción más detallada al respecto, consultar el Manual del Usuario del sistema.

### B) Módulos de Soporte.

Los módulos ejecutivos construyen y dirigen el esquema general de cálculo, sin embargo, no realizan ninguna tarea de simulación. La simulación, propiamente dicha, se lleva a cabo mediante la intervención directa de los módulos unitarios y los

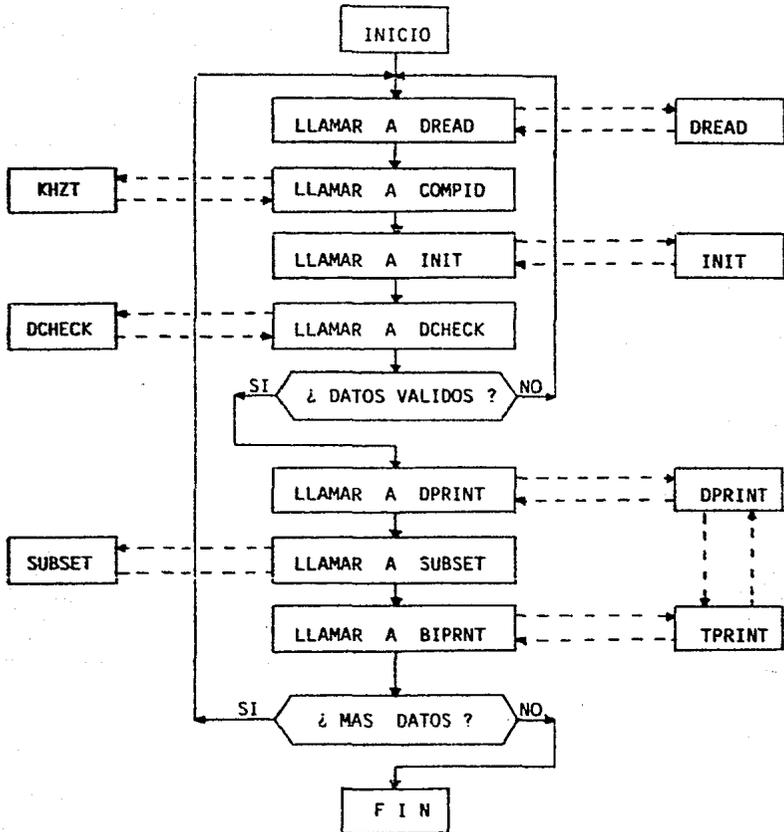


FIGURA 6.2

ESTRUCTURA DEL PROGRAMA PRINCIPAL  
DEL SIMULADOR SGP/ZAR.

módulos de servicio, relacionados armónicamente y de acuerdo a los requerimientos específicos de cada problema.

a) Módulos Unitarios.

Cada módulo unitario simula el comportamiento, ajustado a la realidad en mayor o menor grado, de uno o varios equipos y operaciones unitarias, de acuerdo a procedimientos rigurosos de cálculo. El simulador SGP/ZAR contiene 22 módulos unitarios básicos, referentes a una operación unitaria o a un conjunto de operaciones unitarias afines, agrupadas en una misma rutina de cálculo.

Existe la posibilidad de que el usuario defina sus propias subrutinas de cálculo que no hayan sido incluídas en el conjunto básico, para lo cual dispone de 11 secciones, designadas desde ADD1 hasta ADD11, en cuya definición se deberá respetar el protocolo del sistema para obtener compatibilidad con otros módulos.

b) Módulos de Servicio.

Son módulos que pueden ser tanto de desarrollo matemático, como de manejo de información. Se incluyen también todas rutinas del paquete termodinámico (cálculo de entalpías, constantes de distribución, temperaturas de rocío y burbuja, etc.).

A través del paquete termodinámico, KHZT, se maneja el banco de datos termodinámicos, llamado COMPID, conteniendo parámetros y correlaciones para la evaluación de propiedades termofísicas de los 275 componentes estándar. También se consideran las principales fracciones de petróleo, como componentes estándar, adicionales. Existe la posibilidad de manejar hasta 100 componentes (o mezclas de ellos), especificados por el usuario, permitiendo la introducción de sus parámetros y correlaciones termodinámicas específicas.

El sistema SGP/ZAR requiere cuatro grandes conjuntos de información, con el objeto de iniciar una simulación:

- 1.- Definición de las corrientes de alimentación al proceso.
- 2.- Topología del proceso y red de interconexión entre equipos, incluyendo la matriz del proceso.
- 3.- Algoritmos de solución para cada módulo unitario.
- 4.- Orden de cálculo, indicando la secuencia de llamado para cada módulo unitario aislado y formando parte de un ciclo de recirculación.

El siguiente procedimiento consiste en examinar la matriz de proceso para determinar el estado de las corrientes asociadas con cada módulo unitario. Previamente el sistema ha "etiquetado" cada equipo y sus corrientes, para verificar si estas ya son conocidas totalmente o definir si un equipo ya ha sido calcu-

lado. El simulador inicia la evaluación de los equipos cuyas corrientes de entrada y parámetros de equipo están definidos.

Quando los cálculos no pueden continuar por haber encontrado corrientes de entrada no especificadas, entonces el sistema reconoce un conjunto de equipos y corrientes asociados que participan en un ciclo de recirculación, que será resuelto en forma iterativo. Una vez obtenida la solución del ciclo, dentro de cierto valor de tolerancia, el sistema continúa con el examen de la matriz de proceso, calculando las operaciones unitarias e identificando algún otro ciclo iterativo. Este procedimiento de alternar el examen de la matriz de proceso y la solución de módulos y ciclos iterativos continua hasta que todo el proceso ha sido evaluado, o bien, hasta que la ejecución ya no puede proceder en virtud de encontrar una especificación errónea.

Los detalles referentes a los procedimientos y protocolos para la definición del proceso, datos de las corrientes, tolerancias para convergencia, elección de opciones termodinámicas, modificación de módulos unitarios y de servicio, etc.; deberán ser consultados en el Manual del Usuario del sistema SGP/ZAR (1983).

#### 6.2.2. Programa Ejecutivo de Control.

En esta sección se describe el esquema computacional del programa ejecutivo de control del simulador SGP/ZAR y de sus

principales subrutinas asociadas.

Este conocimiento es muy importante porque este programa es el que se verá directamente influido al realizar la implantación del simulador modular simultáneo SMS/ZAR.

En la Figura 6.2, al presentar la estructura del Programa Principal del SGP/ZAR, se puede apreciar que el proceso de simulación se inicia con el llamado de la subrutina DREAD introduciendo los datos de alimentación al sistema; a continuación se llama a la subrutina COMPID que activa al paquete termodinámico KHZT para obtener los parámetros y correlaciones de cálculo correspondientes a los componentes indicados en los datos de entrada. Se llama a la subrutina INIT, que realiza la identificación de fase y evalúa la entalpía de las corrientes de entrada al proceso. Con esta información se llama a DCHECK para verificar la consistencia de la información de entrada al proceso, de acuerdo a la matriz de proceso y a los resultados de INIT; si se encuentra alguna discrepancia se regresa al control al programa principal de llamado y se cancela la ejecución, solicitando una redefinición del problema. Si la información es consistente, se llama a la subrutina DPRINT para imprimir la información con la que dará inicio la simulación.

Hasta aquí el comportamiento del sistema es funcional tanto para el enfoque secuencial, como para el simultáneo. Sin embargo, cualquiera que sea el esquema que se adopte, existen dos prerequisites muy importantes para un programa de simulación y

diseño: Primero, el acceso eficiente de los parámetros de equipo y las variables de las corrientes; lo que implica que los valores supuestos o aproximados de las variables puedan ser introducidos eficientemente en sus lugares relevantes, acción que en el sistema SGP/ZAR se realiza mediante bloques de información de tipo COMMON. Segundo, la facilidad de llamar y calcular los módulos unitarios de acuerdo a una secuencia predeterminada, capacidad que en el SGP/ZAR está representada por la subrutina SUBSET y sus rutinas subordinadas.

Con el llamado de la subrutina SUBSET se inicia formalmente la etapa de cálculo. Esta subrutina representa el núcleo del programa ejecutivo de control del SGP/ZAR, pues controla el llamado de los módulos unitarios, vigila su solución de acuerdo a los ciclos de recirculación identificados y de la secuencia de cálculo previamente definida. En realidad esta subrutina opera más bien como un director de orquesta, distribuyendo tareas y vigilando su ejecución, aunque sin realizar ningún cálculo numérico por sí misma. En la Figura 6.3 se presenta el diagrama de flujo simplificado para la subrutina SUBSET. Primariamente se llama a SCAN para identificar el tipo de corrientes que maneja cada equipo; a continuación verifica si los datos de entrada provienen de una corrida anterior o si sus variables aún están sin un valor asignado y de ser así se llama a la subrutina RCYCLE. Si un equipo aún no ha sido calculado y todas sus variables de entrada son conocidas, se llama a las subrutinas EQUIP, EQCALL Y REQUIP, para lograr la evaluación de ese equipo.

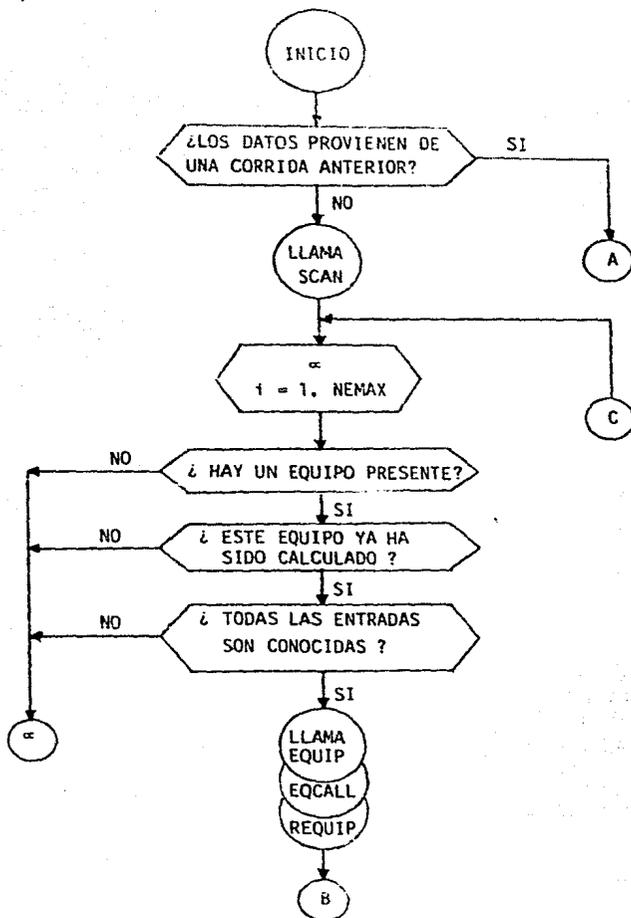


FIGURA 6.3 DIAGRAMA DE FLUJO DE LA SUBROUTINA SUBSET, EN EL SIMULADOR SGP/ZAR.

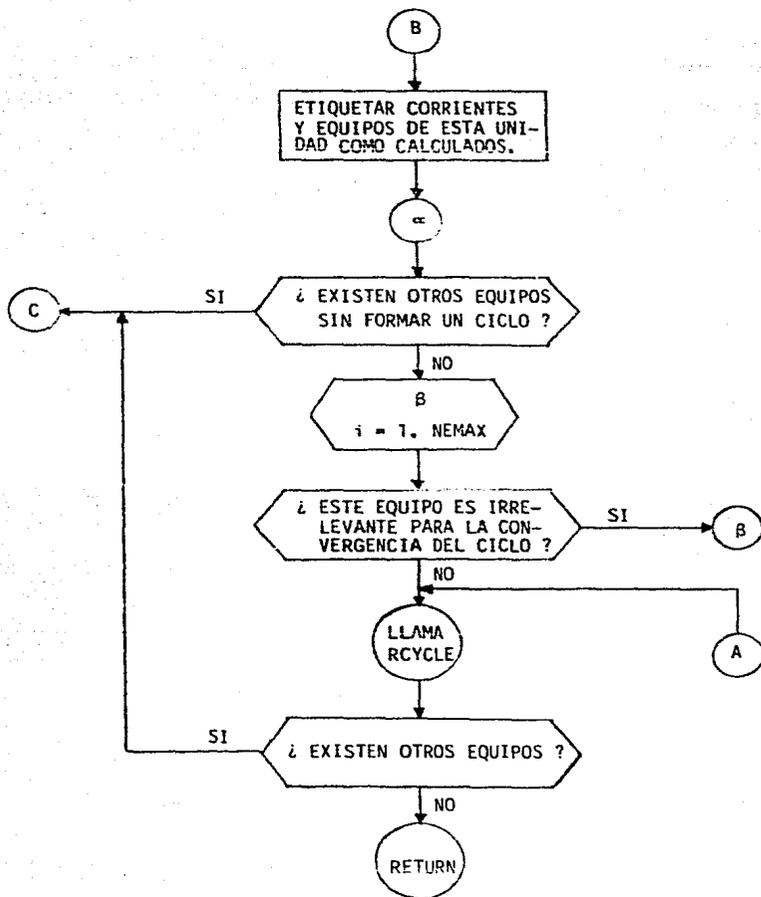


FIGURA 6.3

DIAGRAMA DE FLUJO DE LA SUBROUTINA  
SUBSET, EN EL SIMULADOR SGP/ZAR (CONT.)

La subrutina SUBSET continúa examinando todos los equipos, resolviendolos en forma individual, o en forma de ciclos, hasta alcanzar la solución total del proceso.

La Figura 6.4 muestra el diagrama de flujo simplificado de la subrutina SCAN. A cada una de las corrientes de proceso se les asigna una etiqueta, con un valor de 1 para las corrientes de entrada, 0 para las de salida y -1 para las corrientes no asociadas con el equipo que se examina. Del mismo modo, los equipos ya calculados se etiquetan con 1, los no evaluados con 0, y los equipos no involucrados en un ciclo con -1.

La subrutina EQUIP se encarga de transferir información específica, desde la matriz de datos de corrientes hacia la matriz de entrada del módulo unitario que la requiere. En la Figura 6.5 se muestra el diagrama de flujo simplificado para esta subrutina. Se comienza con la inicialización de los contadores de las corrientes de entrada y salida; a continuación lee el valor de la etiqueta de cada corriente, si es 0 se considera de salida y no sufre cambio, si es menor a cero (-1) es una corriente irrelevante y se sitúa directamente en el arreglo de corrientes de salida sin ningún cambio y si la etiqueta es mayor de cero (+1), es una corriente de alimentación y se transfiere su valor al arreglo de trabajo del equipo correspondiente.

La subrutina REQUIP realiza la tarea inversa a la de EQUIP. Su misión es recuperar los datos de entrada y salida desde las

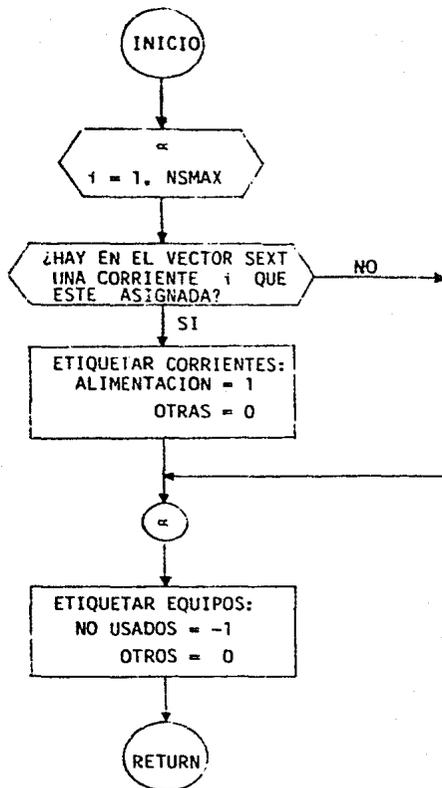


FIGURA 6.4

DIAGRAMA DE FLUJO DE LA SUBROUTINA  
SCAN. EN EL SIMULADOR SGP/2AR.

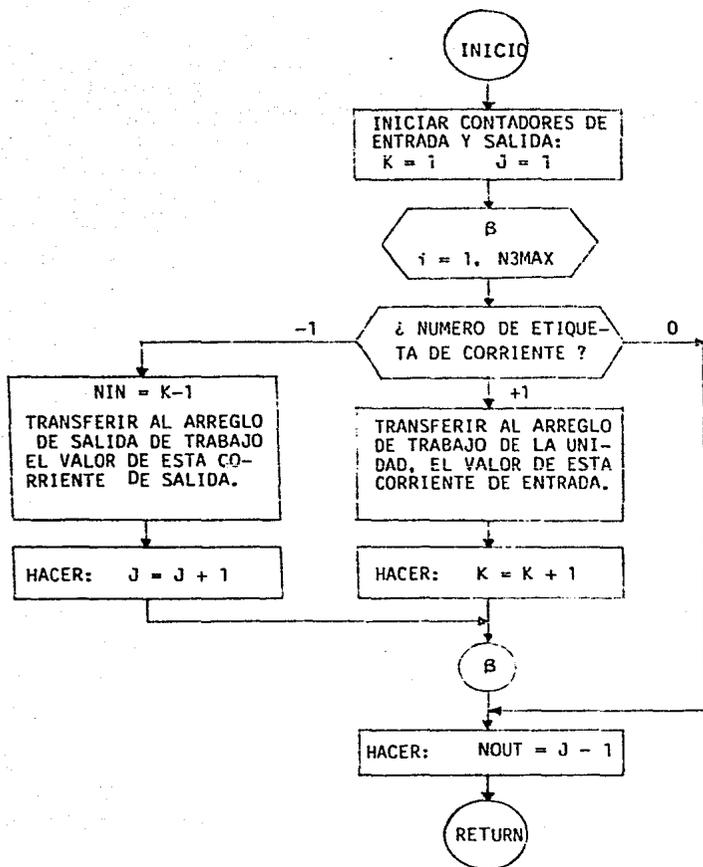


FIGURA 6.5 DIAGRAMA DE FLUJO DE LA SUBROUTINA EQUIP, EN EL SIMULADOR SGP/ZAR.

matrices (arreglos de entrada y salida) de los módulos unitarios hacia los arreglos permanentes del sistema, después de cada cálculo. En la Figura 6.6 se presenta el diagrama de flujo simplificado de esta subrutina, destacando la recuperación de información en dos niveles. Primero, obtiene la información generada en los cálculos del módulo unitario, obteniéndola de los arreglos de salida y situandola en los arreglos permanentes y de impresión. El segundo nivel recupera la información de entrada que cada módulo unitario recibió (por medio de la subrutina EQUIP), situandola nuevamente en los arreglos originales.

La subrutina EQCALL realiza el llamado de los módulos unitarios representativos de cada uno de los equipos contenidos en el proceso. En la Figura 6.7 se presenta el diagrama de flujo simplificado de esta subrutina. Se inicia con la formación de llamada para los equipos, continúa con la selección de opciones de impresión de comentarios y datos y finalmente realiza el llamado, uno a uno, de los módulos necesarios.

La subrutina RCYCLE es llamada por SUBSET cuando encuentra que los cálculos y datos del proceso están incompletos y la simulación no procede debido a la presencia de corrientes de recirculación. En la Figura 6.8 se presenta el diagrama de flujo simplificado de esta subrutina. La secuencia de llamado de los módulos unitarios, representada por las subrutinas EQUIP, EQCALL y REQUIP y la verificación de la convergencia en la subrutina TEST, se repite para el ciclo iterativo, de acuerdo a los números de identificación de unidad, contenido en los vectores KE2 y KE3, que se refieren respectivamente a la lista de números

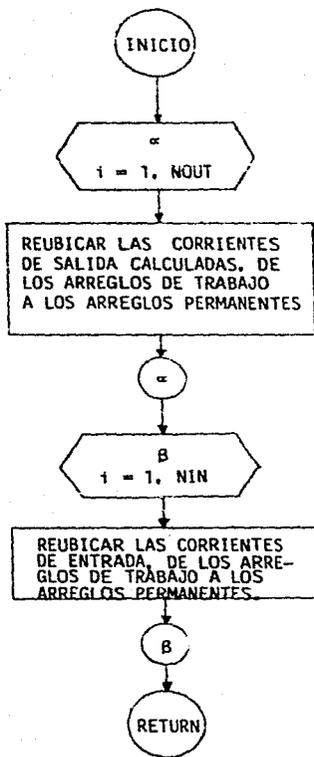


FIGURA 6.6

DIAGRAMA DE FLUJO DE LA SUBROUTINA  
REQUIP. EN EL SIMULADOR SGP/ZAR.

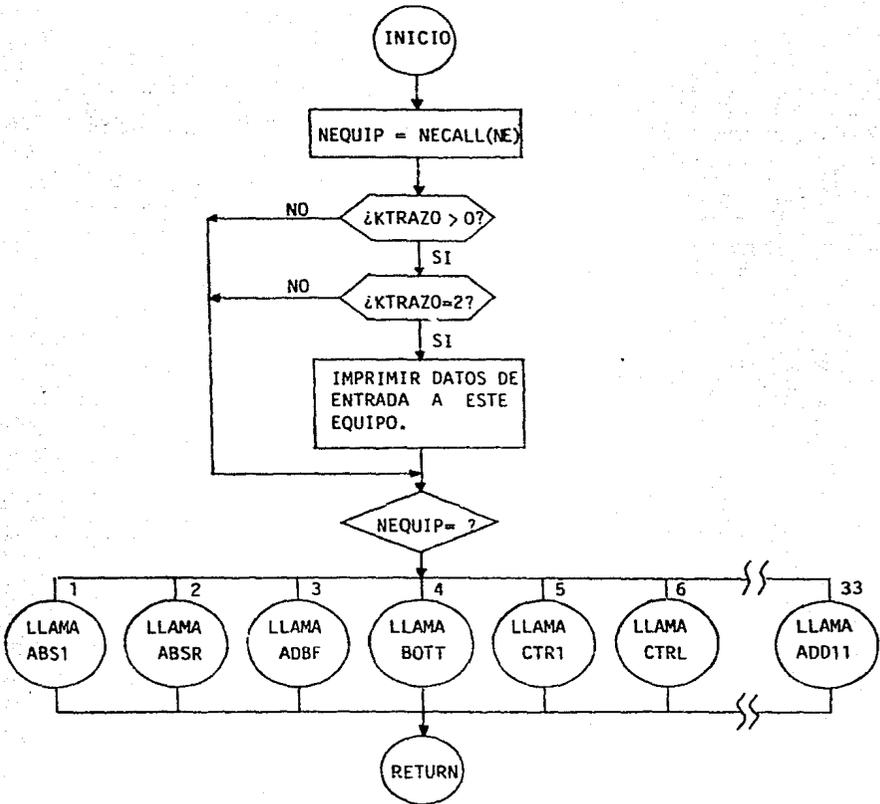


FIGURA 6.7

DIAGRAMA DE FLUJO DE LA SUBROUTINA  
EQCALL, EN EL SIMULADOR SGP/ZAR.

de equipo participantes en el ciclo de recirculación y a la lista de corrientes asociadas a esos equipos. Inicialmente se verifica la existencia de las listas KE2 y KE3, si no existen se indica un error de procedimiento. Al existir, estas listas son transferidas a los arreglos de trabajo, comenzando el proceso iterativo de solución. Al final de cada iteración se evalúa su situación con respecto al número máximo de iteraciones fijado por el usuario; si este límite se ha rebasado, se dá un aviso de no convergencia y retorna el control a SUBSET. Conforme se van resolviendo los equipos y corrientes, se imprimen los resultados parciales y al converger el ciclo se imprimen los resultados finales, junto con el estado general de la simulación.

La subrutina TEST determina la convergencia de un ciclo iterativo, bajo cierta tolerancia. Esta subrutina emplea el método de Wegstein para promover la convergencia, aplicado a algoritmos de tipo Newton-Raphson. También existe la posibilidad de realizar esta evaluación aplicando el método de Broyden, que está implementado en la subrutina TEST1.

RCYCLE hace el llamado de las subrutinas de impresión de resultados, las cuales manejan varias opciones y que están representadas por las subrutinas TPRINT y DPRINT, principalmente.

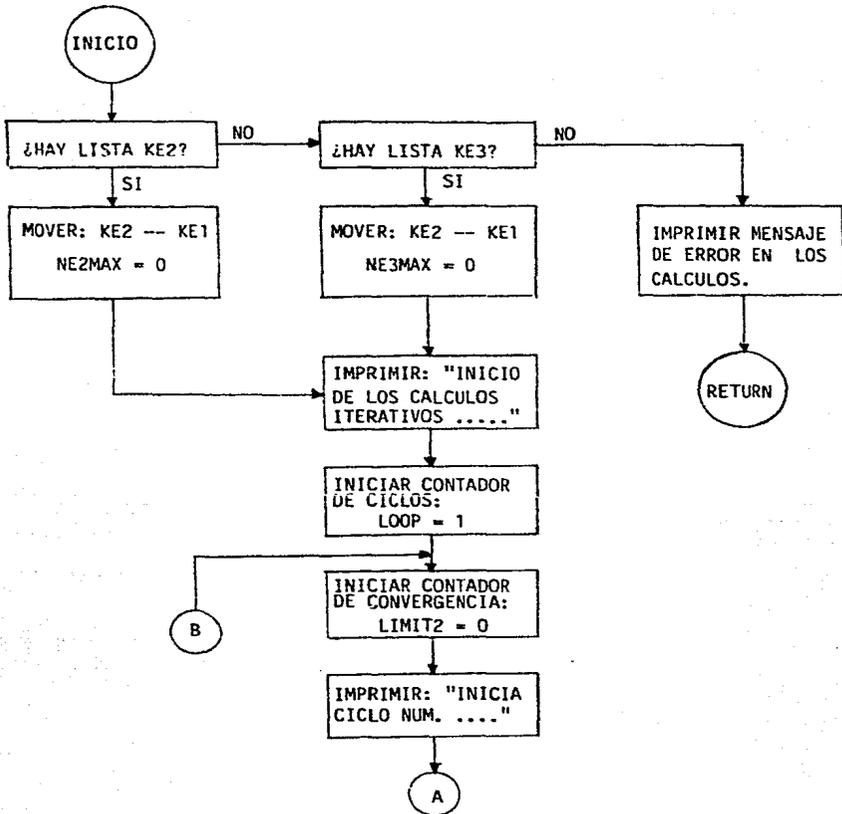


FIGURA 6.8

DIAGRAMA DE FLUJO DE LA SUBROUTINA  
RCYCLE, EN EL SIMULADOR SGP/ZAR.

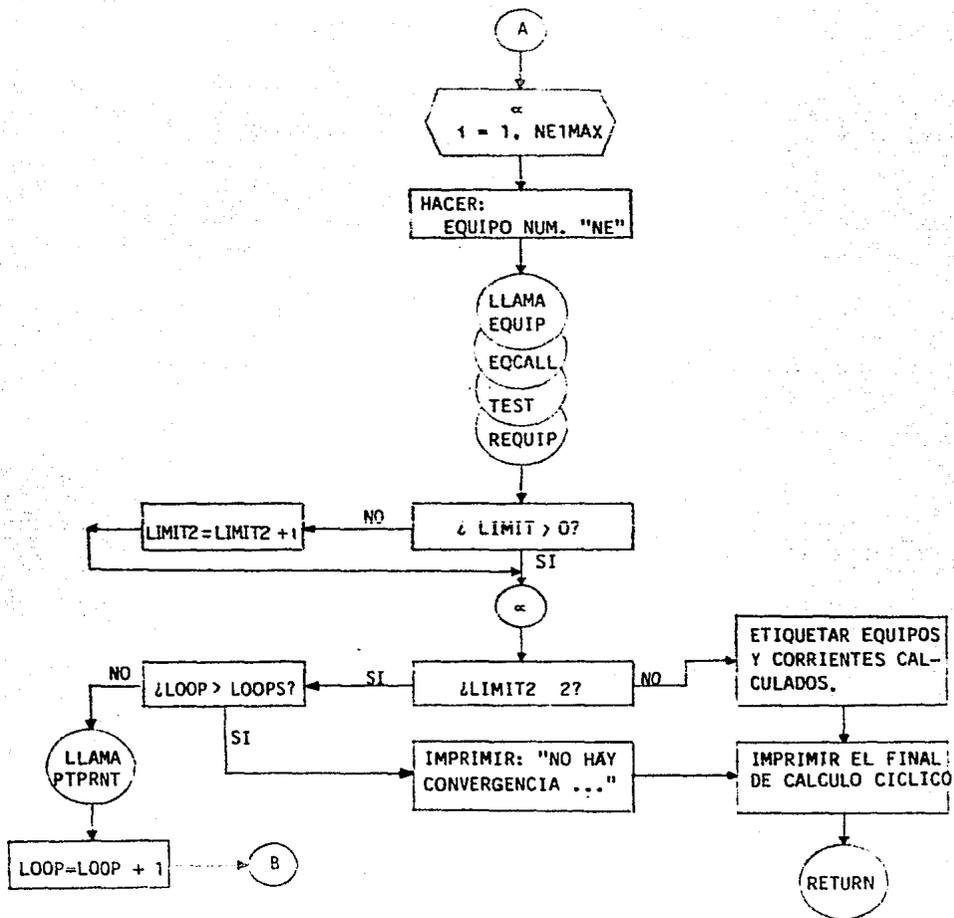


FIGURA 6.8

DIAGRAMA DE FLUJO DE LA SUBROUTINA  
RYCLE, EN EL SIMULADOR SGP/ZAR (CONT.)

### 6.3. EL SIMULADOR MODULAR SIMULTANEO SMS/ZAR.

En esta sección serán presentadas las directrices generales para el desarrollo del simulador modular simultáneo SMS/ZAR.

#### 6.3.1. Conceptos Básicos.

El simulador SMS/ZAR será implementado bajo un enfoque de ciclo interno - ciclo externo, incluyendo la incorporación de modelos aproximados de tipo reducido.

Como se indicó en los Capítulos 4 y 5, el ciclo externo está representado por los cálculos rigurosos sobre las operaciones unitarias; en cambio, el ciclo interno está destinado al cálculo aproximado de éstas. Esto significa que la ejecución repetitiva del ciclo externo es la tarea que actualmente desarrolla el simulador secuencial. El ciclo interno únicamente considera las ecuaciones y variables más relevantes del problema, utilizando métodos de solución más simples que reducen el tiempo de cómputo considerablemente, puesto que maneja sistemas de ecuaciones más pequeños y sin recurrir a llamados continuos del paquete termodinámico, generando valores aproximados que serán utilizados como entradas para los modelos rigurosos de solución. Cada ciclo es manejado independientemente, con su particular número de iteraciones y tolerancias para la convergencia.

Actualmente el programa ejecutivo del sistema SGP/ZAR rea-

liza estas tres tareas, principalmente:

- a) Inicializar las corrientes de corte y de entrada.
- b) Decide el orden en que se ejecutarán los módulos unitarios, de acuerdo a una secuencia predeterminada.
- c) Verifica la convergencia de los ciclos iterativos.

Con el objeto de lograr la implementación modular simultánea, el programa ejecutivo de control deberá ampliar sus funciones realizando las siguientes:

- a) Inicializar las corrientes de corte y entrada.
- b) Iniciar los cálculos mediante la ejecución de uno o más recorridos secuenciales a través del proceso completo.
- c) Dirigir los cálculos a nivel de ciclo externo (modelos rigurosos), con el objeto de generar los parámetros de ciclo interno (modelos reducidos).
- d) Controlar la generación de la información necesaria para hacer converger el ciclo interno (la matriz Jacobiana, principalmente).
- e) Controlar la solución del problema reducido.
- f) Verificar la convergencia del ciclo externo; si éste no converge, transferir las variables del ciclo interno hacia el ciclo externo, para repetir el procedimiento.
- g) Repetir los pasos c, d, e y f hasta que el ciclo externo converja.

A primera vista se puede pensar que es necesario realizar extensas modificaciones en el programa ejecutivo de control; sin embargo, esto no es del todo cierto, por dos razones: Primero, varias de esas tareas son equivalentes entre el simulador existente y el nuevo, puesto que comparten el mismo programa principal. Segundo, algunas de esas tareas no serán realizadas totalmente por las subrutinas del programa de control, sino que serán implementadas dentro de los módulos unitarios.

En efecto, los módulos unitarios tendrán otras tareas, independientemente de las que actualmente realizan, con el objeto de generarlos parámetros reducidos y dar forma a la matriz Jacobiana del problema reducido. Puesto que el nuevo simulador será parcialmente secuencial y parcialmente simultáneo, será necesario indicar a cada módulo el tipo de función que deberá ejecutar.

En el recorrido secuencial los módulos unitarios son ejecutados de acuerdo a una secuencia previamente definida; del mismo modo, la generación de los parámetros del modelo reducido se obtienen con el llamado de esos módulos, siguiendo la misma secuencia. Visto de este modo, la mayoría de la información necesaria para converger el ciclo interno se puede obtener mediante un suplemento al modelo riguroso, que incluya el modelo aproximado.

A continuación se examinan algunos aspectos referentes a la idea conceptual prevaleciente en el simulador SMS/ZAR.

### A) Inicialización y Parámetros Reducidos.

Se debe distinguir el término inicialización, entre dos posibles acepciones dentro de la simulación de procesos. Todo simulador necesita calcular la condición termodinámica y definir la fase de las corrientes de entrada, además de obtener valores iniciales para las corrientes de corte y las variables internas de cada módulo unitario. Sin embargo, para el tema de este trabajo, la inicialización se refiere al procedimiento de encontrar valores iniciales razonables para las variables del ciclo externo, antes de que se generen los primeros grupos de parámetros para los modelos reducidos.

En teoría se puede elegir arbitrariamente una base de cálculo, sin embargo, teniendo aún disponible la estructura secuencial es posible realizar varios recorridos secuenciales sobre todo el proceso, con el objeto de obtener bases de cálculo adecuadas.

Estos recorridos secuenciales tienen características particulares. En un recorrido secuencial normal, los ciclos encerrados dentro de otros ciclos mayores, necesitan resolverse y llevados a la convergencia antes de continuar con los demás ciclos iterativos. En el caso de los recorridos de inicialización, no se requiere la solución del proceso, sino únicamente la obtención de bases de cálculo adecuadas y existentes físicamente, por lo tanto, no se considera necesaria la convergencia de los ciclos iterativos internos.

Esta opción es equivalente a utilizar al simulador modular secuencial como el iniciador del procedimiento de cálculo a partir de un punto alejado de la solución y en determinado momento realizar el cambio hacia la evaluación modular simultánea para la aproximación final a la solución, la cual, como se verá más adelante, se alcanza con un último recorrido secuencial, en el que se evalúa el valor, composición, fase y condición termodinámica de todas y cada una de las corrientes del proceso. El número de recorridos de inicialización puede variar, dependiendo de las características propias del problema, sin embargo, usualmente se realizan entre 2 y 5.

El último recorrido inicial se puede utilizar para obtener la primera generación de parámetros reducidos, una vez resueltos los modelos rigurosos. Es importante considerar la presencia de algunas corrientes cuyo flujo sea igual a cero al momento de empezar a generar los primeros parámetros reducidos, lo cual puede ser indeseable para su correcta especificación, por lo tanto, es recomendable que esos flujos adquieran un valor mínimo, diferente de cero.

#### B) Ciclo Interno.

La ejecución del ciclo interno comienza cuando ha concluido la generación de todos los parámetros y se ha dado forma al pro-

blema reducido. Esta etapa debe ser conducida con una orientación hacia las ecuaciones, con la aplicación de eficientes métodos para la solución simultánea de las ecuaciones reducidas y las restricciones de diseño, que tendrán alguna de las formas y estructuras abordadas en el Capítulo 3. Puesto que la idea de esta implementación es la solución global del ciclo interno, los vectores y matrices que caracterizan el problema deberán contener todas las ecuaciones reducidas del proceso y no únicamente las de una operación unitaria.

Se ha mencionado que existen tres tipos de ecuaciones descriptivas de un proceso, válidas tanto en un esquema riguroso, como reducido: ecuaciones propias del modelo unitario, relaciones de conexión entre módulos y ecuaciones para especificaciones de diseño. En el Capítulo 4 y en el Apéndice C, se describe la forma de generar las ecuaciones reducidas. En cuanto a las ecuaciones de conexión, en la sección 3.4 se presentaron dos formas principales de abordarlas. Una es duplicando cada corriente de proceso en los arreglos del problema reducido, expresando la conectividad en términos de una red de relación, en donde las salidas de ciertas unidades son igualadas con las entradas de otras, con lo que se obtiene cierta independencia entre cada operación unitaria y se conserva una estructura diagonal en bloques para la matriz reducida, con las importantes ventajas que esto representa según se demuestra en el Apéndice B; sin embargo, tiene la desventaja de duplicar el número de estas ecuaciones.

La segunda forma representa las variables de cada corriente una sola vez, al generar las ecuaciones de conexión implícitamente como ecuaciones propias de cada unidad referidas a las variables de corriente apropiadas, ya sea que éstas estén asociadas con una corriente que entra proveniente de otra unidad o con una corriente que se dirige a otra parte del proceso; aproximación que genera sistemas de ecuaciones más pequeños, pero menos estructurados, sin embargo, son susceptibles de un adecuado manejo computacional, según lo reporta Stadtherr y Chen (1981).

Las ecuaciones de diseño adquieren diversas formas y su representación no siempre esta bajo el control del usuario, por lo que su manejo dentro del simulador debe ser amplio, en cuanto a aceptar varios tipos de estructura compatible con el problema que se resolverá simultáneamente, tal y como se describirá más adelante.

Existe un último tópico, que puede ser de vital importancia para el buen desarrollo de una simulación. Se refiere a la elección de las variables del ciclo interno. Esta elección depende en gran medida del tipo de modelo reducido que se pretenda utilizar, puesto que cada uno requiere de un cierto número de variables internas, las cuales difieren según se trate de un modelo lineal o no lineal, por lo tanto, un criterio general es que las variables internas, no esenciales para la convergencia, no aparezcan en el ciclo interno, dicho de otro modo, que las variables de convergencia de ese ciclo sean preferentemente variables

de corrientes de proceso. En la mayoría de los casos basta con incluir el flujo por componente y dos variables que describan su estado termofísico y el modelo reducido obtendrá toda la información que necesita, a partir de estas variables.

Se consideran dos opciones obvias para la caracterización termodinámica de una corriente: presión - temperatura y presión - entalpía. El par presión-temperatura convierte las ecuaciones del balance de entalpía en no lineales y deben agregarse al problema reducido, sin embargo, todos los cálculos para identificación de fase ("flasheo") serán relaciones lineales de presión y temperatura. El par presión-entalpía resulta en un balance de entalpía lineal, pero requiere relaciones de mayor complejidad para la identificación de fase y otros cálculos.

Aún así, ambas opciones son susceptibles de una implementación satisfactoria, por lo tanto, la elección dependerá de otro tipo de restricciones y requerimientos, propios del modelo reducido adoptado, según se indica en el Apéndice C.

### C) Ciclo Externo.

Una vez que el ciclo interno ha convergido se tendrán nuevos valores para las variables de proceso, con los que se verifica la convergencia del ciclo externo. Precisamente, la conver-

gencia del ciclo externo se evalúa en base a la diferencia entre las variables calculadas en el ciclo interno durante dos iteraciones sucesivas del ciclo externo, definiéndose en términos de la diferencia máxima entre las variables escaladas o por la norma de las diferencias de todas las variables escaladas. Se aplica un método de solución (sustitución directa, Wegstein, Broyden, etc.) para obtener nuevos estimados para las variables del ciclo externo, los cuales se sitúan en los arreglos permanentes, que son equivalentes a los que posee el simulador secuencial original, realizándose un nuevo llamado a los módulos de solución rigurosa para solucionarlos y obtener una nueva generación de parámetros reducidos y comenzar una nueva iteración del ciclo interno. La transición del ciclo interno al ciclo externo se lleva a cabo colocando las variables del ciclo interno en los vectores y matrices apropiados para que sean accesibles a la solución rigurosa. Puesto que los vectores de corriente usados en los cálculos rigurosos contienen más variables que los vectores del ciclo interno, las corrientes de proceso necesitan reinicializarse, mediante su identificación de fase, con el objeto de formar un conjunto homogéneo de información.

#### D) Especificaciones de Diseño.

Al solucionar el problema de simulación se obtienen valores para todas las variables del proceso, sin embargo, existen muchas

aplicaciones en que se deben obtener especificaciones alternativas que pueden ser una función de algunas variables de proceso, o más bien dicho, de las variables de simulación. Por ejemplo, se puede desear la especificación de un flujo a la salida de un equipo en particular, o la pureza de un producto, o alguna propiedad de cierta corriente que no sea alimentación al proceso.

De cualquier manera, para un problema de simulación el número total de grados de libertad del proceso siempre es igual a cero, es decir, por cada especificación de diseño se debe "liberar" una variable de estado, ya sea del proceso global o de cierto equipo en particular.

Estas variables manipuladas pueden ser elegidas por el usuario y la solución del problema de simulación con especificaciones de diseño generará valores para las variables manipuladas junto con las demás variables del proceso. Si se emplean métodos de solución orientados hacia las ecuaciones, entonces las restricciones de diseño únicamente vienen a ser un grupo adicional de ecuaciones que deberá agregarse al problema reducido para una solución simultánea. Se debe tomar en cuenta que un simulador modular simultáneo, como el descrito en este capítulo, en realidad posee una etapa modular (ciclo externo) y una etapa orientada hacia las ecuaciones (ciclo interno), por lo tanto, la forma más eficiente de abordar el problema de diseño es agregar sus especificaciones directamente al ciclo interno.

Puesto que el listado del programa fuente del simulador SGP/ZAR (incluyendo su programa principal, programa ejecutivo de control, subrutinas de servicio y módulos unitarios, subrutinas y bases de datos del paquete termodinámico, etc.), no es de dominio público, las modificaciones serán descritas en forma conceptual, con ayuda de los diagramas de flujo de las principales subrutinas del programa ejecutivo de control, que fueron introducidos en la sección 6.2.2, sin profundizar en detalles de programación.

El sistema SGP/ZAR comienza la simulación de un proceso considerando primeramente los módulos que no aparecen en ningún ciclo de recirculación y que no reciben como entrada los resultados de ese ciclo, es decir, considera los módulos unitarios que tienen sus corrientes de entrada y parámetros de equipo totalmente especificados. Por ejemplo, en la Figura 6.9 se muestra una sección de una planta de proceso en la que SGP/ZAR calcula las unidades 1, 2 y 3 antes de comenzar con los cálculos correspondientes al ciclo de recirculación de los demás equipos. Este parece ser el procedimiento más adecuado, y en realidad lo es para problemas de simulación simple; sin embargo, este comportamiento no es deseable si, por ejemplo, la conversión en la unidad 3 (REACTOR), fuera una variable de diseño y el problema por resolver fuera de simulación controlada o diseño.

Si éste fuera el caso, de acuerdo al enfoque secuencial, sería necesario calcular la unidad 3 cada vez que un nuevo estimado

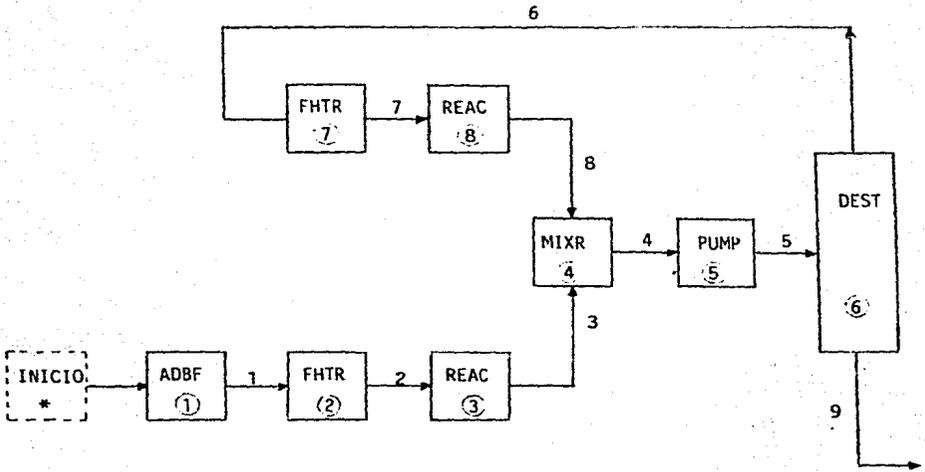


FIGURA 6.9

PROCESO PARA EJEMPLIFICAR LA ENTRADA  
DE PARAMETROS DE DISEÑO.

para la convergencia requerida, fuera generado y si este tipo de situación se repitiera varias veces durante la realización de una simulación, el tiempo de cálculo se incrementaría sustancialmente.

Esto hace que el criterio que emplea el SGP/ZAR para decidir que módulo debe ser ejecutado inicialmente, se debe ampliar para ser compatible con la implementación simultánea del SMS/ZAR, indicando que dicho modelo, además de no pertenecer a ningún ciclo iterativo y de tener sus variables de entrada completas, que no contenga ningún parámetro de diseño no especificado.

Con referencia a la Figura 6.3, que introdujo a la subrutina SUBSET como aquella que controla el llamado y solución de los módulos unitarios, se debe observar que la etapa en que se pregunta por la existencia de algún equipo y si todas sus corrientes de entrada son conocidas, se debe ampliar esa pregunta considerando si todos los parámetros de equipo y diseño también son conocidos.

En caso de que estén incompletos los datos, se debe derivar la solución de este equipo mediante un procedimiento iterativo, incluyendo el llamado a la subrutina RCYCLE.

Es importante hacer notar que la simple ampliación para tomar en cuenta las especificaciones de diseño como criterios de solución iterativa para ese equipo, no basta. Si únicamente se hiciera esa operación, el sistema SGP/ZAR al encontrar un parámetro de equipo no especificado podría truncar la simulación, primero porque no se le notificó previamente de esta posibilidad y segundo, porque no se definieron módulos de control para poder

abordar este problema, de acuerdo al esquema secuencial imperante, y se debe recordar que precisamente el enfoque modular simultáneo pretende eliminar el uso de módulos de control y un esquema iterativo redundante para la solución de los problemas de diseño.

Para aliviar en parte este problema se deben obtener variables de diseño justo al momento de definir el problema, puesto que una fuente muy importante de variables de diseño son precisamente las corrientes de alimentación al proceso. Por ejemplo, la cantidad de cierto producto puede ser conocida, siendo necesario obtener la cantidad y condiciones de las corrientes de alimentación al proceso que las generan. Desafortunadamente, la estructura original del SGP/ZAR, al igual que los simuladores de su tipo, define la alimentación al proceso mediante la especificación de las variables relevantes de todas y cada una de las corrientes de entrada, por lo tanto, es necesario diseñar un sistema que introduzca al procedimiento de cálculo las variables de las corrientes de alimentación en forma de parámetros de equipo, cuyos valores puedan ser variados de acuerdo a los requerimientos del proceso, mediante la iteración de estas variables hasta satisfacer las especificaciones de diseño impuestas.

Una forma de abordar la solución a este problema es diseñar un módulo simple, destinado a recibir toda la información de las corrientes de alimentación al proceso y que opere sobre las ecuaciones descriptivas según el tipo de problema que se abordará, ya sea de simulación simple o de diseño. Para el caso de la

simulación simple, el sistema de ecuaciones no sufrirá ningún cambio; pero en el caso del diseño, la variable o las variables elegidas para controlar el diseño serán consideradas como parámetros de equipos por definir, adquiriendo alguna de las estructuras que fueron introducidas en el Capítulo 3. En la Figura 6.9 este módulo es señalado mediante un trazo segmentado al inicio del proceso, y está designado como INICIO.

La primera llamada que hace la subrutina SUBSET es sobre SCAN, la cual etiqueta las corrientes y equipos de proceso; sin embargo, el módulo INICIO no modifica la forma de identificación, puesto que una variable de equipo continúa siendo una variable de entrada a cierto módulo unitario, sólo que ahora no tendrá un valor predefinido, que deberá ser encontrado mediante la resolución de una unidad o como parte de un ciclo iterativo, dependiendo de la topología del proceso. Esto implica que estas variables también forman parte del sistema de ecuaciones que se resolverán simultáneamente.

Con el objeto de obtener bases de cálculo razonables es necesario realizar algunos recorridos secuenciales, cuyas características fueron descritas en la sección anterior. La subrutina SUBSET es la encargada de dirigir estos recorridos secuenciales, pero la subrutina INICIO deberá impedir el acceso de las especificaciones de diseño a estos recorridos. SUBSET realizará el llamado de los módulos unitarios rigurosos, pero instruyendo a las subrutinas EQUIP, EQCALL, REQUIP, RCYCLE y TEST ó

TEST1 para no buscar la convergencia de los ciclos iterativos, sino unicamente a obtener suficiente información para realizar la inicialización del proceso.

La subrutina EQCALL, que vigila el orden de llamado de los módulos, no sufrirá ningún cambio, puesto que la secuencia de cálculo se respeta y se deberá proporcionar la misma información para el cálculo riguroso, mediante la subrutina EQUIP. Se debe vigilar que la subrutina REQUIP transfiera la información resultante de los recorridos secuenciales justo en los arreglos y posiciones de donde serán llamadas por cada módulo unitario, es decir, se sustituye la información proporcionada por las corrientes de entrada al proceso, por la información generada en los recorridos secuenciales. Esta subrutina también deberá conjuntar la información generada en los módulos unitarios y que se refiere al problema reducido, colocándola en arreglos y posiciones específicas, diferentes a las del problema riguroso.

SUBSET utiliza su propio control de iteraciones para hacer que el número de recorridos secuenciales de inicio coincida con el número preestablecido por el usuario, pero tomando en cuenta que se trata de una cifra independiente del número máximo de iteraciones de los ciclos interno y externo, que también son elegidos por el usuario. Esto significa que el contador de recorridos iniciales es independiente del contador propio de la subrutina RCYCLE, el cual estará desactivado en virtud de que no se busca la convergencia del proceso.

Concluidos los recorridos secuenciales de inicialización, la subrutina SUBSET asumirá una tarea que no existe en el simulador secuencial. Consiste en cambiar la instrucción que dió a sus subrutinas asociadas: ahora la nueva instrucción mantiene el mismo estado para las subrutinas EQUIP, REQUIP, RCYCLE y TEST ó TEST1, o sea, el de realizar un recorrido secuencial sin buscar convergencia, pero ahora a la subrutina EQCALL se le ordena que cada módulo unitario involucrado en el proceso opere en forma rigurosa y que también obtenga los parámetros del modelo reducido para esa unidad. Esto significa que cada módulo unitario adicionalmente al modelo riguroso deberá incluir su respectivo modelo reducido, posibilidad que será abordada más adelante en esta sección.

Una vez generados los parámetros reducidos, se dá forma al sistema de ecuaciones que representan al problema reducido del proceso, sistema que es de menores dimensiones que el riguroso. La información del problema reducido es recuperada por REQUIP, la cual es situada en arreglos específicos, puesto que a pesar de que constituyen una sólo matriz en bloques, no se debe olvidar que cada bloque en realidad es un subsistema representativo de una operación unitaria.

Es precisamente en esta etapa en donde son agregadas las especificaciones de diseño, ya sea por equipo o del proceso global, con el objeto de dar forma al problema reducido total, conteniendo ecuaciones reducidas de cada unidad, ecuaciones de conexión y especificaciones de diseño, que se resolverán simultáneamente.

Hasta este nivel se tienen dos sistemas de ecuaciones descriptivas del proceso: el sistema de ecuaciones rigurosas, tal y como fué planteado para los recorridos secuenciales, y un sistema de ecuaciones reducidas, de menor dimensión que el anterior y que es equivalente al problema riguroso en el punto en que fuéron generados los parámetros reducidos.

Cuando el problema reducido está totalmente formulado, se inician los cálculos del ciclo interno, realizados bajo un esquema simultáneo y para lo cual SUBSET llama a TEST (método de Wegstein), TEST1 (método Broyden), o alguna otra implementación de los métodos señalados en el Apéndice A, con el objeto de realizar la convergencia de este ciclo. Es de suma importancia elegir cuidadosamente las variables que guían la convergencia del ciclo interno, tal y como fué señalado en la sección anterior.

Se debe tomar en cuenta que la posibilidad de incluir un determinado número y tipo de variables de convergencia es una capacidad que se considera al diseñar el modelo reducido, sin embargo, la decisión de que variable o grupos de variables se usarán en la convergencia, es tomada por el usuario al momento de definir el método y el esquema general de solución del problema reducido. Desafortunadamente dentro del simulador no existen reglas predefinidas para elegir con absoluta certeza las variables de convergencia y será el usuario el que las elija de acuerdo a su experiencia y en base a los objetivos de la simulación. De cualquier manera, el problema reducido será una matriz en bloques

(ver Apéndice B), estructurado en menor o mayor grado y que será resuelto con un enfoque orientado hacia las ecuaciones.

A diferencia de la solución secuencial, en donde la subrutina RCYCLE vigila que el número de iteraciones no exceda un valor predefinido, en el caso de la solución del problema reducido será la propia subrutina de solución la que vigile que la convergencia sea obtenida dentro del número máximo de iteraciones fijado para el ciclo externo, por lo tanto, si se utiliza el mismo módulo de solución para el sistema secuencial y para el simultáneo, en este último caso se debe instruir a la subrutina de solución para que realice este control. De lo anterior se destaca que el límite máximo de iteraciones para el ciclo externo es independiente del que se tiene para el ciclo interno y que ambos son indicados por el usuario, considerando que una iteración del ciclo externo se compone de varias iteraciones del ciclo interno.

Si la convergencia del ciclo interno no se obtiene dentro del límite de iteraciones, la subrutina SUBSET despliega un mensaje indicando la falta de convergencia y el estado actual de los cálculos, solicitando la intervención del usuario para que éste sea el que elija la acción a seguir, ya sea que se suspenda la corrida y se redefina el problema, que se realicen más iteraciones, etc.

Si se obtiene la convergencia del ciclo interno, se procede a evaluar la convergencia del ciclo externo, aplicando alguno de

los criterios señalados en la sección anterior. Al no lograr la convergencia del ciclo externo, se aplica algún método de obtención de nuevos estimados para las variables de proceso, del tipo utilizado por el simulador secuencial (sustitución directa, Wegstein, etc.). Estos nuevos valores son desescalados y colocados en los arreglos y posiciones correspondientes a los vectores de parámetros de equipo y corrientes de proceso, tal y como lo requiere el simulador modular secuencial. Puesto que no todas las variables necesarias en el ciclo externo (cálculo riguroso) fueron modificadas en el ciclo interno (cálculo aproximado) es imprescindible identificar y reinicializar todas las corrientes, es decir, definir su fase y características termodinámicas, con el objeto de conjuntar toda la información necesaria para los cálculos rigurosos.

Tal y como fué señalado en el Capítulo 4, el usuario tiene la opción de elegir si todas las corrientes del proceso serán reinicializadas o unicamente las corrientes de corte y las de entrada a cada equipo, siendo el conjunto elegido el mismo que se utilizará para verificar la convergencia del ciclo externo.

A continuación SUBSET indica la ejecución secuencial de los módulos unitarios implicados en el proceso, de acuerdo al mismo orden de cálculo establecido al inicio de la simulación, instruyendo a cada módulo para que realice los cálculos rigurosos y a continuación genere los parámetros del modelo reducido, para

que a continuación se estructure el problema reducido, se agreguen las especificaciones de diseño y se realice el escalamiento de variables para que se llame a la subrutina de solución del ciclo interno, repitiéndose el procedimiento hasta aquí descrito.

Cuando se logra la convergencia del ciclo interno y del ciclo externo se procede a descalar y colocar cada variable en el arreglo y posición correspondientes en el vector de equipos y corrientes, en la forma que lo exige el simulador secuencial.

Si bien es cierto que ya se tienen definidos los valores de las variables más relevantes del proceso al llegar a esta etapa, sin embargo, esto no significa que la totalidad de las variables del proceso lo esten, puesto que no todas las variables están presentes en el ciclo interno. Para lograr la definición total del proceso, primeramente se realiza la identificación de fase de todas las corrientes del proceso (y no únicamente de las corrientes de corte y de entrada a los equipos), y con estos resultados se realiza un último recorrido secuencial que proporcionará el valor para todas y cada una de las variables que intervienen en el proceso, incluyendo variables de proceso, variables internas de los equipos, variables de conexión y especificaciones de diseño, con lo cual concluye la simulación.

Los módulos unitarios originales se modifican lo menos posible, con el objeto de que cumplan la función asignada a la estructura modular simultánea. Estos módulos conservan todas

las características que poseen en la implementación modular secuencial, sin embargo, se les adicionan dos tareas:

- a) Evaluar los parámetros para el modelo reducido.
- b) Dar forma a la matriz Jacobiana del problema reducido.

Cada módulo unitario proporcionará dos tipos de datos diferentes, manejados en forma independiente, de acuerdo a las instrucciones dadas por SUBSET a través de EQCALL: información rigurosa de acuerdo a la estructura modular secuencial, e información aproximada de acuerdo al esquema modular simultáneo.

En el Apéndice C se muestra el desarrollo de varios modelos reducidos, a partir de modificaciones del módulo riguroso correspondiente, en la forma en que pueden ser implementados sobre los módulos unitarios existentes.

Los parámetros del modelo reducido son almacenados en variables de retención locales (independientes de los arreglos propios de la estructura secuencial) y que son modificados una vez por cada iteración del ciclo externo.

CAPITULO 7

CONCLUSIONES

En el desarrollo de este trabajo se ha mostrado un amplio panorama del entorno en que se crean los sistemas de simulación de procesos, abordando temas que constituyen una base para la implementación de un simulador modular simultáneo.

Se destacó la importancia de recopilar una gran cantidad de información fundamental que sirva de respaldo para la creación de un sistema de simulación; información que debe ser integrada a un campo que, si bien es todavía inmaduro, presenta un excepcional crecimiento y expansión.

Sin embargo, todo campo de la ciencia y la tecnología que experimenta crecimientos acelerados debe tener muy en cuenta su ambiente futuro, inmediato y a mediano plazo. En el caso de la simulación de procesos, se ha destacado que desde la idea conceptual hasta la entrega de un simulador pueden transcurrir entre cuatro y siete años, lo que significa que los sistemas que se diseñan actualmente deben tener en cuenta su compatibilidad funcional con las necesidades y condiciones imperantes en la década de los noventas.

Las tendencias generales en cuanto al desarrollo de la Inge-

nería de Sistemas de Proceso, puede ser resumida así:

- a) Mayor integración de todos los procedimientos del diseño (Síntesis, Análisis y Optimización).
- b) Una amplia disposición para sistematizar y computarizar los aspectos más cualitativos del Diseño de Procesos.
- c) Una formulación más extensa de los objetivos del diseño.
- d) Apertura total para adaptar las mejores técnicas y procedimientos provenientes de otras disciplinas.

Algunas de estas tendencias son el resultado del explosivo desarrollo de la computación. sin embargo, debe quedar claro que éste cae en la categoría de disponibilidad de herramientas, más que en un avance conceptual dentro de la Ingeniería de Procesos.

Es verdad que la tecnología de computación, junto con sus costos decrecientes y disponibilidad de recursos, crea condiciones favorables para el uso intensivo de la computadora en el proceso de diseño, pero esta tendencia esta fuertemente reforzada por la creciente competitividad en la industria de procesos, destinada a obtener diseños más eficientes, en un marco de reducción de riesgos económicos y financieros. Por lo tanto, la posibilidad de integrar la información cualitativa acerca del comportamiento de las unidades y especificaciones del proceso, con la información cuantitativa representada por modelos matemáticos, conduce a técnicas orientadas a un manejo computacional eficiente.

Aún así, no debe imperar una euforia exagerada acerca de los recientes desarrollos computacionales como una solución a todos los problemas de diseño. Los modelos facilitan la representación de las ideas y los programas ayudan a orientar la solución de éstos, sin embargo, es a los ingenieros a quienes corresponde elaborar un esquema práctico de solución, siendo los sistemas de simulación únicamente sus herramientas, dicho de otra manera, el darle un bisturí a un hombre, no lo convierte en cirujano.

En base a estas consideraciones, se deben distinguir dos niveles de desarrollo futuro:

- a) Desarrollo conceptual del Diseño y Simulación de Procesos.
- b) Desarrollo de los medios de implementación computacional.

#### 7.1. DESARROLLO CONCEPTUAL DEL DISEÑO Y SIMULACION.

En términos generales, la clase de problemas que se abordarán en la próxima década serán los mismos que se pretenden resolver hoy en día, pero buscando condiciones más ventajosas y que sirvan de base para la ampliación de los objetivos actualmente conocidos, desarrollando estudios de simulación asociados con el diseño y operación de procesos con un eficiente manejo de materia y energía.

Se continuará con la tendencia a extender el uso de la simulación a otras ramas productivas ajenas a las industrias del petróleo y petroquímica, buscando su integración a industrias relacionadas con combustibles sintéticos, pulpa y papel, metales y minerales, producción de alimentos, etc.

Con mayor intensidad se perseguirá la integración del proceso de diseño al ampliar sus objetivos, es decir, los resultados de un programa servirán de base de entrada a otros, y aún más, se buscará que un mismo sistema ejecute varios tipos de análisis.

Todo esto conduce a lo que se considera la tendencia más importante, iniciada en esta década, en cuanto al desarrollo conceptual: considerar el diseño y la optimización del proceso como una tarea adicional del simulador.

En los últimos años han madurado varios polos de desarrollo conceptual del Diseño de Procesos, destacando el modelado de operaciones unitarias individuales, modelado de procesos totales, desarrollo de paquetes para propiedades termodinámicas, creación de algoritmos eficientes para el manejo de modelos, desarrollo de estructuras para el manejo de flujos de información, etc., lo que da forma a nuevas concepciones del problema de simulación, diseño y optimización, tal y como se expuso en este trabajo.

Los primeros simuladores, representativos de la década de los sesentas, fueron meramente intuitivos y muy simples, basados principalmente en el comportamiento físico de las unidades; pero

conforme se fuéron incorporando nuevos desarrollos matemáticos, especialmente en métodos numéricos y modelado matemático, se ampliaron sus capacidades, bajo el estímulo del desarrollo computacional, tanto en soporte físico como lógico.

Desafortunadamente, los métodos numéricos de propósito general no son totalmente adaptables a las tareas del diseñador, por lo tanto, se ha reducido la tendencia al uso de modelos muy sofisticados, en favor de modelos más simples, haciendo resurgir el interés en la intuición física para dar forma a procedimientos más efectivos. Esto no se debe considerar como un retroceso al uso de los antiguos algoritmos, sino más bién a una eficiente unión entre la apreciación física y las herramientas matemáticas modernas.

La tendencia por integrar el diseño como una tarea del simulador responde tanto a una evolución conceptual, como a una necesidad sistemática. Es paradójico, pero en la actualidad se simulan procesos que prácticamente ya han sido diseñados por otros medios, por lo tanto, el distanciamiento entre simulación y diseño obedece más a la falta de herramientas para un manejo adecuado del problema, que a una falla de concepto.

Durante más de dos décadas de desarrollo de la simulación de procesos han existido diferentes tendencias, suscitando controversias acerca de los enfoques más prometedores. El enfoque modular secuencial ha tenido vigencia hasta estos días, sin embargo, en virtud de la ampliación de los objetivos de simula-

ción, de nuevas arquitecturas computacionales, del avance en los métodos de solución de ecuaciones no lineales y la creciente complejidad de los procesos químicos, este enfoque se ve seriamente limitado y es así como se buscan otras alternativas.

La discusión central radica en si las unidades estan mejor representadas por subrutinas con variables específicas de entrada y salida, o si es preferible representarl as mediante conjuntos de ecuaciones que serán manejados y resueltos bajo control de un programa ejecutivo. A primera vista parece más sencillo que un ingeniero simplemente obtenga las ecuaciones descriptivas para cada unidad y que estas se integren a un algoritmo de solución. También es deseable hacer una completa separación entre la descripción del problema y los métodos de manejo y solución, de modo que un cambio para incluir métodos de solución más eficientes no requiera una redefinición de los modelos unitarios. Por lo anteriormente expuesto, parece ser que la orientación basada en las ecuaciones posee todas las ventajas, sin embargo, esta orientación requiere de algoritmos de solución numérica muy poderosos y robustos y excesivos requerimientos computacionales. Por otro lado, los sistemas basados en subrutinas (como el modular secuencial), usualmente son lentos en su convergencia, pero mucho menos susceptibles de fallar.

No se debe considerar esta posición como el enfrentamiento frontal entre dos tendencias, con virtudes y defectos. Por el contrario, a pesar de que cada una ha tenido desarrollos inde-

pendientes existe la posibilidad de conjuntarlas en forma funcional mediante el enfoque modular simultáneo.

No existe mayor urgencia en clasificar sistemas, pero sí en reconocer que los simuladores de la próxima generación deberán ser sistemas bajo múltiples objetivos, parcialmente modulares y parcialmente orientados a las ecuaciones.

Un sistema de esta clase conjuntará los módulos unitarios y de servicio, mientras que los métodos de solución numérica se aplican para el cálculo de las recirculaciones, manejando las variables de iteración y las especificaciones de diseño simultáneamente, en una forma más natural y sin requerir técnicas especiales de control .

Nuevamente se debe recurrir a la intuición del ingeniero para el desarrollo de algoritmos físicamente motivados para lograr esta conjunción de sistemas y la respuesta obvia es la creación de modelos simplificados y que operen como mútuo refuerzo con los modelos rigurosos, estructurando un programa principal que los relacione en la forma más conveniente. Es así como se presenta el concepto modular simultáneo como una opción técnicamente viable, en la forma en que ha sido expuesto en este trabajo.

El esquema elemental implica el uso de modelos simples para obtener una solución aproximada, para que a continuación se utilice un modelo riguroso para lograr un refinamiento de la aproximación inicial. Un método más evolucionado sugiere el uso

de un modelo simple como un predictor, es decir, como un generador de parámetros que serán usados en el modelo riguroso, todo esto dentro de un esquema iterativo que incluya algún método de corrección de las discrepancias entre la solución simple y la rigurosa. Otros modelos simples manejan parámetros que pueden ajustarse de acuerdo al comportamiento local del modelo riguroso.

En años recientes ha cobrado importancia la creación de un esquema adecuado que relacione la etapa simple y la etapa rigurosa en forma de ciclos de cálculo anidados, correspondiendo el ciclo interno a los modelos aproximados y el ciclo externo a los modelos rigurosos. Se busca establecer los criterios adecuados para el manejo de tolerancias en cada ciclo con el objeto de distribuir el esfuerzo de cálculo de acuerdo a las necesidades de precisión y disponibilidad de tiempo, lo que conducirá a una mejor administración de los cálculos.

Se ha mencionado la extensión del simulador hacia el diseño de procesos, sin embargo, esto no implica que el diseño auxiliado por computadora no exista en la industria química. Lo que sucede es que actualmente se pretende que el diseño ya no sea más una actividad independiente, puesto que ya existe la oportunidad de integrarlo a un sistema de simulación y utilizar estructuras y cálculos prácticamente equivalentes.

En la actualidad se reconoce que un diseño orientado al cumplimiento de ciertas especificaciones estrictamente definidas no tiene razón de existir. El diseño debe aplicarse a un

amplio rango de condiciones de operación, es decir, las especificaciones de diseño no deben ser valores fijos, sino que se deben transformar en variables que un simulador debe manejar para obtener procesos viables técnica y económicamente. Esta es una extensión natural para la consideración de plantas químicas de propósitos múltiples, con sus problemas asociados en cuanto a un eventual cambio de especificaciones o en cuanto a su operación por lotes o en forma continua.

El enfoque modular simultáneo en el sentido en que fué abordado en los Capítulos 4, 5 y 6, permite el manejo de las especificaciones de diseño en forma de ecuaciones cuyas variables pueden ser entradas a un módulo unitario o ser agregadas al conjunto de ecuaciones y variables del modelo simple. Es posible iniciar el problema de diseño como un problema de simulación simple y, una vez obtenido el problema base, agregar las especificaciones de diseño en incrementos o una a una, con el objeto de tener un mayor control del procedimiento y poder identificar y trabajar adecuadamente sobre los puntos críticos del problema.

Otra de las grandes motivaciones para el desarrollo modular simultáneo es la posibilidad de abordar problemas de optimización en virtud de la opinión general para abandonar el uso de sistemas destinados exclusivamente a la optimización, enfocando la atención sobre sistemas que obtengan la convergencia de las recirculaciones y la optimización del proceso simultáneamente.

Existen opiniones encontradas acerca del verdadero valor de la optimización, o mejor dicho, del valor del tipo de optimización que actualmente es posible lograr. Es un ideal generalizado el que la optimización comience en la etapa de síntesis y que finalice con un análisis multiobjetivos del proceso diseñado, con bases económicas, operativas, etc. Desafortunadamente, hoy en día la optimización se realiza bajo consideraciones puramente económicas, generando información errónea o poco útil con respecto a otros criterios de evaluación.

Existen varias razones por las que no se alcanza la madurez de la optimización. Primero, reconociendo que existe un renovado ímpetu por el uso de modelos simples, surge la posibilidad de que existan diferencias sustanciales entre el óptimo alcanzado por una aproximación simple y el óptimo encontrado por una solución rigurosa, puesto que todo procedimiento de optimización es extremadamente sensible a las aproximaciones, tanto en la definición del problema, como en su solución numérica. Segundo, es muy difícil definir adecuadamente una función objetivo y en formular un modelo matemático representativo para el problema de ingeniería, en virtud de que la misma definición de "óptimo" aún no está clara y la formulación del procedimiento matemático puede incluir decisiones continuas y discretas junto con definiciones discontinuas y no diferenciables y en muchos casos se requiere información que no está disponible en forma explícita, por lo que se debe recurrir a aproximaciones. Tercero, la opti-

mización requiere la inversión de muchos recursos, especialmente en cuanto a tiempo de cómputo, que puede ser hasta de 6 a 10 veces mayor que el requerido por una simulación simple. Cuarto, puesto que no existen condiciones de operación únicas durante la vida útil de una planta, está en duda el verdadero valor de un estudio de optimización riguroso, bajo condiciones anticipadamente consideradas como "óptimas": en forma adicional, el diseño evaluado en esta forma, fundamentalmente es poco útil y puesto que la planta no opera durante largos periodos de tiempo en condiciones de diseño, la optimización bajo estas condiciones no es un criterio adecuado, tomando en cuenta que los puntos óptimos en condiciones de diseño no necesariamente serán puntos óptimos dentro del rango de condiciones de operación de rutina.

El enfoque modular simultáneo tal vez no ofrezca la solución a varios de los inconvenientes anteriormente citados, en especial a los problemas de definición: sin embargo, presta valiosa ayuda en cuanto a la reducción del tiempo de cómputo y la posibilidad de emplear avanzados modelos y métodos numéricos que actualmente son prohibitivos en los simuladores modulares secuenciales.

En repetidas ocasiones durante el desarrollo de este trabajo se resaltó que en la simulación de procesos es necesario el uso de métodos eficientes para resolver grandes sistemas de ecuaciones algebraicas no lineales y matrices lineales dispersas, cuyos esquemas más representativos se muestran en los Apéndices A y B.

Si bien es cierto que se trata de técnicas de propósito general y que la Ingeniería de Procesos está abierta a la adopción de los avances más promisorios provenientes de otras disciplinas; esto es una clara muestra de que esas aportaciones sólo podrán ser funcionales en la medida que exploten las características estructurales del tipo de sistemas de ecuaciones que se generan al representar procesos químicos, especialmente cuando se trata de abordar el diseño y la optimización como una tarea más del simulador, generando matrices que no son del todo simétricas y que requieren técnicas de solución que preserven su estructura dispersa y sean susceptibles de adoptar un comportamiento computacional satisfactorio.

Sin embargo, varias de esas técnicas, de reconocida eficiencia en otros campos, aún no se integran a la simulación de procesos químicos, en parte debido a la falta de estructuras computacionales que las soporten. Aún así, el panorama en cuanto a las técnicas numéricas de solución es más claro que en otras áreas. Métodos eficientes existen, están ahí en espera de ser implementadas en forma de subrutinas adecuadas y, ante el explosivo desarrollo de la computación, esto parece estar al alcance de la mano.

Es así como se puede afirmar que los sistemas de simulación de la siguiente década tienden a disolver la diferencia funcional entre las orientaciones modulares y no modulares, dando paso a sistemas eclécticos y razonablemente conjuntados.

Esto significa que varios enfoques podrán coexistir dentro de un mismo sistema, siendo aplicables a los mismos modelos del proceso. A un nivel, el usuario podrá emplear módulos rigurosos secuenciales para trabajar en la síntesis del proceso y obtener una base para la simulación. En otro nivel, el modelo de proceso se emplea para realizar el diseño y simulación, en base a modelos rigurosos y modelos aproximados de las operaciones unitarias, aplicando técnicas de solución orientadas al manejo eficiente de las ecuaciones. En un tercer nivel, la arquitectura modular realiza la optimización del proceso, considerándola como un caso particular de simulación, altamente restringida.

## 7.2. DESARROLLO DE LOS MEDIOS DE IMPLEMENTACION.

En los últimos quince años se ha producido una revolución en cuanto a la idea que se tiene de la Ingeniería de Procesos, debido en gran medida a la expansión de la tecnología de computación.

Es bien conocida la importancia que tienen las aplicaciones computacionales en la Ingeniería Química, sin embargo, estas son modestas, si se comparan con las excepcionales perspectivas a fu-

turo. El impacto del desarrollo computacional en la simulación y diseño de procesos puede ser observado desde el punto de vista del soporte lógico (software) y del soporte físico (hardware).

#### A) Soporte Físico.

El soporte físico ha influido decididamente en el avance de la simulación de procesos. Tan sólo en cuanto a costos, se ha experimentado un decrecimiento al menos en un factor de 100 entre 1960 y 1986, como producto de la tecnología de microcircuitos. Actualmente se vive la era de la integración de circuitos a muy alta escala (VLSI, por sus siglas en inglés) junto con una producción masiva de estos componentes, lo que reditua en reducciones sustanciales.

Gran parte de los esfuerzos por crear nuevas estructuras de simulación se destinan a combatir los requerimientos de disponibilidad de grandes volúmenes de memoria. Las computadoras actuales siguen la arquitectura propuesta por Von Neumann, consistente de una memoria principal, un canal de entrada, un canal de salida y un procesador central, desarrollando un flujo de información de tipo secuencial, también conocido como escalar.

De aquí deriva la práctica común de definir la eficiencia computacional en términos de velocidad de ejecución únicamente,

haciendo acopio de una cantidad de memoria prácticamente ilimitada. Precisamente la velocidad de ejecución y la disponibilidad de memoria, son dos puntos críticos en la ejecución de grandes programas, en virtud de que al aumentar indiscriminadamente la cantidad de memoria se produce un efecto de "cuello de botella", al tener que pasar grandes volúmenes de información por un pasaje único y muy estrecho, lo que incide directamente en la velocidad de ejecución de un programa.

Esto ha conducido a redefinir la estructura para las computadoras de la siguiente década. La propuesta más prometedora es la del proceso de datos en paralelo, caracterizado por un manejo vectorial de la información, usando en lugar de un sólo procesador central, varios cientos y hasta miles de pequeños procesadores, cada uno con su propia sección de memoria e interconectados unos con otros, en un sistema que se programa y reprograma sin cambiar esas interconexiones.

Estos avances tecnológicos no se deben pasar por alto al diseñar nuevos simuladores, pues deberán concebirse con la posibilidad de ser ejecutados en computadoras de estructura avanzada.

Visto de este modo, algunas técnicas numéricas que se consideran inadecuadas para implementarse en máquinas escalares, pueden resultar muy superiores en un procesamiento vectorial, existiendo la posibilidad de abandonar algunos métodos de solución de ecuaciones de tipo directo, por técnicas de tipo indirecto

(o iterativo), puesto que se tendría un acceso más rápido a las memorias.

Por lo anteriormente expuesto, se debe tener en cuenta la adaptación de los programas actualmente en uso, y los que serán creados, a las continuas innovaciones en soporte físico y evitar que se amplíe la diferencia entre el desarrollo de las máquinas y de los programas, distanciamiento que hoy en día comienza a ser evidente.

#### B) Soporte Lógico.

Esta área de la computación es la que tiene un efecto más directo sobre el desarrollo de este trabajo. La creación de sistemas de información de interés para la Ingeniería de Procesos es críticamente dependiente de la disponibilidad de programas interactivos de apoyo. Es conocido el explosivo desarrollo de las computadoras, sin embargo, no es posible decir lo mismo acerca del soporte lógico, el cual únicamente ha crecido en un factor de 6 entre 1960 y 1986. Adicionalmente, la diferencia en costos entre los soportes físico y lógico es muy grande, al grado de que en 1986 el costo de realización de un millón de instrucciones por segundo en soporte físico, cayó por debajo del salario de un programador profesional, lo que implica que los costos de programación ahora constituyen aproximadamente el 85% del

costo total de una simulación, todo lo cual da forma a lo que se conoce como el Problema del Soporte Lógico dentro de la simulación de procesos.

Existen múltiples áreas para el desarrollo de soporte lógico que interesan a la Ingeniería de Procesos (programas, lenguajes, estructuras, etc.) , que están íntimamente relacionadas y que se pueden conjuntar en tres aspectos:

a) Creación de programas simples, adecuadamente entrelazados.

La tendencia es la de crear programas con un alto sentido de modularidad y provistos de estándares para ser usados como interface entre ellos y entre programas ejecutivos, con el objeto de dar forma a sistemas más complejos, conjuntados de acuerdo a las necesidades. Puesto que cada elemento individual es rigurosamente verificado y probado, teóricamente es posible obtener programas complejos funcionalmente eficientes, sin embargo, esto no sucede del todo, si bien es cierto que existen excelentes programas de aplicación, aún no se obtiene esta funcionalidad para sistemas completos. Usualmente la dificultad reside en la elección apropiada del conjunto de programas individuales y la correcta formulación del sistema de interconexión.

b) Metodologías y herramientas para el desarrollo de programas.

Este factor en la generación de programas en parte es el resultado de una descomposición del proceso de desarrollo y por

otro lado, de la disponibilidad de procedimientos administrativos y controladores de ese proceso, bajo consideraciones de existencia de leguajes de programación y de programas utilitarios de diseño y control. Los programas para administración de diseño contienen programas de prueba, programas de configuración, analistas de flujo de información, localizadores de fuentes de error, compiladores de bases de datos, etc. En cuanto a los lenguajes de programación, FORTRAN ha sido durante décadas el de mayor aplicación en la Ingeniería Química, sin embargo, se prevee a un futuro no muy lejano, la adopción de lenguajes igualmente poderosos, pero más estructurados y compatibles con el desarrollo del soporte lógico y físico. Se han realizado pruebas con el lenguaje FORTRAN 77, que es una versión del FORTRAN, pero más estandarizado y estructurado, sin embargo se pronostica una especial predilección por el lenguaje PASCAL, que es un lenguaje moderno, altamente estructurado, que exige disciplina en la programación, con excelente tratamiento de asignación y definición de datos y adecuado manejo de información de entrada y salida. Son importantes y dignos de tomar en cuenta, los intentos por constituir al PASCAL (especialmente el ADA, uno de sus lenguajes derivados), como el lenguaje estándar para aplicaciones científicas y tecnológicas, buscando una alta compatibilidad con la mayoría de los sistemas operativos actualmente en uso y en forma muy especial con los sistemas de la próxima década.

c) Alta abstracción en la generación de programas.

En años recientes ha existido gran inquietud en cuanto a las filosofías imperantes en la formulación de un esquema de solución de problemas, auxiliado por computadora. La mayoría de los programas actuales abordan la solución de un problema con un conjunto de instrucciones que en su gran mayoría están dirigidas a definir los detalles que son requeridos por una computadora para integrar las funciones adecuadas para una corrida exitosa, siendo un número comparativamente reducido las instrucciones que abordan realmente la solución del problema, es decir, los actuales sistemas tienen una orientación hacia el sistema computacional, más no una orientación a la solución del problema.

Se hace imperativo un cambio conceptual en la implementación de programas que aborden tópicos de simulación y diseño de procesos químicos, abordando su solución en forma más directa, obviando un excesivo número de instrucciones al sistema y poner más atención a la metodología de solución. Es conocido el hecho de que gran parte del éxito de la orientación hacia la solución de problemas descansa en el desarrollo de lenguajes de programación y códigos de representación con poderosas propiedades algebraicas, sin embargo, existen varios obstáculos para estas implementaciones, correspondientes a la arquitectura de los actuales sistemas de simulación, esencialmente en cuanto a su manejo de la información.

Los sistemas de simulación actualmente son diseñados bajo una estructura enfocada al manejo de un problema mediante la separación de los esquemas de cálculo y manejo de datos, es decir, se pretende abandonar los sistemas orientados a los procedimientos de manejo de información, por sistemas orientados a los procedimientos de solución de problemas, con lo cual la diferencia funcional entre los enfoques modular simultáneo y orientado a las ecuaciones, será muy sutil.

En síntesis, el soporte lógico del futuro próximo será altamente modular, adecuadamente estructurado, estrictamente estandarizado y con un manejo de datos independiente del tratamiento numérico. Varias de las subrutinas de apoyo y de alto consumo de recursos computacionales, sean manejadas fuera del programa principal e, inclusive, fuera del área de memoria asignada al programa. En busca de una mayor independencia de la solución numérica, se puede llegar al tratamiento independiente de las operaciones de tipo escalar-vector, vector-vector y vector-matriz, operaciones que son de excepcional importancia en la Simulación y Diseño de Procesos Químicos Auxiliados por Computadora.

A P E N D I C E      A

METODOS DE SOLUCION DE  
EQUACIONES NO LINEALES.

## A P E N D I C E      A

### METODOS DE SOLUCION DE ECUACIONES NO LINEALES.

En todo Sistema de Simulación de Procesos existe la necesidad de resolver un sistema de  $n$  ecuaciones no lineales en  $n$  variables, del tipo

$$F(x) = 0 \qquad (A.1)$$

en el cual cada ecuación incluye sólo una pequeña proporción del número total de variables del sistema y para conjuntos de ecuaciones muy grandes existen algunos subsistemas acoplados que corresponden a diferentes unidades de proceso, conduciendo a un cierto grado de dispersión del sistema de ecuaciones.

A diferencia de los métodos para resolver sistemas de ecuaciones lineales, los métodos de solución de ecuaciones no lineales en general no son muy robustos, pudiendo conducir sólo a uno de dos o más conjuntos de raíces, algunos de los cuáles pueden ser complejos o idénticos, y casi siempre requieren el uso de procedimientos iterativos. Estos métodos parten de un estimado inicial,  $x^0$ , que conduce al proceso iterativo hacia una solución,  $x^*$ , la cuál se encuentra localizada dentro de cierta región de tolerancia. Sin embargo, disponer de un buen estima-

do inicial no siempre es tarea fácil y por lo tanto se han desarrollado técnicas que promueven la convergencia aún a partir de estimados iniciales alejados de la solución.

Desde este punto de vista, los métodos para resolver sistemas de ecuaciones no lineales se pueden agrupar así:

- 1) Métodos que son localmente convergentes y que requieren muy buenos estimados iniciales.
- 2) Métodos con amplio rango de convergencia y que requieren estimados iniciales medianamente próximos a la solución.
- 3) Métodos con convergencia global y que parten de estimados iniciales elegidos arbitrariamente.

En la Tabla A.1 se presentan algunos métodos pertenecientes a estas categorías.

Antes de iniciar la descripción de algunos métodos que son considerados como representativos de cada grupo, se debe hacer notar que la elección de un método en particular depende no únicamente de la disponibilidad de buenos estimados iniciales, sino también de los requerimientos y condiciones que cada algoritmo impone (velocidad de convergencia, evaluación de derivadas analíticas, tiempo de cómputo, flexibilidad en su estructura, etc.) con el objeto de obtener la solución deseada. No se debe pasar por alto que el problema al que se aplicará un determinado método de solución es quién realmente condiciona el comportamiento del proceso iterativo, siendo muy común que la dificultad en dar solución a los sistemas de ecuaciones no lineales descansa

CONVERGENCIA LOCAL:	Newton y sus variantes Sécante y sus variantes Quasi-Newton: <ul style="list-style-type: none"><li>- Broyden</li><li>- Schubert</li><li>- Brown, etc.</li></ul>
CONVERGENCIA EXPANDIDA:	Híbrido de Powell Levenberg-Marquardt, etc.
CONVERGENCIA GLOBAL:	Algoritmos de Homotopía: <ul style="list-style-type: none"><li>- Kubicek</li><li>- Abbot</li><li>- Chow-Yorke</li><li>- Rheinboldt-Burkardt, etc.</li></ul>

TABLA A.1. CLASIFICACION DE LOS METODOS DE SOLUCION DE ECUACIONES NO LINEALES.

en una o más de estas razones:

- 1) Algunas de las funciones no están definidas para ciertos valores de las variables.
- 2) Pueden existir algunas soluciones del sistema que no sean físicamente posibles.
- 3) Las funciones son altamente no lineales.
- 4) El sistema de ecuaciones es excesivamente grande.

#### A.I. METODOS DE CONVERGENCIA LOCAL Y CONVERGENCIA EXPANDIDA

##### METODO NEWTON-RAPHSON.

En 1690 Joseph Raphson publicó un método iterativo para encontrar la raíz de un polinomio en una variable a partir de un cierto estimado inicial, basado en una propuesta de Isaac Newton en su célebre obra "Principia Matemática" (1687). Este parece ser el origen del método Newton-Raphson, el cual, con algunas modificaciones funcionales, prevalece hasta la fecha.

Si  $f$  es una función de una variable simple, entonces, de acuerdo con Ortega y Rheinboldt (1970), es posible representarla para alguna aproximación,  $x^0$ , de la raíz solución,  $x^*$ , mediante una función lineal del tipo

$$L(x^0) = \gamma (x - x^0) + f(x^0) \quad (\text{A.2})$$

adoptando un valor adecuado para el parámetro  $\gamma \neq 0$ . Repli-

tiendo este procedimiento es posible conformar una ecuación de iteración de este tipo

$$x^{K+1} = x^K - \gamma^{-1} \cdot f(x^K), \quad K = 0, 1, \dots \quad (\text{A.3})$$

en donde K es el contador de iteraciones.

La ecuación A.3 se puede extender a un sistema F, el cuál es n-dimensional y con un dominio  $R^n \rightarrow R^n$ , al reemplazar  $\gamma$  por una matriz constante no singular, A,

$$x^{K+1} = x^K - A_K^{-1} \cdot F(x^K) \quad (\text{A.4})$$

El punto crucial al emplear la ecuación A.4, es elegir un valor apropiado para la matriz A. Para el caso unidimensional, ecuación A.3, la pendiente  $f'(x^0)$  es una razonable elección para  $\gamma$ ; por extensión, para el caso n-dimensional se adopta un valor  $A_0 = F'(x^0)$ , y la ecuación A.4 se transforma en

$$x^{K+1} = x^K - F'(x^K)^{-1} \cdot F(x^K) \quad (\text{A.5})$$

sin embargo, la matriz  $F'(x^K)$  es conocida como matriz Jacobiana, por lo tanto, la ecuación A.5 usualmente se representa así:

$$x^{K+1} = x^K - J_K^{-1} \cdot F(x^K) \quad (\text{A.6})$$

y un reordenamiento de esta ecuación, especialmente orientada a ser utilizada dentro de un proceso iterativo, es

$$J_K \Delta x^K = -F(x^K) \quad (\text{A.7})$$

para la cual

$$\Delta x^K = [\Delta x_1, \Delta x_2, \dots, \Delta x_n]^T \quad (\text{A.8})$$

$$x^K = [x_{1,K} \cdot x_{2,K} \cdot \dots \cdot x_{n,K}]^T \quad (\text{A.9})$$

$$F(x^K) = [f_1(x^K), f_2(x^K), \dots, f_n(x^K)]^T \quad (\text{A.10})$$

$$J_K = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \dots & \frac{\partial f_1}{\partial x_n} \\ \vdots & & \vdots \\ \frac{\partial f_n}{\partial x_1} & \dots & \frac{\partial f_n}{\partial x_n} \end{bmatrix} \quad (\text{A.11})$$

El anterior desarrollo se conoce como método Newton simple, y sugiere este algoritmo:

- 1.- Elegir un valor inicial apropiado para  $x^K$ , en donde  $K=0$ .
- 2.- Evaluar  $J_K = F'(x^K)$
- 3.- Resolver el sistema lineal de la ecuación A.6, para  $\Delta x^K$
- 4.- Obtener  $x^{K+1} = x^K + \Delta x^K$  (A.12)
- 5.- Probar convergencia. Si no se logra hacer  $K = K + 1$  y continuar en el paso 2.

Generalmente, al promover el método hacia una solución, se requiere que  $x^{K+1}$  satisfaga alguna de estas desigualdades:

$$[F_i^2(x^{K+1})]^{1/2} < [F_i^2(x^K)]^{1/2} \quad (\text{A.13})$$

o bien,

$$[f_i(x^{K+1})^T f_i(x^{K+1})]^{1/2} < [f_i(x^K)^T f_i(x^K)]^{1/2} \quad (\text{A.14})$$

Si  $x^{K+1}$  no es un valor satisfactorio, una fracción,  $s_K$ , del

incremento calculado,  $\Delta x^K$ , es usada para modificar la ecuación A.12, es decir, se usa el paso 4 del algoritmo como una dirección de búsqueda lineal:

$$x^{K+1} = x^K + s_K \cdot x^K \quad (\text{A.15})$$

Al resolver el sistema lineal del paso 3 del algoritmo anterior mediante algún método directo (como los mostrados en el Anexo B) y tomando en cuenta que la solución de la matriz Jacobiana es equivalente a  $n^2$  evaluaciones de los elementos de las funciones, entonces los requerimientos computacionales para cada iteración por el método Newton son de  $n^2+2$  evaluaciones de elementos,  $O(n^2)$  operaciones aritméticas y un requerimiento de memoria de  $n^2 + O(n)$  posiciones.

Una desventaja de la anterior implementación es el tener que evaluar un nuevo Jacobiano en cada iteración. Para muchas aplicaciones típicas, la matriz Jacobiana no cambia muy rápidamente y puede ser evaluada únicamente a ciertos intervalos, fijados por anticipado. Por lo tanto, la ecuación A.5, se modifica en esta forma:

$$x^{K+1} = x^K - F'(x^{P(K)})^{-1} \cdot F(x^K) \quad (\text{A.16})$$

en donde  $P(K)$  es un entero menor o igual a  $K$ . Los casos extremos para la ecuación A.16 son cuando  $P(K)=K$ , lo que representa el método Newton simple, y cuando  $P(K)=0$  se tiene un esquema de iteración con la matriz Jacobiana constante. Suponiendo que  $F'(x)$  será reevaluada cada  $m$  iteraciones, entonces el caso para

el que  $m=1$  indica el método Newton original, por lo tanto, el caso general es

$$x^{K,i} = x^{K,i-1} - F'(x^{K,i-1})^{-1} F(x^{K,i-1}) \quad (\text{A.17})$$

$$i = 1, \dots, m$$

$$x^{K,0} = x^K$$

De este desarrollo se obtiene el siguiente algoritmo:

- 1.- Elegir un valor inicial apropiado para  $x^K$ , en donde  $K=0$ .
- 2.- Hacer  $Z_i = x^K$ .
- 3.- Para  $i = 1, \dots, m$ :
  - 3.1.- Evaluar  $F'(x^K) = J_K$
  - 3.2.- Resolver el sistema  $F'(x^K) \Delta Z_i = -F(Z_i)$  (A.18)  
para obtener  $\Delta Z_i$
  - 3.3.- Hacer  $Z_{m+1} = Z_i + \Delta Z_i$
- 4.- Hacer  $x^{K+1} = Z_{m+1}$

Otra posible modificación del método Newton consiste en hacer una aproximación de las derivadas parciales de la matriz Jacobiana, mediante incrementos diferenciales hacia adelante, siendo las formulaciones más comunes:

$$\partial_j f_i(x) = [f_i(x + \sum_{K=1}^j h_{i,K} e^K) - f_i(x + \sum_{K=1}^{j-1} h_{i,K} e^K)] \cdot \left(\frac{1}{h_{ij}}\right) \quad (\text{A.19})$$

o bien,

$$\partial_j f_i(x) = [f_i(x + h_{ij} e^j) - f_i(x)] \cdot \left(\frac{1}{h_{ij}}\right) \quad (\text{A.20})$$

siendo los elementos  $h_{ij}$  parámetros de discretización y  $e_j$  representa la  $j$ -ésima columna de la matriz identidad, lo que indica que  $1 \leq j \leq n$ .

Con respecto a la ecuación A.20, si  $\Delta_{ij}(x, h)$  representa una aproximación diferencial de  $\partial f_i(x)$ , entonces

$$J(x, h) = \Delta_{ij}(x, h) \quad (\text{A.21})$$

y la ecuación de iteración asociada es

$$x^{K+1} = x^K - J(x^K, h^K)^{-1} \cdot F(x^K) \quad (\text{A.22})$$

cuya aplicación se conoce como Iteración Discreta de Newton.

Moré y Cosnard (1979), sugieren que los parámetros  $h$  sean calculados de esta forma

$$\begin{aligned} h &= \epsilon |x| & ; \text{ para } x \neq 0 \\ h &= \epsilon & ; \text{ para } x = 0 \end{aligned} \quad (\text{A.23})$$

siendo  $\epsilon$  el menor número de punto flotante para el cuál  $1 + \epsilon > 1$  expresado en la precisión de la computadora usada en los cálculos.

Un algoritmo simple para esta implementación es:

- 1.- Elegir un valor inicial apropiado para  $x^K$ , en donde  $K=0$ .
- 2.- Construir la matriz  $J(x^K, h^K)$  con las ecuaciones A.20 y A.23.
- 3.- Resolver este sistema lineal para obtener  $\Delta x^K$ :

$$J(x^K, h^K) \cdot \Delta x^K = -F(x^K) \quad (\text{A.24})$$

- 4.- Hacer  $x^{K+1} = x^K + \Delta x^K$

Existen otras variaciones al método de Newton, en el sentido anteriormente aludido. Para más detalles acerca de estas implementaciones se recomienda consultar los textos de Ortega y Rheinboldt (1979) y de Moré y Cosnard (1979). Shacham y Mah (1978) proponen una modificación en base a la aplicación de varios métodos de linealización comunmente usados en problemas de redes de tubería, en los cuáles se manejan simultáneamente ecuaciones lineales y no lineales. Kubicek, Hlavacek y Prochaska (1976) muestran una detallada aplicación del método Newton a procesos de separación y rectificación en plantas complejas.

El método Newton-Raphson tiene varias ventajas, sobresaliendo su buena estabilidad, lo que implica la existencia de un dominio de atracción,  $s$ , el cuál contiene la solución,  $x^*$ , y que las iteraciones permanecerán dentro de ese dominio para que, eventualmente, converjan a  $x^*$ . Otra ventaja es su convergencia local cuadrática, en donde cada iteración cercana a la raíz, duplica el número de cifras significativas en  $x^K$ , y se cumple

$$\lim_{K \rightarrow \infty} \frac{\|x^{K+1} - x^*\|}{\|x^K - x^*\|^2} = C \quad (\text{A.25})$$

siendo  $C$  una constante positiva.

Adicionalmente este método es autocorrectivo en cuanto a que  $x^{K+1}$  sólo depende de  $F$  y de  $x^K$ , y los posibles efectos divergentes de las iteraciones previas no son acumulados, además es muy poco sensible al reescalamiento, lo que significa que su compor-

tamiento teórico no es alterado si las funciones,  $f$ , o las variables,  $x$ , son reescaladas durante el desarrollo del método.

Otra importante ventaja es el manejo lineal de sus subsistemas. Si algunas de las ecuaciones que están siendo resueltas son lineales, entonces toda la secuencia de iteraciones generada por el método Newton las resolverá, de aquí que sea más ventajoso que al realizar la simulación de un proceso químico, usar flujos molares de los componentes en cada corriente en lugar del flujo molar total y fracciones mol, puesto que el balance de materia es lineal en el primer caso, pero será bilineal en el segundo.

Sin embargo, el método Newton-Raphson tiene varias desventajas, siendo la más notable el tener que evaluar la matriz Jacobiana por completo en todas o en la mayoría de las iteraciones y esto se complica cuando esa matriz es difícil de calcular, siendo frecuente que se torne singular y conduzca a severas inconsistencias; además, el tener que realizar la inversión de la matriz Jacobiana en cada iteración es excesivamente costoso en tiempo de cómputo.

Para abordar el problema de la singularidad del Jacobiano se han propuesto varias soluciones, entre las que destacan la aplicación del algoritmo de Levenberg-Marquardt, según la implementación realizada por Moré (1977); la modificación directa de  $J(x^k)$ , según Gill y Murray (1972) y la cancelación de la singularidad modificando las ecuaciones, de Levy y Segura (1979).

Dos estudios recientes acerca del comportamiento del método

Newton-Raphson aplicado sobre problemas de simulación de procesos son los de Stadtherr y Hilton (1982) y Bogle (1983), los cuáles lo aplican a problemas reales de la Ingeniería Química.

#### METODO DEL DESCENSO MAXIMO.

Al aplicar el método de Newton, si  $x^K$  no es lo suficientemente cercana a la solución,  $x^*$ , no se garantiza que  $x^{K+1}$  sea una mejor aproximación que  $x$  y el método no converge.

Un procedimiento que garantiza que la condición de convergencia, indicada por las ecuaciones A.13 ó A.14, será cumplida es el Método del Descenso Máximo.

Considerando la ecuación

$$F(x) = 1/2 f^T(x) \cdot f(x) \quad (A.26)$$

se define una dirección de máximo descenso,  $g$ , cómo

$$g = - \nabla F(x) = -J^T \cdot f(x) \quad (A.27)$$

y conociendo el valor de  $x^K$ , el paso de descenso máximo es

$$\beta^K = x^{K-1} - x^K = \mu^K \cdot g^K \quad (A.28)$$

eligiendo  $\mu^K$  tal que, en la ecuación A.26,  $f(x)$  sea reemplazada por  $f^K(x) + J^K \beta^K$  y se utilice algún procedimiento de minimización. Se ha demostrado que el valor de  $\mu^K$  que minimiza

$$F(x^K) = 1/2 [f^K(x) + \mu^K J^K \beta^K]^T \cdot [f^K(x) + \mu^K J^K \beta^K] \quad (\text{A.29})$$

está dado por

$$\mu^K = \left[ \frac{g^T \cdot g}{g^T \cdot J^T \cdot J \cdot g} \right]^K \quad (\text{A.30})$$

A pesar de que el método del Descenso Máximo evita la divergencia, su comportamiento es muy lento, por lo que se han propuesto técnicas que utilizan una combinación de este método con otros de desarrollo más rápido, tal y como se verá más adelante al describir los algoritmos de Powell y Levenberg-Marquardt.

#### METODO DE WEGSTEIN.

Es muy común, sobre todo cuando la convergencia tiene un comportamiento lineal, desear acelerar el proceso de solución. El método usado con mayor frecuencia es el propuesto por Wegstein (1958) en base a los trabajos de Aitken (1925) y Steffensen (1933). Wegstein sugirió aplicar la fórmula de aceleración  $\rho^2$  en forma separada para cada variable en base a un procedimiento de sustituciones sucesivas.

Partiendo del problema base, que es la solución de un sistema del tipo mostrado por la ecuación A.1, transformado en

$$x = f(x) \quad (\text{A.31})$$

que conforma un esquema iterativo de este tipo

$$x^{K+1} = f(x^K) \quad (\text{A.32})$$

y si la ecuación A.1 no puede ser expresada en la forma de la ecuación A.31, entonces se usa

$$x^{K+1} = x^K + T \cdot F(x^K) \quad (\text{A.33})$$

en donde,  $T \neq 0$ , es una constante.

Wegstein propone modificar el valor de la última iteración,  $x^{K+1}$ , y reemplazarlo por  $\bar{x}^{K+1}$ , dado por

$$\bar{x}^{K+1} = q \cdot x^K + (1-q) \cdot x^{K+1} \quad (\text{A.34})$$

y en donde  $q$  se obtiene así

$$q = \frac{a}{a - 1} \quad (\text{A.35})$$

$$a = \frac{\bar{x}^{K+1} - x^K}{x^K - x^{K-1}} \quad (\text{A.36})$$

todo lo cual logra una convergencia cuadrática.

Este procedimiento es exitoso en los casos en que la interacción entre variables no es muy fuerte, por ejemplo, cuando la matriz Jacobiana es diagonal dominante.

A pesar de que el método de Wegstein es utilizado ampliamente, su ineficiencia para varias aplicaciones típicas es cada vez más reconocida, como lo demuestran los trabajos de Mahalec, Kluzik y Evans (1979) y Perkins (1979).

MÉTODOS DE SECANTE.

Cuando las expresiones analíticas de las derivadas parciales de la matriz Jacobiana no están disponibles, una aproximación a ellas mediante diferencias finitas, puede ser utilizada.

Sea  $\bar{J}(x_s^K)$  una aproximación de secante a la matriz Jacobiana verdadera, obtenida a partir de  $n$  pasos linealmente independientes, junto con sus correspondientes cambios en  $f(x)$ .

Cualquier iteración,  $K$ , tiene  $n$  puntos  $x_{K,j}$ ;  $n$  pasos  $p_{K,j}$   $j = 1, 2, \dots, n$  y los correspondientes cambios en la función

$$q_{K,j} = f(x_{K,j} + p_{K,j}) - f(x_{K,j}) \quad (\text{A.37})$$

y las matrices,  $P$  y  $Q$ , de dimensión  $n \times n$ , se definen así

$$P_K = [p_{K,1} \ p_{K,2} \ \dots \ p_{1,n}] \quad (\text{A.38})$$

$$Q_K = [q_{K,1} \ q_{K,2} \ \dots \ q_{1,n}] \quad (\text{A.39})$$

por lo tanto, la aproximación  $\bar{J}$ , debe satisfacer a

$$Q_K = \bar{J}_K \cdot P_K \quad (\text{A.40})$$

existiendo varias formas de elegir  $x_{K,j}$  y  $p_{K,j}$ . El método discreto de Newton corresponde a  $x_{K,j} = x^K$ , para toda  $j$ , y  $p_{K,j} = h^K \cdot e_j$ . Buscando la convergencia cuadrática, Steffen- sen elige

$$p_{K,j} = f(x^K) \cdot e_j \quad (\text{A.41})$$

$$j = 1, 2, \dots, n$$

y así es posible generar toda una serie de métodos de este tipo, reemplazando  $e_j$  en la ecuación A.41, por algún otro conjunto de vectores linealmente independientes. Por regla general, la mayoría de los algoritmos se concluyen cuando

$$\begin{aligned} \|x_K - x_{K-1}\| &\leq \xi \\ \|f(x_K) - f(x_{K-1})\| &\leq \xi \end{aligned} \quad (\text{A.42})$$

son satisfechas, para alguna pequeña tolerancia  $\xi$ .

Por lo tanto, la convergencia cuadrática no se logra cuando  $\|p_{K,j}\| \leq \xi$  y se requieren  $2n$  evaluaciones de la función de ese vector para determinar  $J_K$ ; pero elecciones más adecuadas de  $e_j$  pueden reducir este número, siendo  $n+1$  el mínimo de evaluaciones, tal y como es logrado por el método discreto de Newton y los métodos de tipo Steffensen.

Reducciones más significativas se pueden obtener usando evaluaciones que tomen en cuenta los puntos iterativos pasados, y si todos los pasos,  $p_{K,j}$ , han sido previamente usados en las iteraciones, entonces sólo una nueva evaluación de la función es requerida en cada iteración, lográndose lo que se conoce como Método de Secante General o Método de Secante  $n+1$ .

Este concepto fué abordado por Bitther (1959) y Barnes (1965), mostrando que la aproximación  $J_K$  podría ser generada en forma recursiva en  $O(n^2)$  operaciones. Wolfe (1959), propuso

elegir  $p_{k,j}$  como la diferencia entre los  $n+1$  puntos  $x_{k,j}$ , evaluados previamente y que estén en posición general. Se dice que los  $n+1$  puntos  $x^0, x^1, \dots, x^n$  están en posición general si los vectores  $x^0 - x^j$ , para  $j=1, \dots, n$ , son linealmente independientes; además, si las  $f(x_{k,j})$  también están en posición general, entonces  $x^{k+1}$  puede ser expresada como una combinación lineal de los puntos  $x_{k,j}$ , con los coeficientes obtenidos al resolver un sistema lineal. Ortega y Rheinboldt (1970), muestran un extenso desarrollo del método de secante y sus variantes.

Crowe y Nishio (1975) propusieron el Método General del Eigenvalor Dominante, el cual puede ser empleado como un procedimiento de aceleración para cualquier método iterativo.

Ellos asumen que los pasos sucesivos satisfacen una ecuación lineal de diferencias

$$x^{k+1} - x^k = A (x^k - x^{k-1}) \quad (\text{A.43})$$

de modo que los  $r+1$  pasos sucesivos deben ser linealmente dependientes para alguna  $r < n$ . Los factores de ponderación en la relación lineal proporcionan suficiente información como para predecir el punto límite de la secuencia  $(x_k)$  al utilizar la ecuación A.43, y éste es tomado como el siguiente punto para el proceso iterativo básico que está siendo empleado. Estos autores sugieren comenzar con una elección a-priori de  $r < n$ , y a continuación evaluar los factores de ponderación bajo un enfoque de mínimos cuadrados, resolviendo

$$\min_{\mu_K} \left\| \sum_{j=0}^r \mu_{K,j} (x_{K-j} - x_{K-j-1}) \right\|^2 \quad (\text{A.44})$$

$$\text{con } \mu_{K,0} = 1$$

y el nuevo punto es obtenido mediante

$$x_{K+1} = x_K - \frac{\sum_{j=0}^{r-1} \sum_{i=j-1}^r \mu_{K,i} (x_{K-i} - x_{K-i-1})}{\sum_{j=0}^r \mu_{K,j}} \quad \text{---- (A.45)}$$

En su trabajo original, estos autores establecen varias formas de verificar que paso de aceleración es el apropiado, siendo su conclusión que aún para pequeños valores de  $r$ , este procedimiento es mucho más efectivo que la aceleración  $\delta^2$  aplicada a cada variable por separado, tal como lo propone Wegstein (1958).

Brown (1966), introdujo un método de tipo secante diferente, en el cuál las aproximaciones hechas a las ecuaciones son resueltas una a una. El último punto encontrado es proyectado hacia la siguiente ecuación, y puesto que esta proyección puede ser realizada en varias formas, una amplia variedad de métodos de este tipo pueden ser generados. En particular, el método de Brown utiliza una forma de eliminación que minimiza los requerimientos computacionales. Muy significativa resulta la modificación de Brent (1973), que bajo el mismo principio, utiliza una proyección ortogonal.

MÉTODOS QUASI-NEWTON

Estos métodos son conocidos también como Métodos de Modificación, ya que transforman el problema original en una secuencia de subproblemas más simples, los cuáles al ser resueltos, conducen a la solución del problema inicial.

El primero de esta clase de métodos fué propuesto por Davidon (1959). Partiendo de una aproximación,  $B$ , a  $\nabla F(x)$ , se desea obtener una mejor aproximación,  $\bar{B}$ , a  $\nabla F(\bar{x})$ , en donde  $\bar{x}$  está dada por

$$\bar{x} = x - t \cdot H \cdot F(x) \quad (\text{A.46})$$

y para un proceso iterativo

$$x^{K+1} = x^K - t \cdot H_K \cdot F(x^K) \quad (\text{A.47})$$

siendo  $H_K = B_K^{-1}$  y  $t$  un factor de amortiguamiento.

Davidon sugirió esta fórmula para actualizar  $B$

$$B^{K+1} = B^K - \frac{(B^K \cdot q^K - y^K) (B^K \cdot q^K - y^K)^T}{(B^K \cdot q^K - y^K)^T \cdot q^K} \quad (\text{A.48})$$

y la inversa,  $H^{K+1}$ , se obtiene así

$$H^{K+1} = H^K - \frac{(H^K \cdot y^K - q^K) (H^K \cdot y^K - q^K)^T}{(H^K \cdot y^K - q^K)^T \cdot q^K} \quad (\text{A.49})$$

$$\text{y en donde} \quad y^K = F(x^{K+1}) - F(x^K) \quad (\text{A.50})$$

$$q^K = x^{K+1} - x^K \quad (\text{A.51})$$

El principal defecto de esta implementación es que tanto  $y$  como  $(H y - q)$  pueden ser ortogonales, por lo tanto, el procedimiento podría no estar definido para ciertas iteraciones.

Algunas modificaciones a este método fueron sugeridas por Fletcher y Powell (1963), mediante las cuáles se evitan los problemas de ortogonalidad, pero incrementan en gran medida los requerimientos computacionales del proceso iterativo.

Una reciente modificación de este método ha sido propuesta por Soliman (1985), el cuál elimina el problema de ortogonalidad y garantiza una definición positiva que es conservada en cada iteración, con un requerimiento computacional moderado.

Este autor actualiza el valor de  $B$ , mediante

$$B^{K+1} = B^K - \theta \frac{(B^K \cdot q^K - y^K) (B^K \cdot q^K - y^K)^T}{(B^K \cdot q^K - y^K)^T \cdot q^K} \quad (A.52)$$

en donde,  $\theta$ , es un parámetro a ser evaluado. Una posible elección de  $\theta$  para el caso en que  $t=1$ , es

$$\theta = \frac{(B^K \cdot q^K - y^K)^T \cdot q^K}{(B^K \cdot q^K - y^K)^T \cdot H^K \cdot F(x^{K+1})} \cdot \frac{q^K^T \cdot F(x^{K+1})}{q^K^T \cdot B^K \cdot q^K} \quad (A.53)$$

Broyden (1965), propuso varios métodos para resolver sistemas de ecuaciones no lineales, en los cuáles no se utiliza una matriz Jacobiana verdadera, sino una aproximación a ella, en cada etapa

del proceso iterativo, para lo cual se define

$$p_K = -B_K^{-1} \cdot f(x^K) \quad (\text{A.54})$$

siendo  $p_K$  un vector que modifica la dirección en cada paso de búsqueda y  $B_K$  es una aproximación a la matriz Jacobiana en  $x^K$ .

A partir de una modificación del método de Newton, se tiene

$$x^{K+1} = x^K + t_K \cdot p_K \quad (\text{A.55})$$

en donde  $t_K$  es un escalar que previene la divergencia del procedimiento, mediante la reducción de alguna norma de  $f(x)$  en cada etapa. Prosiguiendo con el desarrollo de Broyden, se tiene:

$$y_K = f(x^{K+1}) - f(t_K - s_K) = s_K \cdot B^{K+1} \cdot p_K \quad (\text{A.56})$$

en donde

$$s_K = x^{K+1} - x^K \quad (\text{A.57})$$

Si los cálculos son desarrollados reteniendo la matriz  $B$  en la memoria de la computadora, será necesario resolver  $n$  ecuaciones lineales para calcular el vector  $p$  usando la ecuación A.54; pero si la inversa de esa matriz es almacenada, esta operación se reduce a una multiplicación de la matriz por el vector, por lo tanto se define

$$H_K = -B_K^{-1} \quad (\text{A.58})$$

y las ecuaciones A.54 y A.56 se convierten en

$$p_K = H_K \cdot f(x^K) \quad (\text{A.59})$$

$$H_{K+1} \cdot y_K = -s_K \cdot p_K \quad (\text{A.60})$$

y a pesar de que la matriz  $H_K$  cambia en cada iteración, no se requiere evaluar  $f(x)$  en cada etapa, y es así como se obtiene una ecuación de iteración:

$$B^{K+1} = B^K + \frac{(y^K - s_K \cdot B^K \cdot p_K) \cdot p_K^T}{s_K \cdot p_K^T \cdot p_K} \quad (\text{A.61})$$

Sin embargo, como se mencionó anteriormente, se prefiere trabajar con  $H^K$  en lugar de  $B^K$ . Usando la fórmula de modificación de Householder se logra este cambio en la ecuación A.61:

$$H^{K+1} = H^K - \frac{s_K p_K + H^K \cdot y^K \cdot p_K^T \cdot H^K}{p_K^T \cdot H^K \cdot y^K} \quad (\text{A.62})$$

El algoritmo propuesto por Broyden se resume así:

- 1.- Elegir un valor inicial apropiado para  $x^K$ , en donde  $K=0$ .
- 2.- Obtener un estimado inicial para  $H^0$ . Esto se puede lograr calculando la matriz Jacobiana y luego invirtiéndola.
- 3.- Calcular  $f(x^K)$ .
- 4.- Calcular  $p_K = H^K f(x^K)$ .
- 5.- Seleccionar un valor de  $t_K$ , a fin de que la norma para la ecuación  $f(x^K + t_K \cdot p_K)$  sea menor que la norma de  $f(x^K)$ ; de no contar con un valor adecuado se puede emplear  $t_K = 1$ .
- 6.- Calcular:  $x^{K+1} = x^K + t_K \cdot p_K$
- 7.- Calcular  $f(x^{K+1})$  y probar su convergencia. Si no converge pasar al punto 8.

- 8.- Calcular  $y^K = f(x^{K+1}) - f(x^K)$  y  $s_K = x^{K+1} - x^K$ .
- 9.- Calcular  $H^{K+1}$ , mediante la ecuación A.62.
- 10.- Regresar al paso 4.

La gran ventaja del método de Broyden es la drástica reducción de los requerimientos computacionales, en comparación con el método de Newton. En las más eficientes implementaciones se tuvieron  $n$  evaluaciones de los componentes, en comparación a las  $n^2 + n$  del método Newton; Broyden realiza  $O(n^2)$  operaciones aritméticas, en comparación con las  $O(n^3)$  de Newton. Sin embargo, este ahorro se logra a expensas del cambio en la convergencia; en efecto, el método Broyden adquiere un comportamiento de convergencia superlineal, para el cuál se cumple

$$\lim_{K \rightarrow \infty} \frac{\|x^{K+1} - x^K\|}{\|x^K - x^*\|} = 0 \quad (\text{A.63})$$

a diferencia del método Newton, el cuál muestra convergencia cuadrática. El método Broyden es invariante con respecto a los cambios de escala en las funciones, pero no en las variables; Paloschi (1982) y Paloschi y Perkins (1982), han demostrado que una adecuada selección de  $p_K$  puede mejorar el método de Broyden, en cuanto a su eficiencia en el reescalamiento de variables.

Otra seria desventaja del método Broyden es la de requerir el uso de la inversa de la matriz Jacobiana, la cual no conserva la dispersión original del sistema.

Schubert (1970), propuso un método en el cuál se toma en cuenta la presencia de muchos elementos con valor de cero o constantes dentro de las matrices Jacobianas dispersas, los que se definen como elementos conocidos. Este método está destinado a conservar la dispersión del sistema de ecuaciones a lo largo del proceso de solución.

La ecuación A.61 puede ser modificada haciendo sustituciones adecuadas de  $y$  y  $p$ , partiendo de las ecuaciones A.55 y A.56, respectivamente, con lo que se obtiene

$$B^{K+1} = B^K + \frac{[f(x^{K+1})_i - (1-s_K) f(x^K)_i] \Delta x^K^T}{s_K \Delta x_i^K \Delta x^K} \quad (A.64)$$

Schubert impuso la condición de que los elementos conocidos (ceros y constantes), permanecieran sin cambio al realizar la evaluación del Jacobiano y por el uso de esta y otras condiciones, obtuvo una ecuación fila por fila

$$b_i^{K+1} = b_i^K + \frac{[f(x^{K+1})_i - (1-s_K) f(x^K)_i] \Delta x_i^K^T}{s_K \Delta x_i^K \Delta x_i^K} \quad (A.65)$$

en la cuál:

$$i = 1, 2, \dots, n$$

$n$  = orden de la matriz Jacobiana.

$\Delta x_i^K$  = Vector de columna, derivado de  $\Delta x^K$ , haciendo cero cada

elemento de  $\Delta x^K$  que corresponde a algún elemento de  $b_i^K$  el cuál es una constante conocida.

$b_i^K$  = Vector de fila, que contiene los elementos de la  $i$ -ésima fila del Jacobiano aproximado  $B^K$ .

$f(x^K)_i$  =  $i$ -ésimo elemento de  $f(x^K)$ .

Debe notarse que la corrección de Schubert no conduce por sí misma a una eficiente implementación de la matriz Jacobiana inversa. Por tanto,  $B^{K+1}$  debe ser ordenada en cada etapa del proceso iterativo.

Broyden (1971), presentó un trabajo en el que revisa su propuesta original de 1965 e incluye las modificaciones sugeridas por Schubert, desarrollando las propiedades tanto teóricas como experimentales de su algoritmo modificado.

Lam (1978), hace un estudio sobre la convergencia del método de Broyden cuando el Jacobiano es altamente disperso.

Mah y Lin (1980), comparan los métodos de Newton, Broyden-Schubert y un método híbrido resultante de combinar estos dos.

El método híbrido utilizó el método Newton con diferencias finitas para iniciar el proceso y obtener una aproximación a la solución, para aplicar el método Broyden-Schubert con el objeto de promover el acceso final a la solución. El cambio de un

método a otro, dentro del algoritmo híbrido, es indicado por:

$$1) \quad N = \left[ \sum_{i=1}^n [w_i f(x)_i]^2 \right]^{1/2} < 1 \quad (A.66)$$

$$2) \quad \frac{(N^{K+1} - N^K)}{(T^K - T^{K-1})} \geq \frac{(N^{L-1} - N^L)}{(T^L - T^{L-1})} \quad (A.67)$$

en donde:

$N^K$  = Norma Euclideana después de la K-ésima iteración.

$T^K$  = Tiempo de cómputo después de la K-ésima iteración.

$N^L$  = Norma Euclideana después de la última iteración de Newton.

$T^L$  = Tiempo de cómputo después de la última iteración de Newton

$w_i$  = Factor de ponderación de cada ecuación del sistema.

Estos criterios tienen como objetivo realizar el cambio de un método a otro cuando resulta más ventajoso. Los resultados reportados por Mah y Lin indican que, para el caso específico al que fueron aplicados, los tres métodos lograron la convergencia. El método Newton consume más tiempo, pero no presenta inestabilidades, como sucedió con el método Broyden-Schubert, el cual converge más rápido. El método híbrido tuvo un buen comportamiento, siendo el más rápido y el que menos inestabilidad presentó.

Gallum y Holland (1980), propusieron un método que aplicado

al algoritmo de Broyden, resulta ser altamente competitivo con la formulación Broyden-Schubert. Este método evita la principal dificultad de la implementación de Schubert, la cuál consiste en tener que factorizar la matriz en cada etapa del procedimiento iterativo. El método de Gallum y Holland sólo requiere factorizar la matriz inicial  $J_0$  para  $K=0$ .

Lucia (1983), presenta una modificación al algoritmo de Gallum y Holland, en la cuál el Jacobiano no es aproximado en cada iteración, sino que se verifica cada cierto número de iteraciones

Se demuestra que estas modificaciones aún conservan la propiedad de convergencia superlineal.

#### METODO DE LEVENBERG-MARQUARDT.

Cuando no se dispone de un buen estimado inicial hacia la solución de un sistema de ecuaciones y cuando éstas son altamente no lineales, los métodos del tipo Newton y quasi-Newton presentan serios problemas. El más grave de ellos es que la matriz Jacobiana puede tornarse singular, provocando que el subsistema de iteración lineal se transforme en inconsistente. Ante esta situación, se ha propuesto transformar las ecuaciones no lineales

originales, en un problema no lineal de mínimos cuadrados y que tiende a obtener un valor local minimizado para

$$s(x) = \sum_{i=1}^n D_{f,i}^2 \cdot \bar{f}_i^2(x) = \|D_f F(x)\|^2 = \|\bar{F}(x)\|^2 \quad (\text{A.68})$$

en donde,  $D_f$  es una matriz diagonal de escalamiento, con elementos diagonales positivos,  $D_{f,i}$ . El nuevo estimado de la solución puede ser aceptado o rechazado usando la ecuación A.68 como guía.

Levenberg y Marquardt propusieron un método para la solución numérica de la ecuación A.68. Partiendo de un valor para  $x$ , se debe encontrar un factor de corrección,  $p$ , que minimice:

$$\phi(p) = s(x+p) = \|D_f F(x+p)\|^2 = \|F(x+p)\|^2 \quad (\text{A.69})$$

por lo tanto,  $x+p$  puede ser la solución buscada. Puesto que  $\phi$  usualmente es una función no lineal de  $p$ , se debe linealizar  $f(x+p)$  y considerar la función

$$Q(p) = \|D_f F(x) + B \cdot p\|^2 \quad (\text{A.70})$$

en donde  $B$  es el Jacobiano de  $f(x)$  en  $x$ , o una aproximación a él. Por supuesto, esta linealización no es válida para todos los valores de  $p$ , por lo tanto, se debe restringir el tamaño de  $p$  y considerar el siguiente problema lineal restringido de mínimos cuadrados:

$$\min_p Q(p) \quad (\text{A.71})$$

$$\text{sujeto a } \|D_x p\| = \|\bar{p}\| \leq \Delta \quad (\text{A.72})$$

siendo  $D_x$  una matriz diagonal con elementos diagonales positivos y  $\Delta$  es el radio de la región en la cual la linearización deberá ser confinada. La solución de las ecuaciones A.70 y A.71, está dada por

$$p(\lambda) = - (B^T \cdot D_f^2 \cdot B + \lambda \cdot D_x^2)^{-1} B^T \cdot D_f^2 \cdot F \quad (\text{A.73})$$

en donde el parámetro  $\lambda$  debe satisfacer estas condiciones:

$$1) \quad \lambda = 0 \quad \text{y} \quad \|D_x \cdot p(0)\| \leq \Delta \quad (\text{A.74})$$

$$2) \quad \lambda > 0 \quad \text{y} \quad \|D_x \cdot p(\lambda)\| = \Delta \quad (\text{A.75})$$

Estas relaciones sugieren este algoritmo simplificado:

- 1.- Dado  $\Delta_K > 0$ , encontrar  $\lambda_K \geq 0$ , tal que si  $p_K(\lambda_K)$  es definido por la ecuación A.73, entonces tanto A.71 como A.72 se cumplan.
- 2.- Si  $\|D F(x_K + p_K)\| < \|D F(x_K)\|$ , entonces calcular  $x^{K+1} = x^K + p_K$ ; de no cumplirse hacer  $x^{K+1} = x^K$ .
- 3.- Determinar  $\Delta_{K+1}$ , para la siguiente iteración.

Cuando  $\lambda = 0$  y la matriz B es cuadrada y no singular, entonces la ecuación A.73 se convierte en

$$p(0) = -(B^T \cdot D_f^2 \cdot B)^{-1} B^T \cdot D_f^2 \cdot F = -B^{-1} \cdot F = p^N \quad (\text{A.76})$$

el cuál es justamente el paso correctivo del método de Newton, si B es el Jacobiano; o es el paso correctivo del método Broyden, si B es una aproximación al Jacobiano. Para el método Newton la dirección es independiente de la ecuación y variable de escalamiento,  $D_f$  y  $D_x$ , respectivamente; en cambio para el método

de Broyden esta dirección es independiente de  $D_f$ , pero no de  $D_x$ .

Una curva típica en dos dimensiones para  $p(\lambda)$  se muestra en la Figura A.1. De acuerdo a las ecuaciones A.71 y A.72 si  $\Delta \approx \|D_x p(0)\|$ , entonces se tiene el paso de corrección  $p(0)$  representado por el vector  $\overline{OB}$ , desde el origen hasta el punto B de la Figura A.1; de otra forma, se determina el paso de corrección a partir de la intersección de la curva  $D_x p(\lambda)$  y del círculo de radio  $\Delta$ .

### METODO DE POWELL.

La principal idea del método híbrido de Powell (1970) es la de aproximar la curva  $p(\lambda)$  mediante una línea quebrada  $\overline{OAB}$  tal como se muestra en la Figura A.2.

La intersección de  $\overline{OAB}$  y el círculo de radio  $\Delta$ , puede ser determinada analíticamente. En el método de Powell,  $\overline{OA}$  se encuentra en la dirección descendente máxima,  $g$ , definida por

$$p(\lambda) = -\lambda^{-1} \cdot D_x^{-2} \cdot B^T \cdot D_f^2 \cdot F \equiv \lambda^{-1} \cdot g \quad (\text{A.77})$$

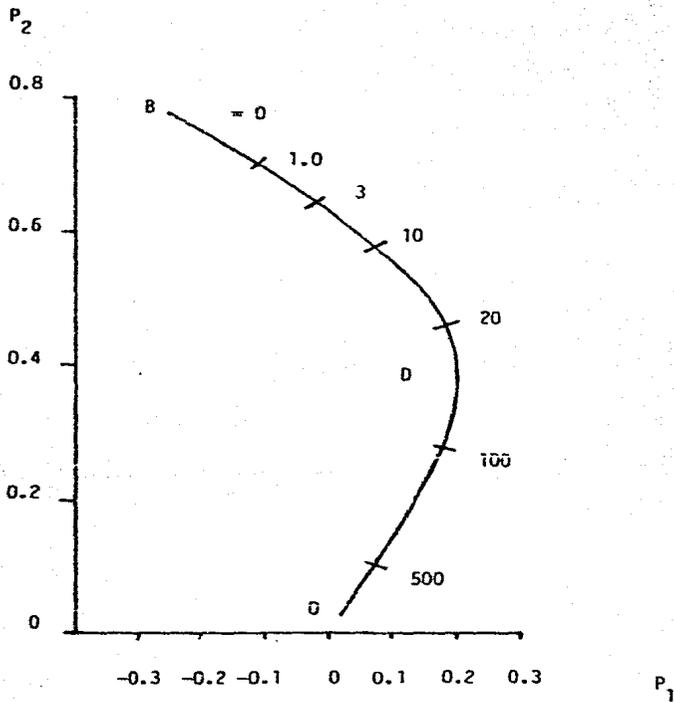


FIGURA A. 1. CURVA TIPICA PARA  $P(\lambda)$  USANDO EL METODO DE LEVENBERG - MARQUARDT.

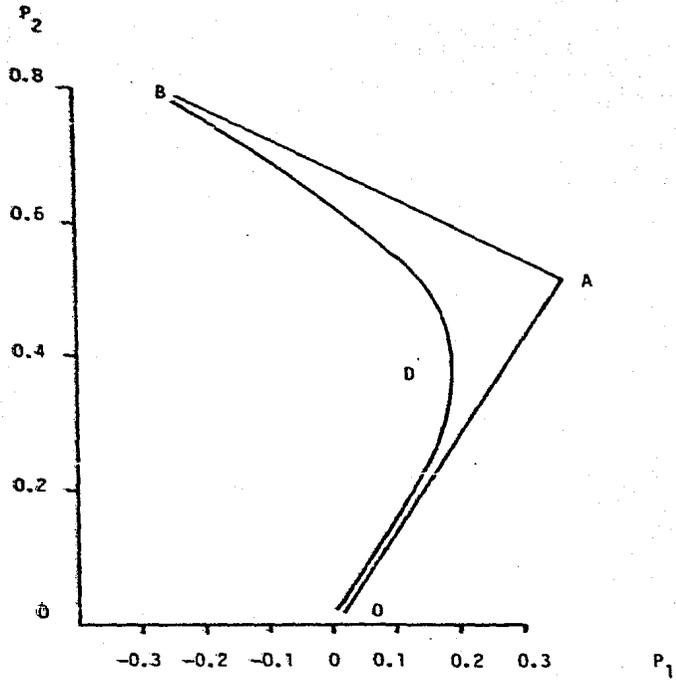


FIGURA A. 2. CURVA TIPICA PARA  $P(\lambda)$  USANDO EL METODO HIBRIDO DE POWELL.

es decir

$$\vec{O}\bar{A} = p^S = u \cdot g \quad (\text{A.78})$$

y en donde  $u$  es elegida para minimizar la función

$$Q(u, g) = \left\| D_f(F + B \cdot u \cdot g) \right\|^2 \quad (\text{A.79})$$

y como  $Q(u, g)$  es una función cuadrática de  $u$ , entonces

$$u = \frac{-F^T \cdot D_f^2 \cdot B \cdot g}{\left\| D_f \cdot B \cdot g \right\|^2} = \frac{\left\| D_x \cdot g \right\|^2}{\left\| D_f \cdot B \cdot g \right\|^2} \quad (\text{A.80})$$

Para simplificar, se definen estas cantidades escaladas:

$$\bar{B} = D_f \cdot B \cdot D_x^{-1} \quad (\text{A.81})$$

$$\bar{F} = D_f \cdot F(x) \quad (\text{A.82})$$

$$\bar{p}^N = D_x \cdot p^N = -B^{-1} \cdot F \quad (\text{A.83})$$

$$\bar{g} = D_x \cdot g = -\bar{B}^T \cdot F \quad (\text{A.84})$$

$$\bar{p}^S = D_x \cdot p^S = \frac{\left\| \bar{g} \right\|^2}{\left\| \bar{B} \cdot \bar{g} \right\|^2} \cdot \bar{g} \quad (\text{A.85})$$

Habiendo calculado  $\bar{p}^N$  y  $\bar{p}^S$ , se ha definido por completo la línea quebrada  $\overline{OAB}$ . En su trabajo original, Powell sugiere varias formas de calcular el paso de corrección escalado,  $p = D_x p$ .

De la Figura A.2 es posible notar que, para el caso bidimensional, los métodos de Levenberg-Marquardt ( $\overline{ODB}$ ) y de Powell ( $\overline{OAB}$ ), son muy aproximados. Para el caso  $n$ -dimensional, la curva  $p_k(\lambda)$ , tal vez no se presente en el plano  $p^N$  y  $p^S$ .

sin embargo, los dos métodos aún generan pasos de corrección similares.

Westerberg y Director (1978), estudiaron las propiedades de los dos métodos anteriormente descritos. Del método Levenberg-Marquardt resaltan sus características de convergencia, en cambio, del método de Powell sobresalen sus propiedades de eficiencia computacional. Estos autores desarrollan un algoritmo que retoma las ventajas de cada método y genera un paso de corrección de longitud máxima, que asegura la convergencia, pero con el inconveniente de requerir  $3n^2 + O(n)$  posiciones de memoria en la computadora.

Chen y Stadtherr (1981), presentaron un trabajo en el que sugieren una modificación al método híbrido de Powell, con el objeto de mejorar su eficiencia y reducir los requerimientos computacionales cuando es aplicado a grandes sistemas de ecuaciones. Las principales modificaciones son:

1) Proporcionar un generador automático de escalamiento de las ecuaciones y/o variables, lo que incrementa drásticamente la eficiencia.

2) Powell utiliza  $n^2 + O(n)$  posiciones de memoria para poder mantener la dirección de búsqueda a fin de que el Jacobiano aproximado,  $B_k$ , eventualmente converja a la solución. Chen y Stadtherr encontraron que esto se puede lograr mediante la

simple reevaluación del Jacobiano por diferencias finitas.

3) Tanto el Jacobiano aproximado,  $B^K$ , como su inversa, son almacenados en el método de Powell, requiriendo  $2n^2 + O(n)$  posiciones de memoria; por su parte, Chen y Stadtherr sólo almacenan los factores L y U de  $B^K$ , utilizando sólo  $n^2 + O(n)$  posiciones de memoria.

Chen y Stadtherr (1984), nuevamente modifican el método de Powell, pero ahora aplicando el algoritmo de Schubert (1970) para evaluar el Jacobiano y con especial orientación a la solución de grandes sistemas dispersos de ecuaciones no lineales.

#### CONCLUSIONES ACERCA DE LOS METODOS DE CONVERGENCIA LOCAL Y EXPANDIDA.

Es difícil designar un método en particular como extraordinariamente eficiente. Cada uno tiene ciertas ventajas que lo hacen adecuado a determinadas situaciones, y ciertas desventajas que lo convierten en prohibitivo para otras.

Usualmente, el método Newton-Raphson se comporta adecuadamente al tener buenos estimados iniciales de  $x$  y cuando el Jacobiano es sencillo y fácil de evaluar, exhibiendo convergencia cuadrática que le confiere una velocidad razonable; sin embargo, uno de sus inconvenientes es el tener que evaluar el Jacobiano por completo en cada iteración, o en la mayoría de ellas.

Por su parte, los métodos quasi-Newton requieren buenos estimados iniciales y puede abordar exitosamente problemas en los que el Jacobiano tiene cierta complejidad, puesto que no evalúa el Jacobiano verdadero, sino una aproximación a él; sin embargo, no explota el carácter disperso de la matriz Jacobiana y en los cálculos sucesivos inclusive se puede perder la dispersión.

La modificación de Schubert induce y preserva la dispersión en el método de Broyden, sin embargo, introduce el problema de tener que factorizar el Jacobiano en cada iteración.

La propuesta de Gallum y Holland, preserva la dispersión sin recurrir a la factorización del Jacobiano, pero todavía falta probar su eficiencia en problemas complejos.

En cuanto a los métodos de convergencia expandida, tales como el de Powell y el de Levenberg-Marquardt, con sus respectivas modificaciones, parten de estimados iniciales medianamente próximos a la solución y pueden abordar problemas con un Jacobiano relativamente difícil de evaluar, puesto que combina las ventajas de convergencia global del procedimiento de Descenso Máximo, con

las propiedades de convergencia local del método Broyden (en el caso del método Powell), y con el método Newton-Raphson (en el caso del método Levenberg-Marquardt); sin embargo, existe cierta dificultad en su implementación sobre problemas muy complejos, en virtud de que el algoritmo de estos métodos es poco flexible a los cambios estructurales del proceso al que se aplica y en ciertos casos sus requerimientos computacionales son altos, existiendo una fuerte tendencia a producir un conjunto de ecuaciones mucho más denso que en el método de Newton.

Se han realizado numerosos intentos por comparar los diferentes métodos existentes; sin embargo, son muy pocos los trabajos cuidadosamente realizados al respecto. La única forma realmente confiable de comparar varios métodos consiste en aplicarlos a problemas típicos reales, pero con una cuidadosa selección de las bases de comparación y requisitos de implementación (precisión de la computadora usada, pruebas de terminación para cada método, etc.), para que un método sea realmente comparable con otro. Un criterio de prueba esencial es la velocidad de convergencia, la cuál no siempre es posible medirla bajo las mismas condiciones para los diversos métodos; Dennis y Moré (1977), elaboraron un estudio acerca de los criterios que deben tomarse en cuenta al evaluar la velocidad de convergencia.

Con respecto a la eficiencia de cada método se puede afirmar que, en forma general, no pueden ser comparados totalmente.

Lo anterior significa que existen numerosas bases de comparación, tales como la mayor o menor complejidad de las estructuras del proceso químico al que se aplicarán, el grado de no linealidad de las ecuaciones descriptivas, velocidad de convergencia, requerimientos computacionales, características de la computadora usada, etc. Además, varios de estos criterios no pueden ser satisfechos simultáneamente; por lo tanto, un método puede cumplir adecuadamente ciertos requisitos y ser totalmente ineficiente para otros. Por todo lo anterior se considera que las pruebas de eficiencia con mayor validez son aquellas que son realizadas sobre problemas reales, y no únicamente en análisis de tipo teórico.

Existen varios trabajos en los cuáles los métodos de solución de ecuaciones no lineales, son probados en problemas reales.

Mah y Lin (1980), realizaron una comparación de los métodos Broyden-Schubert, Newton con diferencias finitas y un método híbrido de los dos anteriores, aplicados a cuatro problemas reales diferentes, bajo diversas situaciones, obteniendo valiosos criterios de comparación y esferas de aplicación de cada uno.

Shacham (1985 a), realizó una comparación entre cinco programas de computadora que manejan los métodos de Brent, Brown, Powell, Homotopía-continuación y un método resultante de combinar los métodos de Newton, Levenberg-Marquardt y Homotopía-continuación, aplicados sobre diez diferentes problemas reales.

Uno de los trabajos más completos es el desarrollado por Hiebert (1982), al comparar ocho programas de computadora que emplean diferentes versiones de los métodos de Brent, Brown, Powell y quasi-Newton, aplicados a una serie de veintitrés problemas cuidadosamente preparados para hacer resaltar las ventajas y desventajas de cada método.

Shacham (1985 b), presentó un trabajo en el cuál empleó doce problemas representativos de algunas áreas de la Ingeniería Química, además de los usados por Moré (1981). Propone un algoritmo, denominado CONLES, el cuál es capaz de manejar ecuaciones bajo restricciones, tanto físicas como absolutas. Las restricciones físicas son aquellas originadas por los fenómenos físicos que representan el sistema de ecuaciones, de aquí que sea posible obtener una solución matemáticamente válida, pero que no sea prácticamente admisible por no satisfacer las restricciones físicas. Las restricciones absolutas son aquellas originadas por el limitado intervalo de definición de algunas funciones matemáticas. En el algoritmo CONLES, las restricciones absolutas son manejadas con los métodos Newton-Raphson o Broyden; en cambio las restricciones físicas son convertidas en restricciones absolutas mediante el uso de funciones de ponderación, o manejadas directamente empleando métodos de Homotopía-continuación, además, al identificar una matriz singular emplea una modificación del método Levenberg-Marquardt. La eficiencia del método se ve incrementada con la incorporación de una nueva técnica

de descomposición de sistemas de ecuaciones en base a operaciones de partición y separación ("tearing"), las cuáles son descritas como parte del Apéndice B.

Lo anterior ilustra una reciente tendencia hacia el uso de varios métodos combinados con el objeto de crear algoritmos más versátiles y eficientes y es precisamente esta idea la que motiva el desarrollo de los métodos de convergencia global, representados por los métodos de Homotopía-Continuación.

## A.II. METODOS DE CONVERGENCIA GLOBAL.

### METODOS DE HOMOTOPIA-CONTINUACION.

En la sección anterior se mostraron varios métodos, los cuáles, en general, convergen a una solución,  $x^*$ , de  $F(x)=0$ , sólo

si las aproximaciones iniciales son suficientemente cercanas a  $x^*$

Esos métodos son esencialmente locales, y a pesar de que sus propiedades de convergencia local estén bien definidas, esto no es garantía de que logren la convergencia a partir de valores iniciales arbitrarios. Si bien es cierto, los anteriores métodos tienden a forzar la convergencia mediante el ajuste del tamaño y dirección de cada paso de avance y búsqueda a fin de asegurar la disminución de una cierta norma en cada iteración, pero aún su reducción de la norma no asegura la convergencia global, presentando en muchos casos cierta inestabilidad, aún para puntos cercanos a la solución.

Recientemente han sido desarrollados una serie de métodos tendientes a ampliar el dominio de convergencia de un cierto método, o alternativamente, operar como un procedimiento para generar estimados iniciales suficientemente cercanos a  $x^*$ , partiendo de valores arbitrarios para  $x^0$ , para luego ser aplicados a los métodos de la sección anterior.

Estos son conocidos bajo varias denominaciones: métodos de homotopía, métodos de continuación, métodos de carga incremental o métodos de inserción. Ficken (1951), introdujo por vez primera el término "método de continuación", referido a los aspectos técnicos de esta técnica al ser aplicados a ecuaciones con operadores. Sin embargo, la primera aplicación como un método práctico en la solución de sistemas de ecuaciones no lineales

fué sugerida por Davidenko (1953). A partir de ese trabajo, se han realizado notables progresos en el perfeccionamiento de este método, a la luz de las actuales facilidades computacionales y aunque su uso es muy común entre matemáticos, su introducción al campo de la Ingeniería Química es reciente.

El problema básico es encontrar la solución de un sistema de ecuaciones no lineales, del tipo

$$F(x) = 0 \quad (\text{A.86})$$

el cuál está definido en  $R^n \rightarrow R^m$ .

Asimismo, se considera un conjunto de funciones

$$G(x) = 0 \quad (\text{A.87})$$

definido en  $R^n \rightarrow R^m$ , el cuál tiene un conjunto de solución ya conocido, o es susceptible de resolverse con facilidad.

La tarea del método de Homotopía-Continuación es la de realizar una deformación continua de la función  $G(x)$ , cuya solución es conocida, hacia la función  $F(x)$ , cuya solución es la deseada.

Esta deformación es acompañada por la creación de

$$H(x,t) \quad (\text{A.88})$$

que es una función definida en  $R^{n+1} \rightarrow R^m$ , la cuál contiene un parámetro adicional,  $t$ , conocido como parámetro o factor de ho-

motopía, y que adquiere valores entre 0 y 1. Por lo tanto, existe un campo unidimensional y continuo,  $T$ , conocido como ruta homotópica y que une la solución del sistema en A.87, con la solución del sistema en A.86. Visto de este modo, la ecuación A.88 es una función de  $x(t)$  y continúa para  $t$ ; por lo tanto, es posible resolver una secuencia de subproblemas con  $t$  decreciendo desde 1 hasta 0, usando la solución de un subproblema como el estimado inicial para el siguiente y eventualmente alcanzar la solución del sistema original,  $F(x)=0$ , en  $x(0)$ .

Las condiciones límite de la ecuación A.88 son:

$$\text{a } t=0 : \quad H(x,0) = G(x) = 0 \quad (\text{A.89})$$

cuya solución, en  $x^0$ , es conocida.

$$\text{a } t=1 : \quad H(x,1) = F(x) = 0 \quad (\text{A.90})$$

que representa el problema original a ser resuelto; por lo tanto, existirá alguna  $x^*$ , tal que  $F(x) = 0$ , y que será la solución.

La familia de funciones representada por la ecuación A.88, puede adquirir varias formas, siendo las más comunes las de tipo lineal, llamadas así porque  $H(x,t)$  es lineal en el parámetro  $t$ :

$$H(x,t) = t F(x) + (1-t) \cdot G(x) = 0 \quad (\text{A.91})$$

o bien:

$$H(x,t) = F(x) + (t-1) \cdot F(x^0) = 0 \quad (\text{A.92})$$

Dos ejemplos de ecuación homotópica lineal son:

1) Homotopía de Punto Fijo.- En este caso la función  $G(x)$  puede ser reescrita como un problema de punto fijo:

$$G(x) = x - x^0 \quad (\text{A.93})$$

y a partir de la ecuación A.91, se genera

$$H(x,t) = tF(x) + (1-t)(x - x^0) \quad (\text{A.94})$$

2) Homotopía de Newton.- Se llama así por su relación con el método de Newton (García y Zangwill (1981)), al definir

$$G(x) = F(x) - F(x^0) \quad (\text{A.95})$$

y a partir de la ecuación A.91, se genera

$$H(x,t) = tF(x) + (1-t)[F(x) - F(x^0)] \quad (\text{A.96})$$

$$= F(x) - (1-t)F(x^0) \quad (\text{A.97})$$

Es posible generar ecuaciones homotópicas no lineales, mediante la adecuada definición de  $G(x)$ .

#### HOMOTOPIA DE NEWTON Y DE PUNTO FIJO.

Para la Homotopía de Newton, una secuencia de valores de  $t$  son seleccionados, resolviendo la ecuación A.97 para  $x$ , usando

el método de Newton en cada valor de  $t$ , con la solución previa utilizada como punto inicial. Este procedimiento es mostrado en la Figura A.3, en donde  $\alpha$  representa el punto inicial, a  $t=t_2$ , para el método Newton, el cuál itera entre  $\beta$  y  $\gamma$ .

En este procedimiento básicamente se va siguiendo la secuencia de cambios en  $t$ , prestando poca atención a la ruta homotópica.

Wayburn (1983), presentó un ejemplo de aplicación del método clásico de homotopía de Newton, en el cuál destaca el principal defecto de éste, y que es él que la ruta homotópica no siempre obtiene una convergencia directa, sino que puede presentar algunos puntos de retorno. Con el objeto de ilustrar esta clase de comportamiento, se elige esta función

$$f(x) = x^3 - 30x^2 + 280x - 860 = 0 \quad (\text{A.98})$$

Al aplicar el método Newton-Raphson a la ecuación A.98, se encontró que, a partir de un estimado inicial  $x^0 = 12.6$ , existe una raíz positiva en  $x^* = 15.5506$ , tal y cómo se muestra en la Figura A.4; sin embargo, para valores de  $x^0 < 12.6$ , el método Newton-Raphson no converge.

Aplicando el método de Homotopía de Newton, ecuaciones A.95, A.96 y A.97, al problema de la ecuación A.98, se encontró la solución mostrada en la Figura A.5. Se pueden observar seis rutas homotópicas, correspondientes a valores iniciales,  $x^0$ , de  $-5, 0, 5, 7.418, 15, 20$  y  $30$ ; las rutas para  $5$  y  $12.5819$

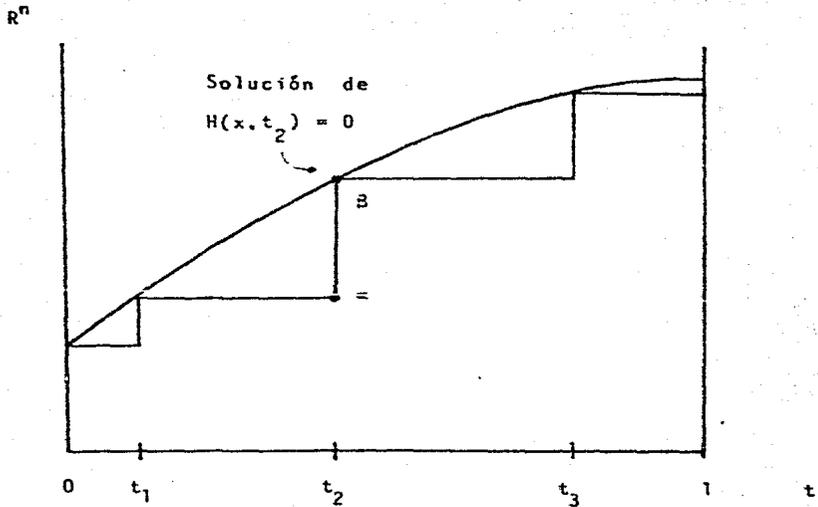


FIGURA A. 3 . EJEMPLO DEL METODO DE HOMOTOPIA - CONTINUACION CLASICO.

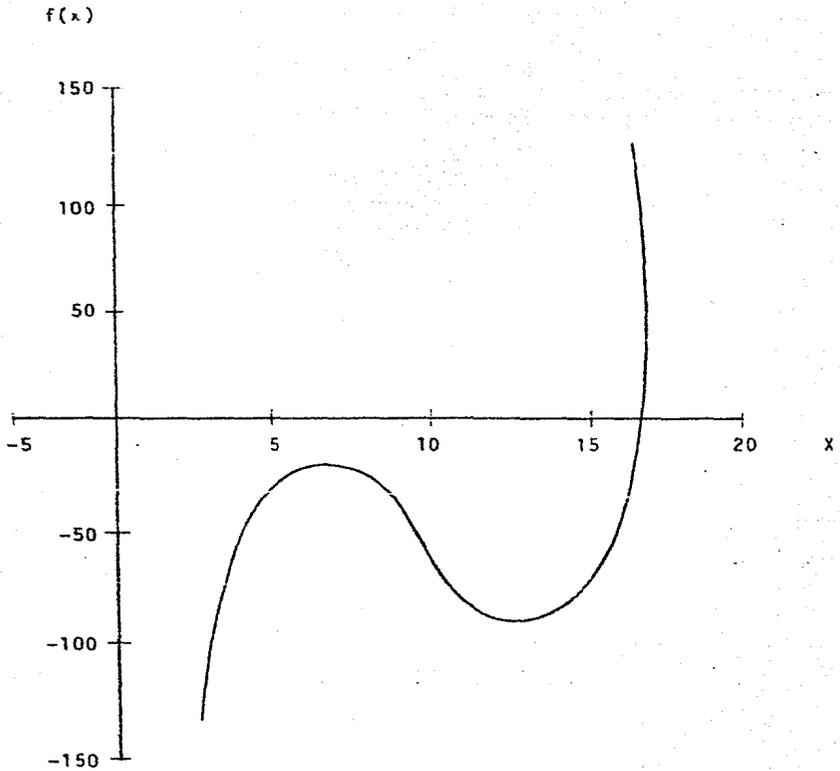


FIGURA A.4. GRAFICA DE LA FUNCION A.98.

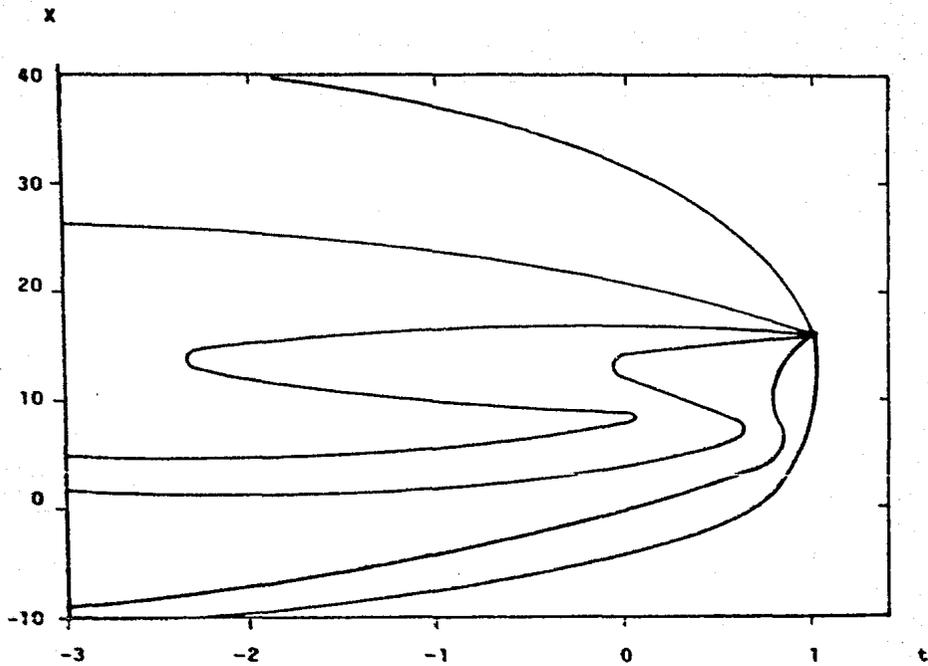


FIGURA A.5. RUTA HOMOTOPICA PARA LA ECUACION A.98  
EMPLEANDO HOMOTOPIA DE NEWTON.

son coincidentes, y las rutas para 7.418 y 15, también coinciden. La secuencia de valores que se usó fué de 0, 0.1, 0.3, 0.6 y 1.0. Debe notarse que para  $x^0$  mayor de 12.5819 no hay puntos de retorno y la convergencia a  $x^*$  es rápida; en cambio, para puntos menores a 12.5819, se logra la convergencia, pero con varios puntos de retorno, e inclusive, para  $x^0 = 5$  se presenta una ruta muy rebuscada. Se puede afirmar que, en este caso, el método de Homotopía de Newton no es competitivo con el método Newton-Raphson.

Aplicando la Homotopía de Punto Fijo, ecuaciones A.93 y A.94, al problema de la ecuación A.98, se encuentra la solución que se muestra en la Figura A.6, observándose una ruta más simple pero que aún presenta varios puntos de retorno.

Para problemas más complejos a veces se generan rutas homotópicas excesivamente rebuscadas y que acrecentan los requerimientos computacionales para su definición, tal como la mostrada en la Figura A.7, que fué generada a partir de un problema propuesto por Kubicek (1976).

Una posible solución a las desventajas de la homotopía de Punto Fijo y de Newton, es seguir más de cerca la ruta homotópica en sí misma, más que seguir los incrementos en  $t$ .

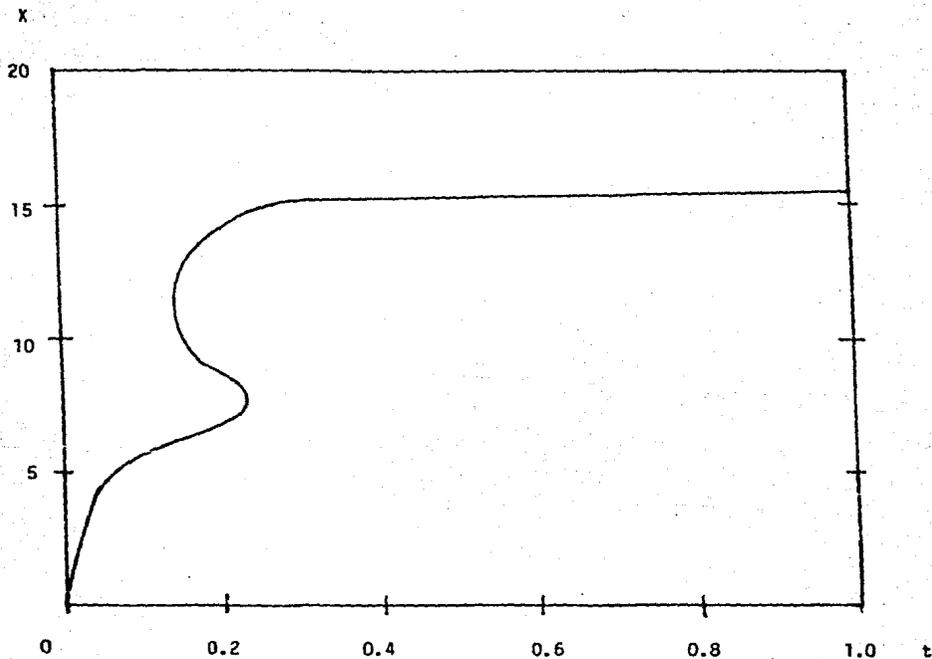


FIGURA A. 6. RUTA HOMOTOPICA PARA LA ECUACION A. 98 ,  
EMPLEANDO HOMOTOPIA DE PUNTO FIJO.

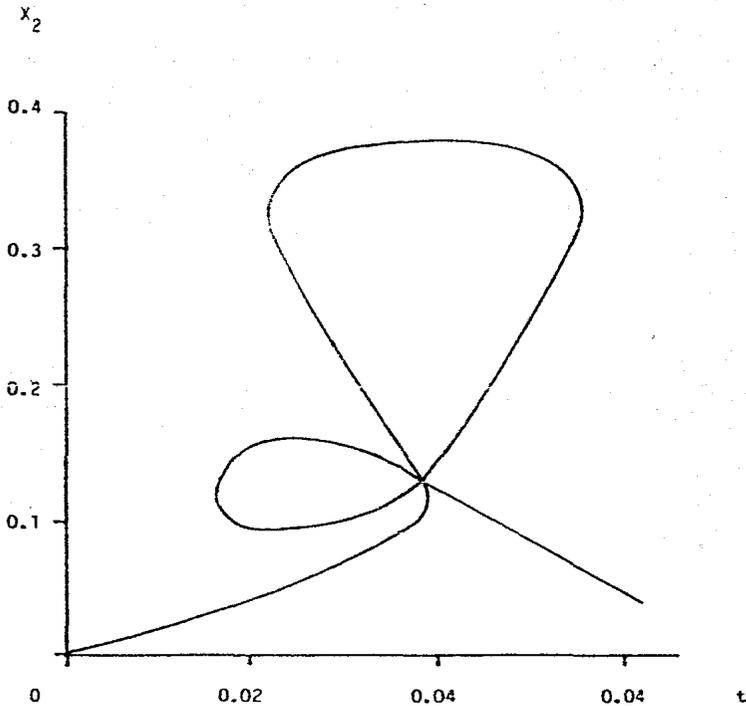


FIGURA A. 7. EJEMPLO DE RUTA HOMOTOPICA PARA UN PROBLEMA COMPLEJO.

### HOMOTOPIA DE LONGITUD DE ARCO DIFERENCIAL.

Wayburn (1983), propuso un método en el que el conjunto de ecuaciones no lineales es reformulado como un conjunto de ecuaciones diferenciales ordinarias simultáneas, que son diferenciadas con respecto a una longitud de arco,  $r$ , de la ruta homotópica. Por fortuna, las ecuaciones diferenciales pueden ser resueltas eficazmente por métodos conocidos.

Partiendo de la ecuación homotópica A.97, se hace una sustitución con  $\psi = 1-t$ , y siguiendo la ruta homotópica,  $T$ , desde  $\psi = 1$  hasta  $\psi = 0$ :

$$H(x, t) = H(x, \psi) = F(x) - \psi \cdot F(x^0) \quad (\text{A.99})$$

a  $\psi = 1$ , la solución es  $x^0$ , mientras que a  $\psi = 0$ , la solución es  $x$ , que representa la raíz buscada para  $F(x)$ .

Diferenciando la ecuación A.99 con respecto a  $r$ , se tiene

$$F'(x) \frac{dx}{dr} - \frac{d\psi}{dr} \cdot F(x^0) = 0 \quad (\text{A.100})$$

que contiene  $n$  ecuaciones para  $n+1$  funciones incógnitas de  $r$ ,  $x_1(r)$ ,  $x_2(r)$ , ...,  $x_n(r)$ ,  $\psi(r)$ .

El vector  $[(dx/dr)^T, d\psi/dr]^T$ , es uno de los números infinitos de vectores nulos de la matriz Jacobiana  $n \times (n+1)$ , correspondiente a la derivación total de  $H(x, \psi)$ .

Si  $\left\{ \left( \frac{dx}{dr} \right)^T, d\psi/dr \right\}^T$

$$\text{satisface a } \left( \frac{dx}{dr} \right)^T \left( \frac{dx}{dr} \right) + \left( \frac{d\psi}{dr} \right)^2 = 1 \quad (\text{A.101})$$

entonces la ecuación A.100 se transforma en:

$$\left( \frac{dx}{dr} \right)^2 + \left( \frac{d\psi}{dr} \right)^2 = 1 \quad (\text{A.102})$$

o sea,

$$\left( dx \right)^2 + \left( d\psi \right)^2 = \left( dr \right)^2 \quad (\text{A.103})$$

lo que representa al Teorema de Pitágoras. Por lo tanto, dr es la longitud de arco diferencial de la ruta entre dos puntos en el plano  $(x, \psi)$ .

Para el caso general:

$$\frac{dx}{dr} = \dot{x} \quad (\text{A.104})$$

$$\frac{d\psi}{dr} = \dot{\psi} \quad (\text{A.105})$$

y el problema se reduce a resolver

$$\left[ F'(x) - F(x^0) \right] \frac{\dot{x}}{\dot{\psi}} = 0 \quad (\text{A.106})$$

$$\dot{x}^T - \dot{x} + \dot{\psi}^2 = 1 \quad (\text{A.107})$$

con las condiciones iniciales:

$$x(0) = x^0 \quad (\text{A.108})$$

$$\psi(0) = 1 \quad (\text{A.109})$$

Las ecuaciones A.106, A.107, A.108 y A.109, constituyen un problema de valor inicial (PVI) para un sistema autónomo de ecuaciones diferenciales ordinarias implícitas y no lineales, en donde  $F'(x)$  es una matriz  $n \times n$ ,  $F(x_0)$  es un vector de columna  $n \times 1$ ,  $x$  es un vector de columna  $n \times 1$ , y  $\psi$  es un escalar. Estas ecuaciones deben ser resueltas mediante un método explícito para ecuaciones diferenciales, tal como el método Euler-Newton (predicor-corrector), o sus similares.

Un algoritmo general para el Método de Homotopía de Longitud de Arco Diferencial fué presentado por Seader (1985):

1.- Partir del problema  $F_i(x_1, x_2, \dots, x_n) = 0$ , para  $i = 1, 2, \dots, n$

2.- Construir una ecuación homotópica, con  $t$

$$H_i(x_1, x_2, \dots, x_n, t) = 0 \quad (\text{A.110})$$

en donde:  $t^0 \leq t \leq t_f$

con:  $x_i = \text{conocida}$   $x_i^0$  a  $t^0$

$$H_i = F_i \text{ a } t_f$$

3.- Convertir a un problema de valor inicial, con ecuaciones diferenciales ordinarias en términos de longitud de arco,  $r$ , y haciendo  $t = x_{n+1}$  :

$$\sum_{j=1}^{n+1} \frac{\partial H_i}{\partial x_j} \cdot \frac{dx_j}{dr} = 0 \quad (\text{A.111})$$

$$\sum_{j=1}^{n+1} \left[ \frac{dx_j}{dr} \right]^2 = 1 \quad (\text{A.112})$$

4.- Seleccionar un valor para una variable independiente,  $\pi$ , para prevenir la singularidad en la matriz Jacobiana y combinar las ecuaciones para obtener derivadas en forma explícita:

$$\frac{dx_\pi}{dt} = - \left[ 1 + \sum_{\substack{j=1 \\ j \neq \pi}}^{n+1} \gamma_j^2 \right]^{-1/2} \quad (\text{A.113})$$

con:  $\gamma = T_\pi^{-1} \cdot \frac{\partial H_i}{\partial x}$

$$T_\pi^{-1} = \text{Jacobiano de } \frac{\partial H_i}{\partial x} \text{, sin } \frac{\partial H_i}{\partial x}$$

$$\frac{dx_j}{dt} = \gamma \cdot \frac{dx}{dt} \text{, } j = 1, 2, \dots, \pi-1, \pi+1, \dots, n+1$$

5.- Calcular el predictor de Euler, para cierto  $\Delta t$ :

$$x_j^\pi = x_j^{\pi-1} + \Delta r \left[ \frac{dx_j}{dr} \right] \text{, } j=1, 2, \dots, n+1 \quad (\text{A.114})$$

6.- Resolver las ecuaciones de homotopía a un  $x_\pi$  fijo, para  $x$ , con  $j=1, 2, \dots, \pi-1, \pi+1, \dots, n+1$ ; mediante el método Newton como corrector para minimizar el error de truncación en la etapa de integración:

$$x_j^m = x_j^{m-1} - \left[ T_\pi^{m-1} \cdot H_j^{m-1} \right] \quad (\text{A.115})$$

- 7.- Prueba de convergencia. Se verifica si la curva homotópica está completada. Si la convergencia no se logra, regresar al paso 4.

### HOMOTOPIA TERMODINAMICA.

Los métodos de homotopía clásicos, a pesar de comportarse exitosamente dentro de un dominio de convergencia expandido, tiene varias desventajas. Una de las más serias es la referida a un comportamiento puramente matemático, que se presenta cuando la ecuación homotópica, en cualesquiera de sus formas, no representa un proceso físicamente existente, en especial durante la solución de las etapas intermedias del procedimiento, encontrándose casos en que al resolver algunos problemas intermedios (es decir, cuando el parámetro de homotopía,  $t$ , no es igual, ni a cero, ni a uno), involucran fracciones molares negativas.

Ante esta situación, Vickery y Taylor (1986), proponen un método que toma mayor ventaja de las características físicas del problema. Puesto que la no idealidad de los valores termodinámicos usados, especialmente los valores para  $K$  y las entalpías, son la causa de la mayoría de las dificultades al resolver las ecuaciones de los modelos del proceso, los autores crearon un tipo de homotopía que remueve la mayoría de las condiciones no

ideales del problema original, con el objeto de simplificar la aproximación inicial, y son agregadas en forma gradual conforme el parámetro de homotopía,  $t$ , se mueve de cero a uno, a fin de llegar a la solución en forma más eficiente.

Vichery y Taylor aplicaron su método a problemas de separación y como aproximación inicial redujeron los valores de  $K$  a su forma ideal y la entalpía fué tomada como la de una mezcla ideal: las condiciones no ideales se agregaron en forma de ecuaciones para  $K$  y la entalpía, en función del parámetro de homotopía.

Durante las pruebas de su método, estos autores encontraron que a pesar que algunos problemas intermedios no fuéron sistemas físicamente reales, el procedimiento los manejó adecuadamente.

El método opera sobre las ecuaciones diferenciales de homotopía (EDH), y en su implementación original utiliza el método Euler-Newton para resolverlas, a través de un procedimiento que los autores describen detalladamente. Un diagrama de flujo resumido de este método se muestra en la Figura A.8.

Como sus autores lo indican, la homotopía termodinámica no pretende sustituir al método Newton-Raphson en la solución de ecuaciones no lineales, sino más bien es un intento por incrementar el número de problemas en los cuáles el método Newton puede ser empleado con eficiencia y versatilidad. De este modo, si dos métodos están disponibles, será posible elegir el más preciso, y si ambos son igualmente precisos, entonces la elección

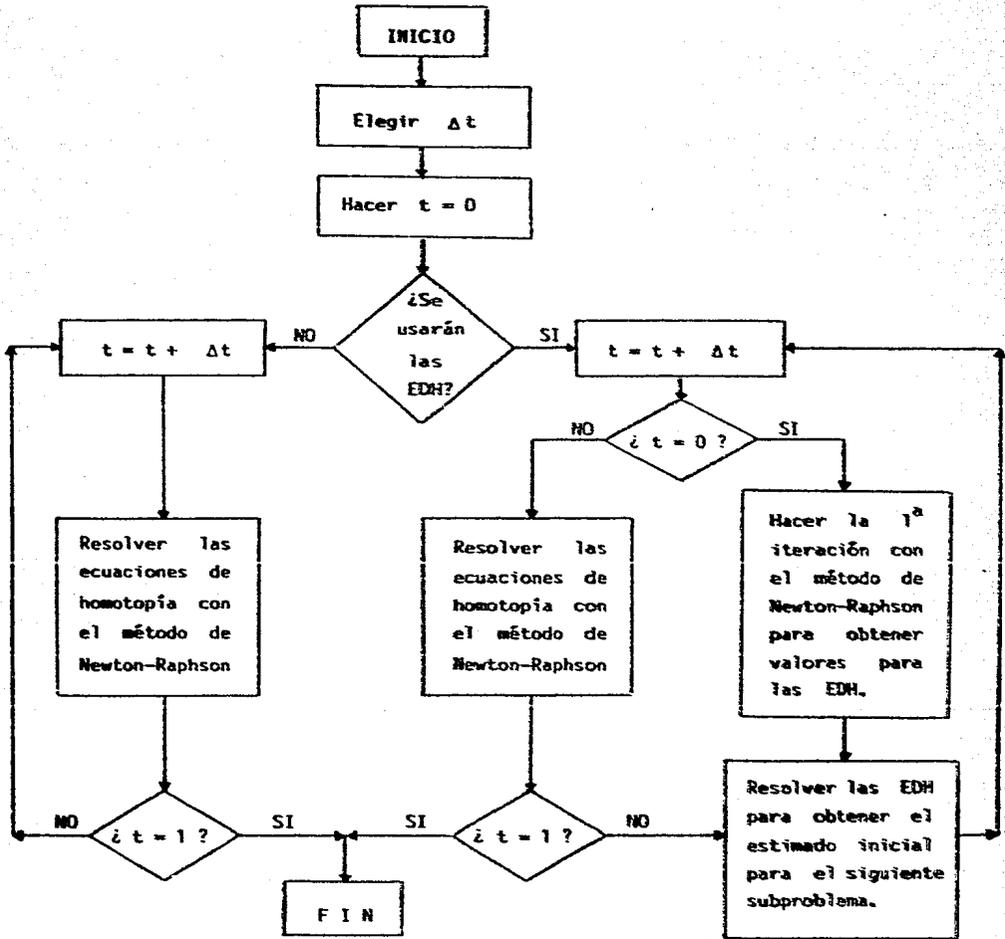


FIGURA A.8. DIAGRAMA DE FLUJO DE LA IMPLEMENTACION DE LA HOMOTOPIA TERMODINAMICA.

será en función del más eficiente en el aspecto computacional.

### CONCLUSIONES ACERCA DE LOS METODOS DE CONVERGENCIA GLOBAL.

Estos métodos están recibiendo una creciente atención y a partir del trabajo de Kubicek (1976), se han presentado muchas variaciones a esta técnica, sobre todo en cuanto a su implementación en base a diferentes métodos de solución de ecuaciones diferenciales o en cuanto a su eficiencia computacional.

Se han elaborado varios algoritmos integrados en grandes programas de computadora, sobresaliendo el ACM 502 de Kubicek (1976), el CJ 110 de Abboit (1980), el ACM 555 de Chow-Yorke (1980) y el ACM 596 de Rheinboldt-Burkardt (1983). Allgower y Georg (1980) presentaron una amplia introducción a los métodos de continuación, en base a un detallado análisis de sus propiedades. Den Heijer y Rheinboldt (1981) estudiaron las propiedades de convergencia de estos métodos junto con técnicas de tipo Newton y quasi-Newton.

Estos métodos son los más promisorios en la solución de ecuaciones no lineales. A pesar de ser manejados por los

matemáticos desde hace más de diez años, en general, son métodos de incorporación reciente a la Ingeniería Química y por lo tanto aún no han sido aplicados con amplitud a problemas típicos y se carece de comprobaciones que promuevan su aceptación.

Shacham (1983), comparó los métodos de homotopía con otros cuatro métodos (Brent, Brown, Newton y Levenberg-Marquardt); su conclusión fué que un método híbrido de Newton-Raphson, Levenberg-Marquardt y de homotopía, fué el que más éxito obtuvo al ser aplicado a diez problemas típicos, y destacó que el método híbrido es el que más se aproxima a un Método General de solución de ecuaciones no lineales.

Los métodos conocidos de homotopía presentan varias desventajas, como son el requerir altos recursos computacionales, y para algunos problemas típicos no son competitivos con métodos más simples. Otra desventaja es su poca flexibilidad a los cambios en la estructura de los problemas a los que se aplican, requiriendo una cuidadosa selección de las funciones de iteración y se requieren reglas confiables en la elección de una adecuada estrategia de cálculo.

Es bien conocido que en muchos casos, el método Newton tiene propiedades divergentes. La mayoría de las veces este problema de divergencia es abordado eligiendo nuevos estimados iniciales con la esperanza de que el procedimiento converja y en casos extremos no se abandona el método y forzando la convergencia.

no importando el número total de iteraciones.

Ante esta situación, los métodos de homotopía tienen una exitosa participación, en cuanto a que generan estimados iniciales adecuados en un número razonable de iteraciones, en comparación con el intento de convergencia forzada.

Es de aceptación general que la aproximación homotópica es más lenta en algunos casos, al menos e un orden de magnitud, en comparación con el método Newton-Raphson. Esto es verdad en cuanto a que se intenta seguir la trayectoria homotópica en forma muy ajustada. Sin embargo, para la mayoría de las aplicaciones en Ingeniería Química, es posible adoptar una estrategia diferente, siguiendo la ruta homotópica en la forma más holgada que sea posible, con la seguridad de que el punto final de las trayectorias es lo suficientemente aceptable como para producir la aproximación final a la solución mediante un método más simple.

Otra ventaja importante de los procedimientos de homotopía es su capacidad para manejar con éxito un Jacobiano singular.

Se puede afirmar que el método de homotopía es superior a los métodos de tipo Newton en cuanto a lograr la convergencia de problemas complicados y que el tiempo de cómputo puede considerarse como comparable.

Se continúa investigando en busca de lograr una mayor efi-

ciencia y competitividad con métodos de estructura más simple y que exploten de mejor manera el uso de los robustos métodos existentes para la solución de ecuaciones diferenciales ordinarias.

Un promisorio esfuerzo está siendo realizado al desarrollar sistemas expertos, los cuáles automáticamente determinan la naturaleza de las ecuaciones a ser resueltas y proporcionan varias estrategias viables de solución. Williams (1982), propuso un paquete conocido como TK! SOLVER, el cuál realiza el análisis de un cierto sistema de ecuaciones y genera una lista de métodos y procedimientos adecuados para su solución.

A P E N D I C E     B

MÉTODOS DE SOLUCIÓN DE  
MATRICES DISPERSAS.

## A P E N D I C E      B

### MÉTODOS DE SOLUCIÓN DE MATRICES DISPERSAS.

Los modelos que describen las diferentes operaciones unitarias de un proceso están compuestos por ecuaciones diferenciales parciales, ecuaciones diferenciales ordinarias, ecuaciones algebraicas no lineales o una combinación de estos tipos; sin embargo, uno o más pasos en el procedimiento de su solución numérica frecuentemente requiere la solución de sistemas de ecuaciones algebraicas lineales.

Estas ecuaciones tienen una estructura peculiar, caracterizada por el hecho de que algunos de sus elementos tienen un valor diferente de cero, y otros son igual a cero, combinados en proporciones variables. Los sistemas de ecuaciones lineales que presentan esta estructura, son los que constituyen las matrices dispersas.

Una matriz se considera como dispersa en términos de su den-

sidad, es decir, de la relación entre el número de elementos diferentes de cero y el número de elementos totales de la matriz. Se han propuesto varias ideas que intentan definir una matriz dispersa. Rice (1983), la define como una matriz en donde la mayoría de sus elementos son cero. Tewarson (1973), clasifica una matriz  $n \times n$  como dispersa, si tiene de dos a diez elementos diferentes de cero en cada fila, para altos valores de  $n$ . Para propósitos prácticos es adecuada la definición propuesta por Duff (1977), quién considera una matriz como dispersa a aquella de la que cierta ventaja puede ser obtenida del porcentaje y/o distribución de los elementos igual a cero, y estas ventajas se hacen evidentes tanto en la reducción del almacenamiento de datos, cómo en el número de operaciones aritméticas realizadas.

La mayoría de los sistemas de ecuaciones lineales tienen una solución, la cuál es obtenida por métodos directos, métodos iterativos o métodos especiales para ciertos tipos de matrices en particular. A mediados de los sesentas, fuéron desarrollados varios métodos directos, basados en las técnicas para matrices completas del tipo de reducción de Gauss-Jordan y de eliminación de Gauss, los cuáles toman poca o nula ventaja de la presencia de los elementos igual a cero.

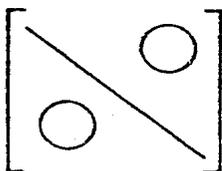
Para sistemas de ecuaciones grandes y medianamente dispersos los métodos del tipo de Iteración de Jacobi y de Iteración de Gauss-Seidel, fueron aplicados para obtener resultados aproxima-

dos. Sin embargo, para aplicaciones típicas dentro de la simulación y diseño de procesos, los sistemas de ecuaciones lineales son de varias decenas de miles de ecuaciones, por lo tanto se hace necesario reordenar las matrices dispersas con el objeto de organizar sus elementos en ciertas estructuras adecuadas, cómo las mostradas en las Figuras B.1 y B.2, y que faciliten su posterior evaluación numérica utilizando métodos directos especialmente desarrollados, usualmente en base a la eliminación de Gauss.

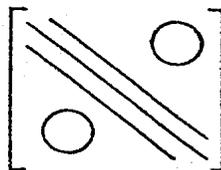
En la Figura B.1 se representan ciertas estructuras organizadas en cuanto a los elementos matriciales; en cambio, en la Figura B.2 la organización es en forma de bloques de submatrices; en ambos casos, se representa con  $x$  a los elementos diferentes de cero,  $0$  es un elemento o región con valor cero, cada cuadrado representa un bloque de elementos, algunos de los cuáles o todos son diferentes de cero, y una línea o triángulo contiene algunos o todos sus elementos diferentes de cero.

Los métodos para resolver en forma eficiente grandes sistemas de matrices dispersas, se orientan primordialmente a:

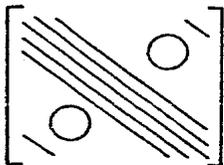
- 1) Reducir el almacenamiento de datos; especialmente los elementos con valor constante o con valor de cero.
- 2) Minimizar el tiempo requerido para recuperar los datos.
- 3) Reducir el número de operaciones aritméticas, especialmente las multiplicaciones y divisiones, las cuáles re-



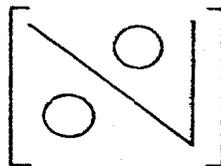
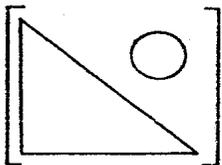
DIAGONAL (D)



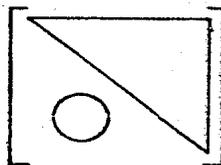
TRIDIAGONAL (TDF)



DIAGONAL MULTIPLE (MDF)

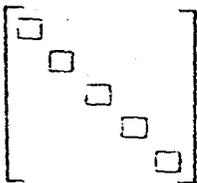
TRIANGULAR CON BORDE  
(BoTF)

TRIANGULAR -INFERIOR (LTF)

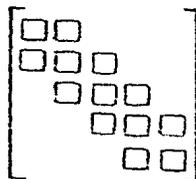


TRIANGULAR SUPERIOR (UTF)

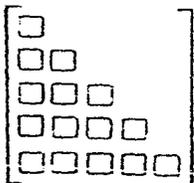
FIGURA B.1. ESTRUCTURAS ORGANIZADAS DE MATRICES DISPERSAS  
ELEMENTO POR ELEMENTO.



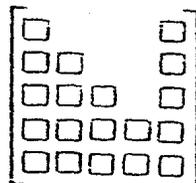
DIAGONAL (BDF)



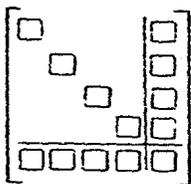
TRIDIAGONAL (BTDF)



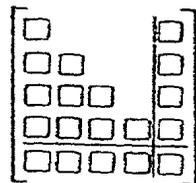
TRIANGULAR INFERIOR (BLTF)



TRIANGULAR INFERIOR CON BORDE (BBLTF)



DIAGONAL CON DOBLE BORDE (DBBDF)



TRIANGULAR INFERIOR CON DOBLE BORDE (DBBTF)

FIGURA B.2. ESTRUCTURAS ORGANIZADAS DE MATRICES DISPERSAS EN FORMA DE BLOQUES.

quieren más tiempo de cómputo que las sumas y restas.

4) Mantener la estructura dispersa durante los cálculos.

Por ejemplo, considerar el problema de resolver un sistema de ecuaciones lineales del tipo  $Ax + b$ , en donde el orden sea de  $n = 1000$ , con  $\Gamma = 2000$  elementos diferentes de cero. Si se utilizara la estructura de matriz completa sería necesario almacenar  $n^2$  elementos, o sea, un millón de datos, pero cerca de 998,000 de esos elementos serían ceros. Al emplear la eliminación de Gauss en la solución numérica, se necesitarían cerca de  $1/3 n^2$  operaciones aritméticas, por lo tanto, para resolver el sistema se realizarían operaciones del orden de un billón; y si una operación básica es realizada en un micro-segundo, entonces el tiempo de procesamiento del sistema será de mil segundos.

En contraste, al utilizar la representación dispersa, sólo los elementos de  $A$  diferentes de cero y sus factores triangulares inferiores y superiores, serán almacenados. Para casos típicos, los factores triangulares pueden ser almacenados con  $k_1 \Gamma$  posiciones de memoria y realizando  $k_2 \Gamma$  operaciones, en donde,  $k_1 = 1$  a 3 y  $k_2 = 5$  a 20. Esto implica almacenar menos de 10,000 elementos y realizar cerca de 20,000 operaciones en aproximadamente 20 mili-segundos.

Usualmente, todas las rutinas para resolver matrices dispersas consideran una fase de reordenamiento y una fase de solución

numérica.

La fase de reordenamiento tiene como finalidad principal el convertir la matriz de ocurrencia, es decir, la matriz que representa la relación simple y directa de las ecuaciones y variables descriptivas de una operación unitaria en particular o de un proceso en general, en una matriz con una estructura bien definida, que facilite su posterior manejo sin que sea reducido su carácter disperso; adicionalmente, se pretende obtener una secuencia de elementos con los cuales realizar un pivoteo durante la etapa numérica y asegurar la estabilidad del procedimiento.

La fase de solución numérica promueve la convergencia del sistema reordenado, hacia una solución que satisfaga a las ecuaciones que contiene.

#### B.1. ETAPA DE REORDENAMIENTO

La tarea inicial de esta etapa es convertir la matriz de ocurrencia en una matriz que adopte alguna estructura del tipo de las mostradas en las Figuras B.1 y B.2.

El rasgo común de estas estructuras es el de poseer un arre-

glo diagonal o triangular de los elementos diferentes de cero.

La elección de estas estructuras no es de ningún modo arbitraria, tiene su fundamento en las características propias de los sistemas dispersos. Como será analizado en la siguiente sección, la mayoría de los procedimientos de solución numérica están basados en la eliminación de Gauss, la cual al ser aplicada a matrices dispersas produce un llenado potencial de la matriz. En efecto, el llenado puede ocurrir potencialmente en las intersecciones de las "columnas diferentes de cero" de la fila de pivoteo a la derecha de la diagonal, con las "filas diferentes de cero" de la columna de pivoteo abajo de la diagonal. La "columna diferente de cero" significa que la fila tiene un elemento diferente de cero en esa columna; y la "fila diferente de cero" significa que la columna tiene un elemento diferente de cero en esa fila. El llenado ocurre "potencialmente" porque puede ya existir un elemento diferente de cero en una intersección dada.

Por ejemplo, en la Figura B.3 A se muestra una matriz de ocurrencia típica, en donde las  $x$  representan la ocurrencia de una variable en una cierta ecuación. En la Figura B.3 B se presenta la misma matriz, después de haber realizado el primer paso de la eliminación de Gauss, notándose que el llenado se produjo en las posiciones (4,4), (4,7) y (4,10), representando con F a los elementos que se "crearon". Esto sugiere que manteniendo el número de tales intersecciones al mínimo, el lle-



nado también puede ser minimizado.

Por todo lo anterior, la estructura cero/diferente de cero más deseable es la mostrada en la Figura B.3 C, la que es conocida como Forma Triangular Inferior (FTI). Se puede notar que no existen "columnas diferentes de cero" a la derecha de la diagonal, por lo tanto, no se producirá llenado durante la solución numérica. Sin embargo, en la práctica no es muy común encontrar esa estructura definida con claridad, por lo tanto, se pretende lograr una estructura lo más próxima posible a la Forma Triangular, de aquí la variedad de formas mostradas en las Figuras B.1 y B.2. Se puede deducir que si no es posible evitar totalmente el llenado de la matriz, al menos se pretende limitarlo a ciertas áreas de la misma.

Otro objetivo importante del reordenamiento es el descomponer un gran sistema de ecuaciones y variables, en una serie de subsistemas más pequeños, susceptibles de ser resueltos uno por uno, en cierto orden y obtener la solución del sistema original. Esto es de gran importancia, puesto que los sistemas que se generan en la simulación y diseño de procesos son de gran tamaño y no es posible resolverlos como un todo, siendo necesario descomponerlos en subsistemas más pequeños y manejables, formando bloques, cada uno de los cuáles puede representar una operación unitaria o una serie de operaciones unitarias interconectadas, lo que da origen a las estructuras de bloques de la Figura B.2.

De manera general, las técnicas de reordenamiento se clasifican en dos categorías: técnicas a-priori (o de dos pasos) y técnicas locales (o de un paso).

Cuando las técnicas a-priori son usadas, la solución es encontrada en dos pasos. Primero, toda o parte (unicamente las filas o unicamente las columnas) de la etapa de reordenamiento es completada, después de lo cual, un pivote es seleccionado entre los elementos diferentes de cero de la matriz reordenada; segundo, se inicia la etapa de solución numérica.

En las técnicas locales, la solución es obtenida en un sólo paso, durante el cual se van alternando las fases de reordenamiento y de solución numérica, es decir, la matriz es evaluada después de cada paso de la eliminación y se elige un pivote que minimice el llenado de la matriz en el siguiente paso.

Es generalmente reconocido (Duff (1977)), que los métodos a-priori, por sí mismos, no son tan efectivos como los métodos locales para preservar la dispersión. Sin embargo, es conveniente tomar en cuenta que el minimizar el llenado de la matriz no es tan importante para el usuario como las necesidades de almacenamiento en memoria y la adaptabilidad del método de almacenamiento fuera del núcleo central de memoria. Además, al aplicar los métodos locales no es posible conocer por anticipado en dónde ocurrirá el llenado de la matriz, por lo tanto, puede

ser necesario emplear hasta  $3\Gamma$  posiciones de memoria únicamente en esta etapa, siendo  $\Gamma$  el número de elementos diferentes de cero de la matriz; en forma adicional, puesto que después de cada paso la matriz resultante es nuevamente revisada para elegir un pivote adecuado, no es conveniente mantener algunas secciones de la matriz fuera del núcleo central de memoria.

Al aplicar los métodos a-priori es posible predecir la localización del llenado de la matriz y se requieren  $2\Gamma$ , o aún, hasta  $\Gamma$  posiciones de memoria, y puesto que el orden de pivoteo es conocido en forma anticipada a la etapa numérica, las filas o columnas de la matriz pueden ser mantenidas fuera del núcleo central de memoria y ser llamadas conforme se vayan necesitando.

Stadtherr y Wood (1982) y Stadtherr (1982), muestran que las técnicas a-priori son más adecuadas para la solución de los sistemas de ecuaciones que se generan en la simulación y diseño de procesos, en forma especial cuando el núcleo central de memoria de la computadora es limitado.

A continuación se presenta una breve introducción a la terminología propia de la etapa de reordenamiento.

### PARTICION

Tal y como es señalado por Lin y Mah (1977), por partición

se entiende el reordenamiento de la matriz de ocurrencia de modo que la estructura resultante sea de la Forma Triangular en Bloques (FTB), siendo cada bloque mínimo en dimensión, es decir, un bloque irreducible.

La partición se realiza en dos etapas. Primero, la matriz es reordenada de modo que se obtenga una diagonal libre de ceros, o en su defecto, obtener la máxima transversal posible. Este procedimiento es conocido como Selección Transversal y el algoritmo más representativo es el implementado por Duff (1981 A, 1981 B), quién presenta además una reseña del desarrollo de esta etapa, así como una amplia bibliografía al respecto. Los requerimientos computacionales de este tipo de algoritmos varía entre  $O(n^2 r)$  y  $O(n^{1/2} r)$ . El segundo paso de la partición consiste en realizar una permutación simétrica que forme una matriz triangular en bloques. Para realizar este paso se han propuesto varios métodos, destacando los de Sargent y Westerberg (1964) y por Tarjan (1972). A continuación se presenta una breve descripción de estos dos métodos, en base a algunos conceptos de la Teoría de Gráficas.

### Teoría de Gráficas.

Para preservar la simetría de una matriz, al realizar su reordenamiento, es necesario realizar permutaciones simétricas, es decir, se deberán elegir elementos pivote (o bloques pivote) de

la diagonal. Los primeros intentos para realizar esta tarea fueron abordados en base a la teoría elemental de gráficas.

Una gráfica es un conjunto de  $n$  nodos o vértices y un conjunto de arcos o líneas, descritos como  $(i,j)$ , entendiéndose que un arco parte del nodo  $i$  y entra al nodo  $j$ . En este contexto, las gráficas no tienen arcos múltiples, es decir, ningún par de arcos distintos tiene los mismos nodos como puntos finales y no existen tampoco arcos que empiecen y terminen en el mismo nodo, por lo tanto, no existe un arco  $(i,i)$ . Realizar una permutación simétrica sobre una matriz  $A$ , corresponde a reetiquetar los nodos de su gráfica. La Figura B.4 muestra una matriz típica y su gráfica asociada.

Una ruta del nodo  $v_1$  al nodo  $v_2$  en la gráfica es una secuencia de arcos  $(v_1, v_2), (v_2, v_3), \dots, (v_{k-1}, v_k)$ , y es conocido como un ciclo si  $v_1 = v_k$ . Una subgráfica consiste de un subconjunto de nodos y todos los arcos, que son pares de nodos, pertenecientes a este subconjunto. Una subgráfica se dice que está fuertemente conectada si hay alguna ruta desde alguno de sus nodos a algún otro, y es un componente fuerte si está fuertemente conectada y no puede ser ampliada a otra subgráfica fuertemente conectada mediante la adición de nodos extras y sus arcos asociados. Claramente, cada nodo puede pertenecer a sólo un componente fuerte, el cuál puede consistir de un nodo simple, por lo tanto, los componentes fuertes definen una partición de la gráfica.

$$\begin{bmatrix} A_{11} & 0 & A_{13} & 0 \\ A_{21} & 0 & 0 & A_{24} \\ A_{31} & 0 & A_{33} & 0 \\ 0 & A_{42} & A_{43} & A_{44} \end{bmatrix}$$

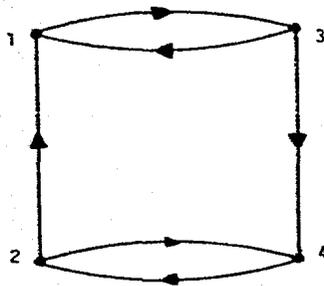


FIGURA B.4. MATRIZ DE OCURRENCIA Y SU GRAFICA ASOCIADA.

### Algoritmo de Sargent y Westerberg.

Este algoritmo hace uso del hecho de que todos los nodos, en algún ciclo de la gráfica, deben descansar en el mismo componente fuerte. El diagrama de flujo de este procedimiento se muestra en la Figura B.5.

El algoritmo traza rutas en la gráfica asociada con la matriz  $A$  y en sus subsecuentes modificaciones. Comienza en cualquier nodo y sigue una cierta ruta a través de la gráfica hasta que encuentra un ciclo, identificado al encontrar un nodo repetido dos veces, o hasta que encuentra un nodo sin arcos que partan de él. Todos los nodos en el ciclo deben pertenecer al mismo componente fuerte y cuando uno es encontrado, la gráfica es modificada al concentrar todos los nodos del ciclo en un sólo "nodo compuesto"; los arcos entre los nodos que constituyen el nodo compuesto son ignorados y los arcos que entran o salen de los nodos constituyentes desde o hacia otras partes de la gráfica fuera del ciclo, son conservados como arcos que entran o salen del nodo compuesto.

Si un nodo, o un nodo compuesto, es encontrado sin arcos que salgan de él, entonces éste debe corresponder a un componente fuerte de la gráfica original, no teniendo ningún arco hacia otro componente fuerte. Este, por lo tanto, corresponderá al primer bloque de la estructura triangular en bloques que se busca. El nodo compuesto y sus arcos asociados son separados de la grá-

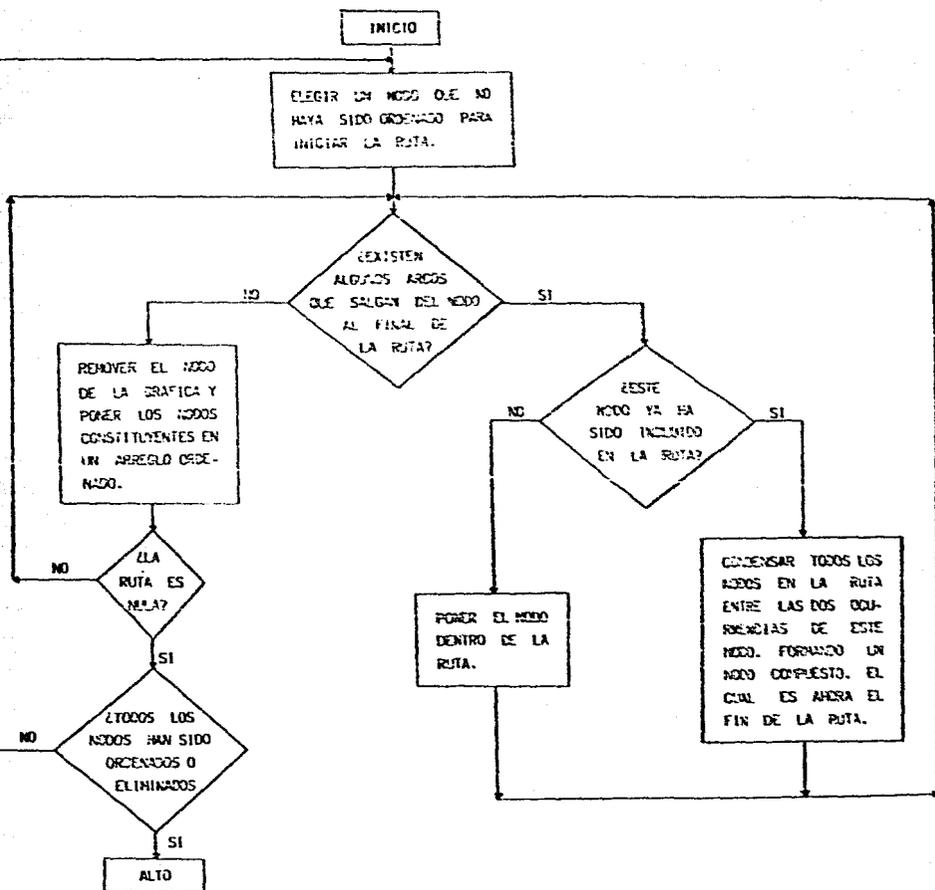


FIGURA B.5 DIAGRAMA DE FLUJO DEL ALGORITMO DE SARGENT - WESTERBERG.

fica original, produciendo una nueva gráfica correspondiente a la matriz  $A$  permutada, con su primera fila y columna de bloques removidas. El algoritmo continúa desde el último paso del resto de la ruta o comienza a partir de otro nodo, si la ruta ha sido terminada. En esta forma, los bloques de la matriz triangular son obtenidos sucesivamente.

Este método tiene la ventaja de que cada elemento diferente de cero fuera de la diagonal (correspondiente a un arco en la gráfica original), debe ser inspeccionado una sola vez; su desventaja reside en sus altos requerimientos de manejo de información. Las más eficientes implementaciones de este algoritmo requieren al menos  $O(n \log n) + O(\Gamma)$  operaciones.

### Algoritmo de Tarjan.

Es un algoritmo similar al de Sargent y Westerberg, pero con la ventaja de evitar la concentración de nodos, la cuál tiene altos requerimientos computacionales. En la Figura B.6 se presenta el diagrama de flujo de este algoritmo, de acuerdo a una implementación realizada por Duff y Reid (1978).

Nuevamente son seguidas las rutas a través de la gráfica.

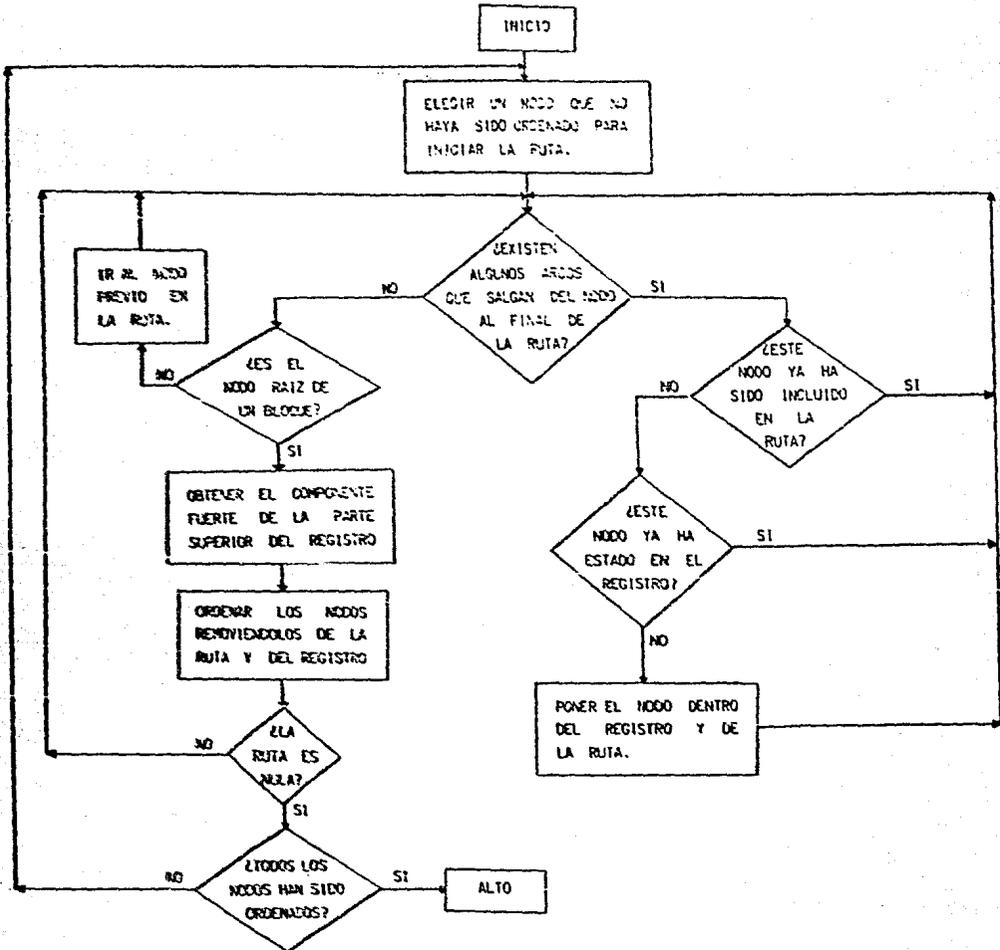


FIGURA B.6 DIAGRAMA DE FLUJO DEL ALGORITMO DE TARJAN.

pero ahora un conjunto de nodos es mantenido en un registro especial. Este registro contiene unicamente los nodos de la ruta que está siendo analizada, o aquellos que serán utilizados en una etapa de "marcha hacia atrás", empleando un señalador para localizar la posición de los nodos en el registro.

El algoritmo consiste de un cierto número (posiblemente uno) de pasos mayores, cada uno de los cuáles comprende una secuencia de pasos menores. Un paso mayor comienza colocando en el registro, y en la ruta, algún nodo que no haya estado en el registro en un paso mayor previo. Entonces se ejecuta una secuencia de pasos menores, cada uno de los cuáles extiende la ruta con otro nodo, o reduce su tamaño en un nodo mediante una marcha hacia atrás. El paso mayor termina cuando cuando la ruta y el registro han quedado vacíos. Un paso menor empieza con una búsqueda de los arcos empezando a partir de un nodo o hasta el final de la ruta, excluyendo aquellos buscados previamente.

Cómo en el algoritmo de Sargent y Westerberg, cada arco es examinado una sólo vez. Debe notarse que cada nodo es colocado en el registro una sólo vez y se ve involucrado en un sólo paso de la marcha hacia atrás, por lo tanto, el ciclo principal no puede ser ejecutado más de  $2n$  veces, reduciendo a  $O(n) + O(1)$  el número de operaciones que deben ser ejecutadas.

Duff (1981 A), probó el algoritmo de Sargent y Westerberg

comparándolo con el de Tarjan, encontrando que este último es superior al primero, en cuanto a reducir los requerimientos computacionales exhibiendo mayor velocidad de ejecución.

Duff (1981 B), presentó un programa de computadora que realiza una eficiente implementación del algoritmo de Tarjan.

Se han propuesto varias comparaciones entre las diferentes etapas en la solución de matrices dispersas. Duff (1977),

realizó una comparación en base al tiempo de cómputo de cada etapa. En la Tabla B.1, y sólo a manera de comparación, se presenta el tiempo, en milisegundos, en una computadora IBM 360/67, para cada etapa de la solución de matrices dispersas, incluyendo las etapas de reordenamiento y solución numérica, en base al promedio obtenido en diez corridas similares sobre matrices del mismo orden y densidad, de donde se puede concluir que la permutación simétrica hacia una estructura triangular en bloques no requiere tanto tiempo de cómputo, comparada con las demás etapas del proceso de solución.

De las anteriores consideraciones se desprende un hecho muy importante. Es ampliamente reconocido que al presentarse el problema de resolver matrices dispersas, se realice un análisis preeliminar para saber si es posible su transformación a una forma triangular en bloques. Existen varios procedimientos para realizar este análisis, siendo los más utilizados los recopilados por Gustavson (1976).

Orden	50	50	100	100	100	100
Elementos fuera de diagonal principal	100	200	300	200	300	500
Selección transversal	41	44	22	78	64	73
Permutación simétrica a una forma triangular en bloques.	15	22	30	32	39	52
Descomposición en L U	471	2099	5834	1459	4323	20938
Solución de ecuaciones	21	42	65	47	75	147

\* Tiempo en milisegundos.

TABLA B.1. COMPARACION ENTRE LAS DIFERENTES ETAPAS EN LA SOLUCION DE UN SISTEMA LINEAL DISPERSO, EN BASE A SU TIEMPO DE COMPUTO.

## SEPARACION

Desafortunadamente, en la práctica son pocos los sistemas que pueden ser descompuestos en forma clara en una matriz triangular en bloques. Como se indicó anteriormente, una de las ventajas principales de ese arreglo es que los bloques sobre la diagonal de la matriz pueden ser colocados en un cierto orden de precedencia, de modo que el sistema pueda ser resuelto paso a paso: sin embargo, para algunos sistemas existen algunas variables que evitan que esta solución progrese y forzan al sistema a ser resuelto en forma total y simultánea. Si estas variables fueran removidas, el sistema restante puede ser reordenado en una forma triangular en bloques.\* Las variables removidas son colocadas en las últimas columnas de la matriz, generando las diversas formas triangulares con margen, al agregar otros elementos, cuidadosamente seleccionados para conservar la simetría de la matriz, tal como se ilustra en las Figuras B.1 y B.2.

La selección de los elementos que deben ser permutados a las últimas columnas, es lo que se conoce como Separación ("tearing")

El objetivo central de los algoritmos de separación es elegir un número mínimo de tales columnas, con la previsión de que el subsistema resultante deberá tener un gran número de bloques pequeños. Adicionalmente, se pretende restringir el llenado de la matriz a ciertas áreas de los bordes.

En la literatura de programación matemática, las filas y columnas que se separan se conocen como puntales ("spikes"), mientras que en la literatura de Ingeniería Química son conocidas simplemente como filas y columnas separadas ("tears"). Sin embargo, existen equivalencias en los términos empleados en cada nomenclatura. Los elementos separados ("tears"), equivalen a los puntales ("spikes"); el conjunto de salida equivale al conjunto de pivotes; el orden de precedencia equivale al orden de pivoteo y las particiones equivalen a rompimientos ("bumps"). La terminología de separación ("tearing") es usualmente asociada con la solución iterativa de sistemas dispersos no lineales; mientras que la terminología de puntales ("spikes"), es usada en conexión con la solución directa de sistemas dispersos lineales, desde que fué propuesta por Hellerman y Zarick (1971, 1972), al desarrollar sus algoritmos de reordenamiento.

### Triangularizaciones

Estas son técnicas muy simples, usadas con el objeto de realizar una triangularización inicial de la matriz, como un paso previo para hacer más eficiente la aplicación de los algoritmos de separación. Normalmente se realizan triangularizaciones hacia adelante y triangularizaciones hacia atrás.

Por ejemplo, si hay una fila con un sólo elemento diferente de cero, podría ser el primer elemento diagonal en una matriz triangular inferior, por lo tanto, para triangularizar hacia adelante:

- 1.- Buscar una fila con un contador de fila de uno. Remover esa fila y la columna en la cuál se encuentra el elemento diferente de cero y ponerlo en la primera posición abierta en la matriz reordenada. Si todos los contadores de fila son mayores de uno, pasar al paso 3.
- 2.- Ajustar el contador de filas, tomando en cuenta la remoción de la columna en el paso 1. Continuar en paso 1
- 3.- Concluir la triangularización hacia adelante.

En forma similar, si hay una columna con un sólo elemento diferente de cero, podría ser el último elemento diagonal en una estructura triangular inferior. Para triangularizar hacia atrás:

- 1.- Buscar una columna que tenga contador de uno. Remover esa columna y la fila en la cuál se presenta el elemento diferente de cero y ponerla en las últimas posiciones abiertas en la matriz reordenada. Si todos los contadores de columna son mayores que uno, ir al paso 3.
- 2.- Ajustar el contador de columnas para la remoción de la fila en el paso 1. Regresar al paso 1.
- 3.- Concluir la triangularización hacia atrás.

### Matrices Triangulares con Bordes.

Al aplicar los algoritmos de separación, se generan varias formas de matrices triangulares con bordes.

La más simple de éstas es la Forma Triangular con Bordes (BoTF, por sus siglas en inglés):

$$A = \left[ \begin{array}{c|c} M & R \\ \hline S & T \end{array} \right] \quad (B.1)$$

en donde M es la matriz triangular del sistema, R y S son bordes diferentes de cero y T es una matriz cuadrada que puede ser cero o diferente de cero.

En la Figura B.7 A, se presenta una matriz de ocurrencia típica, y en la Figura B.7 B, se muestra esa misma matriz reordenada en una Forma Triangular con Bordes.

Westerberg y Berna (1978), sugieren una modificación, al permutar la matriz de coeficientes a la Forma Diagonal en Bloques con Bordes (BBDF, por sus siglas en inglés), con esta estructura:

$$A = \left[ \begin{array}{c|c} D & C \\ \hline V & W \end{array} \right] \quad (B.2)$$

en donde D es una matriz diagonal en bloques, con cada bloque correspondiente a una operación unitaria. Las columnas C y W representan las variables externas que describen los flujos entre las unidades. Las filas que constituyen V y W re-

presentan a las ecuaciones de conexión del proceso, más un grupo de ecuaciones que son elegidas para reducir el llenado de la matriz al resolver D. Este último conjunto de ecuaciones es seleccionado automáticamente por el algoritmo de pivoteo.

Stadtherr y Hilton (1982), proponen la Forma Triangular en Bloques con Bordes (BBTF, por sus siglas en inglés). Con referencia a la ecuación B.1, ahora M es una matriz triangular en bloques. Se pretende que cada bloque contenga información de más de una operación unitaria, de hecho, es deseable que cada operación unitaria pueda ser tratada como un sub-bloque; y los bordes R y T corresponden a las variables de diseño y a las variables separadas, respectivamente. Los elementos fuera de la diagonal de bloques en M indican flujos de información de una unidad a otra, dentro del sistema.

La forma BBTF se debe obtener minimizando el tamaño de los bordes, es decir, reduciendo el número de variables separadas. La forma triangular se puede ajustar mediante la elección adecuada de las variables de diseño. Si esto no fuera posible, se puede intentar conformar una estructura adecuada mediante el manejo de los bloques de la diagonal, haciendo que éstos representen más de una operación unitaria, creando bloques de unidades múltiples, con el objeto de reducir el tamaño de los bordes.

En la Figura B.7 C se representa la matriz de ocurrencia de la Figura B.7 A, permutada a una matriz triangular en bloques con bordes. Es importante hacer notar la diferencia en

el tamaño de los bordes de las matrices reordenadas en las Figuras B.7 B y B.7 C, recordando que el llenado potencial de la matriz está fuertemente restringido en esas áreas.

Stadtherr y Wood (1984), introdujeron la Forma Triangular con Bordes Implícita (i-BoTF, por sus siglas en inglés).

En realidad esta es una extensión de la forma triangular en bloques con bordes. Al obtener el arreglo BoTF, se realiza un proceso similar dentro de cada bloque individual, a fin de reducirlo en la misma forma en que se redujo todo el sistema, dando como resultado la reordenación final de la matriz de ocurrencia en una forma de puntal, la cuál es mostrada en la Figura B.7 D, notándose que las columnas puntal (1,4,2), y las filas (5,10,7) que constituyen los bordes de la forma BoTF, Figura B.7 B, ya están incluidas implícitamente en la forma i-BoTF, Figura B.7 D.

La forma i-BTF es más adecuada para su manejo en computadora, puesto que al realizar la solución numérica el llenado de la matriz sólo puede ocurrir en las columnas puntal y sólo abajo del elemento diferente de cero más alto en esa columna; por lo tanto, el llenado potencial en un puntal depende de su longitud, es decir, la "distancia" desde el elemento diferente de cero más alto en el puntal, hasta la fila inferior del sistema.

	1	2	3	4	5	6	7	8	9	10
1	x	x	x							
2	x				x					x
3	x	x		x						
4				x		x			x	
5	x				x					
6		x	x	x	x		x	x		
7			x	x					x	x
8	x									x
9		x	x		x	x				
10	x			x	x	x	x			

(A)

	10	5	3	6	7	9	8	1	4	2
8	x							x		
2	x	x						x		
1			x					x		
9		x	x	x						x
3	x				x			x	x	x
4				x		x			x	
6		x	x		x		x		x	
5								x		
10								x	x	
7									x	

(B)

FIGURA B.7. FORMAS DE REORDENAMIENTO DE UNA MATRIZ DE OCURRENCIA TIPICA.

	1	5	10	2	3	4	6	7	9	8
2	x	x	x							
5	x	x								
8	x		x							
1	x			x	x					
3	x		x	x		x		x		
4							x	x		x
7			x			x	x			x
9				x	x			x		
10	x	x					x	x	x	x
6		x		x	x	x		x		x

(C)

	10	5	1	3	6	7	4	9	2	8
8	x		x							
2	x	x	x							
5		x	x							
1			x	x						x
9		x		x	x					x
3	x		x			x	x			x
10		x	x		x	x	x			
4					x		x	x		
7	x			x			x	x		
6		x		x		x	x		x	x

(D)

FIGURA B.7. FORMAS DE REORDENAMIENTO DE UNA MATRIZ DE OCURRENCIA TÍPICA (CONTINUACION).

## ALGORITMOS DE REORDENAMIENTO A-PRIORI.

Una vez realizada una introducción a la terminología básica del reordenamiento; será posible entender mejor sus métodos.

Estas técnicas presentan una estructura general muy parecida. por lo tanto, se utilizará el Procedimiento de Partición con Pivote Preasignado (P4), propuesto por Hellerman y Rarick (1972), para describir el procedimiento general de esta metodología.

En la Figura B.8 A se muestra una matriz de ocurrencia típica (similar a la Figura B.7 A), y que será usada como ejemplo.

El primer paso consiste en reordenar esa matriz en una forma triangular inferior, hasta donde sea posible, mediante triangulación hacia adelante y hacia atrás, resultando la matriz mostrada en la Figura B.8 B, en donde las líneas separan la parte ordenada de la parte no ordenada.

A continuación, el algoritmo P4 separa la matriz en bloques irreducibles, creando la matriz mostrada en la Figura B.8 C y en donde las áreas delineadas indican los bloques irreducibles, adoptando una forma triangular en bloques con bordes (BBTF).

El siguiente paso es procesar cada bloque irreducible uno a uno. Esta fase del reordenamiento se conoce como selección de puntales. El objetivo es elegir una columna que una vez removida proporcione la mayor posibilidad de reordenar ese bloque

usando triangularizaciones hacia adelante. En el caso particular del algoritmo P4, se elige como puntal la columna que intersecta con la mayoría de las filas con un contador de fila mínimo. Considerando el bloque 3x3 de la Figura B.8 C, el contador de fila mínimo es 2. La columna 1, la cuál intersecta con dos filas de contador de fila 2, es elegida como puntal. Debe notarse que una vez que la columna 1 es removida, habrá dos filas con un contador de fila igual a uno, lo que da al algoritmo P4 dos oportunidades para completar el reordenamiento mediante triangularización hacia adelante. La columna 1 es removida temporalmente a la última posición en el bloque y será asignada a una fila que esté vacía, es decir, sin elementos diferentes de cero, al final del reordenamiento, situación que es mostrada en la Figura B.8 D.

Ahora, el algoritmo P4 realiza una triangularización hacia adelante dentro del primer bloque. Tanto la fila 5, como la fila 8, tienen un elemento simple; en forma arbitraria se elige la fila 8. La fila 8 y la columna 10 son entonces colocadas en la primera posición disponible en las filas y columnas de la matriz reordenada, produciéndose el arreglo mostrado en la Figura B.8 E. Olvidando por un momento la fila 8 y la columna 10, restan dos filas, cada una con un elemento simple. Se elige la fila 2 y la columna 5 para ser el siguiente pivote, operación que deja a la fila 5 sin elementos, por lo tanto, es asignada a la columna puntal, o sea, a la columna 1. Continuando

con el siguiente bloque, la columna 2 es elegida como un puntal, situación que se muestra en la Figura B.8 F. La triangulación hacia adelante continúa con la fila 1 y columna 3, y con la fila 9 y la columna 6. Nuevamente, todas las filas en la matriz restante tienen dos o más elementos y la columna 4 es elegida ahora como puntal, cómo se muestra en la Figura B.8 G.

Al aplicar la eliminación hacia adelante se intenta encontrar una fila tal que cuando su correspondiente columna sea removida, produzca más de una fila sin elementos. En este caso el algoritmo se aplica a todas las filas restantes. Por ejemplo el elegir la fila 3 y la columna 7 como el siguiente pivote, conduce a que la fila 10 no tenga elementos, por lo tanto, el puntal 4 puede ser asignado a la fila 10. El reordenamiento es concluido al seleccionar a la fila 4 y a la columna 9 como el próximo pivote, lo que conduce a dejar la fila 7 sin elementos, la que es asignada a la columna puntal 2. La matriz de la Figura B.8 H, representa el reordenamiento final.

Debe notarse que las filas puntales (5, 10, y 7) y las columnas puntales (1, 4 y 2), previamente elegidas, están contenidas implícitamente en la matriz de la Figura B.8 G, por tanto, esa matriz es de la Forma Triangular con Bordes Implícita (i-BoTF).

A continuación se describen brevemente algunos de los algoritmos de reordenamiento en dos pasos (a-priori) más usados.

	1	2	3	4	5	6	7	8	9	10
1	x	x	x							
2	x				x					x
3	x	x		x			x		x	
4				x		x			x	
5	x				x					
6			x	x	x	x		x	x	
7				x	x				x	x
8	x									x
9			x	x		x	x			
10	x			x	x	x	x			

(A)

	1	2	3	4	5	6	7	9	10	8
1	x	x	x							
2	x				x				x	
3	x	x		x			x		x	
4				x		x		x		
5	x				x					
7			x	x				x	x	
8	x								x	
9			x	x						
10	x			x	x	x	x			
6		x	x	x	x					x

(B)

FIGURA B.8. DESARROLLO DEL ALGORITMO P4 EN EL REORDENAMIENTO DE UNA MATRIZ DISPERSA.

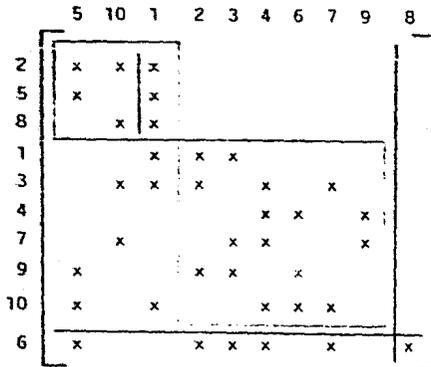
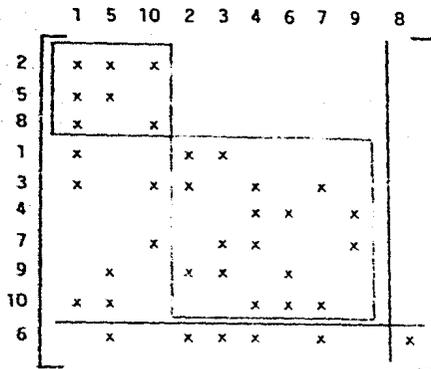


FIGURA B.8. DESARROLLO DEL ALGORITMO P4 EN EL REORDENAMIENTO DE UNA MATRIZ DISPERSA (CONTINUACION).

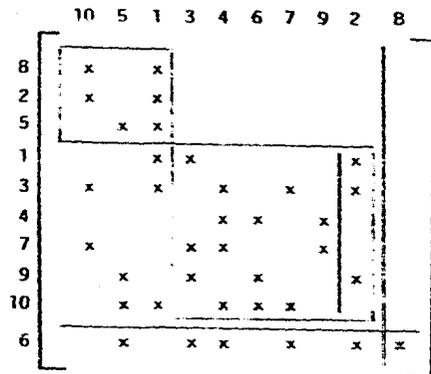
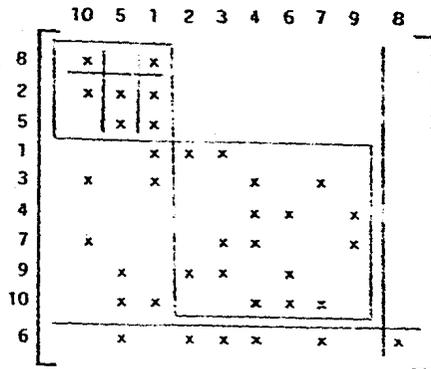
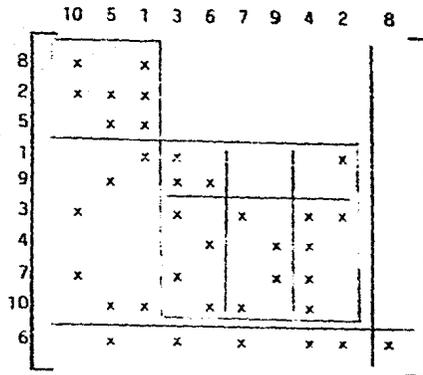
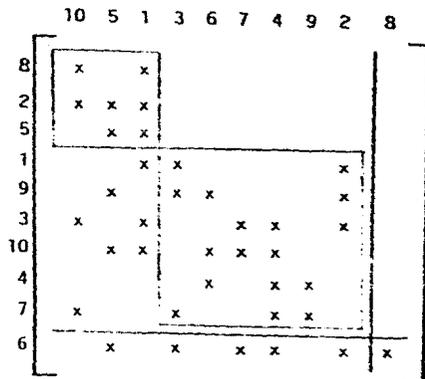


FIGURA B.8. DESARROLLO DEL ALGORITMO P4 EN EL REORDENAMIENTO DE UNA MATRIZ DISPERSA (CONTINUACION).



(G)



(H)

FIGURA B.8. DESARROLLO DEL ALGORITMO P4 EN EL REORDENAMIENTO DE UNA MATRIZ DISPERSA (CONTINUACION).

P4 Procedimiento de Partición con Pivote Preasignado.

Método propuesto por Hellerman y Rarick (1972).

- 1.- Triangularizar hacia adelante y hacia atrás.
- 2.- Realizar la partición de la matriz resultante y ordenar en bloques irreducibles.
- 3.- Procesar el primer bloque; si no hay más bloques ir a 9.
- 4.- Elegir una columna puntal y colocarla en su sitio.
  - Elegir como puntal una columna que interseque con el mayor número de filas de contador mínimo.
  - Si existe más de una de esas columnas, elegir aquella que interseque con el mayor número de filas del siguiente contador mínimo.
  - Otras coincidencias de este tipo deben ser eliminadas, eligiendo la columna con el contador más alto.
- 5.- Si no existe ninguna fila con contador de 1, ir a paso 4.
- 6.- Triangularizar hacia adelante. Si una fila sin elementos es encontrada, debe ser asignada a la última columna puntal. Si la fila es la última en ser asignada, ir al paso 8.
- 7.- Ir al paso 4.
- 8.- Iniciar el procesamiento de otro bloque; ir al paso 3.
- 9.- Fin del algoritmo.

HP    Partición Jerárquica.

Método propuesto por Lin y Mah (1977), junto con sus dos variantes más conocidas, HP 10 y HP 20.

- 1.- Triangularizar hacia adelante y hacia atrás.    Hacer partición de la matriz.
- 2.- Colocar los bloques irreducibles en la diagonal e iniciar el procesamiento del primer bloque.    Si no hay más bloques, ir al paso 10.
- 3.- Realizar la selección inicial de la columna puntal, tal y cómo fué realizada con el algoritmo P4.
- 4.- Triangularización hacia adelante.
- 5.- Realizar la selección inicial de la fila puntal; tomar en cuenta que esta selección es la transpuesta de la selección de la columna puntal.
- 6.- Triangularización hacia atrás.
- 7.- Realizar la partición del resto del bloque, calcular el índice P, que es la suma de los cuadrados del tamaño de los bloques, encontrados durante la partición.
- 8.- Buscar la combinación de filas y columnas puntales que produzca el menor valor de P, aplicando un criterio de exclusión para reducir el número de combinaciones examinadas; cada combinación examinada requiere partición

para evaluar su índice  $P$ .

9.- Reexaminar todas las columnas con el contador de columna mayor que el seleccionado en el paso 8; elegir como columna puntal aquella con el mayor contador de columna, pero que no produzca ningún cambio en el índice  $P$ ; ir al paso 2.

10.- Fin del algoritmo.

#### HP\_10

Es el mismo método HP, excepto en el paso 8, en dónde el número de combinaciones de filas y columnas puntales es reducido. Esto incrementa la velocidad de reordenamiento, pero el valor del índice  $P$  encontrado, no necesariamente será un mínimo absoluto.

#### HP\_20

Es el mismo algoritmo HP, excepto en que el paso 8 es omitido con el objeto de producir un aumento en la velocidad de reordenamiento, en comparación con el método HP 10.

HP 30

Esta es una modificación al algoritmo HP, propuesta por Chen (1981). En esta simplificación, los pasos 8 y 9 son omitidos, con el objeto de dar más velocidad al reordenamiento.

SPK 1

Algoritmo propuesto por Wood (1982), junto con los procedimientos SPK 2 y BLOCK.

- 1.- Triangularizar hacia adelante y hacia atrás.
- 2.- Realizar la partición de la matriz.
- 3.- Procesar el primer bloque; si no hay más bloques, ir a 8
- 4.- Seleccionar un puntal y colocarlo en su sitio.
  - El puntal es seleccionado de una fila con contador mínimo; en caso de existir más de una de estas filas, sumar los contadores de las columnas que intersectan con las filas de contador mínimo, y seleccionar la fila con el mayor valor de la suma.
  - La columna con el contador más pequeño en la fila seleccionada, es asignada a esa fila como el siguiente pivote; las columnas restantes en la fila son coloca-

das en orden decreciente, según su contador.

- 5.- Triangularización hacia adelante. Si una fila sin elementos diferentes de cero es encontrada, se asigna a la última columna puntal; si esa fila es la última del bloque, ir al paso 7.
- 6.- Ir al paso 4.
- 7.- Iniciar el procesamiento de otro bloque; ir al paso 3.
- 8.- Fin del algoritmo.

### SPK 2

Es un algoritmo similar al SPK 1, excepto en el paso 4, el cuál se modifica en esta forma:

- 4.- Seleccionar un puntal y colocarlo en su sitio.
  - Un puntal es seleccionado de una fila con contador mínimo; en caso de existir más de una de estas filas, elegir la fila que, trás suprimir las columnas que intersectan con ella, produce la mayor cantidad de filas de contador mínimo.
  - De estas filas, la duplicidad es rota sumando los contadores de las columnas que ellas intersectan, siendo la fila con la suma mayor, la elegida.

- Las columnas restantes son colocadas en orden decreciente de sus contadores.

### BLOCK

Este algoritmo fué diseñado para explotar la estructura de bloques inherente a las matrices empleadas en la simulación y diseño de procesos.

- 1.- Identificar la estructura de bloques del sistema.
- 2.- Aplicar el algoritmo de reordenamiento SPK 2 a la matriz de coeficientes.
- 3.- Retomar el bloque reordenado del paso 2 y considerar la matriz a un nivel de ecuación-variable.
- 4.- Triangularizar hacia adelante y hacia atrás.
- 5.- Aplicar el algoritmo de reordenamiento SPK 1 a cada bloque, uno por uno: triangularizar hacia adelante fuera del bloque cuando sea posible, es decir, sólo la selección de puntales, paso 4 del método SPK 1, es restringida al interior del bloque; la triangularización hacia adelante puede seleccionar pivotes de la parte exterior del bloque.

Wood (1982) y Stadtherr y Wood (1984), realizaron una comparación de varios de estos algoritmos, aplicados a matrices de varias dimensiones, tanto para ecuaciones de proceso, cómo para ecuaciones generadas aleatoriamente. Los resultados fueron presentados en excelentes tablas y gráficas, bajo diferentes criterios, tales como el número de operaciones realizadas, dimensión del llenado de la matriz, tiempo de solución, etc.

El algoritmo P4 puede ser considerado como una estrategia local, es decir, selecciona puntales y pivotes en base a consideraciones locales, en lugar de intentar identificar una estructura total. Los algoritmos del tipo HP son semilocales, en virtud de que la selección recursiva de filas y columnas puntales trata de identificar una posible estructura total. La estrategia semilocal es particularmente importante, puesto que hace un eficiente uso de la separación de cálculos en pequeños bloques y proporciona flexibilidad en el flujo de información. La desventaja de este tipo de algoritmos, tal y cómo es reconocida por sus autores, es su carácter práctico. HP 20 es una versión simplificada y más rápida de HP 10, la cuál, a su vez, es una versión simplificada y más rápida del algoritmo original HP. Sin embargo, los criterios para evaluar estos algoritmos en su relación efectividad-rapidez, aún no están bien establecidos.

El algoritmo SPK 1 es el más simple de todos, incluso es menos sofisticado que el P4, en cuanto a que no ofrece la posi-

bilidad de verificar el efecto potencial de la remoción de la columna puntal. SPK 2, en cambio, sí ofrece la posibilidad de analizar el efecto de remover todas las columnas, al menos dos, en una fila con contador mínimo, al seleccionar un puntal; SPK 2 encontrará, si es que existen, todos los sub-bloques  $2 \times 2$  al seleccionar una columna puntal.

El algoritmo HP verifica el efecto futuro en cada etapa de selección de puntales, dentro de cada bloque irreducible.

El algoritmo BLOCK es especial en cuanto a que presupone que la estructura de bloques es conocida previamente al reordenamiento de las ecuaciones.

#### Selección de pivotes para preservar la estabilidad.

Tradicionalmente el pivoteo ha sido seleccionado para mantener la estabilidad en los cálculos. Esto es posible lograrlo mediante alguno de estos procedimientos: seleccionando como pivote al elemento con el mayor valor absoluto en una fila o columna determinada, lo que se conoce como Pivoteo Parcial; o bien, eligiendo el mayor valor absoluto en la matriz activa, lo que se conoce como Pivoteo Completo. Sin embargo, al utilizar cua-

lesquiera de estos métodos, todas las ventajas de usar un reordenamiento podrían perderse y en forma adicional, al emplear un esquema de reordenamiento en forma estricta para reducir el llenado de la matriz, podría conducir a inestabilidades durante la etapa de solución numérica.

Este dilema es resuelto al emplear un método llamado Pivoteo Umbral. Al aplicar este método se elige como pivote algún elemento cuyo valor absoluto sea cierto número de veces mayor que una tolerancia de pivote, PTOL, en una determinada fila o columna. La tolerancia de pivote es una fracción tal que se cumpla  $0 \leq \text{PTOL} \leq 1$ . Al aplicar la técnica de Markowitz (1957), la cuál será descrita más adelante, se elige un pivote que minimice el producto  $(r_i^K - 1) \cdot (c_j^K - 1)$ , el cuál también cumple con el criterio de tolerancia umbral.

Al aplicar esta técnica a la forma puntal de la matriz, el pivoteo ocurre sólo si el elemento diagonal no cumple con la prueba umbral. Cuando esto sucede, la columna diagonal es intercambiada con la columna más cercana a la diagonal en la cuál, el elemento diferente de cero, cumple con la prueba.

ALGORITMOS DE REORDENAMIENTO LOCALES.Selección de Pivotes de Markowitz.

Esta es la técnica local de reordenamiento más utilizada y fué propuesta por Markowitz (1957).

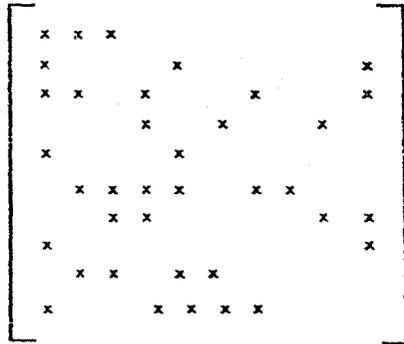
En el  $(K+1)$ -ésimo paso de la eliminación, el algoritmo de Markowitz selecciona un elemento diferente de cero  $a_{ij}^K$ , ( $i > K$ ,  $j > K$ ), que minimice el producto

$$(r_i^K - 1) \cdot (c_j^K - 1) \quad (B.3)$$

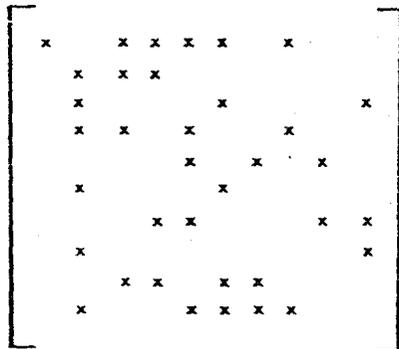
el cuál es la máxima cantidad de llenado que podría ocurrir durante el  $(K+1)$ -ésimo paso, si un elemento  $a_{ij}^K$  fuera elegido como pivote; siendo  $r_i^K$  y  $c_j^K$ , el número de elementos diferentes de cero restantes en la fila  $i$ , y en la columna  $j$ , respectivamente, al final del paso  $K$ , de la eliminación.

Nuevamente se utilizará como ejemplo la matriz mostrada en la Figura B.8 A, reproducida en la Figura B.9 A.

Después de examinar la matriz, se elige el elemento  $a_{68}^0$  cómo el pivote que cumple con el criterio. Aquí,  $r_6 = 6$ ,  $c_2 = 1$  y  $(6 - 1)(1 - 1) = 0$ ; por lo tanto, no se producirá ningún llenado de la matriz en este paso de la eliminación. Reorganizando la matriz para colocar el pivote  $a_{6,8}^0$  en la posición  $(1,1)$  y realizar el primer paso de la eliminación de Gauss, que origina



(A)



(B)

FIGURA B.9. DESARROLLO DE LA SELECCION DE PIVOTES POR EL METODO DE MARKOWITZ.



La forma para  $A$  mostrada en la Figura B.9 B. Considerando únicamente las filas y las columnas de la 2 a la 10 de la matriz de la Figura B.9 B, se elige al elemento  $a_{59}^{(1)}$  como el siguiente pivote. El segundo paso de la eliminación se muestra en la Figura B.9 C, en donde se puede observar la presencia de un elemento de llenado de la matriz, indicado por F. Finalmente, al continuar con esta técnica a través de  $n$  pasos, se origina la estructura de elementos cero/no cero, de la Figura B.9 D.

## B.II. ETAPA DE SOLUCION NUMERICA.

La forma usual de representar los sistemas de ecuaciones lineales es

$$A \cdot x = b \quad (B.4)$$

en donde,  $A$  es una matriz de coeficientes  $n \times n$ , siendo  $x$  y  $b$  vectores de columna de longitud  $n$ .

La solución,  $x$ , del sistema se representa así

$$x = A^{-1} \cdot b \quad (B.5)$$

siendo  $A^{-1}$  la forma explícita de la inversa de  $A$ .

La forma explícita de la inversa de  $A$ , usualmente es calculada mediante la eliminación de Gauss o por la eliminación de Gauss-Jordan. Estos métodos son eficientes en el caso de operar sobre matrices completas. Sin embargo, al aplicarse a la solución de matrices dispersas presentan el gran inconveniente de que la matriz  $A^{-1}$  que generan es completa, perdiéndose el carácter disperso de la matriz original  $A$ .

Para minimizar el llenado de la matriz, se emplean métodos alternativos para evaluar  $A^{-1}$ , conocidos como Formas No Explícitas de la Inversa. Existen dos formas ampliamente usadas para tal efecto: la Forma de Producto de la Inversa (FPI) y la Forma de Eliminación de la Inversa (FEI). La etapa de factorización en la FPI se basa en la eliminación de Gauss-Jordan, y en el caso de la FEI, lo hace en la eliminación de Gauss.

Brayton (1970), demostró que la FPI es teórica y experimentalmente inferior a la FEI, al tratar de mantener la dispersión de las matrices usualmente generadas en los problemas de simulación y diseño; por lo tanto, esta última es la de mayor aceptación actualmente, y será brevemente descrita a continuación.

#### Forma de Eliminación de la Inversa (FEI).

Esta técnica puede ser interpretada como una descomposición

de la matriz  $A$  en varios factores equivalentes.

Por razones de existencia, estabilidad y conservación de la dispersión se introducen las matrices de permutación  $P$  y  $Q$ , produciéndose esta factorización

$$P \cdot A \cdot Q = L \cdot U \quad (\text{B.6})$$

en donde,  $U$  es una matriz triangular superior y  $L$  es una matriz triangular inferior.

Usualmente, para mayor claridad, se omiten las matrices de permutación, por lo tanto

$$A = L \cdot U \quad (\text{B.7})$$

Aplicando la ecuación B.7 a la ecuación B.4, se obtiene

$$L \cdot U \cdot x = b \quad (\text{B.8})$$

La solución de este sistema se realiza en dos etapas.

Primero, a partir de

$$L \cdot y = b \quad (\text{B.9})$$

se obtiene  $y$ , mediante sustituciones hacia adelante; en seguida

$$U \cdot x = y \quad (\text{B.10})$$

se resuelve para  $x$ , mediante sustituciones hacia atrás.

Hasta aquí, la determinación de  $L$  y  $U$  para una matriz  $A$  es equivalente a realizar la eliminación de Gauss. Ambos pa-

Los representan únicamente la solución de sistemas triangulares y, por lo tanto, la estructura dispersa de  $L$  y  $U$  no es alterada en este proceso.

Duff (1979), realizó una revisión de las tres variaciones más comunes de la Eliminación de Gauss (EG).

En la técnica EG estándar, se realiza una eliminación hacia adelante columna por columna, cada paso correspondiente a encontrar una columna de  $L$  y una fila de  $U$ .

La técnica EG operando en filas, corresponde a una eliminación hacia adelante fila por fila, cada etapa correspondiente a encontrar una fila de  $L$  y una fila de  $U$ .

La técnica EG-Crout es similar a la EG estándar en cuanto a que  $L$  es encontrada columna por columna, sin embargo, los cálculos son ordenados de modo que sea posible el uso de registros acumulativos de doble precisión.

Mientras que las tres variantes pueden ser usadas junto con las técnicas de reordenamiento a-priori (o de dos pasos), en las técnicas de reordenamiento locales (de un paso) únicamente se puede utilizar la técnica EG estándar.

Bajo la Forma de Eliminación de la Inversa, se pretende obtener  $2n$  matrices de factores, tales que

$$A^{-1} = [U_1 U_2 \dots U_{n-1} U_n] \cdot [L_n L_{n-1} \dots L_2 L_1] \quad (B.11)$$

en donde, todos los factores de  $U$  y  $L$ , son de orden  $n$ , por lo tanto

$$U^{-1} = U_1 U_2 \dots U_{n-1} U_n \quad (\text{B.12})$$

$$L^{-1} = L_n L_{n-1} \dots L_2 L_1 \quad (\text{B.13})$$

Aplicando ecuaciones B.11, B.12 y B.13 a la ecuación B.5, se obtiene

$$x = U^{-1} \cdot L^{-1} \cdot b \quad (\text{B.14})$$

siendo ésta la solución del sistema representado por la ecuación B.4. A manera de ejemplo se presenta esta matriz:

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \quad (\text{B.15})$$

El factor  $L_1$  es definido tal que al multiplicar a la ecuación B.4, sea equivalente a realizar el primer paso de la eliminación de Gauss, por lo tanto

$$L_1 = \begin{bmatrix} 1/a_{11} & 0 & 0 \\ -a_{21}/a_{11} & 1 & 0 \\ -a_{31}/a_{11} & 0 & 1 \end{bmatrix} \quad (\text{B.16})$$

y el producto  $L_1 A$  es

$$L_1 A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & a_{22} - a_{21} \frac{a_{12}}{a_{11}} & a_{23} - a_{21} \frac{a_{13}}{a_{11}} \\ 0 & a_{32} - a_{31} \frac{a_{12}}{a_{11}} & a_{33} - a_{31} \frac{a_{13}}{a_{11}} \end{bmatrix} \quad (\text{B.17})$$

Ahora,  $U_1$  es definida de modo que en el producto  $L_1 A U_1$ , la fila 1 viene a ser similar a la primera fila de la matriz identidad, es decir,

$$U_1 = \begin{bmatrix} 1 & -\frac{a_{12}}{a_{11}} & -\frac{a_{13}}{a_{11}} \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (\text{B.18})$$

Ahora es conveniente definir una matriz intermedia  $A_1$ , tal que

$$A_1 = L_1 A U_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & a_{22}^{(1)} & a_{23}^{(1)} \\ 0 & a_{32}^{(1)} & a_{33}^{(1)} \end{bmatrix} \quad (\text{B.19})$$

en donde, para  $(i > 1, j > 1)$  :

$$a_{ij}^{(1)} = a_{ij} - a_{i1} \frac{a_{1j}}{a_{11}} \quad (\text{B.20})$$

Continuando el procedimiento con  $A^{(2)}$ , resulta:

$$A^{(2)} = L_2 A^{(1)} U_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{a_{22}^{(1)}} & 0 \\ 0 & \frac{-a_{32}^{(1)}}{a_{22}^{(1)}} & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ a_{22}^{(1)} & a_{23}^{(1)} \\ a_{32}^{(1)} & a_{33}^{(1)} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & \frac{-a_{23}^{(1)}}{a_{22}^{(1)}} \\ 0 & 0 & 1 \end{bmatrix} \quad (\text{B.21})$$

$$A^{(2)} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & a_{33}^{(2)} \end{bmatrix} \quad (\text{B.22})$$

y, finalmente para  $A^{(3)}$

$$A^{(3)} = L_3 A^{(2)} U_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \frac{1}{a_{33}^{(2)}} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & a_{33}^{(2)} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

----- (B.23)

$$A^{(3)} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = I \quad (\text{B.24})$$

por lo tanto, se puede generalizar que:

$$[L_n L_{n-1} \dots L_2 L_1] A [U_1 U_2 \dots U_{n-1} U_n] = I \quad (\text{B.25})$$

Retomando las ecuaciones de B.11 a B.14, la solución puede ser representada como una serie de factores matriciales:

$$x = A^{-1} b = U_1 U_2 \dots U_{n-1} U_n L_n L_{n-1} \dots L_2 L_1 b \quad (\text{B.26})$$

Debe notarse que al reducir la matriz original a una matriz identidad se indica que existe una serie de factores matriciales para ser aplicada al vector  $b$  y encontrar la solución deseada.

El método anteriormente descrito comprende una variación simple de la eliminación de Gauss, destinada a mantener la dispersión de la matriz original. Sin embargo, existen otras formas de implementar la eliminación de Gauss, las cuáles tienen mucho en común con la forma aquí mostrada, variando principalmente en la forma de manejo de los arreglos operacionales y las formas de almacenamiento dentro y fuera del núcleo central de la memoria en computadora.

A continuación se describen brevemente algunos métodos de evaluación numérica de matrices dispersas, basadas en la eliminación de Gauss.

#### MA 28

Duff y Reid (1977), presentaron una rutina conocida como MA 28, la cuál calcula la Forma de Eliminación de la Inversa, bajo el criterio de Markowitz, para preservar la dispersión, y el pivoteo umbral para mantener la estabilidad numérica.

En su implementación completa, MA 28 puede reordenar la

matriz en una forma triangular en bloques y luego procesa cada bloque irreducible por separado. Esta es una técnica representativa de los métodos locales (o de un paso), en la solución total de matrices dispersas. Este algoritmo obtiene la solución numérica actuando sobre la parte activa de la matriz, es decir, sobre la parte del bloque irreducible que ya ha sido descompuesto. El almacenamiento requerido consiste de arreglos reales de longitud  $LICN + n$  y arreglos enteros de longitud  $LIRN + LICN + 13n$ .  $LICN$  es al menos igual al número de elementos diferentes de cero en la inversa;  $LIRN$  es al menos igual al número de elementos diferentes de cero en la matriz original o el máximo número de elementos diferentes de cero en la parte activa de la matriz, eligiendo el que sea mayor; y  $n$  es el orden de la matriz.

#### NSPIV

Sherman (1978), propone un procedimiento en el que emplea la eliminación de Gauss junto con el pivoteo parcial. Aquí el pivoteo parcial consiste en seleccionar como pivote al elemento, en la fila pivote, con el mayor valor absoluto. NSPIV está orientado a resolver un vector  $b$  simple y por lo tanto, una vez

obtenida la inversa de la matriz  $A$ , no la retiene en la memoria.

Sólo la parte de la matriz arriba de la diagonal necesita ser retenida durante los cálculos, es decir, se retiene únicamente el factor  $U$  de la descomposición en  $L$  y  $U$ . A pesar que Sherman proporciona una subrutina de reordenamiento a-priori, PREORD, sugiere que el usuario proporcione su propia matriz ya reordenada, de ser posible. Los requerimientos de almacenamiento consisten de arreglos reales de longitud  $3n + NZ + NU$ , y arreglos enteros de longitud  $6n + NZ + NU$ , siendo  $NZ$  el número de elementos diferentes de cero en la matriz original y  $NU$  es el número de elementos diferentes de cero en el factor  $U$  de la matriz.

### LU 1 PAS

Esta es una proposición de Chen (1981), como una modificación local de la rutina NSPIV, incluyendo pivoteo umbral. Esta modificación elige como pivote al elemento en la fila pivote que está más cercano a la diagonal cuyo valor absoluto excede al más alto valor absoluto en la fila en un número de veces igual a la tolerancia pivote. El requerimiento de almacenamiento en memoria de LU 1 PAS es el mismo que para NSPIV.

LU 1 OUT

Es una modificación a LU 1 PAS, propuesta por Chen (1981).

En esta implementación la matriz original es almacenada fuera del núcleo central de memoria, en un disco, y la procesa fila por fila, lo que ofrece mayor disponibilidad de memoria para el factor  $U$  de la matriz. Para proporcionar capacidad de almacenamiento adicional, el vector  $b$  también se conserva fuera del núcleo central de memoria y no es retenido durante la solución. En todo lo demás, las rutinas LU 1 PAS y LU 1 OUT, son iguales.

El almacenamiento requerido por esta implementación consiste de arreglos reales de longitud total de  $3n + NU$ , y arreglos enteros de longitud  $5n + NU$ .

CBS

Wood (1984), propuso esta rutina como una modificación al método de Stadtherr y Wood (1982), conocido como Sustitución Continua hacia Atrás. Emplea una forma modificada del producto de la inversa, aplicando el pivoteo umbral para retener la dispersión y la estabilidad numérica. CBS intenta resolver un vector  $b$  simple, sin retener la inversa. Esta implementación proporciona dos opciones, elegibles por el usuario.

La primera es  $MAXL$ , que es un número entero entre 0 y  $n$ , inclusive. Las columnas puntal de longitud menor o igual a  $MAXL$ , serán procesadas cuando su pivote correspondiente es encontrado; las columnas de longitud mayor de  $MAXL$  serán retenidas hasta el final de la sustitución hacia atrás. Cuando  $MAXL$  iguala a  $n$ , se emplea el algoritmo CBS por completo, y cuando  $MAXL$  es igual a cero, se emplea la factorización L U. Esta opción es incluida como un "control" para sacrificar espacio por tiempo. El algoritmo CBS minimiza almacenamiento, mientras que la descomposición L U minimiza el número de operaciones.

La segunda opción consiste en manejar un índice de tolerancia,  $CTOL$ . Si el valor absoluto del elemento diagonal es menor que el mayor valor absoluto en la fila, por un cierto número de veces  $CTOL$ , entonces el pivote es rechazado y un pivote nuevo es elegido, mediante un intercambio de columnas; sin embargo el nuevo pivote debe satisfacer la tolerancia de pivote,  $PTOL$ . El propósito de esta opción es la de seleccionar un valor pequeño de  $CTOL$  y un valor alto de  $PTOL$ , mediante un reducido número de intercambio de columnas. Fijar  $CTOL = PTOL$  es equivalente a elegir pivotes umbrales como en LU I PAS. El almacenamiento requerido por CBS consiste de arreglos reales de longitud  $2n + NZ + NV$ , y arreglos enteros de longitud  $6n + NZ$ .

CBS OUT

Esta es una modificación de la rutina CBS que lee un disco y procesa las filas de la matriz original una a una, lo que proporciona mayor capacidad de almacenamiento en el núcleo central de memoria para el factor  $U$  de la matriz; adicionalmente el vector  $b$  es leído de un disco y no es retenido durante la solución. En otros aspectos, CBS OUT es igual a CBS. El requerimiento de almacenaje en memoria es de arreglos reales de longitud total de  $2n + NV$  y arreglos enteros de longitud  $5n$ .

RANK 1

Esta es una implementación del algoritmo RANK ONE BTS, originalmente descrito por Stadtherr (1977). Emplea una sustitución hacia atrás implícita y requiere un reordenamiento de la matriz original en una forma de tipo puntal. El pivoteo umbral es aplicado en todas las filas, excepto en las filas puntal de la matriz original. Cuando el elemento diagonal no cumple con la prueba umbral, la fila y la columna que intersectan el elemento diagonal son designadas y tratadas como una fila o columna puntal. Dentro de las filas puntales, el elemento de columna con el mayor valor absoluto es elgido como pivote.

El almacenamiento requerido consiste de arreglos reales de

longitud  $3n + NZ + NU$ , y arreglos enteros de longitud total  $4n + NZ + NU$ .

Wood (1982), realizó una comparación de algunos programas de solución de matrices dispersas lineales, aplicados tanto a estructuras de matrices generadas aleatoriamente, como a matrices generadas en problemas reales, bajo diferentes criterios y bases de comparación.

Chen y Stadtherr (1984), realizaron una comparación entre diferentes tipos de esquemas de solución de ecuaciones no lineales, en base al análisis de diferentes métodos y estructuras durante la etapa lineal de solución, retomando algunas de las proposiciones de Wood.

En el desarrollo del programa MA 28, para resolver conjuntos de ecuaciones lineales, Duff y Reid (1979 B), hacen una serie de observaciones acerca de los requerimientos que deben cubrir las rutinas de computadora destinadas a la solución de sistemas dispersos, haciendo uso de técnicas eficientes al estructurar y manejar los datos. Algunos de los conceptos que desarrollaron han sido aplicados en la implementación de algunas de las rutinas descritas anteriormente.

Sin embargo, como es señalado por Motard (1985) y Hlavacek (1985) se debe tener en mente que los paquetes existentes para la solución de matrices dispersas, en un futuro no muy lejano, no se ajustarán adecuadamente a la nueva generación de supercomputadoras y al manejo vectorizado de la información, en virtud de que esos paquetes usan diferentes direccionamientos de los datos.

Saunders y Guest (1982), reportan el manejo de más de 140 millones de operaciones de punto flotante por segundo y varios millones de palabras de memoria interna en la solución de ecuaciones lineales completas y multiplicación de matrices, en una máquina CRAY-I.

Ante estas perspectivas, algunas técnicas numéricas que han sido consideradas poco eficientes en las máquinas actuales, podrían resultar con propiedades superiores en las nuevas máquinas, e incluso, abandonar los métodos de solución directa por métodos iterativos con manejo vectorizado de la información.

A P E N D I C E      C

MODELOS REDUCIDOS PARA  
OPERACIONES UNITARIAS.

## A P E N D I C E      C

### MODELOS REDUCIDOS PARA OPERACIONES UNITARIAS

La introducción de los modelos reducidos lineales y no lineales dentro de los métodos modulares simultáneos constituye un gran avance dentro de la simulación de procesos.

A pesar de que el desarrollo de estos modelos es reciente, a corto plazo se prevee un acelerado avance en su implementación, especialmente en algoritmos de tipo ciclo interno-ciclo externo cuya eficiencia está comprobada.

La creación de modelos reducidos está motivada por la necesidad de realizar cálculos sobre las unidades de proceso en una forma eficiente y económica en cuanto a comportamiento computacional, aunque con ello se sacrifique cierto grado de precisión, plenamente conocido de antemano. La adopción de este tipo de modelos no es resultado unicamente del afan en la reducción de recursos necesarios para su funcionamiento, sino que toma en cuenta varias consideraciones de tipo cualitativo, como son:

a) No es necesario poseer un conocimiento detallado de un proceso o de una unidad de él para beneficiarse con un modelo simple o reducido, consideración que se ha comprobado desde los

inicios de la orientación modular simultánea (Vela 1961).

b) Uno de los mayores valores de un modelo simple es mostrar explícitamente las implicaciones de ciertas suposiciones acerca del proceso, tomando en cuenta que siempre habrá algún punto en su desarrollo en que se deberán hacer suposiciones con respecto a su funcionamiento. Por lo tanto, al mostrar las implicaciones de una suposición acerca del proceso, el modelo verifica la consistencia de esa suposición.

c) Cuando se modela una unidad con gran detalle es común adicionar restricciones de "realidad" a la evaluación del proceso; por ejemplo, como se verá más adelante, un absorbedor puede ser modelado en forma muy simple, suponiendo la recuperación de cierta fracción de cada componente removido. En otra etapa de su desarrollo, el modelo del absorbedor puede incluir cálculos como si fuera un proceso de equilibrio entre sus etapas en el que la recuperación dependerá del flujo de líquido y vapor y de los valores ideales del coeficiente de distribución. Finalmente, se puede agregar al modelo un esquema termodinámico más riguroso para estimar los valores de los coeficientes de distribución: es así como en cada etapa de su desarrollo el modelo requiere que el funcionamiento supuesto del proceso sea consistente con creciente rigor al comportamiento real de un absorbedor.

d) El mayor beneficio de los modelos simples o reducidos es el de proporcionar consistencia en las evaluaciones y representar una eficaz herramienta de comunicación. Un modelo simple puede

puede brindar cualquier respuesta acerca de la unidad que representa mediante el adecuado ajuste y manejo de sus suposiciones, sin embargo, no se debe olvidar que tales suposiciones deberán ser explícitas y los resultados deberán ser consistentes con todas y cada una de ellas.

En este apéndice se describen algunos modelos reducidos lineales y no lineales, que representan a las operaciones unitarias más utilizadas en la simulación de procesos, clasificados como modelos para unidades simples y modelos para unidades complejas, en estos últimos proporcionando algunas orientaciones para un desarrollo futuro y haciendo una importante observación acerca de su aplicación.

Algunos de estos modelos fueron originalmente propuestos por Jirapongphan (1981) y posteriormente modificados por Treviño-Lozano (1985).

## C.1. MODELOS REDUCIDOS PARA UNIDADES SIMPLES

### C.1.1. Modelo Reducido para Operación de "Flash"

Este modelo fué descrito en el Capítulo 4, Sección 4.3.3, con el cual se ejemplificó la forma en que se elabora un modelo reducido a partir de un modelo riguroso,

### C.1.2. Modelo Reducido para la Entalpía de una Corriente.

Aquí se presentan tres modelos reducidos para el cálculo de la entalpía de una corriente.

El primer modelo se representa así:

$$H_F = (a + bT)_F \cdot \sum m_n \cdot f_n \quad (C.1)$$

el cual representa la entalpía específica como una función lineal de la temperatura, en donde  $a$  y  $b$  son los coeficientes del modelo reducido. El coeficiente  $b$  tiene un significado físico sobre la capacidad calorífica y puede ser calculado como un promedio ponderado de la capacidad calorífica en base másica de cada componente, en esta forma:

$$b = \frac{\sum x_n \cdot m_n \cdot C_{p_n}}{\sum x_n \cdot m_n} \quad (C.2)$$

en donde  $x_n$  es la fracción mol,  $m_n$  es el peso molecular y  $C_{p_n}$  es la capacidad calorífica molar para el componente  $n$ ; el coeficiente  $a$  es calculado en tal forma que la ecuación C.1 es satisfecha exactamente en la solución del problema base. Este modelo se ajusta satisfactoriamente para corrientes con una fase simple, ya sea de líquido puro o de vapor puro; sin embargo, cuando la corriente tiene una fase líquido-vapor, este modelo no representa con precisión los cambios de entalpía con respecto a la temperatura, en virtud del fuerte efecto de ésta sobre las condiciones de la fase, el cual puede originar cambios drásticos en los coeficientes  $a$  y  $b$ .

El segundo modelo para la entalpía de una corriente remedia las anteriores desventajas al incluir las condiciones de fase en una forma similar a como fué realizada en las ecuaciones descriptivas en la unidad de "flash" (identificación de fase); es así como el modelo para una corriente a dos fases es:

$$\begin{aligned}
 H_F &= (a_L + b_L T) \cdot \sum m_n \cdot l_n + (a_V + b_V T) \cdot \sum m_n \cdot v_n \\
 &= (a_V + b_V T) \sum m_n (f_n - v_n) + (a_V + b_V T) \sum m_n \cdot v_n
 \end{aligned}
 \quad \text{---- (C.3)}$$

en donde  $v_n$  y  $l_n$  son las porciones de  $f_n$  que son vapor y líquido, respectivamente. Ahora, las variables  $v_n$  de la ecuación C.3, pueden ser referidas a la fracción vapor a una presión y temperaturas dadas mediante las ecuaciones de "flasheo" 4.58, 4.59 y 4.50. Sin embargo, la ecuación C.3 también incluye el componente líquido y necesitamos modificarla usando el balance de masa por componente, generando:

$$v_n (1 - \beta + \alpha_n \cdot K_b) - \alpha_n \cdot K_b \cdot \beta \cdot f_n = 0 \quad (\text{C.4})$$

por lo tanto, las ecuaciones C.4, 4.59 y 4.50 constituyen la expresión para el modelo reducido para la entalpía de una corriente a dos fases:

$$\text{ecuación 4.59:} \quad \beta \cdot \sum f_n - \sum v_n = 0 \quad (\text{C.5})$$

$$\text{ecuación 4.50:} \quad \ln (K_b \cdot P) - A - B/T = 0 \quad (\text{C.6})$$

Este modelo agrega  $n+2$  ecuaciones y variables; además

reemplaza los 2 coeficientes reducidos originales por 6 nuevos coeficientes para cada unidad.

El tercer modelo reducido para evaluar la entalpía de una corriente es una variación de los anteriores al emplear funciones de ajuste, del tipo:

$$H = H^{id} + A + B(T - T_{ref}) \quad (C.7)$$

en donde,  $H$  es la entalpía molar de la corriente;  $H^{id}$  es la entalpía de gas ideal;  $T$  es la temperatura de la corriente y  $A$  y  $B$  son parámetros del modelo reducido obtenidos del cálculo riguroso de la entalpía a una temperatura de referencia  $T_{ref}$ .

La introducción del término de entalpía de gas ideal expresa una dependencia de la entalpía sobre la composición en los balances de energía, dentro del ciclo interno de cálculo.

### C.1.3. Modelo Reducido para un Mezclador.

Sólo existen dos tipos de ecuaciones importantes en este modelo reducido, las cuales, de acuerdo a la Figura C.1, son:

Balance de materia:

$$p_n - \sum f(j) = 0 \quad (C.8)$$

en donde  $j$  es el número de la corriente que entra a la unidad.

Balance de energía:

$$(a + bT)_P \cdot \sum m_n \cdot p_n - (a + bT)_{F_j} \cdot \sum (m_n \cdot f_n(j)) = 0 \quad (C.9)$$

en forma general:

$$H_P P - H_F(j) F(j) = 0 \quad (C.10)$$

Siempre se considera que el mezclador opera en forma adiabática y que su presión de salida es la presión más baja de las corrientes de entrada, aunque también puede ser una presión especificada por el usuario. Este modelo contiene  $n+1$  ecuaciones más una ecuación para especificar la presión a la salida del equipo, conteniendo  $(j+1)(n+1)$  variables y  $2(j+1)$  coeficientes reducidos (referentes a la entalpía). Debe notarse que si las entalpías son usadas como variables dentro del ciclo interno de cálculo, el modelo reducido es totalmente lineal.

#### C.1.4. Modelo Reducido para un Divisor de Corriente.

Suponiendo que existen  $j$  corrientes de entrada y  $k$  corrientes de salida en esta unidad, de acuerdo a la Figura C.2, se define la fracción dividida de una corriente  $j$  como  $S_j$ .

Balance de materia:

$$S_j \cdot f_n - \sum p_n(j) = 0 \quad (C.11)$$

además,

$$\sum S_j = 1 \quad (C.12)$$

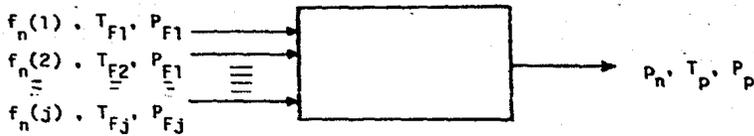


FIGURA C.1. FLUJO DE INFORMACION PARA EL MODELO REDUCIDO DE UN MEZCLADOR.



FIGURA C.2 FLUJO DE INFORMACION PARA EL MODELO REDUCIDO DE UN DIVISOR DE CORRIENTE.

las temperaturas y presiones a la salida son consideradas como iguales a las condiciones de entrada, mediante ecuaciones de la forma:

$$T_P(j) - T_F = 0 \quad (C.13)$$

$$P_P(j) - P_F = 0 \quad (C.14)$$

El número total de ecuaciones del modelo es de  $nk + 2k + 1$ , expresadas en  $3n + 3$  variables, sin ningún parámetro reducido, mientras que la fracción dividida,  $S_j$ , puede ser manejada como variable de decisión.

#### C.1.5. Modelo Reducido para un Calentador.

De acuerdo a la Figura C.3, esta unidad se describe así:

Balance de materia:

$$p_n - f_n = 0 \quad (C.15)$$

Balance de energía global:

$$(a + bT)_P \sum (m_n \cdot p_n) - (a + bT)_F \sum (m_n \cdot f_n) - Q = 0 \quad \text{--- (C.16)}$$

o en forma general:

$$H_P P - H_F F - Q = 0 \quad (C.17)$$

en donde  $Q$  es la carga calorífica agregada a la unidad, además, la presión de salida para esta unidad debe ser especificada por el usuario.

Este modelo contiene  $n+1$  ecuaciones más una ecuación para especificar la presión de salida, con  $2n + 3$  variables ( $P_n, f_n, T_p, P_p, Q$ ) y 4 coeficientes reducidos ( $(a,b)_F$  y  $(a,b)_P$ ). Existen tres posibles variables de decisión: la presión de salida,  $P_p$ , la temperatura de salida,  $T_p$ , y la carga de calor  $Q$ .

### C.1.6. Modelo Reducido para un Compresor.

La relación presión-temperatura en la compresión adiabática de un gas ideal, de acuerdo a la Figura C.4, se representa

$$T_p = T_F \left[ \frac{P_p}{P_F} \right]^{Y_k} \quad (C.18)$$

en donde,

$$Y_k = (k - 1) / k \quad (C.19)$$

$$k = C_p / C_v \quad (C.20)$$

siendo el cambio adiabático de entalpía:

$$H_{ad} = (1/Y_k) \cdot R \cdot T_F \cdot \left[ \frac{P_p}{P_F} \right]^{Y_k} - 1 \quad (C.21)$$

en donde  $R$  es la constante universal de los gases.

En una compresión politrópica, la relación presión-tempera-

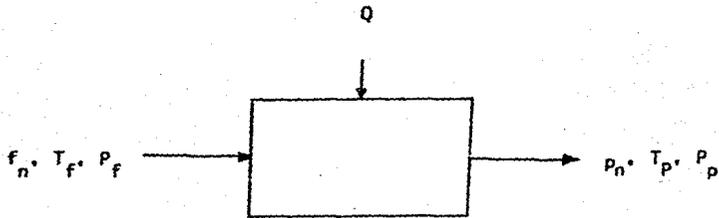


FIGURA C.3. FLUJO DE INFORMACION PARA EL MODELO REDUCIDO DE UN CALENTADOR.

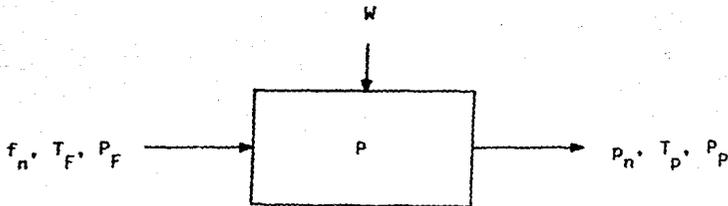


FIGURA C.4 FLUJO DE INFORMACION PARA EL MODELO REDUCIDO DE UN COMPRESOR.

tura y el cambio de entalpía para un gas no ideal puede ser descrita en una forma similar introduciendo un coeficiente politrópico,  $u > k$ , en lugar de  $k$ , por lo tanto:

$$\Delta H = (1/\gamma_u) \cdot Z \cdot R \cdot T_F \cdot \left[ \frac{P_P}{P_F} \right]^{\gamma_u} - 1 \quad (C.22)$$

$$\gamma_u = (u - 1) / u \quad (C.23)$$

$$T_P = T_F \cdot \left[ \frac{P_P}{P_F} \right]^{\gamma_u} \quad (C.24)$$

en donde  $Z$  es el factor de compresibilidad.

El requerimiento de potencia de la unidad de compresión con un 100% de eficiencia mecánica es:

$$h'_p = \Delta H \cdot \sum m_n f_n \quad (C.25)$$

La eficiencia politrópica y la eficiencia mecánica usada para caracterizar una unidad de compresión, se definen así:

$$e_p = \gamma_k / \gamma_u \quad (C.26)$$

$$e_m = h'_p / h_p \quad (C.27)$$

Estos dos factores de eficiencia deben ser proporcionados como un parámetro de entrada para la unidad.

El modelo reducido para el compresor se constituye así:

Balance de masa:

$$p_n - f_n = 0 \quad (\text{C.28})$$

Cambio de entalpía y requerimiento de potencia:

$$\Delta H = (Z \cdot R / \gamma_u) T_F \cdot \left[ \frac{P_P}{P_F} \gamma_u - 1 \right] \quad (\text{C.29})$$

$$h_p - (1/e_m) \Delta H \cdot \sum m_n \cdot f_n = 0 \quad (\text{C.30})$$

Balance de entalpía:

$$(a + bT)_p \sum m_n \cdot p_n - (a + bT)_f \sum m_n \cdot f_n - e_m (hp) = 0 \quad \text{---- (C.31)}$$

en donde  $a$  y  $b$  son los coeficientes del modelo reducido para el cálculo de entalpía, según la sección C.1.2.

En el modelo existen  $n+3$  ecuaciones conteniendo  $2n + 6$  variables ( $p_n, f_n, T_p, P_p, T_F, P_F, \Delta H, hp$ ), con 6 coeficientes reducidos ( $Z, \gamma_u, (a,b)_p, (a,b)_f$ ), en donde  $Z$  es el factor de compresibilidad calculado mediante bloques de cálculo rigurosos; mientras que  $\gamma_u$  es calculado según se indica en las ecuaciones C.26 y C.20 con  $C_p$  y  $C_v$ , también calculados mediante bloques de cálculo rigurosos.

## C.2. MODELOS REDUCIDOS PARA UNIDADES COMPLEJAS.

### C.2.1. Modelos Reducidos para Reactores.

El modelado de reactores para la simulación de procesos todavía está en su etapa de desarrollo. La mayoría de los simuladores existentes sólo ofrecen modelos de reactores que calculan el balance de materia en base a un rendimiento especificado o sobre una determinada conversión.

Los únicos simuladores comerciales que actualmente ofrecen los dos modelos estándar para reactores ideales - Reactor de Tanque Agitado Continuo (RTAC) y Reactor Tubular con Flujo Tapón (RTFT) - son ASPEN (1980) y ASPEN PLUS (1984), y aún así es pertinente recordar que los modelos de reactores ideales son sólo modelos simplificados referentes a los complejos fenómenos que tienen lugar en los reactores reales.

Es necesario reconocer dos tipos diferentes de reactores para los cuales se requieren modelos reducidos: primero, aquellos que contienen sólo ecuaciones algebraicas, tales como los modelos basados en rendimientos específicos y el modelo RTAC estándar; segundo, aquellos modelos que incluyen ecuaciones diferenciales, en donde el modelo de RTFT, es el más representativo.

Los modelos basados en ecuaciones algebraicas que actualmente existen implementados en los principales simuladores, sólo contienen ecuaciones de balance de materia y energía. Los mo-

delos reducidos para estos reactores también pueden consistir de ecuaciones de balance de materia (lineales para los modelos simples de extensión y rendimiento de reacción, no lineales para el RTAC), y ecuaciones del balance global de energía. Un ejemplo de este tipo de modelos es el descrito en C.2.1.1.

El caso de unidades descritas mediante sistemas de ecuaciones diferenciales, es completamente diferente del problema de encontrar ecuaciones algebraicas simples que aproximen el comportamiento de otras ecuaciones algebraicas más complejas. La solución de este problema puede ser la clave para extender el concepto modular simultáneo a otros problemas que contengan ecuaciones diferenciales ordinarias, tales como la simulación dinámica. En la sección C.2.1.2. se presenta un modelo reducido para un reactor con flujo tapón, originalmente propuesto por Treviño-Lozano (1985).

#### C.2.1.1. Modelo Reducido para un Reactor Estequiométrico.

Este es un modelo propuesto por Jirapongphan (1981), bajo estas limitaciones:

- a) Únicamente se verifica una reacción química.
- b) La fracción de conversión se limita a un sólo reactivo.
- c) Sólo existe una corriente de alimentación.
- d) Sólo existe una corriente de producto.

De acuerdo a la Figura C.5, se puede definir un índice de reactivo limitante,  $J$ , como:

$$J = \left[ \min_n \left\{ \frac{f_n}{S_n} \right\} \right] \cdot S_n < 0 \quad (C.32)$$

en donde,  $S_n$  es el coeficiente estequiométrico para el componente  $n$  en la reacción. También se define la relación entre los coeficientes estequiométricos para los componentes  $m$  y  $J$ :

$$B_n = S_n / S_J \quad (C.33)$$

entonces, las ecuaciones del modelo reducido son:

Balance de masa:

$$X - \epsilon \cdot f_J = 0 \quad (C.34)$$

$$P_m - B_m \cdot X - f_m = 0 \quad (C.35)$$

Balance de energía:

$$Q + H_T \cdot X - (a + bT) \sum_m P_m - (a + bT) \sum_n m_n \cdot f_n = 0 \quad (C.36)$$

o bien,

$$Q + H_T \cdot X - H_P \cdot P + H_F \cdot F = 0 \quad (C.37)$$

en donde,  $X$  es la extensión de reacción del reactivo limitante,  $\epsilon$  es la fracción del reactivo limitante siendo convertido,  $H_T$  es el calor de reacción involucrado por mol de reactivo limitante, y  $Q$  es el calor agregado al reactor.

En este modelo existen  $n+2$  ecuaciones expresadas en  $2n+6$  variables ( $p_n, f_n, T_p, T_F, X, \xi, Q, P$ ), con  $n+4$  coeficientes reducidos ( $\beta_n, (a,b)_p, (a,b)_F$ ). Los coeficientes estequiométricos,  $\beta_n$ , son calculados empleando la ecuación C.33.

Treviño-Lozano (1985) propone un modelo reducido en el cual se realizan  $M$  reacciones químicas:

Balance de masa:

$$p_n - f_n + \sum_{m=1}^M (f_J) \cdot (X_m) \cdot \left[ \frac{S_n}{S_J} \right] = 0 \quad (C.38)$$

en donde,  $f_J$  representa el flujo de entrada de un componente clave;  $S_n$  y  $S_J$  son los coeficientes estequiométricos para el componente  $n$  y para el componente clave  $J$ , respectivamente; y  $X_m$  representa la conversión del componente clave en la reacción  $m$ .

Balance de energía:

$$H_F \cdot F - H_P \cdot P + Q = 0 \quad (C.39)$$

siendo  $Q$ , en este caso, el calor neto agregado al reactor; además, la presión de la corriente de salida debe ser especificada por el usuario. El número total de ecuaciones, variables y coeficientes reducidos es el mismo que en el caso anterior.

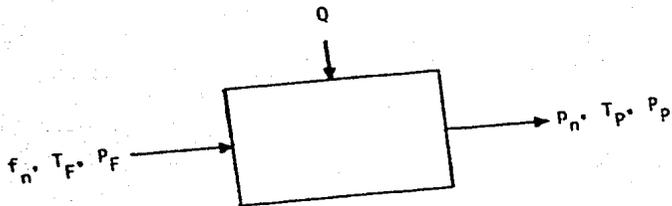


FIGURA C.5 FLUJO DE INFORMACION PARA EL MODELO REDUCIDO DE UN REACTOR ESTEQUIMETRICO.

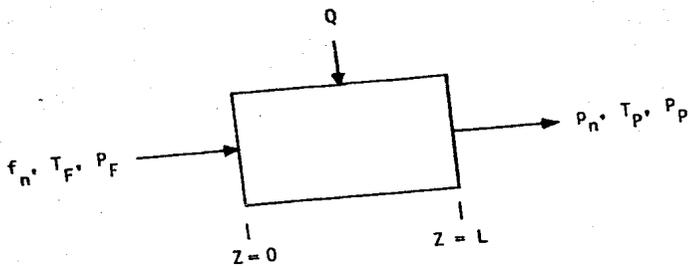


FIGURA C.6 FLUJO DE INFORMACION PARA EL MODELO REDUCIDO DE UN REACTOR DE FLUJO TAPON.

### C.2.1.2. Modelo Reducido para un Reactor con Flujo Tapón.

Dada la complejidad de este problema es necesario realizar una breve introducción, previa al planteamiento del modelo reducido.

De acuerdo a la Figura C.6, se consideran  $n$  componentes incluyendo reactivos y todos los productos posibles. Una corriente de alimentación entra al reactor a una presión  $P$  y una temperatura  $T$  (en algunos problemas pueden existir varias fases a diferentes temperaturas fluyendo a través del reactor; si éste es el caso, un balance diferencial de energía para cada fase se debe calcular para obtener los diferentes perfiles de temperatura). Se representa un vector  $\underline{F}$  que agrupa los flujos por componente a la entrada del reactor,  $f$ . Dentro del reactor tienen lugar una serie de reacciones y cierta cantidad de calor es agregada o removida con el objeto de controlar la temperatura en el reactor.

Los flujos por componente, y la temperatura y presión de las corrientes cambiarán dentro del reactor en una forma que sólo es descrita adecuadamente mediante un sistema de ecuaciones diferenciales no lineales:

$$-\frac{d \underline{F}}{d Z} = H_F (\underline{F}, T, P) \quad (\text{C.40 a})$$

$$\frac{dT}{d Z} = H_T (\underline{F}, T, P) \quad (\text{C.40 b})$$

$$-\frac{d}{dZ} \frac{P}{Z} = H_p (\underline{E}, T, P) \quad (C.40 c)$$

en donde  $Z$  indica la posición dentro del reactor con respecto a su entrada, por lo tanto, a la salida del reactor,  $Z = L$ . las variables de las corrientes tendrán valores  $\underline{E}_{sal}$ ,  $T_p$ ,  $P_p$ .

El propósito del modelo reducido del reactor es aproximar los valores de las variables de las corrientes de salida,  $\underline{E}_{sal}$ ,  $T_p$  y  $P_p$ , a partir de las condiciones de entrada conocidas,  $\underline{E}_{en}$ ,  $T_p$  y  $P_p$  y otros datos externos relevantes, tal como los coeficientes de transferencia de calor. Esta aproximación se debe realizar mediante un sistema de ecuaciones algebraicas que se adapten adecuadamente a las ecuaciones diferenciales originales.

Una forma de lograr esta adaptación es aproximar la composición completa junto con los perfiles de presión y temperatura del reactor. Esto se logra usando polinomios de orden  $m$ , en donde  $m$  es un número elegido por el usuario para un determinado problema, y que tienen la forma:

$$\underline{E} = \underline{A}_{0F} + (\underline{A}_{1F} + \lambda_F)Z + \underline{A}_{2F} Z^2 + \dots \quad (C.41 a)$$

$$T = \underline{A}_{0T} + (\underline{A}_{1T} + \lambda_T)Z + \underline{A}_{2T} Z^2 + \dots \quad (C.41 b)$$

$$P = \underline{A}_{0P} + (\underline{A}_{1P} + \lambda_P)Z + \underline{A}_{2P} Z^2 + \dots \quad (C.41 c)$$

Para el caso más general. el orden de los polinomios que

describen los perfiles de las diferentes variables puede no ser el mismo. En este caso, el número de coeficientes del polinomio cambia, sin embargo, la estructura global de las ecuaciones del modelo reducido permanece sin cambio.

La principal característica de este modelo reducido es que los coeficientes del polinomio,  $\underline{A}_0$ ,  $\underline{A}_1$ ,  $\underline{A}_2$ , etc., no son únicamente parámetros del modelo. Por el contrario, estos coeficientes son variables del modelo reducido que son determinadas a partir de ciertas condiciones iniciales. Los coeficientes  $\lambda$  son parámetros del modelo reducido que son determinados a partir de la solución rigurosa de las ecuaciones diferenciales para el punto base dado por las condiciones iniciales. La función de estos coeficientes será descrita más adelante.

Las ecuaciones necesarias para determinar los coeficientes del polinomio,  $\underline{A}_0$ ,  $\underline{A}_1$ , ... , son obtenidas mediante técnicas de colocación aplicadas a conjuntos linearizados de ecuaciones diferenciales. Dado un conjunto de  $m$  puntos de colocación (elegidos por el usuario),  $Z_1^*$ ,  $Z_2^*$ , ... ,  $Z_m^*$ , es posible definir un sistema de  $m(n + 2)$  ecuaciones algebraicas cuyo lado derecho corresponda a una expansión de primer orden en series de Taylor de la derivada de la función  $H_k$  en el punto de colocación  $Z_j$ , de la forma:

$$\begin{aligned}
H_k^* + \sum_{r=1}^n \frac{\partial H_k}{\partial F_r} \Big|_{Z_J^*} \cdot (A_{0F_r} + A_{1F_r} Z_J^* + \dots - F_r^*) + \\
+ \frac{\partial H_k}{\partial T} \Big|_{Z_J^*} \cdot (A_{0T} + A_{1T} Z_J^* + \dots - T^*) + \\
+ \frac{\partial H_k}{\partial P} \Big|_{Z_J^*} \cdot (A_{0P} + A_{1P} Z_J^* + \dots - P^*) = A_{1k} + 2A_{2k} Z_J^* + \dots
\end{aligned}
\tag{C.42}$$

para:  $k = 1, 2, \dots, n+2$  ( $n = \text{número de componentes}$ )  
 $j = 1, 2, \dots, m$  ( $m = \text{grado del polinomio}$ )

El valor de una variable cuyo perfil es descrito (en forma diferencial) por  $H_k$ , es aproximado mediante el polinomio correspondiente sin el término lineal de corrección  $\lambda_k Z$ . Los asteriscos son usados para indicar los términos evaluados en el punto de colocación  $Z_j$ .

En este sistema de ecuaciones las incógnitas son los coeficientes  $A_1, A_2, \dots, A_m$ . Los coeficientes independientes  $A_0$  son determinados a partir de las condiciones de entrada mediante estas ecuaciones:

$$A_0 = \begin{bmatrix} A_{0F} \\ A_{0T} \\ A_{0P} \end{bmatrix} = \begin{bmatrix} E_{en} \\ T_F \\ P_F \end{bmatrix} \tag{C.43}$$

Los parámetros del modelo reducido que deben ser determinados a partir de la solución rigurosa de las condiciones base a la entrada del reactor, son los elementos de la matriz Jacobiana de la derivada de las funciones  $H$  en cada punto de colocación.

La elección de los parámetros del modelo reducido puede ser justificada por estos argumentos:

a) Las derivadas parciales de primer orden de las funciones  $H$  toman en cuenta la adaptación del sistema original de ecuaciones diferenciales.

b) La forma de los perfiles de las variables dentro del reactor (determinada por los valores de las derivadas parciales a través del reactor) son, usualmente, menos dependientes de las condiciones de entrada que las mismas variables de las corrientes; por lo tanto, estos coeficientes constituyen un mayor conjunto de variables de iteración para los cálculos de convergencia.

c) Los métodos numéricos usados para integrar sistemas de ecuaciones diferenciales ordinarias en aplicaciones prácticas de ingeniería (GEAR, EPISODE, etc.) requieren el cálculo de derivadas en cada punto a lo largo de la integración, según lo destaca Carnahan y Wilkes (1980); por lo tanto, la evaluación de los coeficientes reducidos del modelo no involucra trabajo adicional, puesto que la matriz Jacobiana de las funciones diferenciales en cada punto de colocación generalmente es calculada en forma automática como parte del paquete de solución.

Es importante recordar que uno de los requerimientos de los

modelos reducidos usados en los cálculos modulares simultáneos, es que sean exactos en el punto base en donde fueron generados.

En el caso del reactor de flujo tapón, las ecuaciones de colocación no garantizan por sí mismas que la solución del modelo reducido coincida con la solución rigurosa en las condiciones base a la entrada del reactor, puesto que las soluciones dadas por los modelos rigurosos y los modelos reducidos solo se requiere que coincidan a la salida del reactor.

Con el objeto de hacer coincidir ambas soluciones, se agrega una función lineal de corrección a los polinomios, y los parámetros de corrección  $\lambda$  son calculados a partir de la diferencia entre las soluciones rigurosas y reducidas evaluadas a la salida del reactor para esas condiciones base iniciales, es decir,

$$\lambda = \begin{bmatrix} E \\ T \\ P \end{bmatrix} = \begin{bmatrix} F_{sal} \\ T_p \\ P_p \end{bmatrix} - \begin{bmatrix} A_{0F} + A_{1F} L + \dots \\ A_{0T} + A_{1T} L + \dots \\ A_{0P} + A_{1P} L + \dots \end{bmatrix}_{base} \quad (C.44)$$

Las funciones de corrección son agregadas a los polinomios de colocación para obtener los perfiles aproximados de las variables, descritos por las ecuaciones C.41.

Un desarrollo futuro en esta área podría ser la adaptación de este tipo de modelos de reactor dentro de procesos que contengan columnas de absorción continuas u otras unidades descritas mediante ecuaciones diferenciales ordinarias.

### C.2.2. Modelo Reducido para una Columna de Absorción.

Este modelo fué desarrollado para absorbedores adiabáticos de etapas múltiples. A pesar de que este modelo aún no ha sido integrado a un simulador modular simultáneo, fué probado bajo una amplia variedad de condiciones y los resultados publicados por Treviño-Lozano, Kisala y Boston (1984). Este modelo cumple con los requerimientos descritos en la sección 4.3.1.1., referidos a una unidad compleja real.

Considerando una columna de  $N$  etapas, tal y como se muestra en la Figura C.7, se obtiene:

Balance de masa global:

$$(l_{n,N+1} - l_{n,1}) - (v_{n,N} - v_{n,0}) = 0 \quad (C.45)$$

Balance de energía global:

$$L_{N+1} \cdot H_{N+1}^L + V_0 \cdot H_0^V - L_1 \cdot H_1^L - V_N \cdot H_N^V = Q \quad (C.46)$$

en donde,  $H_j$  se refiere al flujo de entalpía molar de la corriente  $j$ . En general, la entalpía de una corriente es una función de la temperatura, presión y composición.

Puesto que el efecto de la temperatura es mayor que el de otras variables y la capacidad calorífica es una función relativamente poco dependiente de la temperatura, la entalpía molar puede ser aproximada mediante esta expresión:

$$H = H^{id} + A + B(T - T_{ref}) \quad (C.47)$$

la cual representa la ecuación C.7, empleada para modelar la entalpía de una corriente, manteniendo la composición constante.

Los balances globales de materia y energía proporcionan relaciones que las corrientes de entrada y salida deben satisfacer; sin embargo, ellas no determinan la distribución de componentes en las corrientes de salida. La distribución es indicada mediante dos tipos adicionales de ecuaciones, empleadas en el modelo reducido: ecuaciones de equilibrio de fases para las etapas y ecuaciones abreviadas que proporcionan una descripción aproximada de la separación multicomponente en multietapas.

Para las relaciones de equilibrio, las volatilidades relativas se eligieron como parámetros del modelo reducido debido a su pequeña dependencia de la temperatura, presión y composición; por ejemplo, para un componente  $n$  en la etapa 1, la constante de equilibrio líquido-vapor (coeficiente de distribución),  $K_{n,1}$ , puede ser expresada en términos de la volatilidad relativa, así:

$$K_{n,1} = \alpha_{n,1} \cdot K_{b1} \quad (C.48)$$

en donde,  $\alpha_{n,1}$  es la volatilidad relativa del componente  $n$  y  $K_{b1}$  es una constante de equilibrio de referencia. En forma similar, para un componente en la etapa  $N$ :

$$K_{n,N} = \alpha_{n,N} \cdot K_{bN} \quad (C.49)$$

y las constantes de equilibrio de referencia,  $K_{b1}$  y  $K_{bN}$ , son determinadas a partir de la solución rigurosa del caso base como

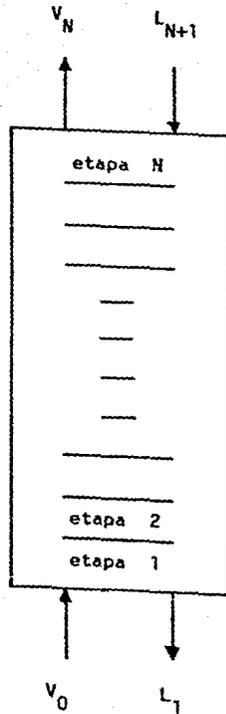


FIGURA C.7 FLUJO DE INFORMACION PARA EL MODELO REDUCIDO DE UNA COLUMNA DE ABSORCION.

un promedio ponderado de las constantes de equilibrio líquido vapor por componente, de acuerdo al desarrollo propuesto por Boston y Britt (1978); por lo tanto, para cada etapa:

$$\ln K_b = \sum_{i=1}^n w_i \cdot \ln K_i \quad (\text{C.50})$$

en donde,  $w_i$  son los factores de ponderación. Se pueden usar las fracciones mol de vapor,  $y_{n,N}$ , como factores de ponderación para  $K_{b,i}$  y las fracciones mol de líquido,  $x_{n,1}$ , para  $K_{b,1}$ .

El error introducido al asumir que las volatilidades relativas de todos los componentes en ambas etapas permanece constante cuando la temperatura, presión y composición han cambiado a partir de las condiciones base, típicamente es muy pequeño, en virtud de la poca dependencia de la volatilidad relativa en esas variables. Sin embargo, los valores de las constantes de equilibrio de referencia pueden depender fuertemente de la temperatura y de la presión, y en menor grado de la composición, por lo tanto, para tomar en cuenta esta dependencia se utilizan ecuaciones de constantes de equilibrio similares a aquellas empleadas en el modelo reducido para la operación de "flash", Capítulo 4, sección 4.3.1.3.:

$$\ln (K_b P) = a + b \left( -\frac{1}{T} - \frac{1}{T^*} \right) \quad (\text{C.51})$$

En el modelo reducido, tanto  $K_{b,1}$  como  $K_{b,N}$ , son representadas mediante ecuaciones de este tipo, eligiendo como temperaturas de referencia,  $T_1^*$  y  $T_N^*$ , las temperaturas del caso base, en-

fatizando el hecho de que estas ecuaciones calculan las constantes de equilibrio mediante desviaciones de las condiciones iniciales, por lo tanto:

$$\ln (K_{b1} \cdot P_1) = a_1 + b_1 \left( -\frac{1}{T_1} - -\frac{1}{T_1^*} \right) \quad (C.52)$$

$$\ln (K_{bN} \cdot P_N) = a_N + b_N \left( -\frac{1}{T_N} - -\frac{1}{T_N^*} \right) \quad (C.53)$$

Las constantes de equilibrio de referencia a las condiciones base,  $K_{b1}^*$  y  $K_{bN}^*$ , pueden ser evaluadas mediante la ecuación C.50 y los valores de las constantes de equilibrio y las fracciones mol para cada componente. Estas cantidades usualmente están disponibles como parte de la solución dada por el modelo riguroso del absorbedor, o pueden ser calculadas a partir de los perfiles de composición a lo largo de la columna. Puesto que las volatilidades relativas,  $\alpha_{n,1}$  y  $\alpha_{n,N}$ , se consideran constantes, sus valores se calculan en forma simple usando datos del caso simplificado, por ejemplo:

$$\alpha_{n,1} = \frac{K_{n,1}^*}{K_{b1}^*} \quad (C.54)$$

$$\alpha_{n,N} = \frac{K_{n,N}^*}{K_{bN}^*} \quad (C.55)$$

Los parámetros  $b_i$  y  $b_N$  de las ecuaciones C.52 y C.53 pueden obtenerse a partir de valores para las constantes  $K$  a dos temperaturas, manteniendo constante la composición.

Una vez que se han definido los parámetros del modelo simple necesarios para representar las constantes de equilibrio líquido-vapor en las etapas inferior y superior de la columna, ahora es posible elaborar una ecuación para cada etapa que describa el equilibrio físico entre las corrientes de líquido y vapor que abandonan cada etapa. El desarrollo de esa ecuación para la etapa superior se inicia al considerar esta identidad:

$$\sum_{i=1}^n x_{i,N} = \sum_{i=1}^n y_{i,N} = 1 \quad (\text{C.56})$$

en donde,  $x_{i,N}$  y  $y_{i,N}$  se refieren a la fracción mol del componente  $i$  en las corrientes de líquido y vapor que abandonan la etapa  $N$ , respectivamente. Usando el criterio de equilibrio líquido-vapor, la expresión anterior se convierte en:

$$\sum_{i=1}^n y_{i,N} \cdot \left[ \frac{1}{K_{i,N}} - 1.0 \right] = 0 \quad (\text{C.57})$$

Sustituyendo la ecuación C.49 en C.57 y representando las fracciones mol de la fase vapor en términos de flujos molares

$$\sum_{i=1}^n v_{i,N} \cdot \left[ \frac{1}{K_{bN} \alpha_{i,N}} - 1.0 \right] = 0 \quad (\text{C.58})$$

y mediante un desarrollo similar, para la etapa 1 se tiene:

$$\sum_{i=1}^n l_{i,1} (K_{b1} \cdot \alpha_{i,1} - 1.0) = 0 \quad (C.59)$$

por lo tanto, las ecuaciones C.58 y C.59, representan el equilibrio de fase combinado y las ecuaciones constitutivas.

En adición a las ecuaciones de equilibrio para las etapas inferior y superior, se hace necesario aproximar el comportamiento de la separación multicomponente en un sistema multietapa.

La ecuación de Kremser (1963), representa una solución analítica al problema, bajo ciertas condiciones. Esta ecuación describe la distribución de un componente relacionando las corrientes de salida, las corrientes de entrada, el número de etapas y las propiedades promedio de la mezcla en la columna.

Para un componente  $n$ , la ecuación de Kremser puede ser escrita como sigue:

$$(v_{n,0} + l_{n,N+1}) \cdot (1.0 - \bar{S}_n^N) + l_{n,N+1} \cdot (\bar{S}_n^N - \bar{S}_n) - l_{n,1} \cdot (1.0 - \bar{S}_n^{N+1}) = 0$$

----- (C.60)

en donde,  $\bar{S}_n$ , es el factor promedio de separación para el componente  $n$  en la columna, definido como

$$\bar{S}_n = \frac{K_n V}{L} \quad (C.61)$$

en donde,  $K_n$ ,  $V$  y  $L$  son la constante de equilibrio promedio,

flujo de vapor y flujo de líquido, respectivamente.

Debe notarse que los factores promedio de separación serán funciones altamente dependientes de la composición de la corriente de alimentación y de las condiciones de operación. Con el objeto de introducir parámetros de modelo que sean menos dependientes de estas variables, es posible seguir un desarrollo paralelo como en el caso de las constantes de equilibrio por componente y definir un factor de separación de referencia,  $S_n$ , y parámetros de modelo simples,  $\beta_n$ , que relacionen el factor de referencia con los factores de separación individuales para cada componente:

$$\bar{S}_n = \beta_n \cdot S_b \quad (C.62)$$

Los parámetros del modelo reducido,  $\beta_n$ , son calculados al resolver la ecuación C.61 para  $S_n$  para cada componente y calculando  $S_b$  a las condiciones base mediante una ecuación análoga a la ecuación C.50.

Es razonable asumir que para desviaciones en las condiciones de operación a partir del caso base, los factores de separación por componente cambiarán bastante, pero en forma tal que el cambio relativo de todos los componentes será mucho menor; por ejemplo, cambios en la presión y temperatura afectarán cualitativamente las constantes de equilibrio y los flujos de líquido y vapor en la misma forma, de modo que  $S_1$  variará para todos los componentes; además, se ocasiona una pequeña dependencia de  $\beta_1$  con respecto a las condiciones de operación.

Para tomar en cuenta las variaciones del factor de separación en el modelo, se introduce  $S_b$  como una variable a ser determinada y no como un parámetro del modelo reducido. Esto permite al modelo reducido responder apropiadamente a los cambios en temperatura, presión y condiciones en las corrientes de entrada; por lo tanto, al resolver el modelo reducido, los factores de separación efectivos,  $S_n$ , son calculados a partir de la ecuación C.62 usando un valor de  $S_b$  el cual es determinado en la solución del modelo reducido.

Anteriormente se destacó la importancia del análisis de grados de libertad en el desarrollo de un modelo reducido. Con el objeto de ilustrar este punto, en las Tablas C.1 y C.2, se muestra el análisis de grados de libertad para el modelo riguroso y para el modelo reducido de una columna de absorción, respectivamente. Lo primero que se debe notar es que el modelo reducido está formado por  $2n + 5$  ecuaciones, contra  $(3n + 3)n$  ecuaciones del modelo riguroso. Esta reducción en el número de ecuaciones es muy significativa, especialmente para columnas grandes y que manejan un gran número de componentes. Otra nota importante es que en ambos modelos existe el mismo número de grados de libertad, con la única diferencia de que el modelo reducido requiere sólo la presión de la primera y de la última etapa, mientras que el modelo riguroso necesita un perfil de presión completo de la columna. Finalmente, las variables de las corrientes de salida (flujos y temperaturas) pueden ser calculados con ambos modelos una vez que son especificadas las variables de las corrientes de entrada, número de etapas y perfil de presión.

1. Ecuaciones

Balances de materia por componente	$n \times N$
Ecuaciones de equilibrio	$n \times N$
Modelos de constante de equilibrio	$n \times N$
Balances de energía	$N$
Modelos de entalpía de vapor	$N$
Modelos de entalpía de líquido	$N$
	<hr/>
	$(3n + 3) N$

2. Variables

Flujos por componente líquido	$n \times N$
Flujos por componente vapor	$n \times N$
Constantes de equilibrio	$n \times N$
Variables en corrientes de entrada	$2n \times N$
Temperatura en cada etapa	$N$
Presión en cada etapa	$N$
Entalpías de líquido y vapor	$2N$
Número de etapas	$1$
	<hr/>
	$(3n + 4) N + 2n + 5$

3. Grados de libertad

$$2n + 3 + N$$

En el modelo riguroso se requieren  $N$  grados de libertad para especificar el perfil de presión.

TABLA C.1. ANALISIS DE GRADOS DE LIBERTAD DEL MODELO RIGUROSO DE UNA COLUMNA DE ABSORCION.

1. Ecuaciones

Balance de materia por componente	n
Balance de energía	1
Modelos para constante de equilibrio	2
Ecuaciones de equilibrio	2
Ecuaciones constitutivas	n
	<hr/>
	2n + 5

2. Variables

Flujos por componente ( $v_{n,0}; l_{n,1}; v_{n,N}; l_{n,N+1}$ )	4n
Temperaturas ( $T_0; T_1; T_N; T_{N+1}$ )	4
Presiones ( $P_1; P_N$ )	2
Número de etapas (i)	1
Otras: ( $K_{b1}; K_{bN}; S_D$ )	3
	<hr/>
	4n + 10

3. Grados de libertad

2n + 5

4. Parámetros de modelo reducido

3n + 12

( $\alpha_{n,1}; \alpha_{n,N}; \beta_1; a_1; b_1; a_N; b_N; A_1; B_1; A_0; B_0;$   
 $A_N; B_N; A_{N+1}; B_{N+1}$ )

TABLA C.2. ANALISIS DE GRADOS DE LIBERTAD DEL MODELO REDUCIDO DE UNA COLUMNA DE ABSORCION.

### C.2.3. Modelo Reducido para una Columna de Destilación.

El desarrollo de un eficiente modelo reducido no lineal para columnas de destilación es fundamental para la implementación práctica del concepto modular simultáneo. Este modelo debe ser lo suficientemente general como para describir columnas muy complejas; el mínimo requerimiento es que el modelo considere múltiples corrientes de alimentación y de producto, calentadores internos y configuraciones no estandarizadas, tales como derivaciones laterales. Adicionalmente, el modelo deberá prever la existencia de especificaciones de diseño sobre las variables de las corrientes internas.

Para cumplir estos objetivos, Treviño-Lozano propuso una aproximación en base a la reunión de varios bloques de operaciones unitarias, en especial los modelos reducidos para separadores instantáneos ("flash") y columnas de absorción.

Por ejemplo, una columna de destilación simple con una alimentación, un rehervidor y un condensador podría formarse por seis bloques: el plato de alimentación podría ser descrito con un modelo de mezclador y uno de operación de "flash"; las secciones de rectificación y separación se representan por modelos de absorbedor y dos modelos de operación "flash" son usados para describir al rehervidor y al condensador. La Figura C.3 ilustra gráficamente las diferentes secciones de un modelo de columna de destilación. Columnas más complejas pueden ser descritas adecuadamente, agregando más secciones (o sea, más mo-

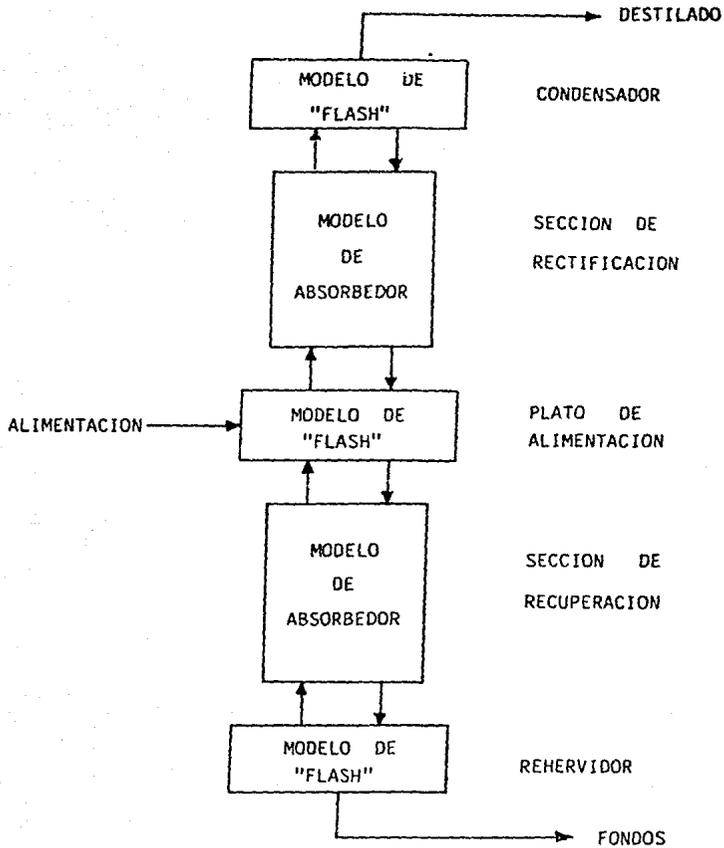


FIGURA C.8. FLUJO DE INFORMACION PARA EL MODELO REDUCIDO DE UNA COLUMNA DE DESTILACION.

de los reducidos de operaciones unitarias más simples), al esquema básico. De esta manera las ecuaciones de los modelos reducidos son manejadas de acuerdo al problema, evitando estructuras demasiado complicadas.

La idea de construir el modelo para una columna de destilación mediante la combinación de unidades más simples fué desarrollada y probada por Lee (1983). Sus resultados muestran que los modelos reducidos generados para una amplia variedad de configuraciones son funcionales y predicen en forma aceptable el comportamiento de la columna, aún en condiciones alejadas del caso base, y esto se cumplió aún para sistemas moderadamente no ideales. El único caso en donde esta aproximación fué poco satisfactoria fué para destilación azeotrópica, en donde se requieren modelos reducidos más complejos para representar esos sistemas altamente no ideales.

El concepto de ciclo interno-ciclo externo dentro de la simulación modular simultánea ha creado varios algoritmos para destilación bajo este enfoque. Estos modelos son, en general, más complicados que los modelos reducidos generados a partir de la adición de modelos más simples, en virtud de que los cálculos simplificados deben ser realizados etapa por etapa.

Boston y Sullivan (1974) y Boston (1980), presentan una detallada discusión sobre los algoritmos de destilación bajo un enfoque de ciclo interno-ciclo externo. La alta complejidad de algunos de estos modelos contrasta con el eficiente manejo que

hacen de los casos altamente no ideales.

Cho (1982), realizó un desarrollo de modelos reducidos, tanto dinámicos como en estado estacionario, para problemas de absorción y destilación en base a una aproximación polinomial; sin embargo, aún falta comprobar su efectividad en la solución de problemas reales complejos.

Si bien es cierto que los resultados publicados por Jirapongphan (1980), Pierucci (1982) y Treviño-Lozano (1985), indican que los modelos reducidos son claramente superiores a las aproximaciones modulares secuenciales convencionales y que pueden manejar problemas reales más eficientemente que los actuales simuladores orientados a las ecuaciones, se debe tener especial cuidado al trabajar con unidades complejas, especialmente con reactores y unidades de separación, puesto que de no existir formulaciones adecuadas pueden rendir resultados contraproducentes.

El afán de modelar unidades complejas con modelos reducidos puede conducir a la creación de modelos simples que sean demasiado limitados e inexactos y que pueden causar fallas y retrasos en la convergencia, no sólo de esas unidades, sino del proceso en general, tal y como fué señalado por Chmowitz (1983).

**NOMENCLATURA**

## NOMENCLATURA

### MAYUSCULAS

- A Matriz de coeficientes lineales. Parámetro del modelo reducido polinomial.
- A<sup>o</sup> Matriz constante y no singular.
- B Submatriz Jacobiana de especificaciones de diseño. Matriz aproximada.
- C Matriz Jacobiana aproximada de un sistema.
- C<sub>p</sub> Capacidad calorífica.
- D Aproximación diagonal de una matriz Jacobiana.
- E Vector de errores.
- F Flujo molar total como alimentación a un proceso o a un ciclo de éste.
- H Inversa de una matriz Jacobiana. Vector de restricciones de diseño. Flujo total de entalpía.
- I Matriz identidad.
- J Matriz Jacobiana. Índice de reactivo limitante.
- K Contador global de iteraciones. Coeficiente de distribución de fases.
- K<sub>b</sub> Coeficiente de distribución entre fases en el estado de referencia.
- L Flujo molar total del producto en fase líquida. Factor triangular inferior de una matriz. Función lineal.

- M Número total de reacciones químicas dentro de un reactor.
- N Parámetro de cambio del método híbrido Newton - Broyden - Schubert.
- P Presión. Flujo molar total de productos. Vector de parámetros de equipo fijos.
- Q Carga térmica. Conjunto de las funciones de residuos.
- R Constante universal de los gases. Vector de variables de retención.
- S Vector de las variables en las corrientes no cortadas.
- $S_n$  Coeficiente estequiométrico para el componente n en una reacción.
- T Temperatura. Vector de las variables en las corrientes de corte y de las variables libres. Ruta homotópica.
- U Conjunto de todas las variables de entrada y salida de cada unidad en el proceso. Factor triangular superior de una matriz.
- V Vector de variables internas aludidas por las ecuaciones de conexión de cierta unidad. Flujo molar total del producto en fase vapor.
- W Vector de parámetros de equipo libres (manipulados). Trabajo.
- X Vector de variables en las corrientes de entrada. Extensión de la reacción del reactivo limitante.
- Y Vector de variables en las corrientes de salida.
- Z Factor de compresibilidad. Posición dentro de un reactor tubular.
- $Z_1$  Vector que incluye variables internas, de retención y de salida.
- $Z_2$  Vector que incluye variables de entrada y parámetros de equipo fijos.

MINUSCULAS

- a      Coeficiente del modelo reducido.      Parámetro del método de Weyburn.
- $a_{ij}$     Elementos de una matriz de ciclo.
- b      Coeficiente del modelo lineal sobre los residuos de Jirapongphan.
- $b_{ki}$     Elementos de la matriz Jacobiana aproximada.
- $c_n$     Flujo molar para el componente  $n$  en las corrientes internas.
- e      Factor de perturbación de una variable.      Número de parámetros de diseño del modelo evolutivo.
- $e_p$     Eficiencia politrópica.
- $e_m$     Eficiencia mecánica.
- $f_n$     Flujo molar para el componente  $n$  en la corriente de alimentación.
- g      Dirección de avance en el método del Descenso Máximo.
- $h_{ij}$     Parámetros de discreción del método de Newton.
- hp    Requerimiento de potencia en un compresor.
- j      Número de corrientes de alimentación diferentes de cero.
- k      Relación  $C_p/C_v$
- $l_n$     Flujo molar del producto  $n$  en fase líquida.
- m      Contador de iteraciones del ciclo interno al aplicar el MAR.      Orden de un polinomio.
- $n_i$     Número de corrientes internas para el  $i$ -ésimo módulo unitario.
- $n_c$     Número total de componentes en un sistema.
- p      Parámetro de equipo fijo.      Factor de corrección en el método Levenberg-Marquardt.

$P_n$	Flujo molar total para el componente $n$ en alguna corriente de producto.
$P_K$	Vector que modifica a la $K$ -ésima iteración.
$q$	Función de residuos.      Parámetro del método de Wegstein.
$r$	Variable de retención.      Longitud de arco en la homotopía diferencial Cierta número de iteraciones sucesivas.
$s$	Coefficiente de control usado para forzar el avance de una función hacia un valor extremo.
$s_K$	Diferencia en el valor de una misma variable en dos iteraciones consecutivas.
$t$	Factor de amortiguamiento del método Quasi-Newton.      Factor de homotopía .
$t_K$	Escalar usado para prevenir la divergencia de la $K$ -ésima iteración.
$u$	Coefficiente politrópico de compresión.
$v_n$	Flujo molar del producto $n$ en fase vapor.
$w$	Parámetro de equipo libre (manipulado).      Factor de ponderación.
$x$	Variable de una corriente de entrada al proceso, ciclo o unidad.
$x_n$	Fracción mol del componente $n$ en fase líquida.
$y$	Variable de una corriente de salida del proceso, ciclo o unidad.
$y_n$	Fracción mol del componente $n$ en fase vapor.

INDICES

- ad Indica que una variable pertenece a un proceso adiabático.
- F Indica una propiedad de la corriente de alimentación a un proceso.
- h Indica la variable sobre la que se impone una restricción de diseño.
- i Elemento de fila de una matriz.  $i$ -ésimo módulo unitario.
- id Valor de una variable en condiciones ideales.
- j Elemento de columna de una matriz.  $j$ -ésima entrada a un módulo unitario.
- k Contador de iteraciones en el ciclo externo de cálculo.  $k$ -ésima salida de un módulo unitario.
- L Última iteración.
- m Contador de iteraciones en el ciclo interno de cálculo. Variable que pertenece a la  $m$ -ésima reacción química en un reactor.
- mm Elemento de una matriz situado en su diagonal principal.
- n  $n$ -ésima especie química en una corriente.
- P Indica una propiedad de la corriente de productos de un proceso.
- R Indica una propiedad de una corriente de recirculación.
- ref Indica el valor de una variable tomada como referencia.
- t Indica el valor de una variable tomado como prueba.
- T Transpuesta de una matriz.
- ( $\prime$ ) Indica el valor de una variable en la iteración inmediata anterior.
- ( $^0$ ) Indica valor inicial o solución conocida de una variable o función.
- \*
- Indica que el valor de esa variable es la solución buscada en un proceso iterativo.

GRIEGAS

- $\alpha$  Volatilidad relativa. Extremo inicial de la iteración por homotopía.
- $\alpha_K$  Coeficientes generalizados del MAR en cada iteración.
- $\beta$  Fracción de vapor. Extremo final de la iteración por homotopía.
- $\delta$  Parámetro de diseño en el modelo evolutivo.
- $\Delta$  Incremento aplicado a una variable. Radio de la región en la que se confina la linearización de variables.
- $\epsilon$  Fracción de reactivo limitante convertido. Parámetro de Moré-Cosnard.
- $\tau$  Número de elementos diferentes de cero en una matriz dispersa.
- $\theta$  Parámetro del método quasi-Newton de Soliman.
- $\lambda$  Parámetro reducido obtenido en el punto base. Parámetro del método de Levenberg-Marquardt.
- $\mu'$  Parámetros a minimizar en el método del Máximo Descenso.
- $\xi$  Tolerancia en la convergencia de un proceso iterativo.
- $\pi$  Conjunto de variables internas desconocidas del modelo evolutivo.
- $\sigma$  Secuencia en la solución de las operaciones unitarias.
- $T$  Tiempo de cómputo. Jacobiano modificado por la homotopía diferencial.
- $\phi$  Coeficiente de fugacidad.
- $\psi$  Factor de homotopía diferencial.
- $w$  Parámetro del modelo evolutivo que indica su secuencia de ejecución.
- $\nabla$  Gradiente de una función.

**BIBLIOGRAFIA**

## BIBLIOGRAFIA

- Abbot, J.P., "Algorithm 110: Computing Solution Arcs of Nonlinear Equations with a Parameter", Computer Journal, v.23, p.85 (1980).
- Allgower, E. y K.Georg, "Simplicial and Continuation Methods for Approximating Fixed Points and Solutions to Systems of Equations", SIAM Review, v.22, p.28 (1980).
- ASPEN Ver referencia de Gallier, P.W. y otros (1980).
- ASPEN PLUS - Introductory Manual, propiedad de ASPEN Technology Inc., Cambridge, Massachusetts (1984).
- Bañares-Alcántara, R., A.W. Westerberg y M.D. Rychener, "Development of an Expert System for Physical Property Predictions", Comp. Chem. Eng., v.9, p.127 (1985).
- Barkeley, R.W. y R.L. Motard, "Decomposition of Nets", Chem. Eng. Journal, v.3, p.265 (1972).
- Barnes, J.G.P., "An Algorithm for Solving Nonlinear Equations Based on the Secant Method", Computer Journal, v.8, p.65 (1965).

- Barreto, G.F. y I.H. Farina. "An Efficient Univariate Method to Solve Systems of Nonlinear Equations", Chem. Eng. Science v.33, p.63 (1979).
- Berna, T.J., "A Newton-Raphson Based Procedure for Optimization of Large Chemical Processes", Tesis de Doctorado, Carnegie Mellon University (1979).
- Berna, T.J., M.H. Locke y A.W. Westerberg. "A New Approach to Optimization of Chemical Processes", AIChE Journal, v.26, p.37 (1980).
- Biegler, L.T., "Simultaneous Modular Simulation and Optimization", Proceeding of the Second Int. Conference on Foundations of Computer Aided Process Design, CACHE Publications, (1985).
- Bogle, I.D.L., "The Numerical Solution of Flowsheeting Problems Described by Large Sparse Jacobian Matrices", Tesis de Doctorado, University of London (1983).
- Boston, J.F., "Inside-Out Algorithms for Multicomponent Separation Process Calculations", Computer Applications to Chemical Engineering, ACS Monograph, ed. Squires and Reklaitis, (1980).
- Boston, J.F. y H.I. Britt, "A Radical Formulation and Solution of the Single Stage Flash Problem", Computers and Chemical Engineering, v.41, p.109 (1978).

- Boston, J.F. y S.L. Sullivan, "A New Class of Solution Methods for Multicomponent, Multistage Separation Processes", Canadian Journal of Chemical Engineering, v.52, p.52 (1974).
- Broyden, C.G., "A Class of Methods for Solving Nonlinear Simultaneous Equations", Math. Comput., v.19, p.577 (1965).
- Broyden, C.G., "The Convergence of an Algorithm for Solving Sparse Nonlinear Systems", Math. Comput., v.25, p.285 (1971).
- Carnahan, B. y J.O. Wilkes, "Numerical Solution of Differential Equations - An Overview", Engineering Foundation Conference on Computer-Aided Chemical Process Design, Henniker New Hampshire (1980).
- Cavett, R.H., "Applications of Numerical Methods to the Convergence of Simulated Processes Involving Recycle Loops", American Petroleum Institute, n.04-63 (1963).
- Crowe, C.M., "On a Relationship Between Quasi-Newton and Dominant Eigenvalue Methods for the Numerical Solution of Nonlinear Equations", Computer and Chemical Engineering, v.8, p.35 (1984).
- Crowe, C.M. y M. Nishio, "Convergence Promotion in the Simulation of Chemical Processes - The General Dominant Eigenvalue Method", AIChE Journal, v.21, p.528 (1975).
- Chen, H.S., "Computational Strategies for Chemical Processes Flowsheeting and Optimization", Tesis de Doctorado, University of Illinois (1982).

- Chen, H.S. y M.A. Stadtherr, "A Modification of Powell's Dogleg Method for Solving Systems of Nonlinear Equations", Computers and Chem. Eng., v.5, p.143 (1981).
- Chen, H.S. y M.A. Stadtherr, "On Solving Large Sparse Nonlinear Equations Systems", Computers and Chem. Eng., v.8, p.1 (1984).
- Chen, H.S. y M.A. Stadtherr, "A Simultaneous-Modular Approach to Process Flowsheeting and Optimization - Part I.- Theory and Implementation", AIChE Journal, v.31, p.1843 (1985a)
- Chen, H.S. y M.A. Stadtherr, "A Simultaneous-Modular Approach to Process Flowsheeting and Optimization - Part II.- Performance on Simulation Problems", AIChE Journal, v.31, p.1857 (1985 b).
- Chen, H.S. y M.A. Stadtherr, "A Simultaneous-Modular Approach to Process Flowsheeting and Optimization - Part III.- Performance on Optimization Problems", AIChE Journal, v.31, p.1868, (1985 c).
- Chimowitz, E.M., S. Macchietto, T.F. Anderson y L.F. Stutzman "Local Models for Representing Phase Equilibria", I & E C Proc. Des. Dev., (1983).
- Cho, Young-Sang, "Reduced-Order Dynamic Models for Separation Processes", Tesis de Doctorado, Washington University (Sever Institute of Technology), (1982).

- Chow, S.N., J. Mallet-Paret y J.A. Yorke, "Finding Zeroes of Maps: Homotopy Methods That are Constructive with Probability One", *Mathematics of Computation*, v.32, p.887. (1978)
- Davidenko, D.F., "On a New Method of Numerical Solution of Systems of Nonlinear Equations", *Doklady Akad. Nauk. SSSR*, (NS) v.88, p.601 (1953).
- Davidon, W.C., "Variable Metric Method for Minimization", A.E.C. Research and Development Report, ANL-5990 (Rev. TID-4500, 14th ed.) (1959).
- Den Heijer, C. y W.C. Rheinboldt, "On Steplenght Algorithms for a Class of Continuation Methods", *SIAM Journal*, v.18, p.925 (1981).
- Dennis, J.E. y E.S. Marwil, "Direct Secant Update of Matrix Factorization", *Math. Comp.*, v.38, p.459. (1982).
- Dennis, J.E. y J.J. Moré, "Quasi-Newton Methods, Motivation and Theory", *SIAM Review*, v.19, p.46 (1977).
- Dichover, M.E., C.L. Mc Gregor y D.T. Ross, "Software Design Using SADT", *Proceedings of ACM National Conference*, Seattle, Washington, (1977).
- Duff, I.S., "A Survey of Sparse Matrix Research", *Proc. IEEE*, v.65, p.500 (1977).
- Duff, I.S. y J.K. Reid, "Algorithm 529.- Permutation to Block Triangular Form", *ACM Trans. on Math. Software*, v.4, p.189 (1978 a).

- Duff, I.S. y J.K. Reid, "An Implementation of Tarjan's Algorithm for the Block Triangularization of Matrix", ACM Trans. on Math. Software, v.4, p.137 (1978 b).
- Duff, I.S. y J.K. Reid, "Some Design Features of a Sparse Matrix Code", ACM Trans. on Math. Software, v.5, p.18 (1979).
- Evans, L.B., "Advances in Process Flowsheeting Systems", Foundations of Computer-Aided Chemical Process Design, Vol. 1, e. Engineering Foundation, Nueva York (1981).
- Evans, L.B., B. Joseph y W.D. Seider, "Systems Structures for Process Simulation", AIChE Journal, v.23, p.658 (1977).
- Ficken, F.A., "The Continuation Method for Functional Equations" Comm. Pure Appl. Math., v.4, p.435 (1951).
- FLOWTRAN User's Manual, Monsanto Co. (1974).
- Gaines, L.D. y J.L. Gaddy, "Process Optimization by Flowsheet Simulation", Ind. Eng. Chem. Process Des. Dev., v.15, p.206 (1976).
- Gallier, P.W., L.B. Evans, J.F. Boston, H.I. Britt y P.K. Guptz, "ASPEN: Advanced Capabilities for Modeling and Simulation of Industrial Processes", Computer Applications to Chemical Engineering, ACS Symposium Series, (1980).
- Gallum, S.E. y C.D. Holland, "A Modification of Broyden's Method for the Solution of Sparse Systems", Computers Chem. Eng., v.4, p.93 (1980).

- Genna, P.L. y R.L. Motard. "Optimal Decomposition of Process Networks". AICHE Journal, v.21, p.656 (1975).
- Gill, P.E. y W. Murray. "Two Methods for the Solution of Linearly Constrained and Unconstrained Optimization Problems". NA25, NPL Report (1972).
- Gorczyński, E.W. y H.P. Hutchinson. "Towards a Quasilinear Process Simulator: I.- Fundamental Ideas". Comp. Chem. Eng. v.2, p.189 (1978).
- Gorczyński, E.M., H.P. Hutchinson y A.R.M. Wajih. "Development of a Modularly Organized Equation-Oriented Process Simulator". Comp. Chem. Eng., v.3, p.353 (1979).
- Gordon, E. y K.D. Simpson. "Nonsequential Modular Flowsheeting". Discusión Invitada en Proceeding of the Second International Conference on Foundations of Computer Aided Process Design, CACHE Publications (1985).
- Grossman, I.E.. "Problems in the Optimum Design of Chemical Plants", Tesis de Doctorado, University of London (1977).
- Gustavson, F.G.. "Some Basic Techniques for Solving Sparse Systems of Linear Equations", Symposium on Sparse Matrices and Their Applications, e. Plenum Press, Nueva York (1972).
- Gustavson, F.G.. "Some Aspects of Computation with Sparse Matrices", Foundations of Computer-Aided Chemical Process Design Vol. 1, e. Engineering Foundation, Nueva York (1981).

- Hellerman, E. y D. Rarick, "The Partitioned Preassigned Pivot Procedure (P<sup>4</sup>)". en *Sparse Matrices and Their Applications* e. Plenum Press. Nueva York (1972).
- Hiebert, K.L., "An Evaluation of Mathematical Software That Solves Systems of Nonlinear Equations". *ACM Trans. on Math. Software*, v.8, p.5 (1982).
- Hilton, C.H., "Numerical Studies in Equation-Based Chemical Process Flowsheeting". Tesis de Doctorado, Massachusetts Institute of Technology MIT (1982).
- Hlavacek, V.H., "Analysis of a Complex Plant.— Steady State and Transient Behavior", *Comput. Chem. Eng.*, v.1, p.75 (1977)
- Hlavacek, V.H., "Nonsequential Modular Flowsheeting", *Discusión Invitada en Proceeding of the Second International Conference on Foundations of Computer Aided Process Design, CACHE Publications* (1985).
- Jain, Y.V.S., "Executive Programs for the Analysis and Steady State Simulation of Chemical Process", *University of Nebraska* (1974).
- Jirapongphan, S., "Simultaneous Modular Convergence Concept in Process Flowsheet Optimization", Tesis de Doctorado, Massachusetts Institute of Technology MIT (1980).
- Johnson, A.I., "GEMCS Manual and Application Studies", *Mc Master University, Ontario, Canada* (1960).

- Ketchum, R.G., "A Combined Relaxation-Newton Method As a New Global Approach to the Computation of Thermal Separation Processes", Chem. Eng. Science, v.34, p.387 (1979).
- Kilkas, A.C. y H.P. Hutchinson, "Process Optimization Using Linear Models", Comp. Chem. Eng., v.4, p.39 (1980).
- Kjaer, J., "Solution of Process Flowsheets by Formal Gauss Elimination", Comp. Chem. Eng., v.9, p.153 (1985).
- Kluzik, H.A., "A Study of the Simultaneous Modular Convergence of Chemical Process Flowsheets", Tesis de Maestría, Massachusetts Institute of Technology MIT (1979).
- Kubicek, M., "Algorithm 502.- Dependence of Solution of Nonlinear Systems on a Parameter", ACM Trans. on Math. Software, v.2, p.98 (1976).
- Kubicek, M., V. Hlavacek y F. Prochaska, "Global Modular Newton-Raphson Technique for Simulation of an Interconnected Plant", Chem. Eng. Science, v.31, p.277 (1976).
- Lam, B., "On the Convergence of a Quasi-Newton Method for Sparse Nonlinear Systems", Math. Comp., v.32, p.447 (1978).
- Lee, L.S., "Development of a Generalized Reduced Model of a Fractionation Column", Tesis de Maestría, Massachusetts Institute of Technology MIT (1983).

- Levy, A.V. y A.C. Segura, "Stabilisation of Newton's Method for the Solution of a System of Nonlinear Equations", Presentado en la Dundee Biennial Conference on Numerical Analysis (1979).
- Lin, T.D. y R.S.H. Mah, "Hierarchical Partition.- A New Optimal Pivoting Algorithm", Math. Programming, v.12, p.260 (1977).
- Locke, M.H. y A.W. Westerberg, "An Improved Successive Quadratic Programming Algorithm for Engineering Design Problems", Presentado en la AIChE National Meeting, (1982).
- Lucía, A., "A Note on the Broyden-Householder Update", Computers Chem. Eng., v.7, p.129 (1983).
- Lucía, A. y L.F. Stutzman, "On a New Method for Sparse Matrix Updating in Unconstrained Minimization", Computers Chem. Eng., v.5, p.57 (1981).
- Macchietto-Pinotto, S., "Solution of Equation Systems for Process Simulation", Tesis de Doctorado, University of Connecticut (1982).
- Mah, R.S.H., "Application of Graph Theory to Process Design and Analysis", Process Systems Engineering Symposium, Tokio, (1982).
- Mah, R.S.H. y T.D. Lin, "Comparison of Modified Newton's Methods", Computers Chem. Eng., v.4, p.75 (1980).

- Mahalec, V., H. Kluzik y L.B. Evans, "Simultaneous Modular Algorithm for Steady State Flowsheet Simulation and Design", 12th Eur. Sym. Computers in Chem. Eng., Montreux, Suiza, (1979).
- Markowitz, H.M., "The Elimination Form of the Inverse and Its Application to Linear Programming", Management Sci., v.3, p.255 (1957).
- Marquardt, D.M., "An Algorithm for Least Squares Elimination of Nonlinear Parameters", J. Soc. Ind. Appl. Math., v.2, p.431 (1963).
- Moré, J.J. y M.Y. Cosnard, "Numerical Solution of Nonlinear Equations", ACM Trans. Math. Software, v.5, p.65 (1978).
- Motard, R.L., "Computer Technology in Process Systems Engineering", Proceedings PSE'82, The Society of Chemical Engineers, Japón (1982).
- Motard, R.L., "Nonsequential Modular Flowsheeting", Discusión Sumaria en Proceeding of the Second Int. Conference on Foundations of Computer Aided Process Design, CACHE Publications (1985).
- Motard, R.L., M. Shacham y E.M. Rosen, "Steady State Chemical Process Simulation", AIChE J., v.21, p.417 (1975).
- Myers, G.J., Reliable Software Through Composite Design, e. Petroselli, Nueva York (1975).

- Nagiev, N.F., "Material Balance in Complex and Multistage Recycle Chemical Plants". Chem. Eng. Progr., v.53. p.297 (1975).
- Naphtali, L.M., "Process Heat and Material Balances". Chem. Eng. Progr., v.60. p.9 (1964).
- Nishida, N., G. Stephanopoulos y A.W. Westerberg, "A Review of Process Synthesis", Reporte No. DRC-06-17-80, Carnegie Mellon University, Pittsburgh (1980).
- Nishimura, H., Y. Hiraizumi y S. Yagi, "Kagaku Kogaku", v.31 p.183 (1967).
- Ortega, J.M. y W.C. Rheinboldt, "Iterative Solution of Nonlinear Equations in Several Variables", e. Academic Press, Nueva York (1970).
- Paloschi, J.R., "The Numerical Solution of Nonlinear Equations Representing Chemical Processes", Tesis de Doctorado, University of London (1982).
- Paloschi, J.R. y J.D. Perkins, "Robustness of Nonlinear Equation Solvers for Flowsheeting Packages", Presentado en IChemE Annual Research Meeting, Londres (1982).
- Perkins, J.D., "Efficient Solution of Design Problems Using a Sequential-Modular Flowsheeting Program", Comput. Chem. Eng., v.3, p.375 (1979).

- Perry, R.H. y C.H. Chilton, Chemical Engineer's Handbook, 5a. ed., e. Mc Graw-Hill Book Co. (1973).
- Peters, L.J. y L.L. Tripp, "Comparing Software Design Methodologies", Datamation, v.23, p.11 (1977).
- Pho, T.K. y L. Lapidus, "Topics in Computer Aided Design. Part I.- An Optimum Tearing Algorithm for Recycle Streams", AIChE Journal, v.19, p.1170 (1973).
- Pierucci, S.J., E.M. Renzi y G.E. Biardi, "Solution of Recycle Problems in a Sequential Modular Approach", AIChE J., v.28, p.820 (1982).
- Powell, M.J.D., "A Hybrid Method for Nonlinear Equations", Numerical Methods for Nonlinear Algebraic Equations, por P. Rabinowits, e. Gordon & Branch, Nueva York (1980).
- Proctor, S.I., "Simulation's Role and Value", Process Simulation Forum. Atlanta (1980).
- Rheinboldt, W.C. y J.V. Burkardt, "Algorithm 596: A Program for a Locally Parameterized Continuation Process", ACM Trans. on Math. Software, v.9, p.236 (1983).
- Rose, D.J. y R.A. Willoughby, Symposium on Sparse Matrices and Their Applications, e. Plenum Press, Nueva York (1972).
- Rosen, E.M., "A Machine Computation Method for Performing Material Balances", Chem. Eng. Progr., v.58, p.10 (1962).

- Rosen, E.M., "Steady State Chemical Process Simulation - State of the Art Review", Computer Applications to Chemical Engineering, ACS Symposium Series, v.124, p.3 (1980).
- Rosen, E.M. y A.C. Pauls, "Computer Aided Chemical Process Design: The FLOWTRAN System", Comp. Chem. Eng., v.1, p.11 (1977).
- Sandler, S.I., "Thermodynamic Models and Process Simulation", Foundations of Computer-Aided Chemical Process Design, V. 2 e. Engineering Foundation, Nueva York (1981).
- Sargent, R.W.H., "A Review of Methods for Solving Nonlinear Algebraic Equations", Foundations of Computer-Aided Chemical Process Design, Vol. 1, e. Engineering Foundation (1981).
- Sargent, R.W.H., "Process Synthesis Engineering: Challenges and Constraints in Computer Science and Technology", Proceeding to the Second International Conference on Foundations of Computer Aided Process Design (1983).
- Saunders, V.R. y M.F. Guest, "Applications of the CRAY-1 for Quantum Chemistry Calculations", en Computer Physics Communications.- Vector and Parallel Processors in Computational Science, Volumen Especial (1982).
- Schubert, L.K., "Modification of a Quasi-Newton Method for Non-linear Equations with a Sparse Jacobian", Math. Comput., v.25, p.27 (1970).
- Seader, J.D., "Computer Modeling of Chemical Processes", AIChE Monograph Series, v.81, p.15 (1985).

- Seader, J.D., W.D. Seider y A.C. Pauls, FLOWTRAN Simulation - An Introduction, CACHE Publications (1977).
- SGP/ZAR Simulador General de Procesos Zaragoza, Manual del Usuario, (1982).
- Shacham, M., "Equation Oriented Approach to Process Flowsheeting", Comp. Chem. Eng., v.5, p.79 (1981).
- Shacham, M., "Comparing Software for the Solution of Systems of Nonlinear Algebraic Equations Arising in Chemical Engineering", Comp. Chem. Eng., v.9, p.103 (1985 a).
- Shacham, M., "Recent Developments in Solution Techniques for Systems of Nonlinear Equations", Proceedings of the Second International Conference on Foundations of Computer Aided Process Design, CACHE Publications (1985 b).
- Shacham, M. y E. Kehat, "A Direct Method for the Solution of Large Sparse Systems of Linear Equations", Comp. Journal, v.19, p.353 (1975).
- Shacham, M. y R.S.H. Mah, "A Newton Type Linearization Method for the Solution of Nonlinear Equations", Comp. Chem. Eng., v.2, p.64 (1978).
- Sherman, A.H., "Algorithm 533: NSPIV, A Fortran Subroutine for Sparse Gaussian Elimination with Partial Pivoting", Trans. on Math. Software, v.4, p. 391 (1978 a).

- Sherman, A.H., "Algorithms for Sparse Gaussian Elimination with Partial Pivoting", *ACM Trans. Math. Software*, v.4, p.530 (1978 b).
- Shewchuk, C.F., "Nonlinear Algebraic Equations", en *Foundations of Computer-Aided Chemical Process Design*, Vol. 1, e. Engineering Foundation, Nueva York (1981).
- Soliman, M.A., "Quasi-Newton Methods for Convergence Acceleration of Cyclic Systems", *Can. J. Chem. Eng.*, v.57, p.643 (1979).
- Soliman, M.A., "A New Update for the Solution of Nonlinear Algebraic Equations", *Comp. Chem. Eng.*, v.9, p.407 (1985).
- Stadtherr, M.A., "Maintaining Sparsity in Process Design Calculations", *AIChE Journal*, v.25, p.609 (1979).
- Stadtherr, M.A. y H.S. Chen, "Strategies for Simultaneous-Modular Flowsheeting and Optimization", en *Proceeding of the Second Int. Conference on Foundations of Computer Aided Process Design*, CACHE Publications (1985).
- Stadtherr, M.A. y C.H. Hilton, "Development of a New Equation Based Process Flowsheeting System: Numerical Studies", *Selected Topics on Computer-Aided Process Design and Analysis*, AIChE Symposium Series, v. 78, p.12 (1982).
- Stadtherr, M.A. y C.H. Hilton, "On Efficient Solution of Large Scale Newton-Raphson-Based Flowsheeting Problems in Limited Core", *Comp. Chem. Eng.*, v.6, p.115 (1982).

- Stadtherr, M.A. y E.S. Wood, "Exploiting Bordered Structure Involving Large Sparse Linear Systems in Bordered Triangular Form", *Comp. Chem. Eng.*, v.4, p.191 (1980).
- Stadtherr, M.A. y E.S. Wood, "Sparse Matrix Methods for Equation-Based Chemical Process Flowsheeting. I.- Reordering Phase", *Comp. Chem. Eng.*, v.8, p.9 (1984 a).
- Stadtherr, M.A. y E.S. Wood, "Sparse Matrix Methods for Equation-Based Chemical Process Flowsheeting. II.- Numerical Phase", *Comp. Chem. Eng.*, v.8, p.9 (1984 b).
- Stephanopoulos, G., "Synthesis of Process Flowsheets: An Adventure in Heuristic Design or a Utopia of Mathematical Programming?", en *Foundations of Computer-Aided Chemical Processes Design*, Vol. 1, e. Engineering Foundation (1981).
- Steward, D.V., "On an Approach to Techniques for the Analysis of the Structure of Large Systems", *SIAM Review*, v.4, p.321 (1962).
- Steward, D.V., "Partitioning and Tearing Systems of Equations", *J. SIAM Numer. Anal.*, v.8, p.2 (1965).
- Takamatsu, T., "The Nature and Role of Process Systems Engineering", *Comp. Chem. Eng.*, v.7, p.203 (1983).
- Tarjan, R., "Depth-First Search in Linear Graph Algorithms", *SIAM J. Comput.*, v.1, p.146 (1972).

- Treviño-Lozano, R.A., "Simultaneous Modular Concept in Chemical Process Simulation and Optimization", Tesis de Doctorado, Massachusetts Institute of Technology MIT, (1985).
- Treviño-Lozano, R.A., T.P. Kisała y J.F. Boston, "A Simplified Absorber Model for Nonlinear Simultaneous Modular Flow-sheet Calculations", *Comp. Chem. Eng.*, v.8, p.10 (1984).
- Umeda, T. y M. Nishio, "Comparision Between Sequential and Simultaneous Approaches in Process Simulation", *Ind. Eng. Chem. Process Des. Dev.*, v.11, p.153 (1972).
- Upadhye, R.S. y E.A. Grens, "An Efficient Algorithm for Optimum Decomposition of Recycle Systems", *AIChE Journal*, v.18 p.533 (1972).
- Upadhye, R.S. y E.A. Grens, "Selection of Decompositions for Chemical Process Simulation", *AIChE Journal*, v.21, p.136 (1975).
- Vasek, V., M. Dohmal, J. Klemes y J. Lutcha, "Methods of Chemical Plant Balance Simulation Problem Solving", *Proceedings of Int. Symp. on Computer Applications in Process Development*, Erlangen, Dinamarca (1974).
- Vela, M.A., "Use Split Fractions for Recycle Balances, Parts I and II", *Petrol. Refiner*, v.40, p.5 (1961).
- Vickery, D.J. y R. Taylor, "Path-Following Approaches to the Solution of Multicomponent, Multistage Separation Process Problems", *AIChE Journal*, v.32, p.547 (1986).

- Watson, L.T. y D. Fenner. "Algorithm 555: Chow-Yorke Algorithm for Fixed Points or Zeroes of  $C^r$  Maps", ACM Trans. on Math. Software, v.6, p.252 (1980).
- Wayburn, T.L., "Modelling Interlinked Distillation Columns by Differential Homotopy-Continuation", Tesis de Doctorado, University of Utah, (1983).
- Wegstein, J.H., "Accelerating Convergence of Iterative Processes" Communication Association Computing Machines, v.1, p.9 (1958).
- Westerberg, A.W., H.P. Hutchinson, R.L. Motard y P. Winter, "Process Flowsheeting", e. Cambridge University Press, Cambridge (1974).
- Westerberg, A.W., "Optimization in Computer Aided Design", en Foundations of Computer-Aided Chemical Process Design, V.1, e. Engineering Foundation, Nueva York (1981).
- Westerberg, A.W. y T.J. Berna, "Decomposition of Very Large Scale Newton-Raphson Based Flowsheeting Problems", Comp. Chem. Eng., v.2, p.61 (1978).
- Westerberg, A.W. y S. Director, "A Modified Least-Squares Algorithm for Solving Sparse  $n \times n$  Sets of Nonlinear Equations", Comp. Chem. Eng., v.2, p.61 (1978).
- Westerberg, A.W. y F.C. Edie, "Computer Aided Design: I.- Enhancing Convergence Properties by the Choice of Output Variable Assignments in the Solution of Sparse Equation Sets" Chem. Eng. Journal, v.2, p.9 (1971 a).

Westerberg, A.W. y F.C. Edie, "Computer Aided Design: II.- An Approach to Convergence and Tearing in the Solution of Sparse Equation Sets", Chem. Eng. J., v.2, p.17 (1971 b).

Wood, E.S., "Two Pass Strategies for Sparse Matrix Computations in Chemical Process Flowsheeting Problem", Tesis de Doctorado, University of Illinois (1982).

Yourdon, E. y L.L. Constantine, "Structured Design", e. Yourdon Inc., Nueva York (1975).