



UNIVERSIDAD LA SALLE

ESCUELA DE QUIMICA

Incorporada a la U. N. A. M.

**"ESTUDIO, COMPLEMENTACION Y APLICACION DE UN  
PROGRAMA DE SIMULACION DE PROCESOS"**

TESIS CON  
FALLA DE ORIGEN

**TESIS PROFESIONAL  
QUE PARA OBTENER EL TITULO DE:  
INGENIERO QUIMICO  
P R E S E N T A :  
LEONOR DORIS HERRERAMORO GOMEZ**

MEXICO, D. F.

1986



Universidad Nacional  
Autónoma de México



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

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

Todo el material contenido en esta tesis 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   G E N E R A L

I	INTRODUCCION	...	1
	Referencias	...	23
II	OBJETIVOS	...	24
III	DISEÑO DE PROCESOS POR COMPUTADORA	...	25
	3.1 Simulación de Procesos	...	25
	3.1.1 Generalidades	...	25
	3.1.2 Tipos de Simulación para Procesos Químicos	...	30
	3.1.3 Estructuración de la Simulación	...	33
	3.1.3.1 Programas de Diagramas de Flujo	...	38
	Variables de las Corrientes	...	38
	Diagramas de Flujo de Información y su Representación Numérica	...	40
	3.1.4 Errores Factibles dentro de la Simulación	...	44
	3.2 Actividades Relacionadas con la Simulación	...	48
	3.2.1 Optimización (Investigación de Operaciones)	...	48
	3.2.2 Diseño, Investigación y Operación	...	49
	3.3 El Método Secuencial Iterativo para el Cálculo de Problemas con Recirculación - Método de Convergencia	...	52
	Referencias	...	55
IV	GEMCS	...	56
	4.1 El Programa Maestro	...	57
	4.1.1 Breve Descripción	...	57
	4.1.2 Variables del Programa Maestro	...	59
	4.1.3 Bases de Datos	...	61
	Hojas de Codificación	...	63

4.1.4 Descripción Funcional	...	68
4.1.5 Diagrama de Flujo del Programa Maestro	...	71
Listado del Programa Maestro	...	72
4.2 Subrutinas del Programa Maestro	...	76
4.2.1 La Subrutina de Lectura de la Base de Datos DLOAD1	...	76
Listado de la Subrutina DLOAD1	...	81
4.2.2 La Subrutina de Parámetros de Equipo DISKIO	...	84
Listado de la Subrutina DISKIO	...	85
4.2.3 La Subrutina de Almacenamiento de Corrientes STREAM	...	86
Listado de la Subrutina STREAM	...	87
4.3 Programas de Soporte	...	88
4.3.1 Código Numérico de los Equipos	...	89
4.3.2 Simbología para las Unidades Computacionales	...	90
4.3.3 Nomenclatura para los Módulos en GEMCS	...	91
Listado de la Subrutina MODULE	...	94
4.3.4 Módulo de Computación JUNCO1	...	95
Listado de la Subrutina JUNCO1	...	96
4.3.5 Módulo de Computación CONTL1	...	97
Listado de la Subrutina CONTL1	...	99
4.3.6 Módulos de Computación SEPA01, SEPA02 y SEPA03	...	101
Listado de la Subrutina SEPA01	...	103
Listado de la Subrutina SEPA02	...	104
Listado de la Subrutina SEPA03	...	105
4.3.7 Módulo de Computación REACT1	...	106
Listado de la Subrutina REACT1	...	108
Referencias	...	109

V	IMPLEMENTACION DE DOS NUEVAS SUBROUTINAS	... 110
	5.1 Formulación Matemática de una Subrutina Correspondiente a un Balance de Materia y Energía	... 110
	5.1.1 Diagrama de Flujo de la Subrutina JUNCO2	... 117
	Variables Particulares de la Subrutina JUNCO2	... 122
	5.1.2 Módulo de Computación JUNCO2	... 125
	Listado de la Subrutina JUNCO2	... 127
	5.2 Formulación Matemática de una Subrutina Correspondiente a una Destilación Flash	... 130
	5.2.1 Diagrama de Flujo de la Subrutina FLASH1	... 144
	Variables Particulares de la Subrutina FLASH1	... 153
	5.2.2 Módulo de Computación FLASH1	... 158
	Listado de la Subrutina FLASH1	... 159
	Referencias	... 164
VI	MANUAL DE OPERACION DEL PROGRAMA 'GEMCS'	... 165
	6.1 Como Correr el Programa de Simulación de Procesos Químicos 'GEMCS'	... 166
	6.2 Datos	... 168
	6.2.1 Cambio del Archivo de Datos	... 173
	6.2.2 Generación, por Parte del Usuario, de un Nuevo Archivo de Datos que él Desea Resolver	... 175
	6.3 Modificación de Textos	... 178
	6.4 Adición de Nuevas Subrutinas	... 178
VII	APLICACION DEL PROGRAMA DE SIMULACION DE PROCESOS QUIMICOS 'GEMCS'	... 180
	Ilustración I.- Aplicación de las Subrutinas REACT1, SEPA01, SEPA02 y SEPA03 del	

Programa de Simulación de Procesos Químicos 'GEMCS'	... 181
Base de Datos Correspondiente a la Ilustración I	... 184
Ejecución de la Ilustración I	... 185
Análisis de Resultados	... 187
Ilustración II.- Aplicación de las Subrutinas JUNCO1 y CONTL1 del Programa de Simulación de Procesos Químicos 'GEMCS'	... 188
Base de Datos Correspondiente a la Ilustración II	... 192
Ejecución de la Ilustración II	... 193
Análisis de Resultados	... 195
Ilustración III.- Aplicación de las Subrutinas JUNCO2 y FLASH1 del Programa de Simulación de Procesos Químicos 'GEMCS'	... 196
Base de Datos Correspondiente a la Ilustración III	... 200
Ejecución de la Ilustración III	... 201
Análisis de Resultados	... 203
Referencias	... 205
VIII CONCLUSIONES	... 206
BIBLIOGRAFIA	... 211

## INDICE DE FIGURAS

- 1.1 Diagrama de Flujo de Proceso de las Operaciones Básicas en una Planta de Alquilación
- 1.2 Relación entre el Ingeniero y el Programa Maestro
- 3.1 Estrategia Global Propuesta
- 3.2 Organización de las Principales Líneas de Comunicación
- 3.3 Diagrama de Flujo de Proceso de un Proceso Petroquímico
- 3.4 Diagrama de Flujo de Información Correspondiente a la Figura 3.3
- 3.5 Diferencias entre el Diseño y la Simulación
- 3.6 Relación entre las Actividades Ingenieriles
- 4.1 Diagrama de Flujo del Programa Maestro en GEMCS
- 5.1 Mezclador-Separador
- 5.2 Vaporización Flash Continua
- 7.1 Diagrama de Flujo de Información de la Ilustración I
- 7.2 Diagrama de Flujo de Información de la Ilustración II
- 7.3 Diagrama de Flujo de Información de la Ilustración III

## INDICE DE TABLAS

- 1.1 Beneficios y Limitaciones de la Simulación de Procesos
- 1.2 Ventajas y Desventajas de la Simulación de Procesos
- 1.3 Principales Usos de la Simulación de Procesos
- 3.1 Formato para el Flujo de Información en una Corriente
- 3.2 Matriz de Proceso para el Proceso de la Figura 3.4
- 4.1 Formato para los Conjuntos de Datos de los Módulos en GEMCS
- 4.2 Formato para los Parámetros de las Propiedades en GEMCS
- 4.3 Formato para los Parámetros de las Corrientes en GEMCS
- 4.4 Módulos Usados en GEMCS
- 7.1 Condiciones de Alimentación para el Problema de la Ilustración III
- 7.2 Coeficientes para el Cálculo de Entalpías en Fase Líquida
- 7.3 Coeficientes para el Cálculo de Entalpías en Fase Vapor
- 7.4 Coeficientes para el Cálculo de la Constante de Equilibrio a 250 p.s.i.a.



## CAPITULO I

### I N T R O D U C C I O N

La meta de las profesiones ingenieriles es la traducción del conocimiento científico en mejoras tangibles en la calidad de vida de la raza humana. La ingeniería química es, a grandes rasgos, el diseño, construcción y operación de procesos en los que las operaciones esenciales son las reacciones químicas encargadas de convertir la materia prima en productos útiles. Así, con el objeto de lograr un cambio en estado, cantidad de energía y/o composición, los ingenieros químicos utilizan las ciencias físicas conjuntamente con la economía y las relaciones humanas para desarrollar, diseñar y llevar a cabo la ingeniería tanto del proceso completo como del equipo usado en él (1). Para ello deben elegir la materia prima adecuada; operar sus plantas eficiente, segura y económicamente y ver que sus productos satisfagan la calidad y requisitos establecidos por los consumidores.

Consistente con la ingeniería en general, la ingeniería química es tanto un arte como una ciencia pues, con frecuencia, el ingeniero deberá usar su iniciativa, ingenio, buen juicio y experiencia para dar una respuesta completa a sus problemas; por lo tanto, su nivel profesional dependerá de su destreza para combinar todas las fuentes de información que le permitan lograr soluciones prácticas para sus problemas de proceso.

La industria química se encuentra en una fase de gran desarrollo y día con día se proponen nuevos procesos o modificaciones a los ya existentes. Para poder introducir rápidamente las nuevas ideas se necesitan técnicas

de evaluación rápida de las innovaciones propuestas. Una de estas técnicas es la simulación matemática de procesos, entendiéndose por este tipo de simulación la representación de los mismos mediante modelos matemáticos que deben resolverse para obtener información acerca del comportamiento de los sistemas involucrados (2).

La tendencia a usar técnicas como la simulación ha persistido debido al creciente uso de las computadoras electrónicas que han reemplazado a las calculadoras de escritorio en la solución de sistemas complejos de ecuaciones matemáticas por su habilidad para manejar grandes cantidades de información. Este acercamiento computarizado para la resolución de problemas de ingeniería crea un horizonte considerablemente más amplio en la investigación de diseños alternativos al ayudar a un mayor entendimiento de los mecanismos internos de los procesos estudiados. Conviene comentar que no existe tal cosa como el proceso químico típico sino que cada proceso representa una solución particular de ingeniería a un problema único.

El uso que el ingeniero químico le da a las computadoras depende directamente de su posición en la industria de procesos. Al participar en toda una gama de decisiones, el profesionista consciente debe manifestar una preocupación constante por los desarrollos tecnológicos que afecten las operaciones diarias y a largo plazo en las que se ve envuelto de ahí que, en muchos casos, se haya incorporado el uso de las computadoras no sólo a la simulación de procesos sino también al diseño de equipo.

El desarrollo de la simulación matemática aplicada a los procesos químicos empezó a mediados de los 50's y el primer simulador, Flexible Flowsheet (2), fue publicado en 1958. Este simulador se implementó en base a

una estructura global de tal forma que sólo simulaba un proceso específico, como todos los de esa época. La persona encargada de desarrollar estos programas, muchos de ellos elaborados para operaciones unitarias individuales, debía ser una experta en ingeniería química, matemáticas y computadoras, incluyendo lenguajes de máquina, y por lo general era la única que podía emplear dichos programas para la resolución de problemas prácticos.

A mediados de los 60's surgió la estructura modular como un nuevo concepto en la simulación de procesos químicos al crearse simuladores generales del tipo de CHESSE, PACER, GEMCS, FLOWTRAN y muchos otros sistemas de diseño por computadora que representaron durante los 70's la estructura más popular. Estos simuladores siguieron tres pasos durante su desarrollo:

1. La codificación se hacía en un lenguaje de nivel elevado (por ejemplo, FORTRAN) y debía ser altamente modular (característica principal).
2. Las correlaciones para las propiedades físicas debían ser lo más rigurosas y exactas posibles.
3. El sistema debía poder utilizarse en forma fácil con poca o ninguna intervención de expertos.

En nuestros días la estructura modular es la más desarrollada caracterizándose por un alto grado de participación del usuario, un rápido acceso entre componentes y una gran capacidad de transferencia de arreglos masivos con datos del proceso (2) (3).

Por último, a fines de los 70's, se empezaron a proponer los simuladores con base en solución de ecuaciones. Actualmente, es hacia este tipo de estructura hacia donde se están enfocando los esfuerzos de la mayoría de los investigadores en el área; su objetivo es la creación de sistemas para el diseño de plantas de proceso totalmente integrados que incorporen toda

la información de un proyecto. Sin embargo, aun se encuentra en etapa de desarrollo (4).

En función del uso que se les da, podemos distinguir dos tipos de simuladores: industriales y didácticos. Los primeros generalmente tienen paquetes de propiedades y módulos de proceso más exactos y eficientes que los segundos que, en un momento dado, no requieren de tanta precisión.

Reflexionando acerca de los progresos que se han hecho en el área de la simulación de procesos a régimen permanente desde la publicación del primer simulador, podrían plantearse dos preguntas:

1. ¿Qué tan bien ha cumplido la simulación de procesos químicos con las expectativas planteadas en la industria y en la educación?
2. ¿Cuáles son las principales áreas de investigación a las que debe extenderse la utilidad de la computadora en el campo de la simulación?

Probablemente la experiencia más importante obtenida en los últimos años es que la simulación de procesos no representa un fin por sí misma sino que es simplemente una nueva herramienta para el ingeniero químico ya que no puede reemplazar su criterio en la resolución de problemas de proceso.

Hasta ahora, el grado de aceptación y aplicación de la simulación de procesos en la industria ha variado ampliamente pues muchos usuarios industriales suelen preferir técnicas con las que estén familiarizados además de que no todos los trabajos de ingeniería de procesos necesitan de un programa de simulación ya que a algunos les basta con renovar o pulir viejas soluciones manuales.

Sin embargo, podría decirse que a partir de 1970 los simuladores se convirtieron en herramientas legítimas para el ingeniero de proceso aunque su uso se restringía, principalmente, a las compañías que, por contar con recursos suficientes y con un buen equipo de ingeniería, habían podido desarrollar y documentar su propio simulador. En México, algunas firmas de ingeniería como el Instituto Mexicano del Petróleo y Bufete Industrial poseen simuladores de tipo industrial.

Teóricamente, la simulación de procesos puede resultar útil en cualquier etapa del desarrollo de un proyecto químico ya que las diferentes etapas del mismo requieren de programas de simulación con diferentes niveles de sofisticación pudiendo mencionarse:

1. Etapa de Investigación y Desarrollo. Puede usarse un programa de simulación simple que requiera de un mínimo de datos para probar la factibilidad económica de diferentes procesos.

2. Etapa del Examen Crítico. Una vez que se ha encontrado un proceso financieramente atractivo, deben probarse diferentes alternativas de los proyectos de la planta y de sus condiciones de operación para su optimización.

3. Etapa de la Planta Piloto. El uso de un simulador de procesos más sofisticado puede ayudar en la obtención de buenos estimadores que permitan inferir las condiciones de operación de la planta a gran escala a partir de relativamente pocos resultados en la planta piloto.

4. Etapa de Diseño. El simulador de procesos puede proporcionar al ingeniero todos los datos del proceso requeridos para el diseño detallado de los diferentes equipos. De esta manera pueden reducirse los factores de seguridad debidos a la incertidumbre en el diseño del equipo.

5. Simulación de Plantas Existentes. Este tipo de simulación puede ser muy útil cuando existe la necesidad de cambiar las condiciones de operación de una planta porque puede ayudar a encontrar la mejor estrategia para aumentar la producción y la eficiencia de operación a través de una mejor integración energética y a adaptar la planta existente a una materia prima diferente o a la demanda de productos con composiciones distintas.

En lo que respecta a la educación, las principales barreras para una amplia aceptación de la simulación parecen ser su costo y la disponibilidad de datos razonables y suficientes.

En este campo pueden distinguirse cuatro usos diferentes de la simulación de procesos:

1. enseñanza de la simulación de procesos;
2. uso de los programas de simulación como una ayuda en la enseñanza de otras materias;
3. proyectos de simulación y diseño;
4. investigación sobre las distintas técnicas de simulación.

La aplicación pedagógica más importante de los programas de simulación parece ser el diseño avanzado de un proyecto de simulación en el que la disponibilidad de esta herramienta y el deseo de lograr un buen diseño obligan al estudiante a usar la mayor parte de sus conocimientos y a traducirlos en un diseño práctico y económicamente factible, obteniendo al mismo tiempo cierta experiencia industrial. No obstante, el instructor de estos cursos deberá afrontar ciertas dificultades debido a la falta de problemas realistas pero no triviales que sean lo suficientemente sencillos para ser manejados por los estudiantes. Así, la colaboración industrial en un proyecto estudiantil aumentaría la veracidad del

mismo al proporcionar información de diseños ya resueltos que permanezcan en los archivos de la compañía. En México, ciertas instituciones de educación superior cuentan ya con simuladores de tipo académico.

Aunque existen algunos problemas para entrenar a los ingenieros en el uso profuso de la simulación de procesos, debido a la gran demanda de consulta hacia los grupos que desarrollaron algún programa de simulación, en las universidades de Estados Unidos ha crecido un interés especial en la utilización de estos programas así como en su investigación con lo cual se espera proporcionar al campo profesional elementos humanos adecuadamente preparados. Es por ésto que se cree que en los próximos años el empleo de la simulación de procesos en la industria química puede verse fuertemente acelerado, más aun si se considera que hoy en día las poderosas computadoras se encuentran prácticamente disponibles a nivel mundial tanto en la escuela como en la industria.

Así, la investigación académica deberá concentrarse en la optimización de procesos, en aplicaciones para la conservación de recursos y en la simulación dinámica; no olvidando, por supuesto, las principales aplicaciones industriales y los problemas más críticos a los que se enfrenta el usuario industrial.

Debe notarse que en el campo de la simulación de procesos industriales existe una gran preocupación por el área de las propiedades físicas (especialmente a temperaturas criogénicas) ya que su desarrollo y uso a lo largo de amplios rangos de temperatura y presión frecuentemente constituyen uno de los principales tropiezos para lograr una simulación exitosa. Consecuentemente, se plantea la necesidad de ampliar las bases de datos,

las correlaciones de las propiedades físicas que maneja el simulador y la confiabilidad de los métodos de estimación de las mismas pues aunque actualmente se manejan en forma adecuada los fluidos no polares, se espera que en el futuro pueda lograrse la habilidad para tratar en forma rigurosa tanto el procesamiento de sólidos (incluyendo polímeros) y su manejo como el de componentes polares. Por otro lado, también los procedimientos de cálculo rápidos y confiables de las unidades siguen siendo un problema, especialmente los que se refieren a destilación no ideal y absorbedores, sucediendo algo similar en lo relacionado con cálculos de flashes trifásicos por no poderse encontrar datos apropiados para los modelos de equilibrio vapor-líquido no ideales.

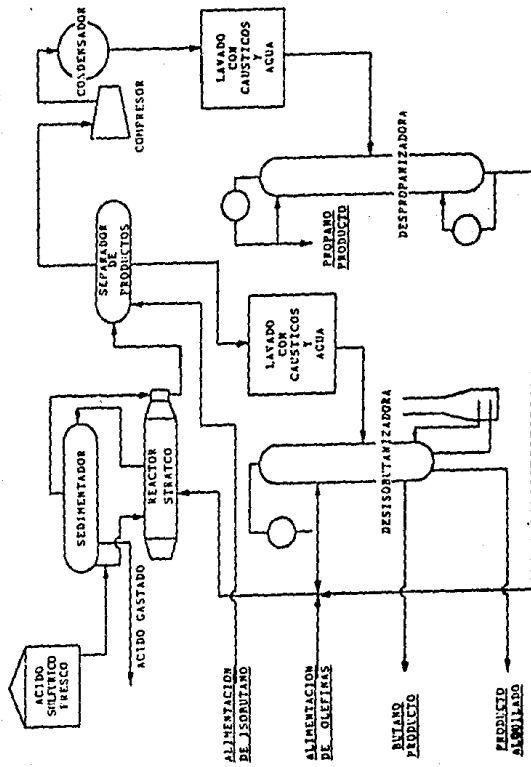
Para ver la forma en que pueden capitalizarse los adelantos que plantea el uso de computadoras dentro de la simulación de procesos, conviene considerar la manera en que un ingeniero maneja los cálculos por efectuar, ya sea que los haga a mano o mediante la ayuda de una computadora, y establecer una estrategia de resolución.

Para analizar un proceso (como podría ser el representado en la Figura 1.1 que corresponde a las operaciones básicas que se efectúan en una planta de alquilación) en busca de posibles mejoras, el ingeniero idea una descripción matemática de las innovaciones propuestas, esto es, crea una simulación. Tradicionalmente, en la simulación modular (que es la que nos interesa) de cualquier proceso, el ingeniero comienza con ciertas condiciones de entrada y utiliza los principios de las operaciones unitarias que describen el fenómeno que ocurre en la unidad para calcular las condiciones de salida de la misma. Esto se repite para las unidades subsecuentes con las condiciones de salida de una o varias unidades



Figura 1.1

DIAGRAMA DE FLUJO DE PROCESO DE LAS OPERACIONES BASICAS  
EN UNA PLANTA DE ALQUILACION



siendo las condiciones de entrada de la(s) siguiente(s) por lo que deberá conocerse la secuencia del proceso así como los arreglos entre sus unidades (ver Figura 4.1). Así, el ingeniero elabora un diagrama de flujo en base al proceso de la planta a simular que le sirve para llevar un control de los cálculos y resumir los resultados y a partir del cual tratará de especificar la secuencia mediante la que deberán efectuarse los cálculos de forma que se minimice el trabajo. El arreglo de simulación puede establecerse en otro diagrama en el que para cada juego de condiciones de entrada existirá el correspondiente juego de respuestas calculadas. Finalmente, el ingeniero de proceso deberá ser capaz de reconocer la presencia de problemas de recirculación o bien, de prueba y error.

De estas características, solamente el equipo, el arreglo del mismo y las condiciones de entrada a la planta varían según el problema pues el resto — el manejo de números y la transmisión de información de un cálculo a otro, la identificación de recirculaciones y la especificación de la secuencia de cálculo a seguir — es común a todos los problemas por lo que puede manejarse ventajosamente mediante un programa maestro de computadora que permita efectuar rápidamente los cálculos repetitivos necesarios en los una vez temidos problemas con recirculación. No hay que olvidar que en las plantas químicas las recirculaciones se presentan con frecuencia para conservar material y para mejorar la eficiencia global del proceso.

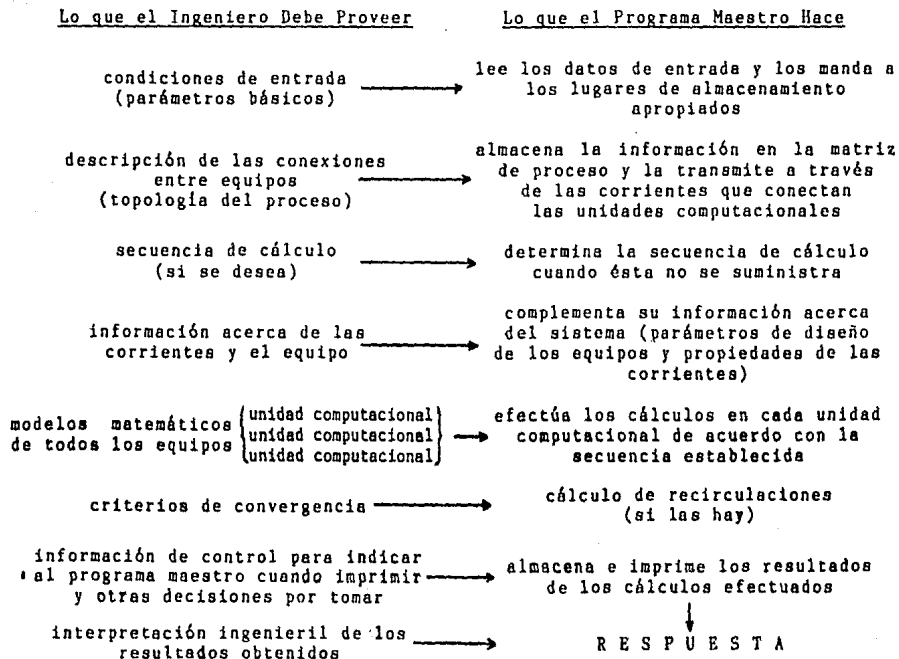
Con un sistema modular como el descrito, el ingeniero químico provee las condiciones específicas de entrada, indica el arreglo del equipo y suministra el modelo matemático que describa la operación unitaria en cada

unidad del equipo; a su vez, el sistema manipulará los datos para regresar al usuario información de acuerdo a la Figura 1.2 (adaptada de (3)). Dentro de este tipo de arquitectura, el programa maestro es como un director de proyecto que obtiene resultados realistas para el proceso completo pidiendo a cada ingeniero que lo provea con el cálculo de una unidad de proceso.

Convendría mencionar que, para facilitar su trabajo, el ingeniero químico divide un proceso de manufactura en operaciones unitarias (cambios producidos por procesos físicos como flujo de fluidos y transferencia de calor) cuyas características de comportamiento pueden estudiarse independientemente del material que se esté procesando y de otras características del sistema en particular. Por otro lado puede decirse que una unidad de proceso es un aparato ó un conjunto de equipos que forma parte de un proceso químico y que logra ciertos cambios físicos, químicos o ambos en el material que pasa a través de ella. La mayoría de las unidades de proceso llevan a cabo más de una operación unitaria.

Debe recordarse que el ingeniero químico relacionado con la simulación de procesos u otra de las aplicaciones de las computadoras está trabajando en un ambiente muy dinámico. El rápido desarrollo del "hardware" y del "software" de las computadoras ofrece nuevos usos de las mismas sugiriendo posibles soluciones a nuevos tipos de problemas de manera que los problemas inaccesibles de hoy podrían resolverse mañana al desarrollarse una computadora más grande o un lenguaje de computación más eficaz que permita resolver los problemas más complicados con un máximo de eficiencia y un mínimo de esfuerzo y conocimientos por parte del usuario.

Figura 1.2 RELACION ENTRE EL INGENIERO Y EL PROGRAMA MAESTRO



Es importante notar la diferencia entre el usuario y el creador de un programa de simulación. Mientras que el usuario podría ser cualquiera de nosotros ingenieros químicos con un problema por resolver; el creador de un simulador generalmente es un equipo de profesionistas constituido por varios grupos de expertos en distintos campos, que complementan de esta forma sus conocimientos, y, en ocasiones, algunos estudiantes universitarios y/o de postgrado supervisados por un coordinador.

En base a esto puede preverse que en el futuro el usuario de los programas de simulación tendrá que saber cada vez menos acerca de los métodos de cómputo, los lenguajes y la estructura de los programas de simulación pues un breve entrenamiento será suficiente para que el ingeniero químico adquiera la habilidad para utilizar la simulación de procesos en la resolución de muchos de sus problemas. Por otro lado, el equipo de desarrollo y mantenimiento tendrá que ser mucho más experto en métodos numéricos y de optimización y en lenguajes de programación y de control de ejecución, además de serlo en las materias clásicas de la ingeniería química (entre ellas, termodinámica y operaciones unitarias). Así el programa de simulación podrá irse actualizando simultáneamente a otros desarrollos en el campo de la computación.

El tremendo impacto de la simulación sobre la industria de los procesos químicos se debe principalmente a los beneficios, citados en la Tabla 1.1, que de ella se derivan. No obstante, el ingeniero de proceso deberá estar consciente de que existen ciertas limitaciones a las que también se hace referencia en la Tabla 1.1.

Resulta lógico pensar que los mencionados beneficios son consecuencia directa de las ventajas planteadas al

Tabla 1.1

BENEFICIOS Y LIMITACIONES DE LA SIMULACION DE PROCESOS

<u>Beneficios</u>	<u>Limitaciones</u>
* generación de mejores diseños con capital y costos de operación menores y en menos tiempo que si se hicieran a mano o combinando cálculos manuales con programas de computación aislados	* falta de buenos datos
* aumentar las utilidades producidas por la planta	* desconocimiento de los mecanismos del proceso
* mejorar la comunicación entre distintos grupos de ingeniería	* el carácter de las herramientas computacionales (existen algunos tipos de sistemas de ecuaciones que aun resultan un problema para el ingeniero como ciertos sistemas algebraicos no lineales y algunos sistemas de ecuaciones diferenciales parciales no lineales)
* es una forma conveniente de investigar los efectos de las variaciones en los parámetros del sistema o de alteraciones en el proceso sobre la operación del sistema	* el peligro de olvidar las consideraciones hechas al modelar el proceso
* ayuda a comprender comportamientos y mecanismos del proceso significativo lográndose un mayor aprendizaje del mismo	
* auxiliar en el chequeo y optimización de diseños antes de su construcción (objetivo: evitar errores costosos y "garantizar" diseños seguros)	

Tabla 1.2

VENTAJAS Y DESVENTAJAS DE LA SIMULACION DE PROCESOS

Ventajas

- \* permite la experimentación controlada
- \* admite la compresión del tiempo
- \* no perturba al sistema real
- \* constituye una herramienta efectiva por su poderosa aplicación a nivel educacional y de entrenamiento
- \* permite realizar análisis de sensibilidad mediante la manipulación de las variables de entrada

Desventajas

- \* resulta muy costoso construir y validar los modelos de simulación para computadora, especialmente en términos de horas-hombre y de tiempo de máquina
- \* dado que la mayoría de los modelos de simulación son bastante largos, consumen mucho tiempo de computadora al correrse lo que también los encarece
- \* el modelo puede divergir de la realidad debido a consideraciones críticas que permanecieron ocultas
- \* los parámetros del modelo pueden resultar difíciles de inicializar lo que implicaría la dedicación de mucho tiempo para su colección, análisis e interpretación
- \* la tendencia de la gente a utilizar esta técnica aun cuando no sea el mejor método de análisis

Tabla 1.3

PRINCIPALES USOS DE LA SIMULACION DE PROCESOS

- \* predecir el efecto ocasionado por:
  - a) condiciones variables
  - b) distribuciones físicas cambiantes
  - c) capacidades distintas a las normalmente manejadas
- \* ver la sensibilidad de los resultados a cambios en los parámetros de diseño
- \* efectuar balances de materia y energía
- \* optimizar la operación de una planta
- \* detectar y eliminar cuellos de botella en producción
- \* proveer suficientes conocimientos acerca del comportamiento del sistema completo
  - \* mejorar el control del proceso
- \* investigar la factibilidad del control por computadora
- \* facilitar los cálculos de costos, el control del presupuesto y la planeación de operaciones
  - \* entrenar operadores e ingenieros
- \* hacer factible y financieramente atractiva la resolución de problemas muy complejos que, de otra forma, no podrían ser atacados como:
  - a) cuando se complican por la presencia de recirculaciones
  - b) aquellos que involucran muchos números
  - c) los que requieren del análisis de un gran número de alternativas



emplear la simulación. En la Tabla 1.2 se consideran tanto sus ventajas como sus desventajas.

Finalmente, la Tabla 1.3 corresponde a algunos de los usos que pueden dársele a una simulación por computadora ayudada por un programa maestro.

Además de considerar beneficios, limitaciones, ventajas, desventajas y aplicaciones; para decidir si crear o no una simulación deberán tomarse en cuenta la necesidad de corroborar datos, el acceso inmediato a la computadora, la habilidad para describir el comportamiento del equipo y la disponibilidad de gente con experiencia en computación y programación.

Ahora bien, vale la pena aclarar que en función del tipo de computadora utilizado, la simulación puede considerarse analógica, digital o híbrida. La elección entre una u otra dependerá, principalmente, de las instalaciones a las que se tenga acceso. En nuestro caso, se desarrolló una simulación digital al resolverse las ecuaciones matemáticas de los modelos del programa mediante una microcomputadora digital por contar la universidad con este tipo de computadoras.

Cuando se emplea una computadora para correr un modelo, se plantea la necesidad de utilizar un lenguaje de computación apropiado. Considerando como supremo, para la selección de un lenguaje, el criterio que relaciona los aspectos de flexibilidad operacional, facilidad de transferencia que presente el lenguaje y su eficiencia se eligió FORTRAN como lenguaje de programación no sólo porque el programa original estaba escrito en él sino también por:

- a) su disponibilidad;

- b) familiaridad del programador con el lenguaje o facilidad de aprendizaje y uso si no se le conoce;
- c) eficiencia con que puede correrse un programa.

Otra de las ventajas del FORTRAN, lenguaje con fines generales designado para uso científico y que ha sido profusamente utilizado por los equipos de simulación experimentados, es su universalidad ya que casi todos los graduados de ingeniería y los programadores lo conocen y prefieren emplearlo a aprender un nuevo lenguaje porque tanto técnicas numéricas sencillas como complicadas trabajan a la perfección. Todo ésto sin olvidar que incluso instalaciones de cómputo pequeñas suelen contar con computadores para el mismo y que los modelos desarrollados en FORTRAN standard son fácilmente transportables por la similitud entre sus diferentes implementaciones.

Por último, habría que tomar en cuenta que los modelos desarrollados en base a lenguajes de alto nivel suelen costar menos al correrse y obligan al programador a estar en contacto con los detalles del modelo para su adecuada codificación.

Para cerrar esta introducción quisiera referirme un poco al programa de simulación de procesos que se escogió para su estudio, complementación y aplicación: GEMCS.

En 1974, Peters y Barker (6) compararon tres programas de simulación diferentes:

PACER: uno de los primeros y más conocidos programas de diagramas de flujo académicos,

GEMCS: en aquella época, uno de los programas más nuevos y compactos que, por lo mismo, gozaba de gran popularidad en las universidades y

CONCEPT: el único paquete completamente interactivo disponible, en Gran Bretaña, para usuarios interesados;

de acuerdo a su valor para los estudiantes, tanto universitarios como de postgrado, y para los ingenieros practicantes.

Ellos concluyeron que PACER resultaba de interés solamente para estudiantes universitarios pero, por su gran tamaño, era difícil de implementar en computadoras pequeñas; que GEMCS era superior a los otros dos para propósitos pedagógicos pues resultaba ideal tanto para el estudiante como para el investigador aunque también tenía ciertas aplicaciones industriales y que CONCEPT constituía un buen paquete de simulación industrial.

GEMCS se destacó por su tamaño relativamente compacto, sus requerimientos de entrada breves y fácilmente modificables y su estructuración simple que lo hace sencillo de entender y utilizar. Además, los catedráticos parecían preferir su carencia de una subrutina encargada de la evaluación del orden de cálculo de las unidades al emplearlo en la enseñanza. Por otro lado, la secuencia de cálculo alimentada puede ser pasada por alto temporalmente cuando, al presentarse recirculaciones, es necesario repetir los cálculos de ciertas unidades varias veces.

Estas habilidades lo hacen muy útil para los trabajos de optimización y diseño pues también cuenta con un control de impresión que si el usuario lo desea, permite "monitorear" el desarrollo de la simulación.

Por lo tanto, se optó por este programa, que la Universidad del Oeste de Ontario gentilmente prestó para

su estudio y desarrollo con fines educativos, cuya arquitectura secuencial modular es la más usada en la actualidad y que posiblemente sea la más fácil de comprender.

Con base en lo expuesto a lo largo de este capítulo creo que la simulación de procesos es un tema muy importante, ilustrativo e interesante para nosotros los estudiantes universitarios mexicanos que, ante la imposibilidad de generar nuestro propio programa de simulación, podemos estudiar y tratar de adaptar a nuestras necesidades un programa elaborado desde mediados de los 60's, el programa de simulación de procesos químicos GEMCS, para así reafirmar nuestros conocimientos, aprender a aplicar los y a comunicarnos a través de ellos complementando de esta forma nuestra educación para lograr un mejor desarrollo profesional.

GEMCS (General Engineering and Management Computation System) es un programa maestro de manejo de información, interactivo, orientado por archivos, creado en 1966 en la Universidad de McMaster (en Hamilton, Ontario, Canadá), por el profesor A. I. Johnson y sus asociados y alumnos para ser utilizado en el análisis y diseño de sistemas complejos de ingeniería y administración. La programación original la llevaron a cabo el Profesor Johnson y el señor O. Orbach; posteriormente, en 1968, el señor T. Toong recibió un año de licencia de la Compañía General Electric de Canadá para continuar con el desarrollo del sistema y crear la documentación básica. El programa GEMCS es un sistema modular que puede manejar cerca de 100 corrientes (cada una conteniendo hasta 30 componentes) y 50 unidades computacionales (cada una con 85 parámetros de equipo). Elaborado inicialmente para una computadora GE 265, el programa de simulación GEMCS ha sido adaptado a varias computadoras de tiempo compartido y "batch" entre

las que se pueden mencionar CDC (6400 y CYBER 73), IBM, PDP 11/34 y otras minicomputadoras que manejen FORTRAN como lenguaje de programación (3) (5).

El programa maestro fue diseñado de manera que sistemas enteros pudieran ser fácilmente estudiados usando unidades computacionales aproximadas que posteriormente pudieran ser reemplazadas sin dificultad por modelos más comprensivos de manera que se les pudiera analizar y aplicar dentro del sistema total del proceso. Los datos de entrada del proceso son "conversacionales" (el programa pregunta al usuario por la información de las corrientes y las unidades computacionales); además, se puede almacenar previamente información en los archivos para simplificar así la evaluación de muchas alternativas (corrientes y parámetros de equipo).

Actualmente, GENCS es un programa de simulación de procesos químicos que SACDA (Systems Analysis, Control and Design Activity de la Universidad del Oeste de Ontario) vende a industrias cuyos procesos poseen cierta complejidad por la variedad de equipos y de corrientes que se manejan, en forma integrada, en el diagrama general de flujo de las instalaciones en las plantas. Por lo mismo, su importancia radica en la versatilidad que presenta en cuanto a variación de condiciones de operación, a la adición o supresión de equipos dentro de un proceso dado y a las posibilidades de intercambio de corrientes de unos equipos a otros. Todas estas ventajas permiten obtener un registro expedito de las condiciones en que se encuentra operando un proceso sin necesidad de alterar el proceso en existencia o el proyecto y diseño del proceso bajo estudio.

Consciente de la imposibilidad de ignorar el desa

rrollo tecnológico que rodea a los ingenieros químicos y deseando poder despertar en mis compañeros interés por la simulación de procesos y sus múltiples usos, la presente tesis pretende describir el programa GEMCS para su aplicación a cualquier proceso después de estudiar la metodología para la introducción de los datos necesarios de manera que las nuevas subrutinas desarrolladas (solución de un balance de materia y energía en un mezclador monofásico y solución de una destilación flash adiabática para multicomponentes) se encuentren en forma adecuada a las secciones del programa original. El empleo del programa se ejemplifica mediante la simulación de tres casos que suelen presentarse en procesos típicos tomados de la vida real.

Como podrá verse a lo largo de este trabajo, GEMCS intenta ser un programa sencillo, flexible y orientado hacia el usuario. De hecho, se sabe que casi todas las personas que lo han utilizado le han introducido algunas modificaciones para adecuarlo a sus necesidades e instalaciones de cómputo (5), en este caso la Universidad La Salle. De ahí que la información contenida en esta tesis se suministre con la esperanza de que resulte útil para sus estudiantes en la resolución de distintos problemas, intentando motivarlos a usar, y quizá desarrollar, programas de simulación en su trayectoria profesional.

Referencias.

1. American Institute of Chemical Engineers Constitution.
2. Motard, R. L.; Shacham, M. & Rosen, E. M.; "Steady State Chemical Process Simulation", [A. I. Ch. E. Journal, 21 (1975), No. 3, pp. 417-436].
3. Crowe, C. M. et al.; "Chemical Plant Simulation", [McMaster University Press, Canada (1969) & Prentice Hall, U. S. A. (1971)].
4. Westerberg, A. W. et al.; "Process Flowsheeting", [Cambridge University Press, England (1975)].
5. Johnson, A. I.; Peters, N. & Associates; "GEMCS/CYBER 73", [S. A. C. D. A., University of Western Ontario, Canada (1979)].
6. Peters, N. & Barker, P. E.; "An Appraisal of the Use of PACER, GEMCS and CONCEPT for Chemical Plant Simulation", [Chem. Eng., (Junio 1978), pp. 145-152].

## CAPITULO II

### OBJETIVOS

- I. Proporcionar a la Universidad La Salle un programa de simulación de procesos con el fin de introducir a los estudiantes de Ingeniería Química a esta disciplina básica.
- II. Facilitar el uso de dicho programa mediante un manual comprensible y de fácil acceso, aun para personas sin experiencia en programación.
- III. Desarrollar dos nuevas subrutinas para el programa de simulación utilizando métodos numéricos aplicados a operaciones unitarias dentro de la Industria de Procesos.
- IV. Integrar algunos casos típicos de simulación tomados de la vida real usando tanto el programa base como las nuevas subrutinas incorporadas al mismo.



## CAPITULO III

# DISEÑO DE PROCESOS POR COMPUTADORA

### 3.1 SIMULACION DE PROCESOS.

#### 3.1.1 Generalidades.

Aunque al escuchar el término simulación se tiende a asociarla con la computación, ésto no siempre resulta acertado pues etimológicamente la palabra simulación proviene del latín simulatio y significa la acción de fingir por lo que, en esencia, cualquier modelo o representación de una cosa es una forma de simulación.

Para ampliar este concepto y enfocarlo hacia los objetivos del presente trabajo se cita la definición dada por C. M. Crowe et al en CHEMICAL PLANT SIMULATION An Introduction to Computer-Aided Steady-State Process Analysis (1): "Una simulación es un conjunto de programas de computadora que calcula valores realistas de las variables de operación significativas del proceso que se está simulando". En este caso, en la misma forma en que una planta está constituida por unidades de equipo conectadas por líneas de tubería u otros medios de flujo de materia y/o energía, una simulación está compuesta por conjuntos de cálculos, llamados unidades computacionales, enlazados por corrientes que transmiten la información entre ellos. De esta analogía se puede inferir que así como el flujo de materia se efectúa a través de las tuberías debido a las leyes de la naturaleza, la información fluye a través de la red de corrien

tes-unidades computacionales de acuerdo a lo estipulado en el programa maestro.

Así, una simulación tiene tres partes, cuya relación se indicó en la Figura 1.2, que son: el paquete de datos, el programa maestro y los conjuntos de modelos matemáticos o unidades computacionales.

La base de datos comprende las condiciones de entrada, la secuencia de procesamiento (muchas veces) y la información relativa a los cálculos por efectuar en las unidades computacionales.

El conjunto de cálculos que predicen lo que ocurre dentro de un equipo se conoce como unidad computacional. En los simuladores modulares, la unidad computacional (que se designará U. C.) está representada por una subrutina que es llamada por el programa maestro para desarrollar cierta función deseada. Además de los módulos de computación que representan dispositivos de la planta, también pueden existir U. C. que contengan funciones matemáticas para probar o forzar convergencias o para efectuar procesos de optimización (módulos de "control"); U. C. responsables de la lectura de los datos de las propiedades físicas y otros parámetros y U. C. que crean, para el usuario del programa, reportes adecuados una vez que se ha completado el caso en estudio. Obviamente, todas estas U. C. deberán tener una estructura compatible con el esquema de manejo de información del programa maestro. Entre sus sinónimos se pueden mencionar: subrutinas de equipo; módulos de proceso, de cálculo o de computación y bloques de construcción.

El programa maestro transmite la información a través de las corrientes y, en base a los datos sumi

nistrados, llama a las diferentes unidades computacionales en función de la secuencia de procesamiento (ya sea que se trace intrínsecamente o que se alimente en los datos) calculando y almacenando los valores de las variables de operación para todas las corrientes del sistema.

A grandes rasgos, una unidad computacional que representa un equipo en particular es un modificador de información ya que recibe los datos que describen a las corrientes de entrada (alimentaciones o bien, corrientes que provienen de otras unidades), los procesa matemáticamente de acuerdo al modelo del módulo empleando para ello los parámetros de la unidad asociados al mismo y genera la información de salida que es enviada a otra U. C. o que corresponde a los datos acerca de los productos finales de la planta.

Además de poder contar con distintos niveles de sofisticación (complejidad y precisión) para cada tipo de unidad computacional, pueden identificarse varias clases de U. C. para cada operación unitaria. Así, los modelos matemáticos de un equipo pueden ser:

a) Analíticos: en ellos las ecuaciones matemáticas describen las leyes de física y química referentes al proceso.

b) Estadísticos: se basan en correlaciones, a menudo empíricas, que relacionan las salidas con valores de las variables de la(s) corriente(s) de entrada y con los parámetros del equipo de acuerdo a estudios de planta previos y análisis de regresión adecuados.

c) Mixtos: en ellos los parámetros de los modelos analíticos tienen valores que dependen de la experimentación y de estudios estadísticos.

En esta tesis se hace énfasis en los modelos analíticos porque la mayoría de los cálculos de ingeniería involucran correlaciones, como es el caso de las capacidades caloríficas, que se basan en leyes físicas y químicas. Sin embargo, también se les podría considerar mixtos por ser función de correlaciones que surgieron de la experimentación.

Un modelo debe crearse con un propósito específico y su validez deberá evaluarse solamente en términos de este fin. El objetivo de una simulación es generar un modelo que cree los mismos problemas y tenga las mismas características de comportamiento que el proceso o sistema que se está estudiando y es precisamente ésto lo que hay que probar. Por consiguiente, la validez de un modelo no es una variable binaria sino que debe considerarse en una escala del 0 al 1 en donde 0 representa total invalidez y 1, totalmente válido.

Además, un buen modelo de simulación deberá ser:

1. de fácil comprensión para el usuario;
2. enfocado hacia una meta u objetivo;
3. robusto en cuanto a que no deberá dar respuestas absurdas (por robustez debe entenderse la cualidad de los programas que permite su operación aún bajo condiciones difíciles de cálculo proporcionando resultados confiables y permitiendo la continuidad en la simulación);
4. eficiente (entendiéndose por eficiencia la utilización de la menor cantidad de memoria y el menor tiempo de cómputo posibles);
5. fácil de manipular y controlar para el usuario, es decir, de fácil comunicación;
6. completo en lo que respecta a la impresión de resultados importantes;
7. capaz de adaptarse, mediante un procedimiento sencillo, a su modificación o actualización;
8. evolutivo en cuanto a que debe empezar en forma

simple e irse haciendo cada vez más complejo en conjunción con el usuario.

Cuando se elige la simulación por medio de computadoras, el modelo consistirá esencialmente en un conjunto de variables que correspondan a las variables del problema, restringidas por el "hardware" analógico o por el "software" digital, de manera que sólo puedan cambiar de acuerdo a un conjunto de ecuaciones que correspondan a aquellas que obedecen las variables del problema.

En conclusión, la simulación no es una teoría sino una metodología, experimental y aplicada, que busca resolver un problema a través de:

1. la descripción del comportamiento de el (los) sistema(s);
2. la construcción de hipótesis y teorías que expliquen y representen el comportamiento observado y
3. el uso de estas teorías para predecir un comportamiento futuro, esto es, los efectos que producirán cambios en el sistema o en su método de operación.

Más aún, la simulación es una de las varias aproximaciones valiosas disponibles para la resolución de problemas con las que cuenta el analista de sistemas.

Por lo tanto, se deberá considerar el empleo de la simulación cuando una o más de las siguientes condiciones se presenten:

1. La formulación matemática completa del problema no existe.
2. Los métodos analíticos están disponibles pero los procedimientos matemáticos son complejos.
3. Se desea observar una historia simulada del pro

ceso durante cierto período de tiempo.

4. La simulación puede ser la única posibilidad debido a la dificultad en la conducción de experimentos.

5. Puede requerirse la compresión del tiempo para procesos con marcos de tiempo largos.

### 3.1.2 Tipos de Simulación para Procesos Químicos.

En el caso de las simulaciones de procesos químicos, la representación del proceso se logra por medio de modelos matemáticos, que suelen ser programas de computadora también conocidos como programas de diagramas de flujo (Sección 3.1.3.1), constituidos por ecuaciones (algebraicas, diferenciales y/o lógicas) que reproducen en forma aproximada el comportamiento de un equipo o conjunto de equipos presentando sus resultados en forma tabular o gráfica.

Los sistemas a modelar pueden agruparse de manera muy amplia en continuos y discretos: un sistema en el que los cambios en su estado ocurren ininterrumpidamente en el tiempo es un sistema continuo mientras que aquellos en los que los cambios suceden en cuantía o saltos finitos son sistemas discretos. Dentro de los sistemas continuos, cuyos modelos contienen ecuaciones diferenciales, se encuentran la mayoría de las industrias de elaboración como son las fábricas de productos químicos. Asimismo existen algunos sistemas, conocidos como híbridos, que pueden poseer ambas propiedades.

Los sistemas también pueden clasificarse en determinísticos y estocásticos o probabilísticos. Un sistema determinístico es aquel en el que el nuevo estado del

sistema es fijado completamente por el estado previo y la actividad que ocurre mientras que un sistema estocástico contiene cierto grado de aleatoriedad en sus transiciones de un estado a otro por lo que no puede ser predicho inequívocamente a partir del estado precedente y el estímulo conocidos.

Una tercer clasificación divide a los sistemas en abiertos y cerrados siendo éstos últimos aquellos en los que los cambios de estado son ocasionados por actividades endógenas en contraste con los primeros en los que dichos cambios son una respuesta a actividades tanto internas como externas.

En lo que se refiere al ingeniero químico, por lo general deberá tratar con simulaciones de sistemas continuos determinísticos aunque también puede llegar a simular eventos discretos y sistemas estocásticos.

Dentro de la definición de simulación, la palabra proceso se utiliza para representar una serie de cambios de tipo físico y/o químico en algún material en específico. Los procesos pueden clasificarse a grandes rasgos en "batch" o por lotes cuando no son continuos y a régimen permanente o estacionario cuando lo son.

Profundizando un poco, un proceso químico es un conjunto de unidades de proceso tales como reactores, intercambiadores de calor, separadores y mezcladores en el que la unidad de proceso lleva a cabo una o más de las siguientes operaciones (se mencionan las más importantes):

1. reacción química (núcleo del proceso químico),
2. separación o purificación,
3. cambio en la temperatura,

4. cambio en la presión,
5. cambio de fase,
6. mezclado.

Así, a grandes rasgos, los simuladores de procesos se clasifican de acuerdo a:

a) su estructura intrínseca en:

particulares de un proceso, es decir, preparados específicamente para simular un proceso dado con una distribución de planta fija como por ejemplo los simuladores de plantas de destilación primaria de crudo;

modulares como SIMPRO, GEMCS, CHESS y muchos otros simuladores empleados para el diseño y la simulación y en los que cada paso del proceso químico se representa como un módulo (unidad computacional);

b) su aplicación en:

régimen permanente;

régimen no permanente.

Conviene comentar que aunque los simuladores particulares de un proceso son ventajosos, desde un punto de vista computacional, pues se puede lograr la máxima eficiencia en el cálculo y la memoria utilizada se reduce al mínimo, su uso se encuentra limitado a casos especiales por su falta de flexibilidad. En contraposición, los simuladores modulares, al estar formados por un conjunto de subrutinas que calculan equipos específicos y en los que sólo se emplean aquellas subrutinas necesarias para representar el proceso en estudio, ofrecen la ventaja de poder simular prácticamente cualquier proceso pero su eficiencia computacional es inferior y requieren mayor cantidad de memoria.

A través de los simuladores de procesos químicos se ha investigado el comportamiento, tanto dinámico como a régimen estacionario, de muchos sistemas químicos.



En el segundo caso, es necesario efectuar balances de materia y energía para la operación a régimen permanente de la planta química mientras que en el primero se simula la operación de la planta de acuerdo a su variación en el tiempo. La simulación dinámica es la más complicada y cara, computacionalmente, de las dos (2).

### 3.1.3 Estructuración de la Simulación.

Uno de los aspectos más beneficiosos de la simulación radica en que es un proceso de aprendizaje tanto para el modelador como para el usuario ya que en la elaboración del modelo debe ponerse especial cuidado en la planeación y desarrollo del mismo tomando en cuenta las necesidades y psicología del último usuario; éste, conforme adquiera mayores conocimientos del proceso, podrá ir mejorándolo.

La secuencia de pasos que deben seguirse para la construcción y aplicación de un modelo dependen del tipo y tamaño del problema que se está investigando; sin embargo, existen algunos puntos comunes e independientes del problema que, para la simulación de sistemas complejos, pueden resumirse en el diagrama de flujo de la Figura 3.1 (1) (3) (4) (5) (6).

Además, la Figura 3.2 muestra una posible forma de organización del personal durante la creación de la simulación (1).

Teniendo en mente que la simulación se usará para investigar las propiedades del sistema real se pueden distinguir las siguientes etapas:

Figura 3.1 ESTRATEGIA GLOBAL PROPUESTA

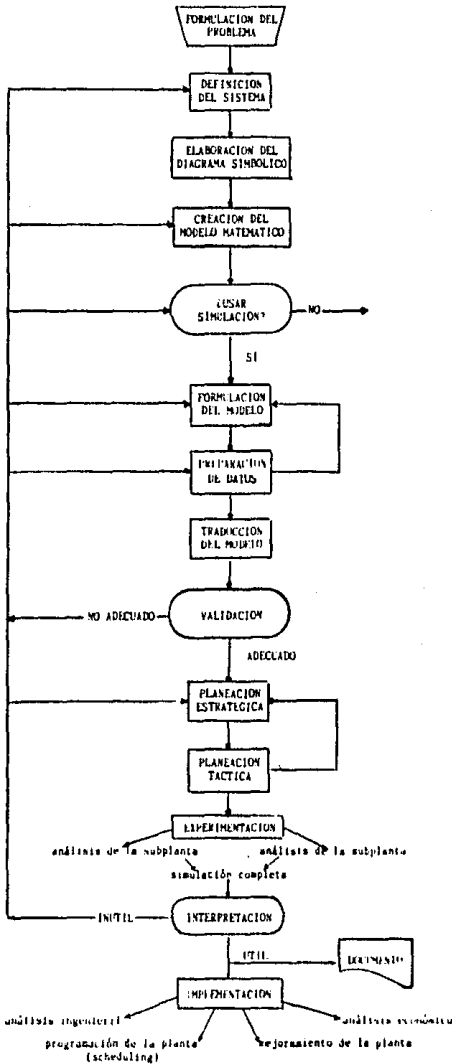
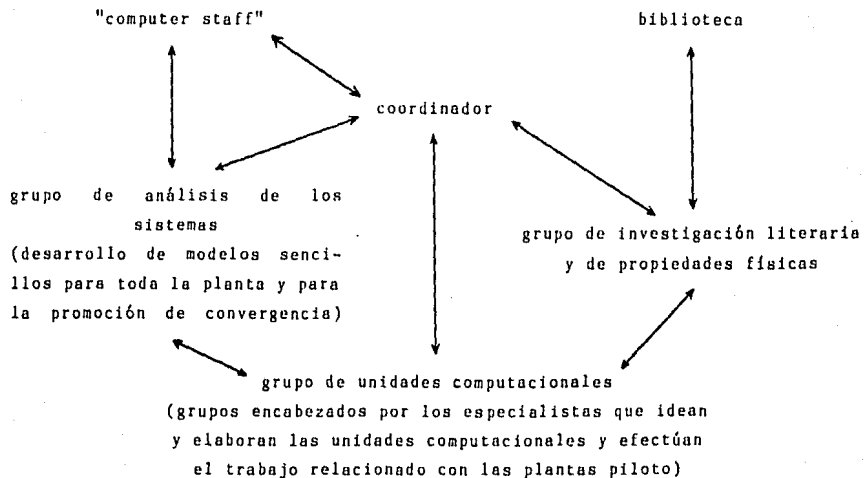


Figura 3.2 ORGANIZACION DE LAS PRINCIPALES LINEAS DE COMUNICACION



1. Formulación del Problema: en ella se considera el problema en general especificándose objetivos; a menudo no es un paso matemático, podría ser simplemente "dado un proceso en particular, determinar las condiciones a las que puede operarse con máxima eficiencia desde el punto de vista de las utilidades a obtener".

2. Definición del Sistema: determinación de los límites, restricciones y medida de efectividad que se utilizarán para definir el sistema por estudiar.

3. Elaboración de Diagramas Simbólicos: algunas veces el sistema o proceso por estudiarse se representa como una serie de bloques en la que cada bloque corresponde a cierto elemento del sistema.

4. Creación del Modelo Matemático: sus ecuaciones no sólo deben describir al sistema sino también a los objetivos que se persiguen; si las ecuaciones encontradas en esta etapa resultan ser muy sencillas, podría optarse por una solución analítica.

5. Formulación del Modelo: reducción o abstracción del sistema real a un diagrama de flujo lógico que muestre las unidades y las interacciones necesarias para modelar el sistema.

6. Preparación de Datos: identificación y recolección de los datos requeridos para la resolución del modelo; reducción de los mismos a una forma apropiada para su uso.

7. Traducción: en algunos casos, descripción del modelo en un lenguaje aceptable por la computadora que se va a emplear — desarrollo de las unidades computacionales.

8. Validación: aumentar a un nivel aceptable la confianza de que una inferencia acerca del sistema real obtenida a partir del modelo sea correcta; resulta

esencial cuando se estudian problemas por métodos computarizados el checar la validez de las soluciones alcanzadas, dichos chequeos deben relacionar directamente la solución final/interpretación con el problema original (o las ecuaciones del mismo) para eliminar errores de trabajo.

9. Planeación Estratégica: diseño de un experimento que permita obtener la información deseada.

10. Planeación Táctica: determinación de cómo cada una de las corridas especificadas en el diseño experimental debe ejecutarse.

11. Experimentación: ejecución de la simulación para generar los datos deseados y llevar a cabo el análisis de sensibilidad.

12. Interpretación: obtención de conclusiones a partir de los datos generados mediante la simulación.

13. Implementación: poner el modelo y/o resultados en uso.

14. Documentación: registrar las actividades del proyecto y los resultados documentando así el modelo y su utilización.

Indiscutiblemente, entre las partes más difíciles y críticas de cualquier estudio que emplee simulación se encuentran la definición del sistema y la fase de formulación del modelo. El éxito o fracaso en etapas posteriores dependerá en gran parte de qué tan bien se haya definido el problema, establecido los objetivos del estudio, definido los límites del sistema, determinado los componentes y variables de relevancia, hecho las hipótesis y abstraído las relaciones entre los componentes y las variables y estimado los valores de los parámetros pertinentes. Por otro lado, una vez construido el

modelo, uno de los problemas más serios que presenta un estudio de simulación, sobretodo si es por computadora, es precisamente el darle validez al mismo.

### 3.1.3.1 Programas de Diagramas de Flujo.

Para complementar algunas de las ideas y definiciones básicas del acercamiento modular en la simulación a régimen permanente, en esta sección se incluyen aspectos acerca de las corrientes, los diagramas de flujo y su representación numérica.

#### Variables de las Corrientes

En la técnica de simulación bajo estudio se dice que la información fluye entre las U. C. en corrientes, generalmente en el mismo sentido del flujo de materia, por lo que se les considera portadoras de información. Cuando se requiere, la lista de corrientes corresponde a una relación ordenada de números referentes a la información que describe el flujo de una corriente en particular.

En la Tabla 3.1 se ilustran las variables de las corrientes (cantidades necesarias para especificar el estado de la corriente) más importantes y útiles para proyectos estudiantiles de simulación y diseño de plantas químicas. Nótese que a partir del sexto elemento los flujos de los componentes se listan en el orden en que fueron definidos y que, por lo tanto, será una lista de final abierto ya que depende del número de componentes presentes. Para simulaciones más complejas también podrían introducirse otros datos como entalpía, fracción de vapor, temperatura de algún medio de enfriamiento, etc.

Las unidades usadas para los flujos, la temperatura y la presión son arbitrarias ya que dependen

Tabla 3.1

FORMATO PARA EL FLUJO DE INFORMACION EN UNA CORRIENTE

1. Número de la corriente\*
2. Bandera de la corriente\*\*
3. Flujo total  
 $\left(\frac{\text{lbmol}}{\text{hr}}\right)$
4. Temperatura  
(°F)
5. Presión  
(psias)
6. Flujo del componente 1  
 $\left(\frac{\text{lbmol}}{\text{hr}}\right)$
7. Flujo del componente 2  
 $\left(\frac{\text{lbmol}}{\text{hr}}\right)$
8. Flujo del componente 3  
 $\left(\frac{\text{lbmol}}{\text{hr}}\right)$
9. Etc.

Notas:

\* Corresponde a un esquema de numeración arbitrario usado en el diagrama de flujo de información.

\*\* Indica el tipo de la corriente; en nuestra versión se le utiliza para definir la fase de la corriente.

de los datos disponibles pero deberán ser consistentes; no obstante, al considerar las reacciones químicas del proceso y cuando éste involucra sistemas gaseosos resulta ventajoso manejar los flujos en  $\frac{\text{moles}}{\text{unidad de tiempo}}$ .

Las unidades computacionales deberán ser compatibles con sus listas de corrientes asociadas y mientras que en los ejemplos de esta tesis se utilizará este formato, ésto no es necesario. En el caso en que las banderas se empleen para llevar un número de código que sirva para identificarlas, se les podrá usar para verificar que las corrientes que entren y salgan de cada U. C. sean las correctas.

Una corriente se considerará "conocida" cuando se trate de una alimentación, cuando sus variables se hayan calculado como salida de otra unidad o cuando se haya asumido temporalmente su valor.

#### Diagramas de Flujo de Información y su Representación Numérica

El diagrama de flujo de proceso (Figura 3.3), diagrama que corresponde al equipo' y las líneas de tubería que constituyen la planta y en donde éstas últimas se representan mediante flechas que indican la dirección del flujo de masa, puede codificarse numéricamente para su uso en una computadora. Esto se logra en dos pasos: primero se construye el diagrama de flujo de información análogo utilizando un conjunto apropiado de símbolos para representar las diversas U. C. y después se le pone en forma numérica.

Al construir un diagrama de flujo de información (Figura 3.4) suele existir una correspondencia uno a uno entre las unidades computacionales y las etapas del proceso. El arreglo general de los símbolos



Figura 3.3

DIAGRAMA DE FLUJO DE PROCESO DE UN PROCESO PETROQUIMICO

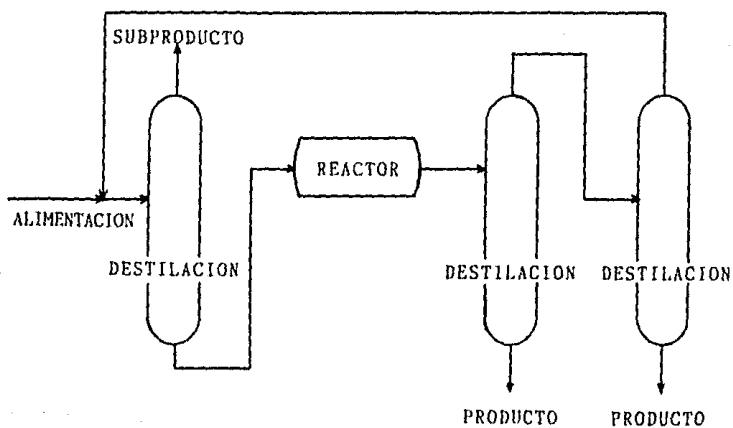
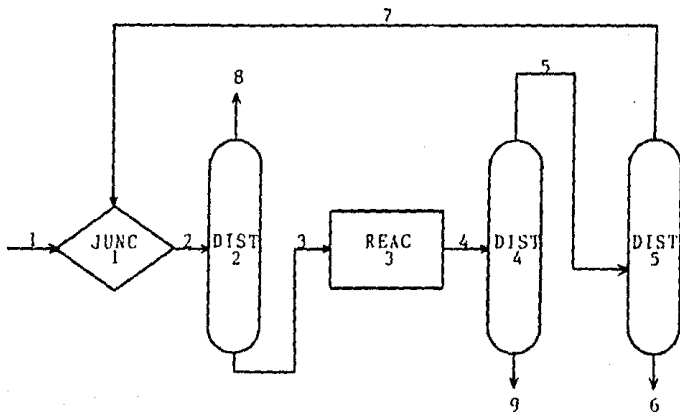


Figura 3.4

DIAGRAMA DE FLUJO DE INFORMACION CORRESPONDIENTE A LA FIGURA 3.3



de los módulos será pues similar al que aparece en el diagrama de flujo del proceso y en él las corrientes de flujo de información se dibujan con flechas que muestran la dirección del flujo de información más importante.

Cada símbolo empleado recibe un nombre que deberá corresponder a la subrutina módulo que represente y que deberá estar disponible en la librería del sistema de simulación.

Las corrientes y los símbolos se numeran por separado y aunque dicha numeración es arbitraria se sugiere que se haga en forma ascendente siguiendo la dirección del flujo de información principal y, por supuesto, evitando repeticiones.

Para facilitar el aspecto computacional, el diagrama de flujo de información se codifica normalmente en forma numérica. El motivo de esta codificación es que se estima que una computadora digital no es capaz de interpretar una representación gráfica (7). Existen varias alternativas para representar dicho diagrama numéricamente de las que se considera la más conveniente la matriz de proceso (Tabla 3.2) por contener toda la información del mismo.

La matriz de proceso consiste en una lista de las U. C. que corresponden a los símbolos en el diagrama de flujo de información ordenadas, por lo general, numéricamente. Dentro de esta matriz, para cada módulo de proceso existe un renglón que contiene su número en el diagrama, su nombre (o su número de código) y los números de las corrientes de entrada (positivas) seguidos de los números de las corrientes de salida (negativas); sin olvidar que estas corrientes pueden ser únicas.

Cuando, como a menudo sucede, dos ó más corrientes

Tabla 3.2

MATRIZ DE PROCESO PARA EL PROCESO DE LA FIGURA 3.4

Equipo		Corrientes Asociadas		
Número	Nombre*			
1	JUNC	<del>1</del>	<del>7</del>	-2
2	DIST	2	-8	-3
3	REAC	3	-4	
4	DIST	4	-5	-9
5	DIST	5	-7	-6

Notas:

En DIST, la primer corriente de salida corresponde al domo y la segunda a los fondos.

\* En GEMCS se utiliza un código numérico equivalente a los nombres de los módulos.

entran a o salen de un módulo de cálculo, el orden de los números que las representan en una fila de la matriz de proceso es importante para la transferencia de información entre U. C. por lo que se han elaborado reglas sencillas relacionadas con el ordenamiento de estas corrientes. Por lo tanto, la matriz de proceso incluye un "ordenamiento local" de las corrientes sencillo pero satisfactorio. Por ejemplo, en un intercambiador de calor la primer entrada y salida corresponderían al fluido de proceso mientras que las segundas al fluido intercambiador de calor; en los procesos de destilación, la primer salida representa al producto (destilado) mientras que la segunda al producto en los fondos (residuo).

La matriz de proceso descrita es probablemente la representación numérica más simple de los arreglos entre las unidades de manufactura de un proceso en base a una lista de sus módulos y de las corrientes asociadas que los unen. Así, por contener su estructura completa, es posible reconstruir un diagrama de flujo de información a partir de su matriz de proceso. Una ventaja más es que el ingeniero puede leerla y entenderla fácilmente por lo que se le puede checar y modificar sin mayores dificultades.

Las otras representaciones numéricas del diagrama de flujo de información, como son la matriz de conexión de corrientes, la matriz de adyacencia y la matriz de incidencia que no se discuten en este trabajo, se pueden derivar de la matriz de proceso (1).

#### 3.1.4 Errores Factibles dentro de la Simulación.

Al usar la simulación para estudiar un sistema complejo pueden presentarse diferentes clases de errores,

cada uno de los cuales puede llevarnos a una conclusión equivocada; por lo tanto, la exactitud de la resolución total del problema dependerá de:

1. Errores en el Diseño: los problemas de los libros, generalmente, se preparan de manera que contengan la información necesaria para su solución; en la práctica, los errores al especificar las variables de diseño pueden generar soluciones inconsistentes o físicamente imposibles o bien, a veces, la solución no existe.

2. Conformidad de las Ecuaciones del Modelo Matemático con el Problema Original: los errores en esta etapa pueden ser de cualquier tamaño; frecuentemente los sistemas no lineales se aproximan mediante ecuaciones lineales mientras que en otros casos, el problema no se ha entendido completamente por lo que la formulación se encuentra incompleta.

3. Conformidad del Modelo Teórico de la Computadora con las Ecuaciones Matemáticas: pueden presentarse errores de programación en esta etapa si la computadora es capaz (teóricamente) de llevar a cabo las operaciones matemáticas necesarias. Sin embargo algunas de ellas son difíciles de simular, particularmente por el "hard-ware" analógico, por lo que puede necesitarse de alguna forma de aproximación entre el modelo de computadora y las ecuaciones del modelo.

4. Errores en la Solución del Modelo por Computadora: éstos dependen en forma compleja de la exactitud de las unidades analógicas usadas y la velocidad de operación de las mismas o del tamaño del paso empleado en la integración digital. Estos errores pueden ir desde 0.01% ó menos hasta 1% ó más dependiendo del método, equipo, tamaño y naturaleza del problema.

5. Errores en los Datos Utilizados o en el Uso del Modelo.

6. Errores en la Toma de Resultados: las soluciones de la computadora analógica deben de medirse y/o registrarse por lo que estos errores dependerán del tipo de equipo disponible para la medición y pueden ir desde 0.01% para voltímetros digitales y registradores de datos hasta 10%, por lo menos, para osciladores pobres.

El error global resulta ser una función compleja del sistema, la computadora y el equipo de registro. Las computadoras analógicas no son dispositivos de alta precisión pero son lo suficientemente buenas para muchos de los problemas que se necesitan estudiar por simulación computarizada. De hecho, los errores ocasionados por el registro de resultados o por la computadora pueden ser menores en un orden que los errores asociados con la formulación del problema. Las computadoras digitales también introducen errores en la solución de problemas que involucran ecuaciones diferenciales debidos a la longitud en el paso de integración y a la aproximación numérica. En algunos casos estos errores pueden exceder a aquellos involucrados en la simulación analógica si no se tiene cuidado al seleccionar la técnica de integración a emplear y la longitud del paso de integración de la misma.

Además de difícil, a menudo la parte menos exacta del estudio de simulación es la formulación de las ecuaciones matemáticas del modelo. Por lo mismo, antes de publicar los resultados es conveniente suprimir las cifras innecesarias o carentes de significado. Esta es una de las ventajas de presentar los resultados en forma gráfica ya que las gráficas sencillas pueden leerse casi con la misma precisión que la obtenida en muchos estudios de simulación que involucran sistemas dinámicos.

Finalmente, es importante mencionar que si uno es hábil y cuidadoso en el diseño del modelo de la simulación y se le desarrolla de manera modular, conforme se adquiriera una mayor experiencia del sistema, se le podrán hacer las correcciones y mejoras necesarias. A pesar de ello, entre más errores se cometan inicialmente, más complicadas y costosas serán las modificaciones posteriores.

### 3.2 ACTIVIDADES RELACIONADAS CON LA SIMULACION.

#### 3.2.1 Optimización (Investigación de Operaciones).

El conjunto de técnicas matemáticas que proporciona los medios para determinar normas eficaces con la mayor eficiencia posible suele agruparse, en forma bastante amplia, bajo el nombre de investigación de operaciones.

La simulación puede considerarse un método de optimización, comprendido dentro de las mencionadas técnicas de investigación de operaciones, cuya finalidad es reconocer que existe un gran número de maneras de lograr algo y que entre ellas hay una que es la mejor tomando en cuenta la medida de efectividad elegida.

Consideremos ahora el caso de la optimización de la simulación de una planta. Después de que se ha completado la simulación, ésta puede usarse para perfeccionar el comportamiento de la planta; esta búsqueda del mejor comportamiento de acuerdo a ciertos criterios de beneficio y sujeta a determinadas restricciones en cuanto a los posibles valores de las variables se conoce como optimización.

Para la optimización de un proceso deberán tomarse cuatro decisiones principales:

1. Elegir el criterio de comportamiento, función objetivo, que se espera mejorar.

2. Seleccionar las variables que cambiarán en forma independiente durante la optimización.

3. Especificar las restricciones que existen en las magnitudes de las variables del proceso.



4. Elegir el método de optimización.

3.2.2 Diseño, Investigación y Operación.

Se podría sugerir que el diseño se refiere a ideas, planes y dibujos ingenieriles para plantas y equipo por construirse mientras que la simulación es un análisis de una planta existente. Esta definición es demasiado restrictiva ya que las técnicas de simulación también se utilizan para diseñar. En vez de eso, el diseño se podría definir como el dimensionamiento del dispositivo para lograr el cambio requerido a la salida en las variables de las corrientes de entrada en tanto que en la simulación se calculan las variables de las corrientes de salida dados los valores de las de entrada y el comportamiento del dispositivo.

En otras palabras, el diseño es un problema con valores en la frontera a diferencia de la simulación que es el problema análogo con valores iniciales (ver Figura 3.5). En el diseño se responde a preguntas acerca del equipo mientras que en la simulación, acerca de las corrientes (1).

Existen algunas similitudes entre la estrategia seguida para el diseño y la seguida para la simulación. En el caso de la primera es necesario efectuar balances de materia y energía para poder dimensionar el equipo pudiendo emplear para ello la simulación; incluso en la etapa de definición del tamaño del equipo se pueden emplear técnicas de simulación. Así, aunque el diseño y la simulación son diferentes, se encuentran íntimamente relacionados.

Figura 3.5

DIFERENCIAS ENTRE EL DISEÑO Y LA SIMULACION

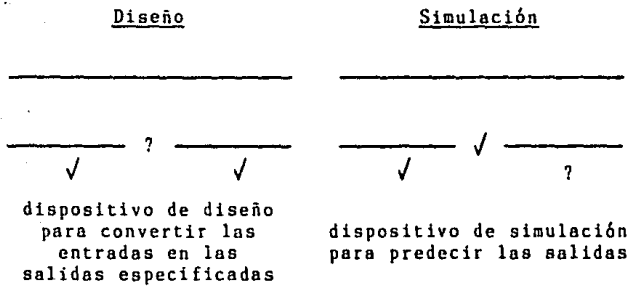
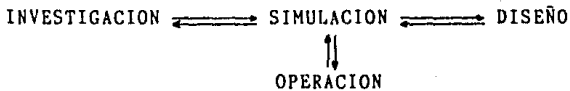


Figura 3.6

RELACION ENTRE LAS ACTIVIDADES INGENIERILES



Para concluir podría decirse que la simulación representa un nexo sumamente importante entre tres actividades ingenieriles tradicionales: operación, diseño e investigación (Figura 3.6); nexo que también debe existir entre la gente y los departamentos de cualquier organización (1).

### 3.3 EL METODO SECUENCIAL ITERATIVO PARA EL CALCULO DE PROBLEMAS CON RECIRCULACION - METODO DE CONVERGENCIA.

Como se mencionó en la introducción, muchos procesos químicos presentan recirculación de materia, energía y/o información; ésto es, la(s) salida(s) de una etapa forma(n) parte de las entradas a la misma ya sea directamente o después de pasar a través de otra(s) etapa(s). Esta retroalimentación puede presentarse en varios puntos del proceso.

Al estudiar el diagrama de flujo de información de un proceso se puede elaborar una secuencia de cálculo factible; para plantas pequeñas que contienen unas cuantas corrientes (entre 20 y 30) es posible obtener este orden de cálculo por simple inspección. La necesidad de un programa de secuenciación automática surge cuando los usuarios disponen de muy poco tiempo para encontrar dicho orden de cálculo o no están lo suficientemente familiarizados con la elaboración de diagramas de flujo y sus conceptos como para entender los requerimientos de la secuencia de cálculo de las unidades y, sobretodo, cuando la determinación del orden de cálculo se plantea difícil por la presencia de varios ciclos de recirculación anidados (8) (9).

Los métodos más útiles en la obtención de secuencias de cálculo consisten en localizar las corrientes por suponer inicialmente y seguir las a través de la planta para obtener la secuencia factible. Hay que notar que el orden de cálculo óptimo corresponde a aquel en que se minimiza el número total de corrientes que se deben suponer para poder comenzar los cálculos del sistema y cuyos valores deberán converger en la respuesta final para lograr la "verdadera" simulación de la planta (10).

El programa GEMCS emplea un procedimiento secuencial iterativo para resolver los problemas que presentan recirculaciones. Para ello el usuario del programa especifica el orden en que se pasa de una U. C. a otra arreglando también que, al término de un ciclo de recirculación, el programa regrese a la primer unidad en el orden de cálculo de dicho ciclo para repetir la secuencia. De esta forma puede lograr que la información sea reprocesada hasta que, a su satisfacción, no existan más cambios y se hayan cumplido las condiciones planteadas por todos los módulos en el diagrama de flujo de información.

Obviamente, para determinar cuando deberá detenerse el procedimiento iterativo se requiere de una prueba. En GEMCS el usuario selecciona una de las corrientes dentro del ciclo en base a sus conocimientos del proceso, generalmente alguna corriente de la que cuenta con información de la planta para confrontar los resultados de su modelo, y el programa, después de haber ejecutado todas las unidades que componen el ciclo, compara los nuevos valores de la corriente seleccionada con los obtenidos en la iteración anterior. Si el valor absoluto del cambio fraccional,

$$\text{cambio fraccional} = \frac{\text{nuevo valor} - \text{valor antiguo}}{\text{nuevo valor}},$$

para cada elemento de la corriente es menor que la tolerancia especificada, entonces se dice que el procedimiento secuencial iterativo convergió.

La convergencia se define como el estado alcanzado en el proceso cuando el cambio fraccional absoluto en las variables prescritas en las corrientes de flujo de información que conectan a los módulos es menor que cierta tolerancia fijada de antemano.

Como la mayoría de los esquemas de programación

modular, GEMCS utiliza la iteración directa para encontrar el conjunto de variables que satisfagan a todos los módulos en el ciclo de recirculación (11). Este método, también llamado de substituciones sucesivas, es el más sencillo para lograr la convergencia y consiste en considerar el valor calculado como nueva suposición para la siguiente iteración:

$$S_{i+1} = C_i$$

en donde  $S_i$  es el valor supuesto en la iteración  $i$  y  $C_i$ , el calculado y empleado para la siguiente iteración.

En 1974, Shacham y Motard (12) demostraron que el método de substituciones sucesivas es el más estable pues converge para cualquier sistema con recirculaciones, físicamente realizable, siempre que el valor inicial estimado se encuentre cercano a la solución. Su desventaja radica en que en algunos casos puede ser lento, en especial cuando dichos problemas con recirculación requieren de muchas iteraciones a lo largo de la secuencia de cálculo para aproximarse a los valores de convergencia, lo que implicaría un gran consumo tanto de tiempo de computación como de memoria (1) (13) (14).

Referencias.

1. Crowe, C. M. et al.; "Chemical Plant Simulation", [McMaster University Press, Canada (1969) & Prentice Hall, U. S. A. (1971)].
2. Kehat, E. & Shacham, M.; "Chemical Process Simulation Programs - 1", [Process Technol., 18 (1973a), No. 1/2, pp. 35-40].
3. Shannon, R. E.; "Systems Simulation", [Prentice Hall, U. S. A. (1975)].
4. Ord-Smith, R. J. & Stephenson, J.; "Computer Simulation of Continuous Systems", [Cambridge University Press, England (1975)].
5. Ramirez, F. W.; "Process Simulation", [D. C. Heath and Company, U. S. A. (1976)].
6. Graybeal, W. J. & Pooch, U. W.; "Simulation: Principles and Methods", [Winthrop Publishers, U. S. A. (1980)].
7. Johnson, A. I.; Peters, N. & Associates; "GEMCS/CYBER 73", [S. A. C. D. A., University of Western Ontario, Canada (1979)].
8. Lee, W. et al.; "Design Variable Selection to Simplify Process Calculations", [A. I. Ch. E. Journal, 12 (1966), No. 6, pp. 1104-1110].
9. Lee, W. & Rudd, D. F.; "On the Ordering of Recycle Calculations", [A. I. Ch. E. Journal, 12 (1966), No. 6, pp. 1184-1190].
10. Forder, G. J. & Hutchinson, H. P.; "The Analysis of Chemical Plant Flowsheets", [Chem. Eng. Sci., 24 (1969), pp. 771-785].
11. Kesler, M. G. & Griffiths, P. R.; "A Computer System for Process Simulation", [Proc. Am. Petrol. Inst., Div. of Refining, 43 (1963), sec. III, pp. 49-56].
12. Shacham, M. & Motard, R. L.; "Application of the Theory of Linear Recycle Systems", [A. I. Ch. E. 78<sup>th</sup> National Meeting, U. S. A. (Salt Lake City, 1974)].
13. Borden, R. L. et al.; "Numerical Analysis", [Prindle, Weber & Schmidt, U. S. A. (1981)].
14. Cavett, R. H.; "Application of Numerical Methods to the Convergence of Simulated Processes Involving Recycle Loops", [Proc. Am. Petr. Inst., Div. of Refining, 43 (1963), sec. III, pp 57-76].

## CAPITULO IV

### G E M C S

El programa que se describe en esta tesis fue escrito originalmente como herramienta para la enseñanza del curso de Ingeniería de Sistemas en la Universidad de McMaster; sin embargo, debido a su flexibilidad, tamaño y facilidad de uso, encontró amplia aceptación en muchas instituciones académicas canadienses y europeas así como en la industria en donde pasó a formar la base de sistemas computacionales más elaborados.

En esta forma el programa de simulación de procesos GEMCS se convirtió en un sistema de computación en ingeniería y administración, simple pero general, utilizado para el análisis, diseño y optimización de procesos industriales que operan a régimen permanente.

Su ventaja sobre los programas de simulación convencionales (que son grandes y de estructura fija) radica sobre todo en su flexibilidad, consecuencia directa del acercamiento modular, y en el hecho de que cualquier usuario puede aprender todo acerca de su código y lógica en unas cuantas horas pudiendo modificarlo después para aplicarlo a una amplia variedad de problemas.

Un sistema simulador modular puede dividirse en dos tipos de programas:

- a) programas ejecutivos      y      b) programas de soporte.

Los programas ejecutivos controlan la entrada y salida de información, la secuencia de ejecución y, en algunos casos, el cálculo de recirculaciones pudiendo forzar la convergencia.



Los programas de soporte están constituidos por los módulos de proceso y el paquete de propiedades termodinámicas (cuando se cuenta con éste).

Mientras que los programas ejecutivos del sistema controlan el esquema global del cálculo de un proceso simulado, la simulación específica de los equipos es efectuada por las unidades computacionales.

El sistema GEMCS consta de un programa principal o maestro que gobierna y ejecuta cada una de las secciones del programa general y de una librería de módulos de proceso. La subrutina que llama a las diferentes unidades computacionales que representan parte del proceso simulado, las subrutinas que almacenan los datos de las corrientes y los parámetros del equipo y la subrutina que maneja las pruebas de convergencia en el caso de la presencia de recirculaciones son inherentes al programa y le permiten operar independientemente del proceso que se vaya a simular. Por otro lado, la librería de U. C. comprende subrutinas módulo representativas de diferentes operaciones unitarias.

Gracias a su modularidad, al ejecutarse, el programa se puede integrar de manera que todos los equipos operen simultáneamente o bien, se pueden omitir los equipos que no intervengan en el proceso sin necesidad de modificar el programa principal o las subrutinas que no se vayan a emplear.

#### 4.1 EL PROGRAMA MAESTRO.

##### 4.1.1 Breve Descripción.

La versión de GEMCS aquí descrita es una adapta

ción del programa original para poder utilizarlo en las microcomputadoras digitales de la Universidad La Salle (Franklin ACE 1000 y 1200 y Apple II).

El programa maestro de GEMCS es el responsable del almacenamiento y manejo de la información relacionada con la simulación de una planta en particular así como de llamar y ejecutar, en el orden prescrito, a las unidades computacionales necesarias para llevar a cabo el estudio en la forma deseada.

Para lograrlo, MASTER GEMCS se apoya en las siguientes cuatro subrutinas del sistema:

DLOAD1: para leer la información del archivo de datos;

STREAM: para localizar la información que describe a las corrientes de la planta dentro de la matriz de corrientes acumulativa;

DISKIO: para transferir los parámetros del equipo al vector de almacenamiento principal o viceversa y

MODULE: para llamar a las unidades computacionales apropiadas.

Todas ellas se describen con detalle más adelante.

A continuación se listan algunas de las características deseables en un programa maestro y que se encuentran presentes en GEMCS:

a) El sistema debe ser flexible y, en lo que se refiere al tamaño de la planta, de final abierto (número de equipos en el proceso fijado por el usuario) de modo que la simulación se maneje en base a la información contenida en las variables de las corrientes y

los parámetros de equipo.

b) El chequeo de los datos que representan a la planta antes y durante la ejecución del programa deberá ser sencillo para el usuario de manera que pueda detectar y rastrear errores si éstos llegaran a presentarse.

c) Para poder estudiar modos de operación alternativos de una planta, los parámetros de las unidades computacionales deberán poder alterarse sin dificultad.

d) Los resultados de la simulación deberán ser fáciles de interpretar.

#### 4.1.2 Variables del Programa Maestro.

La siguiente es una relación alfabética de las variables usadas por MASTER GEMCS; las señaladas con un asterisco pueden ser redefinidas, para utilizarlas con otros propósitos en las subrutinas que corresponden a las unidades computacionales, sin afectar los cálculos que se efectúan dentro del programa principal.

EEN(N)	Enésimo elemento en la lista acumulativa de parámetros de equipo.
EN(N)	Enésimo elemento en la lista de parámetros de cierto equipo.
I*, IKE	Número o subíndice de la corriente.
III	Número máximo de corrientes que puede almacenarse en SN (en este caso 17).
IS*	Posición de una corriente en la matriz SN (renglón).
J*	Índice del elemento en una corriente.
JJ	Longitud del arreglo de corrientes en función del número de elementos que lo describen; corresponde a NOCOMP+5

pero como se le fija en la subrutina DLOAD1, el usuario la puede modificar si utiliza una descripción diferente de las corrientes.

- KPRINT Control de impresión de resultados.
- LLST(NC) Número de equipo del equipo NC en la lista de orden de cálculo LLST.
- LOOP Número de la iteración, se emplea cuando el proceso presenta recirculaciones; LOOP = 999 en problemas de resolución directa.
- MM Ver NE.
- NC Subíndice de la lista de secuencia de ejecución LLST.
- NCALC Número total de equipos en LLST.
- NCOUNT Punto de referencia en la ubicación de los parámetros de equipo en el vector EEN.
- NE Número del equipo en el diagrama de flujo de información, corresponde a EN(1) o a LLST(NC).
- NIN Número de corrientes de entrada a un equipo, corresponde a EN(6).
- NN Longitud de la lista de parámetros de equipo, corresponde a EN(3).
- NOCOMP Número de componentes de flujo en el sistema.
- NOUT Número de corrientes de salida de un equipo, corresponde a EN(11).
- NTYPE Número del tipo de equipo (código), corresponde a EN(2).

- S Corriente a localizar en la matriz SN.
- SI(I,J) Información en la corriente de entrada I para el elemento J.
- SO(I,J) Información en la corriente de salida I para el elemento J.
- SN(I,J) En la matriz de corrientes acumulativa, información en la corriente I para para el elemento J.

#### 4.1.3 Bases de Datos.

Cada base de datos corresponde a una sencilla descripción de la planta a simular por lo que incluye:

- 1) el número de componentes que se manejarán;
- 2) el número y el orden en que se calcularán los módulos de proceso (especificado por el usuario en esta versión);
- 3) información acerca de la(s) corriente(s) de alimentación y de la(s) corriente(s) con datos iniciales por leerse;
- 4) información relativa a las unidades computacionales (comprende el tipo de módulo, sus corrientes asociadas y los parámetros del mismo) y
- 5) propiedades físicas, químicas y/o termodinámicas (cuando se requieren).

Una vez leída, la base de datos, que en GEMCS se almacena en bloques COMMON en el corazón de la computadora, puede imprimirse para verificar su contenido(1).

Originalmente los datos se leían de tarjetas por lo que hubo necesidad de modificar ligeramente el programa para lograr el acceso de GEMCS a un archivo

que tuviera almacenada toda la información. De manera similar, es decir, mediante sencillas modificaciones, los formatos de las variables en la base de datos también pueden adaptarse a las necesidades del usuario.

Las variables utilizadas para alimentar los datos y cuya función se describe con más detalle al hablar de la subrutina DLOAD1 (Sección 4.2.1) son:

EN(N),N=1,EN(3) Parámetros de equipo para cada unidad computacional (ver Tabla 4.1).

ITITLE Título de la simulación.

KPRINT Control de impresión.

LLST(I),I=1,NCALC Secuencia de ejecución.

MSN Número máximo de corrientes en el diagrama de flujo de información.

NCALC Número de equipos por calcular.

NOCOMP Número de componentes en el sistema.

NOE Número de conjuntos de parámetros por leer.

NP Número de propiedades físicas, químicas y/o termodinámicas por leerse.

NSR Número de corrientes con datos iniciales por leer.

PP(N,I),N=1,NP;I=1,NOCOMP Propiedades termodinámicas, químicas y/o físicas para todos los componentes (ver Tabla 4.2).

SN(I,J),I=1,NSR;J=1,JJ Información relativa a las corrientes con datos iniciales (ver Tabla 4.3).

Para facilitar la creación de una base de datos se elaboraron 2 hojas de codificación en donde se indica el orden en que deberán suministrarse dichos datos.







Tabla 4.1

FORMATO PARA LOS CONJUNTOS DE DATOS DE LOS MODULOS EN GEMCS  
(Parámetros de los Equipos en el Proceso Simulado)

N	EN(N)
1	Número del equipo en el diagrama de flujo
2	Tipo o código del equipo (número asignado a la subrutina módulo)
3	Número de parámetros en la lista
4	(Libre para usarse en la subrutina en particular; si no se requiere, utilícese 0.0)
5	(Libre para usarse en la subrutina en particular; si no se requiere, utilícese 0.0)
6	Número de corrientes de entrada ( $\leq 4$ )
7	Números de las corrientes de entrada cuyo
8	orden estará definido por la subrutina módulo
9	(si no se requieren todos utilícese 0.0)
10	
11	Número de corrientes de salida ( $\leq 4$ )
12	Números de las corrientes de salida cuyo
13	orden estará definido por la subrutina módulo
14	(si no se requieren todos utilícese 0.0)
15	
16	De aquí en adelante, los demás parámetros estarán
...	definidos por la subrutina en particular ...
EN(3)	

Tabla 4.2

FORMATO PARA LOS PARAMETROS DE LAS PROPIEDADES EN GEMCS

N	PP(N,I)
1	Tipo de datos de equilibrio vapor-líquido suministrados*
2	Datos de equilibrio vapor-líquido
3	(coeficientes de la función polinomial;
4	si no se requieren todos utilícese 0.0)
5	
6	Tipo de datos suministrados para el cálculo de entalpías en la fase líquida ( $h_L$ )*
7	Datos para el cálculo de $h_L$
8	(coeficientes de la función polinomial;
9	si no se requieren todos utilícese 0.0)
10	
11	Tipo de datos suministrados para el cálculo de entalpías en la fase vapor ( $H_V$ )*
12	Datos para el cálculo de $H_V$
13	(coeficientes de la función polinomial;
14	si no se requieren todos utilícese 0.0)
15	
	* Usese 1.0 para polinomios de grado 3.

Tabla 4.3

FORMATO PARA LOS PARAMETROS DE LAS CORRIENTES EN GEMCS

J	SI(I,J), SN(I,J) y SO(I,J)
1	Número de la corriente
2	Fase de la corriente; úsese 1.0 si es vapor y 2.0 si es líquida
3	Flujo total
4	Temperatura (si no se requiere, utilícese 0.0)
5	Presión (si no se requiere, utilícese 0.0)
6	Flujo del componente 1
7	Flujo del componente 2
8	Flujo del componente 3
.	
.	
.	
JJ	Flujo del componente NOCOMP

#### 4.1.4 Descripción Funcional.

Al iniciarse la simulación, el programa maestro llama a la subrutina DLOAD1, encargada de la lectura de la base de datos, para obtener toda la información necesaria para llevar a cabo el cálculo de las unidades computacionales representativas del proceso. La secuencia a seguir estará en función de la lista de orden de cálculo LLST alimentada en los datos.

Para la ejecución de cada módulo, MASTER GEMCS llama primero a la subrutina DISKIO para transferir el paquete de datos correspondiente a los parámetros de la U. C. del área de almacenamiento acumulativo de todos los datos de los módulos (EEN), en donde estos conjuntos de datos se encuentran apilados cabeza-concola, a un vector EN ("número de equipo"). El vector EEN contiene, entre otros datos, los comprendidos por la matriz de proceso descrita en la Sección 3.1.3.1.

El formato para los parámetros de equipo se muestra en la Tabla 4.1 que está constituida por la información relativa a las corrientes asociadas a la U. C. y los parámetros requeridos para su ejecución. El tipo de datos que pueden almacenarse en los arreglos EN se especificará cuando se trate cada subrutina módulo en particular.

A partir del vector EN, el programa maestro obtiene el número de corrientes de entrada (NIN) y de salida (NOUT) relacionadas con la U. C. así como los números de las corrientes que entran a y salen de la misma. Entonces MASTER GEMCS llama a la subrutina STREAM para que copie la información relativa a cada una de las NIN corrientes de entrada en la matriz SI en la que el primer renglón

lo ocupa la lista de datos de la primer corriente de entrada, la segunda fila contiene la lista de la segunda corriente y así sucesivamente según se hayan alimentado al vector EN.

Tradicionalmente, el programa GEMCS está dimensionado para manejar hasta cuatro corrientes de entrada y cuatro de salida para cada módulo pero, en caso de necesitarse, podría modificarse dicho dimensionamiento.

Una vez ejecutada la unidad computacional, la matriz SO contendrá la información acerca de las corrientes de salida en forma similar a la estipulada para la matriz SI. La información para todas las corrientes se almacena en la matriz de corrientes acumulativa SN.

Usualmente todas las corrientes del proceso a simular se adhieren a un formato consistente. Así, para los ejemplos de esta tesis el formato seguido y recomendado para uso futuro se describe en la Tabla 4.3; de acuerdo con él, cada corriente está representada por cinco números más los flujos de los componentes que intervienen en el proceso y puesto que el número de éstos corresponde a NOCOMP, la longitud de cada arreglo de corrientes estará dada por  $JJ = NOCOMP + 5$ .

Los flujos de los componentes pueden ser másicos o fraccionales (en unidades molares cuando se usan ciertos paquetes de propiedades en conjunción con GEMCS) pudiendo tener cualesquiera unidades.

Después de que la información acerca de las corrientes es copiada en la matriz SI, el programa maestro llama a la subrutina MODULE pasándose el tipo de módulo (NT) que modela a la unidad. MODULE a su vez llama

a la subrutina de la U. C. que simulará el equipo; ésta llevará a cabo las operaciones planteadas y almacenará las corrientes resultantes en la matriz SO.

Al regresar al programa principal, éste llama nuevamente a la subrutina STREAM para que guarde los valores de las corrientes de salida en la matriz de corrientes central SN.

A continuación se busca la siguiente posición dentro de la lista de orden de cálculo para obtener el número de la unidad que corresponde al próximo módulo por ejecutar repitiéndose el procedimiento anterior hasta que todos los equipos en el proceso se hayan simulado.

Cuando en dicho proceso se presentan recirculaciones, al final de cada ciclo se llamará a la subrutina CONTL1 que en la primer iteración únicamente registra la información de la corriente de prueba y regresa el señalador del orden de cálculo al primer módulo en el mismo para que se repitan los cálculos. En las iteraciones siguientes, este módulo de control también examinará el cambio fraccional en la corriente de prueba seleccionada y parará automáticamente el proceso de cálculo iterativo cuando todos los flujos de ésta hayan pasado la prueba de convergencia o al excederse el número máximo de iteraciones preestablecido, en cuyo caso se indicará la no convergencia.

La tolerancia para la convergencia se puede especificar para todos los elementos en la lista de corrientes; en esta versión de GEMCS se usa una sola tolerancia para todos los flujos en la corriente de prueba elegida.

#### 4.1.5 Diagrama de Flujo.

La siguiente figura corresponde a un diagrama de flujo simplificado del programa principal en GEMCS.

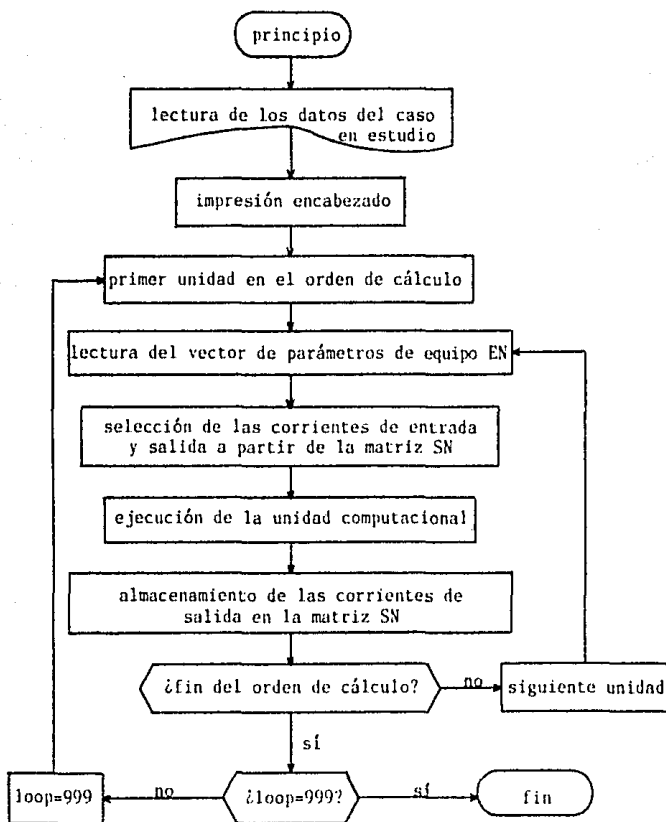


Figura 4.1 PROGRAM MASTER — DIAGRAMA DE FLUJO

Listado del Programa Maestro de GEMCS

```
#USES UFLASH1 IN FLASH1 CODE OVERLAY
#USES USTREAM IN #5:STREAM CODE OVERLAY
#USES UDISK10 IN DISK10 CODE OVERLAY
#USES UJUNC02 IN #5:JUNC02 CODE OVERLAY
#USES USEPA01 IN #5:SEPA01 CODE OVERLAY
#USES USEPA02 IN #5:SEPA02 CODE OVERLAY
#USES USEPA03 IN #5:SEPA03 CODE OVERLAY
```

```
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
```

```
UNIVERSIDAD LA SALLE
ESTUDIO, COMPLEMENTACION Y APLICACION DE UN PROGRAMA DE
SIMULACION DE PROCESOS
TESIS QUE, PARA OPTAR POR EL TITULO DE INGENIERA QUIMICA, PRESENTA
LEONOR DORIS HERRERAMORO GOMEZ
DICIEMBRE, 1984
```

```
PROGRAM MASTER
```

```
C
C
C
C
C
C
C
C
C
C
C
```

```
***
***          PROGRAMA MAESTRO GEMCS          ***
***
```

```
C *** BLOQUE COMMON ***
```

```
C
```

```
CHARACTER*1 RES,REZ
COMMON /SET1/EN(100), JJ, NTN, NOUT, S1(4, 30), S0(4, 30), PP(15, 25)
COMMON /SET2/KPRINT
COMMON /SET3/III, IS, LOOP, NC, NE, SN(17, 30)
COMMON /SET4/LLST(50), NCALC, NCOUNT
COMMON /SET5/NPOINT(50, 2)
COMMON /SET13/EEN(600)
```

```
C
```

```
102 FORMAT(/, 33X, ' ERROR IN SN ')
103 FORMAT(5F14. 3)
104 FORMAT(///, 30X, ' END OF EXECUTION ')
105 FORMAT(5F14. 3)
106 FORMAT(///, 22X, ' EQUIPMENT PARAMETERS FOR UNIT ', I2, /)
107 FORMAT(///, 21X, ' INPUT STREAMS FOR EQUIPMENT UNIT ', I2, /)
109 FORMAT(///, 20X, ' OUTPUT STREAMS FOR EQUIPMENT UNIT ', I2, /)
114 FORMAT(///, 17X, ' FINAL OUTPUT STREAMS FOR EQUIPMENT UNIT ', I2, /)
```

```
C
```

```
CALL HOLA
CALL CONT
30 CALL HOME
```



```
      REZ='*'
      WRITE(*,1)
1  FORMAT(3X,' DESEA CORRER EL PROGRAMA...? (S/N)',3(</>))
      READ(*,2)REZ
2  FORMAT(A1)
      IF(REZ.EQ.'S')GOTO 875
      IF(REZ.EQ.'N')GOTO 456
      IF(REZ.EQ.'*')GOTO 30
      GOTO 30
875 WRITE(*,871)
871 FORMAT(7(</>),3X,' RESULTADOS EN LA IMPRESORA...? (S/N)',3(</>))
      READ(*,872)RES
872 FORMAT(A1)
      IF(RES.EQ.'S')GOTO 873
      IF(RES.EQ.'N')GOTO 874
      GOTO 875
873 OPEN(6,FILE='PRINTER:')
      GOTO 876
874 OPEN(6,FILE='CONSOLE:')
876 OPEN(5,FILE='#5.DATOS')
C
      NCOUNT=0
      III=17
      CALL DLOAD1
C
C *** CALCULANDO EL EQUIPO EN LA LISTA DE ORDEN DE CALCULO ***
C
      NC=1
      707 NE=LLST(NC)
C
C *** LEYENDO EL VECTOR EN ***
C
      MN=NE
      CALL DISKIO(1,MN)
      NN=EN(3)
      NIN=EN(6)
      NOUT=EN(11)
      NTYPE=ABS(EN(2))
      IF(NIN)47,47,42
      42 IF(KPRINT)40,40,711
      711 WRITE(6,106)NE
          WRITE(6,105)(EN(I),1=L,MN)
C
C *** LOCALIZANDO E IMPRIMIENDO LAS CORRIENTES DE ENTRADA ***
C
      40 DO 41 I=1,NIN
          S=EN(I+6)
          CALL STREAM(-S)
          IF(IS-III)43,43,143
      143 SI(1,1)=S
```

```
DO 45 J=2, JJ
45 SI(I, J)=0.0
GOTO 41
43 DO 44 J=1, JJ
44 SI(I, J)=SN(15, J)
41 CONTINUE
IF(KPRINT)714, 714, 713
713 WRITE(6, 107)NE
DO 108 IKE=1, NIN
108 WRITE(6, 105)(SI(IKE, J), J=1, JJ)
714 CONTINUE
C
C *** LOCALIZANDO LAS CORRIENTES DE SALIDA ***
C
DO 46 I=1, NOUT
S=EN(I+1)
CALL STREAM(-S)
IF(III-IS)46, 146, 146
146 CONTINUE
DO 49 J=2, JJ
49 SO(I, J)=SN(15, J)
46 SO(I, 1)=S
C
C *** LLAMANDO A LOS MODULOS DE CALCULO
C (UNIDADES COMPUTACIONALES) ***
C
47 CALL MODULE(HTYPE)
C
C *** ALMACENANDO E IMPRIMIENDO LAS CORRIENTES DE SALIDA ***
C
IF(NOUT)99, 99, 168
168 DO 60 I=1, NOUT
S=EN(I+1)
CALL STREAM(-S)
IF(15-III)62, 62, 171
171 CALL STREAM(0.0)
IF(15-III)62, 62, 163
163 WRITE(6, 102)
GOTO 60
62 DO 63 J=1, JJ
63 SN(15, J)=SO(I, J)
60 CONTINUE
IF(LOOP-999)799, 97, 97
97 WRITE(6, 114)NE
GOTO 110
799 IF(KPRINT)99, 99, 715
715 WRITE(6, 109)NE
110 DO 116 IKE=1, NOUT
116 WRITE(6, 105)(SO(IKE, J), J=1, JJ)
99 NC=NC+1
```

```
IF(NC-NCALC)/707,707,708
708 CONTINUE
C
C *** FINALIZANDO LA SIMULACION ***
C
IF(LOOP-999)/71,70,70
71 NC=0
LOOP=999
KPRINT=1
GOTO 93
70 WRITE(6,104)
456 STOP
END
SUBROUTINE HOLA
WRITE(*,1)
1 FORMAT(7(/),2X,36(1H*),/,2X,'*',34X,'*',/,2X,
' ',34X,'*',/,2X,
' ',7X,'UNIVERSIDAD LA SALLE',7X,'*',/,2X,
' ',34X,'*',/,2X,' ',6X,'PROGRAMA DE SIMULACION',6X,'*',/,2X,
' ',34X,'*',/,2X,' ',8X,' 'GEMCS' ',4X,'*',/,2X,
' ',34X,'*',/,2X,' ',2X,'LEONOR DORIS HERRERAMORO GOMEZ',2X,
' ',/,2X,
' ',34X,'*',/,2X,' ',5X,' INGENIERIA QUIMICA ',7X,'*')
WRITE(*,2)
2 FORMAT(2X,
' ',34X,'*',/,2X,' ',5X,' DICIEMBRE, 1984. ',7X,'*',/,2X,
' ',34X,'*',/,2X,' ',34X,'*',/,2X,36(1H*))
CALL WAIT
RETURN
END
SUBROUTINE WAIT
DO 10 I=1,30
DO 10 J=1,100
10 CONTINUE
RETURN
END
SUBROUTINE CONT
CHARACTER*1 A
WRITE(*,1)
1 FORMAT(5(/),7X,'PRESS <RETURN> TO CONTINUE')
READ(*,2)A
2 FORMAT(A1)
RETURN
END
SUBROUTINE HOME
WRITE(*,1)
1 FORMAT(24(/))
RETURN
END
```

#### 4.2 SUBROUTINAS DEL PROGRAMA MAESTRO.

A continuación se describen la subrutina de lectura de la base de datos, la subrutina de parámetros de equipo y la subrutina de almacenamiento de corrientes; la subrutina de los módulos utilizados en GEMCS se encuentra en la Sección 4.3.3 precediendo las descripciones de éstos.

##### 4.2.1 La Subrutina de Lectura de la Base de Datos DLOADI.

La primer subrutina llamada por MASTER GEMCS es DLOADI cuyo objetivo es leer y transferir en forma directa los datos de la simulación a los lugares de almacenamiento apropiados dentro de los bloques COMMON.

En nuestro caso, lee la información del archivo DATOS que previamente se ha guardado en uno de los diskettes empleados. Los datos almacenados se hallan acomodados de manera que simulan a las tarjetas de la versión original por lo que toda referencia a las mismas será aplicable al dato correspondiente del archivo (una tarjeta equivale a un renglón del archivo).

Como ya se mencionó, el usuario puede crear otras versiones de DLOADI que se adapten a sus necesidades; una opción sería desarrollar una versión que fuera interactiva con el usuario a través del teclado de un teletipo.

Al preparar la base de datos para correr el programa GEMCS deberá tenerse en cuenta que, a excepción de las variables enteras (formato I3), todos los números suministrados deberán ser reales, esto es, deberán incluir punto decimal y no podrán estar en forma exponencial. El formato de lectura empleado por DLOADI es

5F14.3 salvo para el conjunto de parámetros de las propiedades físicas, químicas y/o termodinámicas que utiliza el formato 4F17.11 (ver Sección 6.2).

Para poder otorgar un enunciado a la corrida, una tarjeta de título antecede a los datos numéricos. La variable ITITLE puede contener hasta 72 caracteres empezando en la columna 1 lo que asegura que no se exceda la línea del teletipo cuando el programa se corre en una terminal de esta clase.

A continuación iría la tarjeta que contiene el caracter de control de impresión KPRINT que actúa como un regulador ya que si es uno, se produce una impresión extensa que incluye resultados parciales, comentarios descriptivos adicionales y los datos de entrada lo que permite la tarea de irlos checando conforme van entrando mientras que si es cero, se obtienen únicamente los resultados finales. Por lo mismo, KPRINT deberá adquirir el valor de uno cuando se desee seguir de cerca el desarrollo de la corrida y/o rastrear errores (propósitos de diagnóstico).

La siguiente tarjeta incluye el número de unidades en el orden de cálculo (NCALC), que generalmente corresponde al número de unidades de equipo que integran el proceso, y el número de componentes en el sistema que se está simulando (NOCOMP).

A menos que se use un paquete de propiedades físicas, químicas y termodinámicas que especifique lo contrario, NOCOMP deberá alimentarse como número negativo pues GENCS lo volverá positivo. En esta versión, NCALC no podrá ser mayor de 50 y NOCOMP no podrá exceder 25 a menos que se redimensionen las variables

LLST y NPOINT para aceptar más equipos y las matrices SI, SN y SO para más componentes.

Si NCALC es negativo, la subrutina DLOAD1 asignará a LOOP el valor de 999 y GEMCS efectuará los cálculos de las unidades una sola vez (en caso de solicitarla, CONTLI sería ineficaz) imprimiéndose las corrientes de salida encontradas — problemas de resolución directa. Por el contrario, NCALC deberá ser positivo cuando el proceso contenga recirculaciones.

Después de estas dos porciones de datos, se alimenta el orden de cálculo en una o más tarjetas (cinco datos por tarjeta); para ello se suministran los números de las unidades, que se van almacenando en el arreglo LLST, en el orden en que deberán calcularse. Así, el vector LLST comprende la lista de los números que identifican a los equipos en la secuencia de ejecución.

Siguiendo al orden de cálculo se introducen por separado, es decir, cada uno en su propia tarjeta, el número máximo de corrientes presentes en el diagrama de flujo de información (MSN) y el número de corrientes con datos iniciales por leerse ( $NSR \leq 17$ ).

Las corrientes cuyos datos iniciales van a suministrarse se alimentan entonces, cada una empezando una nueva tarjeta y con cinco datos por tarjeta. Cada corriente estará descrita por  $JJ = NOCOMP + 5$  números por lo que se deberán usar ceros para llenar los espacios de componentes no presentes o de elementos no utilizados en la corriente. Mientras que el orden en que se introducen las corrientes carece de importancia, el formato recomendado para las mismas por considerarse adecuado para la simulación de plantas químicas se detalla en la Tabla 4.3.

La siguiente entrada corresponde al número de conjuntos de parámetros de equipo (NOE) que habrán de leerse y que por lo general es igual al número de unidades en el sistema; no obstante, cuando se requiera, deberá incluir las rutinas de control (por ejemplo, para el caso en que existan recirculaciones en el proceso) y la de generación de reportes (subrutina adicional).

Los parámetros de equipo para cada unidad se suministran siguiendo el formato mostrado en la Tabla 4.1, considerando también cinco por tarjeta. El orden en que se alimentan los conjuntos de parámetros no tiene importancia pero cada uno deberá iniciar una nueva tarjeta. El límite en el número total de parámetros que pueden manejarse en cada simulación es de 600 (arreglo EEN) y cada conjunto de parámetros puede contener hasta 100 de éstos (vector EN). La otra restricción dimensional es que el número de un equipo no puede ser mayor de 50 aunque existan menos de 50 unidades.

Muchos programas de simulación de procesos químicos trabajan en conjunción con paquetes de propiedades físicas, químicas y termodinámicas. Estos paquetes son programas adicionales que efectúan cálculos de las mencionadas propiedades para las corrientes del proceso en base a su composición.

En el caso de las propiedades termodinámicas, comúnmente se les evalúa a partir de correlaciones o ecuaciones de estado que, obviamente, tienen sus limitaciones pues no se pueden predecir las propiedades de cualquier compuesto, sobre todo en la fase líquida debido a condiciones de inmiscibilidad y miscibilidad parcial.

En este trabajo se prefirió alimentar los datos para el cálculo del equilibrio vapor-líquido y de las entalpías, tanto en fase líquida como en fase vapor, por ser éstas las únicas propiedades termodinámicas requeridas y, principalmente, para no hacer uso innecesario del limitado espacio de memoria de las microcomputadoras. De esta manera, las mencionadas propiedades se calculan dentro de las unidades computacionales empleando funciones de declaración aritmética que representan, por ejemplo, la capacidad calorífica como función de la temperatura.

Por lo tanto, es necesario alimentar en una tarjeta el número de parámetros (NP) que, para cada componente, contendrá la matriz de propiedades  $PP(N,I)$ . Dichos parámetros se introducirán, cuatro por tarjeta, siguiendo el orden utilizado para los parámetros de las corrientes e iniciando una nueva tarjeta con cada componente. El formato sugerido se puede apreciar en la Tabla 4.2.

Existe una versión de GEMCS — la elaborada para una computadora CDC CYBER 73 — que incorpora un paquete de propiedades conocido como CIMPP (Comprehensive Industrial Physical Property Package) desarrollado por SACDA para el cálculo de propiedades físicas, químicas y termodinámicas. CIMPP es una librería de subrutinas de computadora, relacionadas con una base de datos, capaz de efectuar más de 150 tipos de cálculos diferentes para cerca de 300 componentes.



Listado de la Subrutina DLOAD1

```
SUBROUTINE DLOAD1
C
C *** LA SUBROUTINA DLOAD1 LEE TODOS LOS DATOS Y LOS ALMACENA EN LOS
C *** ARREGLOS CORRESPONDIENTES
C
C *** CARACTERES DE CONTROL DE IMPRESION ***
C KPRINT=1 IMPRESION INCREMENTADA, INCLUYE RESULTADOS PARCIALES
C Y SE UTILIZA PARA PROPOSITOS DE DIAGNOSTICO
C KPRINT=0 IMPRESION DE RESULTADOS FINALES EXCLUSIVAMENTE
C
C *** BLOQUE COMMON ***
C
CHARACTER *4 ITITLE(18)
COMMON /SET1/EN(100), JJ, NIN, NOUT, SI(4, 30), SO(4, 30), PP(15, 25)
COMMON /SET2/KPRINT
COMMON /SET3/I11, IS, LOOP, NC, NE, SN(17, 30)
COMMON /SET4/LLST(50), NCALC, INCOUNT
COMMON /SET5/NPOINT(50, 2)
COMMON /SET13/EEN(600)
C
100 FORMAT(18A4)
200 FORMAT(//, 79(1H*), //, 40(2H* ), //)
201 FORMAT(///, 4X, 18A4, //, 40(2H* ), //, 79(1H*))
203 FORMAT(//, 36X, ' DATA ', //)
207 FORMAT(/, 26X, ' NUMBER OF COMPONENTS =', I3, ///, 22X,
* ' NUMBER OF UNITS IN THE PLANT =', I3, //)
209 FORMAT(/, 30X, ' CALCULATION ORDER ', //, 10(//, 2X, 5I15))
211 FORMAT(//, 24X, ' INITIAL VALUES FOR ', I2, ' STREAMS ', //)
213 FORMAT(5F14. 3)
219 FORMAT(//, 21X, ' INITIAL VALUES FOR EQUIPMENT UNIT ', I2, //,
*5(F14. 3))
220 FORMAT(2X, SE14. 5)
221 FORMAT(//, 25X, ' NO PHYSICAL PROPERTIES USED ')
222 FORMAT(//, 14X, I3, ' PHYSICAL PROPERTY PARAMETERS FOR EACH
*COMPONENT ', //)
470 FORMAT(5F10. 3)
471 FORMAT(I3)
472 FORMAT(2I3)
475 FORMAT(5I3)
476 FORMAT(4F17. 11)
C
C *** INICIALIZANDO EL VECTOR DE PARAMETROS DE EQUIPO Y LAS MATRICES DE
C PARAMETROS DE LAS CORRIENTES DE ENTRADA, DE SALIDA Y ALMACENADAS ***
C
DO 7070 I=1,100
7070 EN(I)=0.0
DO 7073 I=1,20
```

```
DO 7072 J=1,4
SI(J, I)=0.0
7072 SO(J, I)=0.0
DO 7073 IK=1,17
7073 SN(IK, I)=0.0
C
C *** LEYENDO EL ENCABEZADO DE LA SIMULACION Y EL CARACTER DE CONTROL
C DE IMPRESION ***
C
READ(5, 100)(ITITLE(KJ), KJ=1, 10)
WRITE(6, 201)(ITITLE(JK), JK=1, 10)
READ(5, 471)KPRINT
IF (KPRINT)9, 9, 7
7. WRITE(6, 203)
C
C *** LEYENDO EL NUMERO DE MODULOS Y DE COMPONENTES PRESENTES EN EL
C PROCESO ***
C
9 READ(5, 472)NCALC, NOCOMP
IF(-NCALC)510, 509, 509
509 NCALC=-NCALC
LOOP=999
GOTO 11
510 LOOP=1
11 IF(KPRINT)511, 511, 10
10 WRITE(6, 207)NOCOMP, NCALC
C
C *** LEYENDO EL ORDEN DEL CALCULO ***
C
511 READ(5, 475)(LLST(I), I=1, NCALC)
IF(KPRINT-1)14, 13, 14
13 WRITE(6, 209)(LLST(I), I=1, NCALC)
14 CONTINUE
C
C *** LEYENDO EL NUMERO MAS GRANDE DE CORRIENTES Y LA DESCRIPCION DE
C LAS CORRIENTES DE ENTRADA ***
C
READ(5, 471)MSH
JJ=NOCOMP+5
READ(5, 471)NSR
IF(KPRINT-1)20, 19, 20
19 WRITE(6, 211)NSR
20 DO 530 I=1, NSR
READ(5, 470)(SN(I, J), J=1, JJ)
IF(KPRINT-1)22, 21, 22
21 WRITE(6, 213)(SN(I, J), J=1, JJ)
22 CONTINUE
530 CONTINUE
C
C *** LEYENDO LOS PARAMETROS DE EQUIPO PARA LOS EQUIPOS CON DATOS
C INICIALES ***
```

```
C      READ(5, 471)NOE
      DO 540 I=1, NOE
      DO 541 IZ=1, 100
541  EN(IZ)=0.0
      READ(5, 470)(EN(N), N=1, 5)
      NN=EN(3)
      NNN=EN(1)
      NCOUNT=NCOUNT+NN
      READ(5, 470)(EN(N), N=6, NN)
      IF(KPRINT-1)26, 25, 26
25  WRITE(6, 219)NNN, (EN(N), N=1, NN)
26  MM=EN(1)
      NPOINT(MM, 1)=NCOUNT-NN
      NPOINT(MM, 2)=NN
540  CALL DISKIO(2, MM)
C
C *** LEYENDO LOS PARAMETROS DE LAS PROPIEDADES FISICAS ***
C
      READ(5, 471)NP
      IF(NP. EQ. 0. AND. KPRINT. EQ. 1)WRITE(6, 221)
      IF(NP. EQ. 0)GOTO 543
      WRITE(6, 222)NP
      DO 542 I=1, NCOMP
      READ(5, 476)(PP(N, I), N=1, NP)
      IF(KPRINT. EQ. 1)WRITE(6, 220)(PP(N, I), N=1, NP)
542  CONTINUE
543  IF(KPRINT. EQ. 1)WRITE(6, 200)
      RETURN
      END
```

#### 4.2.2 La Subrutina de Parámetros de Equipo DISKIO.

En GEMCS los datos de los parámetros de los módulos se almacenan como registros en el vector acumulativo EEN. En la Tabla 4.1, que describe el conjunto de datos de las unidades computacionales, el tercer elemento del vector EN es el número total de parámetros, NN, que corresponde también a la longitud del registro guardado en EEN para cada módulo.

Al examinar la subrutina DLOAD1 puede observarse que conforme se lee cada conjunto de datos de un módulo, la variable NCOUNT se ve incrementada por el valor de NN. Además, en la matriz NPOINT(MM,1) se encuentra la localización en el arreglo EEN del principio del registro para el módulo MM mientras que en la matriz NPOINT(MM,2) se halla su longitud, NN.

La subrutina DISKIO(IPNT,MM) registra, para cada unidad de equipo, los datos de sus parámetros en el vector EEN para después leerlos de éste hacia el vector EN cuando se vayan a efectuar los cálculos en la U. C. correspondiente.

Por lo tanto, para el módulo MM,:

si IPNT=1, la subrutina lee del vector EEN hacia el vector EN y

si IPNT=2, los datos se transfieren del vector EN hacia el vector EEN para su adecuada ubicación en este último.

Listado de la Subrutina DISKIO

```
SUBROUTINE DISKIO(IPNT,MM)
C
C *** LA SUBROUTINA DISKIO ALMACENA O EXTRAE EL ARREGLO DE PARAMETROS
C *** "EN" EN O DE EL ARREGLO ACUMULATIVO "EEN"
C
C *** ESTA SUBROUTINA SIMULA EL ALMACENAMIENTO EN DISCO DEL
C VECTOR "EN"; POR LO TANTO,
C SI IPNT = 1.0 LEE DE LA TABLA DE PARAMETROS DEL EQUIPO Y
C SI IPNT = 2.0 ESCRIBE EN DICHA TABLA ***
C
C *** BLOQUE COMMON ***
C
COMMON /SET1/EN(100), JJ, NIN, NOUT, SI(4, 30), SO(4, 30), PP(15, 25)
COMMON /SET5/NPOINT(50, 2)
COMMON /SET13/EEN(600)
C
100 FORMAT(5X, ' DISKIO ERROR IPNT= ', I4, ' MM = ', I4, '/')
MQ=NPOINT(MM, 1)
ML=NPOINT(MM, 2)
IF(IPNT-1)2, 1, 2
1 DO 11, I=1, ML
11 EN(I)=EEN(MQ+I)
GOTO 70
2 IF(IPNT-2)4, 3, 4
3 DO 33 I=1, ML
33 EEN(MQ+I)=EN(I)
GOTO 70
4 WRITE(6, 100)IPNT, MM
70 RETURN
END
```

#### 4.2.3 La Subrutina de Almacenamiento de Corrientes STREAM.

En el arreglo de almacenamiento de corrientes SN, el índice I en SN(I,J) no es el número de la corriente (es decir,  $I \neq SN(I,1)$ ) sino que para obtener una corriente de la matriz SN es necesario buscar SN(I,1) desde I = 1 hasta I = III. Puesto que este procedimiento se utiliza con frecuencia se escribió la subrutina STREAM.

Así, para buscar una corriente de número S se requiere una declaración CALL STREAM(-S) que permita localizarla en SN(IS,J), J=1, JJ en donde IS es la ubicación de la corriente S, en la matriz SN, encontrada por la subrutina. En caso de que la corriente no esté en SN, se imprimirá un mensaje indicando el error.

Listado de la Subrutina STREAM

```

SUBROUTINE STREAM(S)
C
C *** LA SUBROUTINA STREAM LOCALIZA LA POSICION DE UN ARREGLO
C *** DE CORRIENTES EN LA MATRIZ ACUMULATIVA "SN"
C
C *** BLOQUE COMMON ***
C
COMMON /SET3/III, IS, LOOP, NC, NE, SN(17,30)
C
400 FORMAT(5X, ' ERROR:  STREAM NO ', I3, ' NOT IN SN MATRIX')
C
DO 1 KK=1, III
  IS=KK
  IF (ABS(SN(IS,1))-ABS(S))>.001)2,1,1
1 CONTINUE
  IS=IS+1
  IF(S)2,2,3
3 WRITE(6,400)S
2 CONTINUE
  RETURN
END
```

#### 4.3 PROGRAMAS DE SOPORTE.

Los módulos de cálculo en GEMCS tienen una variedad de posibles usos aunque su función principal la desempeñan como modificadores de información.

Así, independientes del programa maestro pero muy relacionadas con él, las unidades computacionales representativas de operaciones unitarias con que cuenta GEMCS y que se analizarán posteriormente son JUNCO1, SEPA01, SEPA02, SEPA03 y REACT1 que representan, respectivamente, un mezclador, tres separadores y un reactor simple.

Además se adicionarán dos nuevas subrutinas: JUNCO2 que corresponde a la solución del balance de materia y el balance de energía para un mezclador que maneja corrientes en la misma fase (Capítulo V, Sección 1) y FLASH1 que simula una destilación flash adiabática para multicomponentes (Capítulo V, Sección 2).

Asociadas con cada módulo de proceso se encuentran las listas de las corrientes de entrada (matriz SI), las listas de las corrientes de salida (matriz SO) y un conjunto de parámetros (vector EN). Normalmente la U. C. recibe las listas de información de las corrientes de entrada y procesa estos datos de acuerdo a sus ecuaciones matemáticas utilizando los parámetros pertinentes para el empleo del módulo en un lugar particular del diagrama de flujo de información. Los resultados de las operaciones llevadas a cabo con estos parámetros suelen aparecer en las listas de información de las corrientes de salida.

Las subrutinas módulo también pueden emplearse para ajustar parámetros en otras U. C. o para generar reportes.



El uso imaginativo de estos bloques de construcción es uno de los desafíos que plantea el acercamiento modular a la resolución de problemas mediante sistemas como GEMCS.

De acuerdo con este principio, un módulo sólo estará interesado en sus corrientes de entrada y salida asociadas por lo que no hará referencia al diagrama de flujo en busca de números de corrientes o de equipo pues es el programa maestro el que se encarga de buscar en dicho diagrama y suministrar los valores de las corrientes de entrada almacenando la información de las corrientes de salida evaluadas.

Los formatos de las listas de corrientes y de equipo estipulados para GEMCS se describieron en la Sección 4.1.3 en donde se estableció que parte de la lista de equipo debe ser definida por el módulo en cuestión. El formato de las corrientes puede redefinirse en cada simulación y aunque todas las U. C. involucradas suelen utilizar el mismo formato, ésto no es necesario.

#### 4.3.1 Código Numérico de los Equipos.

A cada módulo se le asigna, arbitrariamente, un código que también se conoce como tipo de número del equipo y que corresponde al segundo elemento en el vector EN; por ejemplo, JUNCO1, CONTL1 y JUNCO2 tienen códigos 1, 2 y 3 respectivamente.

Un equipo es llamado para su ejecución por la subrutina MODULE haciendo referencia a su código mediante una declaración GO TO condicionada en la que el número de la declaración ramificada es el mismo que el tipo de número del equipo. Si se desea hacer el programa accesible a una nueva subrutina módulo puede

usarse alguna de las etiquetas vacías o bien, deberá insertarse otra declaración GO TO con su etiqueta apropiada.

Se sugiere que cada vez que se emplee el programa GEMCS para resolver cierto problema se revise la subrutina MODULE(NT), que es la encargada de llamar a las demás subrutinas, para confirmar que los códigos de los equipos correspondan a éstos, sobre todo cuando se hayan hecho modificaciones.

#### 4.3.2 Símbología para las Unidades Computacionales.

En un intento por homogeneizar la nomenclatura y mantener cierta consistencia dentro de los módulos de proceso se utilizan los siguientes símbolos:

a) Subíndices:

I	Número de la corriente.
J	Número del elemento de la corriente.
N	Número del elemento del equipo.

b) Variables:

EN(N)	Parámetros del equipo. (incluyendo corrientes asociadas)
JJ	NOCOMP+5
LOOP	Número de la iteración.
NE	Número del equipo.
NIN	Número de corrientes de entrada.
NOCOMP	Número de componentes de flujo.
NOUT	Número de corrientes de salida.
P	Presión.

b) Variables (continuación):

SI(I,J)	Información de la corriente de entrada.
SO(I,J)	Información de la corriente de salida.
T	Temperatura.
X	Fracción mol en la fase líquida.
Y	Fracción mol en la fase vapor.

4.3.3 Nomenclatura para los Módulos en GEMCS.

Por lo general, los nombres de las unidades computacionales constan de cuatro caracteres alfabéticos y dos dígitos. Los primeros corresponden a las cuatro primeras letras de un nombre descriptivo del equipo que se está manejando mientras que los dos números irán aumentando conforme el módulo se vaya haciendo más sofisticado o bien, servirán para indicar la existencia de algunas diferencias entre las subrutinas como es el caso de los separadores.

Cuando los cuatro caracteres alfabéticos no son suficientes para describir el equipo adecuadamente, el primer dígito se substituye por un quinto carácter por lo que el nombre asignado a la subrutina módulo también podrá estar compuesto por cinco letras y un solo número.

Las U. C. que se usan en esta tesis (y que se describen a continuación) junto con el tipo de número del equipo con que se les ha designado se muestran en la Tabla 4.4

Tabla 4.4

MODULOS USADOS EN GEMCS

Código	Nombre de la Unidad Computacional
1	JUNCO1
2	CONTL1
3	JUNCO2
.	
5	SEPA01
.	
.	
.	
12	FLASH1
.	
.	
.	
21	SEPA02
22	SEPA03
23	REACT1

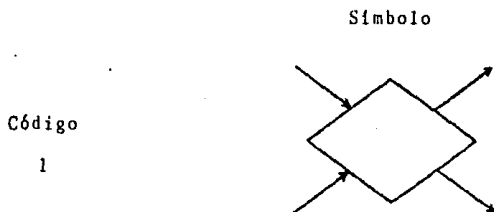
Se recomienda revisar esta lista frecuentemente de manera que conforme se emprendan nuevos estudios se le pueda actualizar al igual que a la subrutina MODULE con la que está íntimamente ligada como puede observarse en el listado de la misma que sigue.

Listado de la Subrutina MODULE

```

SUBROUTINE MODULE(NT)
C
C *** LA SUBROUTINA MODULE LLAMA AL MODULO DE CALCULO (U. C.) APROXIMADO
C *** ADECUADO PARA LA SIMULACION DE LA OPERACION UNITARIA PLANTEADA O
C *** PARA EL CONTROL DE LOS CICLOS DE RECIRCULACION
C
      GOTO (1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21,
      *22, 23), NT
      STOP
1     CALL JUNC01
      RETURN
2     CALL CONTL1
      RETURN
3     CALL JUNC02
      RETURN
4     STOP
5     CALL SEPA01
      RETURN
6     STOP
7     STOP
8     STOP
9     STOP
10    STOP
11    STOP
12    CALL FLASH1
      RETURN
13    STOP
14    STOP
15    STOP
16    STOP
17    STOP
18    STOP
19    STOP
20    STOP
21    CALL SEPA02
      RETURN
22    CALL SEPA03
      RETURN
23    CALL REACT1
      RETURN
      END
```

#### 4.3.4 Módulo de Computación JUNCOI.



Esta subrutina representa un balance de materia muy sencillo sobre un mezclador/separador monofásico que combina NIN corrientes de entrada y divide el flujo resultante en NOUT corrientes nuevas de idéntica composición (NIN y NOUT deben ser menores de cinco).

Aunque la unidad computacional efectúa exclusivamente el balance de materia, permite al usuario fijar la temperatura y la presión de salida. Nótese que, además de la fase de la(s) corriente(s) de salida, en caso de no especificarse, tanto su temperatura como su presión corresponderán a las de la primer corriente de entrada.

Los parámetros que definen a la unidad son el número de éstos {EN(3)}, el número del equipo [EN(1)], su código [EN(2)], el número de corrientes de entrada [EN(6)] seguido por los números de estas corrientes [EN(7) a EN(10)], el número de corrientes de salida [EN(11)] seguido por los números de dichas corrientes [EN(12) a EN(15)], la temperatura de salida [EN(16)], la presión de salida [EN(17)] y la fracción del flujo de entrada total que se desea en cada corriente de salida [EN(18) a EN(21)].

Listado de la Subrutina JUNC01

```

SUBROUTINE JUNC01
C
C *** LA SUBROUTINA JUNC01 COMBINA NIN CORRIENTES DE ENTRADA
C *** REPARTIENDO EL FLUJO RESULTANTE EN NOUT CORRIENTES DE
C *** SALIDA
C
C *** VECTOR DE PARAMETROS DE EQUIPO "EN" ***
C EN( 1) A EN(15) = FORMATO STANDARD
C EN(16)          = TEMPERATURA DE SALIDA
C EN(17)          = PRESION DE SALIDA
C EN(18) A
C EN(17+NOUT) = PROPORCION DEL FLUJO TOTAL DE ENTRADA
C              EN CADA CORRIENTE DE SALIDA
C
C *** BLOQUE COMMON ***
C
C DIMENSION TOTAL(20)
C COMMON /SET1/EN(100), JJ, NIN, NOUT, SI(4, 30), SO(4, 30), PP(15, 25)
C
C *** INICIALIZANDO LOS VALORES DE LAS CORRIENTES DE SALIDA ***
C
C DO 10 I=1, NOUT
C DO 10 J=3, JJ
C SO(I, J)=0
C 10 TOTAL(J)=0
C
C *** SUMANDO TODOS LOS FLUJOS DE ENTRADA PARA CADA COMPONENTE ***
C
C DO 11 J=6, JJ
C DO 11 I=1, NIN
C 11 TOTAL(J)=TOTAL(J)+SI(I, J)
C DO 12 I=1, NOUT
C SO(I, 2)=SI(1, 2)
C IF(EN(16))30, 30, 40
C 30 SO(I, 4)=SI(1, 4)
C GOTO 50
C 40 SO(I, 4)=EN(16)
C 50 IF(EN(17))60, 60, 70
C 60 SO(I, 5)=SI(1, 5)
C GOTO 80
C 70 SO(I, 5)=EN(17)
C
C *** MULTIPLICANDO LOS FLUJOS TOTALES DE LOS COMPONENTES POR SU
C *** FRACCION EN CADA CORRIENTE DE SALIDA ***
C
C 80 DO 12 J=6, JJ
C SO(I, J)=TOTAL(J)*EN(I+17)
C 12 SO(I, 3)=SO(I, 3)+SO(I, J)
C RETURN
C END
```



#### 4.3.5 Módulo de Computación CONTLI.

Símbolo

Código

2



Este módulo es muy importante ya que constituye la unidad de control de las iteraciones en un ciclo (conjunto de módulos que no pueden resolverse sin suponer las propiedades de alguna de las corrientes dentro del mismo) por lo que se usa cuando éstos se presentan debido a la existencia de recirculaciones. El tener que suponer la(s) corriente(s) de corte implica que la solución del problema sea iterativa de ahí la necesidad de contar con un módulo de control.

Para que pueda actuar, se le coloca en la lista de orden de cálculo como el último equipo dentro del ciclo que está controlando.

Su función es evaluar los cambios fraccionarios en los elementos que describen una corriente específica entre dos cálculos sucesivos del ciclo y, considerando su valor absoluto, compararlos contra una tolerancia preestablecida. El cambio fraccional se definió (Sección 3.3) como la diferencia de los elementos descriptivos de la corriente dividida entre el valor más reciente de éstos.

Si todos los elementos descriptivos pasan la prueba de convergencia, los cálculos del ciclo terminan

y GEMCS avanzará a la siguiente unidad dentro de la lista de secuencia de ejecución. En caso contrario, el contador NC que indica el equipo que se está evaluando en el momento es retrocedido al primer equipo dentro del ciclo para así poder reiniciar los cálculos.

Para detener el proceso después de un número razonable de iteraciones y previendo que no se presente la convergencia se especifica un número máximo de iteraciones.

En GEMCS, CONTL1 se trata como un equipo pues aunque no tiene corrientes de entrada ni de salida, controla una corriente elegida por el usuario. El conteo de las iteraciones empieza en el valor almacenado en EN(20) y se va incrementando, de uno en uno, hasta lograr la convergencia o hasta exceder el número máximo de iteraciones EN(17). EN(16) define el número de la corriente que va a ser probada mientras que EN(19) corresponde a la tolerancia fraccional especificada para la convergencia.

Los valores de la corriente bajo prueba se almacenan en el vector EN en las posiciones EN(21) a EN(18+JJ) por lo que cuando se alimenta el conjunto de datos para el módulo CONTL1 es necesario acomodar en el vector EN las  $20 + \text{NOCOMP} + 3$  variables definiéndolas desde el principio.

Listado de la Subrutina CONTL1

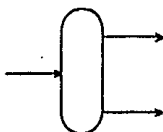
```
SUBROUTINE CONTL1
C
C *** LA SUBROUTINA CONTL1 SE EMPLEA CUANDO SE GENERAN CICLOS POR
C *** LA PRESENCIA DE RECIRCULACIONES PARA CHECAR, EN CADA CICLO,
C *** TODAS LAS VARIABLES DE UNA CORRIENTE ELEGIDA HASTA QUE EL
C *** % DE DIFERENCIA ENTRE DOS ITERACIONES SUCESIVAS ES MENOR
C *** QUE LA TOLERANCIA FIJADA
C
C *** VECTOR "EN" ***
C EN( 1) A EN( 3) = FORMATO STANDARD
C EN( 4)          = CERO
C EN( 5)          = CONTROL DE IMPRESION
C EN( 6) A EN(15) = CEROS
C EN(16)          = NUMERO DE MODULOS CONTROLADOS POR CONTL1
C EN(17)          = NUMERO MAXIMO DE ITERACIONES
C EN(18)          = NUMERO DE LA CORRIENTE DE PRUEBA
C EN(19)          = TOLERANCIA FRACCIONAL
C EN(20)          = NUMERO DE IDENTIFICACION DE LA ITERACION
C
C *** BLOQUE COMMON ***
C
COMMON /SET1/EN(100), JJ, NIN, NOUT, SI(4, 30), SO(4, 30), PP(15, 25)
COMMON /SET2/KPRINT
COMMON /SET3/III, IS, LOOP, NC, NE, SN(17, 30)
C
102 FORMAT(11X, ' FRACTIONAL CHANGE IN TOTAL FLOW IN STREAM ', I2,
* ' IS', F8. 5)
103 FORMAT(/, 19X, ' J ', 7X, ' VALUE ', 6X, ' FRACTIONAL CHANGE ', /)
104 FORMAT(18X, I3, 6X, F9. 4, 11X, F7. 5)
105 FORMAT(/, 9X, 20(1H*), 5X, ' CONVERGED ', 5X, 20(1H*), /)
106 FORMAT(/, 16X, ' ----- NOT CONVERGED ----- ', /)
107 FORMAT(/, 17X, ' CONTROLLER (UNIT NO ), I3, ', 8X, ' LOOP NO. ', I3)
C
IF (LOOP.GE. 999) RETURN
IF (EN(20) EQ. 0. 0) EN(20)=1. 0
LOOP=EN(20)
D=FLOAT(LOOP)/EN(5)
C=FLOAT(INT(D))-D
E=EN(18)
INTE=INT(E+0. 001)
IF(INTE)5, 6, 5
C
C *** RECIBIENDO LA CORRIENTE DE LA MATRIZ SN ***
C
5 CALL STREAM(E)
IF(LOOP-2)6, 7, 7
7 L=0
```

```
DO 15 J=3, JJ
S=SN(15, J)
IF(INT(S*0.001))0, 15, 8
8 TEST=(S-EN(J+18))/S
C
C *** CHECANDO EL % DE DIFERENCIA VS LA TOLERANCIA ***
C
IF(ABS(TEST)-EN(19))10, 10, 9
9 L=L+1
10 IF(C)15, 11, 11
11 IF(J-3)14, 12, 14
12 WRITE(6, 107)NE, LOOP
WRITE(6, 102)INTE, TEST
WRITE(6, 103)
14 WRITE(6, 104)J, S, TEST
15 CONTINUE
IF(L)17, 16, 17
16 WRITE(6, 105)
EN(20)=EN(20)+1.0
LOOP=1
GOTO 19
17 IF(LOOP-INT(EN(17)+0.001))6, 18, 6
18 WRITE(6, 106)
EN(20)=EN(20)+1.0
LOOP=999
GOTO 19
6 EN(20)=EN(20)+L.0
LOOP=LOOP+1
NC=NC-INT(EN(16))-1
IF(INTE)19, 23, 19
C
C *** REEMPLAZANDO LOS ANTIGUOS VALORES DE LA CORRIENTE ***
C
19 DO 20 J=3, JJ
20 EN(J+18)=SN(15, J)
MM=NE
CALL DISKIO(2, MM)
23 RETURN
END
```

#### 4.3.6 Módulos de Computación SEPA01, SEPA02 y SEPA03.

Símbolo

Códigos  
5, 21 y 22



Estas unidades computacionales corresponden a tres tipos de separadores que permiten repartir la corriente de entrada en las corrientes del domo y los fondos. Gracias a ello, se les podría utilizar como una representación muy simplificada de una columna de destilación, un evaporador flash, un separador de mezclas bifásicas, un absorbedor e, incluso, un secador (2).

La subrutina SEPA01 permite fijar la temperatura, la presión y la composición de las corrientes del domo y los fondos. Para ello distribuye cada componente presente en la alimentación en los flujos del domo y los fondos basándose en los factores de separación para la corriente indicada en EN(12) señalados en el vector EN [EN(20) a EN(19+NOCOMP)]; de manera similar se especifican las temperaturas [EN(16) para la corriente EN(12) y EN(17) para la corriente EN(13)] y las presiones [EN(18) para la corriente EN(12) y EN(19) para la corriente EN(13)] de salida.

Por otro lado, la subrutina SEPA02 calcula los flujos de las corrientes del domo y los fondos en base al porcentaje de recuperación EN(16) de cierto componente [especificado en EN(17)] para una determinada

corriente [EN(12)] y en las composiciones fraccionarias de todos los componentes en dicha corriente [EN(18) a EN(17+NOCOMP)].

Finalmente, la subrutina SEPA03 también calcula los flujos de las corrientes del domo y los fondos pero basándose en las composiciones de todos los componentes [EN(18) a EN(17+NOCOMP)] en una determinada corriente [EN(12)] y la composición EN(16) de cierto componente especificado en EN(17) en la otra corriente [EN(13)].

Listado de la Subrutina SEPA01

```
SUBROUTINE SEPA01
C
C *** LA SUBROUTINA SEPA01 FIJA LA TEMPERATURA, PRESION Y COMPOSICION
C *** EN LAS CORRIENTES DEL DOMO Y DE LOS FONDOS
C
C *** VECTOR "EN" ***
C EN( 1) A EN(15) = FORMATO STANDARD
C EN(16)          = TEMPERATURA DE SALIDA DE LA CORRIENTE EN(12)
C EN(17)          = TEMPERATURA DE SALIDA DE LA CORRIENTE EN(13)
C EN(18)          = PRESION DE SALIDA EN LA CORRIENTE EN(12)
C EN(19)          = PRESION DE SALIDA EN LA CORRIENTE EN(13)
C EN(20) A
C EN(19+10COMP) = FRACCION DE LA ALIMENTACION EN LA CORRIENTE
C                ESPECIFICADA EN EN(12) PARA CADA COMPONENTE
C
C *** BLOQUE COMMON ***
C
C COMMON /SET1/EN(100), JJ, NIN, NOUT, SI(4, 30), SO(4, 30), PP(15, 25)
C
C *** DIVIDIENDO LA CORRIENTE DE ENTRADA EN DOS CORRIENTES DE SALIDA
C EN BASE A LAS FRACCIONES ESTABLECIDAS EN EL VECTOR "EN" ***
C
C   E=0.0
C   D=0.0
C   DO 1 J=6, JJ
C     SO(1, J)=SI(1, J)*EN(J+14)
C     E=E+SO(1, J)
C     SO(2, J)=SI(1, J)-SO(1, J)
C 1  D=D+SO(2, J)
C     SO(1, 3)=E
C     SO(2, 3)=D
C
C *** FIJANDO LAS TEMPERATURAS Y LAS PRESIONES DE LAS CORRIENTES DE
C SALIDA ***
C
C   SO(1, 4)=EN(16)
C   SO(2, 4)=EN(17)
C   SO(1, 5)=EN(18)
C   SO(2, 5)=EN(19)
C   RETURN
C   END
```

Listado de la Subrutina SEPA02

```
SUBROUTINE SEPA02
C
C *** LA SUBROUTINA SEPA02 CALCULA LOS FLUJOS DE LAS CORRIENTES DEL DOMO
C *** Y DE LOS FONDOS EN BASE A LAS COMPOSICIONES Y AL PORCENTAJE DE
C *** RECUPERACION FIJADOS
C
C *** VECTOR "EN" ***
C EN( 1) A EN(15) = FORMATO STANDARD
C EN(16)          = RECUPERACION DEL COMPONENTE ESPECIFICADO EN
C EN(17)          EN LA CORRIENTE SEÑALADA POR EN(12)
C EN(18) A
C EN(17+HOCOMP) = COMPOSICION DE TODOS LOS COMPONENTES EN LA
C CORRIENTE INDICADA EN EN(12)
C
C *** BLOQUE COMMON ***
C
COMMON /SET1/EN(100), JJ, NIN, NOUT, SI(4, 30), SO(4, 30), PP(15, 25)
C
C *** INICIALIZANDO LOS VALORES DE LAS CORRIENTES DE SALIDA ***
C
DO 1 I=1, JJ
SO(1, I)=0.0
1 SO(2, I)=0.0
C
C *** DETERMINANDO LAS CORRIENTES DE SALIDA ***
C
SO(1, 1)=EN(12)
SO(2, 1)=EN(13)
ICPT=EN(17)+5
SO(1, ICPT)=SI(1, ICPT)*EN(16)
SO(1, 3)=SO(1, ICPT)/EN(12 +ICPT)
SO(2, 3)=SI(1, 3)-SO(1, 3)
DO 2 I=6, JJ
SO(1, I)=SO(1, 3)*EN(12+I)
2 SO(2, I)=SI(1, I)-SO(1, I)
RETURN
END
```



Listado de la Subrutina SEPA03

```

SUBROUTINE SEPA03
C
C *** LA SUBROUTINA SEPA03 CALCULA LOS FLUJOS DE LAS CORRIENTES DEL
C *** DOMO Y DE LOS FONDOS EN BASE A LAS COMPOSICIONES DESEADAS
C
C   *** VECTOR "EN" ***
C   EN( 1) A EN(15) = FORMATO STANDARD
C   EN(16)         = COMPOSICION DEL COMPONENTE ESPECIFICADO EN
C   EN(17)         EN LA CORRIENTE INDICADA POR EN(13)
C   EN(18) A
C   EN(17+NOCOMP) = COMPOSICION DE TODOS LOS COMPONENTES EN LA
C   CORRIENTE SENALADA EN EN(12)
C
C *** BLOQUE COMMON ***
C
COMMON /SET1/EN(100), JJ, NIN, NOUT, SI(4, 30), SO(4, 30), PP(15, 25)
C
C *** INICIALIZANDO LOS VALORES DE LAS CORRIENTES DE SALIDA ***
C
DO 1 I=1, JJ
  SO(1, I)=0.0
  1 SO(2, I)=0.0
C
C *** DETERMINANDO LAS CORRIENTES DE SALIDA ***
C
SO(1, 1)=EN(12)
SO(2, 1)=EN(13)
ICPT=EN(17)+5
SO(1, 3)=(SI(1, ICPT)-SI(1, 3)*EN(16))/(EN(12+ICPT)-EN(16))
SO(2, 3)=SI(1, 3)-SO(1, 3)
DO 2 I=6, JJ
  SO(1, I)=SO(1, 3)+EN(12+I)
  2 SO(2, I)=SI(1, I)-SO(1, I)
RETURN
END
```

#### 4.3.7 Módulo de Computación REACT1

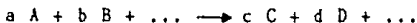
Símbolo

Código

23



Esta subrutina modela un reactor simple (sin considerar el balance de energía) en el que ocurre una reacción del tipo



Los coeficientes estequiométricos molares de la reacción  $a, b, c, d, \dots$  [EN(18) a EN(17+NOCOMP)] se alimentan como números negativos para los reactivos, como positivos para los productos y nulos para los componentes que no intervienen en la reacción debiendo estar referidos al reactivo limitante.

La extensión de la reacción está determinada por la conversión EN(16) del componente clave EN(17) lo que permite al módulo calcular por separado los flujos de salida de cada componente así como el flujo total de la corriente que abandona el reactor. (Los flujos deberán estar dados en unidades molares).

Así la distribución de los productos estará dada en función del conjunto de factores estequiométricos y la conversión del reactivo limitante según:

$$(\text{salida})_i = (\text{entrada})_i + c (\text{entrada})_k F_i$$

en donde  $c$  = conversión del componente clave,

$k$  = número del componente clave,

$i$  = número del componente y

$F_i$  = factor estequiométrico para el componente  $i$   
por mol de reactivo limitante.

Listado de la Subrutina REACT1

```

SUBROUTINE REACT1
C
C *** LA SUBROUTINA REACT1 CORRESPONDE A UN REACTOR SIMPLE
C
C *** VECTOR "EN" ***
C EN( 1) A EN(15) = FORMATO STANDARD
C EN(16)          = CONVERSION DEL COMPONENTE ESPECIFICADO EN
C EN(17)          (REACTIVO LIMITANTE)
C EN(18) A
C EN(17+NOCOMP) = FACTORES ESTEQUIOMETRICOS DE LA REACCION
C EN BASE A EN(17)
C
C *** BLOQUE COMMON ***
C
COMMON /SET1/EN(100),JJ,NIN,NOUT,S1(4,30),S0(4,30),PP(15,25)
C
S0(1,1)=EN(12)
ICPT=EN(17)+5
S0(1,3)=0.0
DO 2 I=6, JJ
S0(1, I)=S1(1, I)+S1(1, ICPT)*EN(16)*EN(12+I)
2 S0(1,3)=S0(1,3)+S0(1, I)
RETURN
END
```

Referencias.

1. Organick, E. I.; "A FORTRAN IV Primer", [Addison-Wesley Publishing Co., U. S. A. (1966)].
2. Johnson, A. I.; Peters N. & Associates; "GEMCS/CYBER 73", [S. A. C. D. A., University of Western Ontario, Canada (1979)].

## CAPITULO V

### IMPLEMENTACION DE DOS NUEVAS SUBROUTINAS

#### 5.1 FORMULACION MATEMATICA DE UNA SUBROUTINA CORRESPONDIENTE A UN BALANCE DE MATERIA Y ENERGIA.

El balance de materia constituye un prerrequisito esencial en la ingeniería de detalle de una planta para poder diseñar hasta las operaciones más simples, tanto para calcular el tamaño del equipo como para considerar las interrelaciones entre varias piezas del mismo.

Para una planta que ya se encuentra operando, el balance de masa es una herramienta muy útil y efectiva para mejorar la eficiencia del proceso ya que ayuda a identificar la naturaleza, magnitud y localización de cada fuente de pérdida de eficiencia.

Siempre que se pueda, conviene preparar el balance de materia de manera diagramática debiendo cumplirse el principio de conservación de la misma, es decir, que la suma total de masas que salen del proceso sea igual a la suma total de masas alimentadas, exceptuando situaciones que involucran sustancias radiactivas y reacciones nucleares. Cuando el material no sale del sistema sino que simplemente se recircula, los datos resultan indispensables para el balance pero no suelen dibujarse en el diagrama.

La ventaja de preparar un balance de materia completo es que se puede aplicar a procesos donde ocurren cambios físicos y químicos mientras que, por otro lado,

un balance sencillo y limitado es de considerable valor cuando las circunstancias permiten su uso pues simplifica lo que de otro modo sería innecesariamente complejo. Un balance de materia bien hecho no corrige ninguna falla pero destaca claramente la existencia y magnitud de éstas.

Una vez elaborado el balance de materia, las masas pueden utilizarse para calcular el balance de energía. Los niveles de presión y temperatura en varios puntos importantes del proceso, en particular en cada pieza principal del equipo, servirán como guía para efectuar el balance de calor. La descripción de estos balances, cuyos resultados generalmente se expresan en  $\frac{\text{unidades de energía}}{\text{unidad de tiempo}}$ , también puede realizarse en forma de diagramas de flujo análogamente a lo que sucede con el balance de materia.

Los elementos principales de un balance de energía son el calor de reacción (calor neto entre el calor de formación de los productos y el calor de formación de los reactivos a la temperatura de reacción), el calor latente (requerido siempre que exista un cambio de fase) y los calores específicos (necesarios para calcular cambios de temperatura).

Vale la pena aclarar que hay muchas maneras de representar diagramáticamente estos balances dependiendo de la persona que los hace y del tipo de información que se quiere destacar más. Se deberá recordar que un ingeniero busca comprender como trabaja una unidad de proceso para lo cual suele construir un modelo matemático tomando en cuenta los principios de conservación (de masa, de energía, de momentum), las ecuaciones de movimiento (Navier-Stokes, Poiseuille y otras genera

lizaciones de la segunda ley de Newton para fluidos en movimiento), las ecuaciones que describen el flujo de calor (ley de Fourier para la conducción del mismo, ley de Stefan Boltzman para la radiación, etc.) y muchas otras ecuaciones básicas de física y química. Como a menudo el proceso es tan complicado que no puede analizarse en términos de principios elementales, muchas de estas ecuaciones son relaciones empíricas obtenidas de experiencias con procesos similares.

Las ventajas de las computadoras para la solución de problemas son bien conocidas por lo que, cuando se aplica el acercamiento modular para resolver balances de materia y energía en los procesos químicos, suelen elaborarse programas que efectúan, varias veces, las mismas series de cálculos que representan a las unidades de proceso ya que los valores de las variables de diseño de cada una de ellas cambian sistemáticamente para converger en una solución para el proceso completo.

Así, un programa de computación digital para un balance de materia y/o energía sobre una unidad de proceso se elabora en base a un conjunto específico de variables de diseño y tendrá la siguiente estructura:

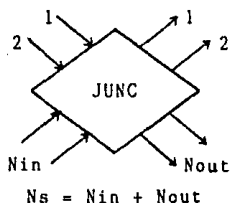
1. Los valores de las variables de diseño se leen (tarjetas o archivo) y se imprimen.
2. Los valores estimados (si se requieren) de las variables desconocidas pero necesarias para las soluciones iterativas de las ecuaciones no lineales se leen o suponen e imprimen.
3. Se resuelven las ecuaciones.
4. Se imprime el conjunto de valores solución.

Para el desarrollo de la subrutina correspondiente



a un balance de materia y energía se escogió un mezclador-separador (ver Figura 5.1) que maneja exclusivamente  $N_s$  corrientes en la misma fase con un número arbitrario de corrientes de alimentación,  $N_{in}$ , y de salida,  $N_{out}$ , éstas últimas de idéntica composición. El número de componentes,  $N_c$ , es variable pero no podrá exceder de 25.

Figura 5.1 MEZCLADOR-SEPARADOR



Utilizando los conceptos expuestos en los libros clásicos de balance de materia y energía (1) (2) (3) y tomando en cuenta que el equipo opera a régimen permanente y que las especies involucradas no reaccionan entre sí, el balance de materia puede expresarse como:

$$\sum_{i=1}^{N_s} \sum_{j=1}^{N_c} F_i x_{ij} = 0 \quad \dots$$

$$\sum_{i=1}^{N_{in}} \sum_{j=1}^{N_c} F_{i_{in}} x_{ij_{in}} = \sum_{i=1}^{N_{out}} \sum_{j=1}^{N_c} F_{i_{out}} x_{ij_{out}}$$

en donde  $F_i$  = flujo total de la corriente  $i$  y

$x_{ij}$  = fracción mol del componente  $j$  en la corriente  $i$ .

En este caso los flujos de los componentes en cada corriente así como el flujo total de la misma se alimen

tan directamente y puesto que

$$f_{ij} = F_i x_{ij} \quad \text{entonces} \quad F_{j_{in}} = \sum_{i=1}^{N_{in}} f_{ij_{in}} = F_{j_{out}}$$

en donde  $f_{ij}$  = flujo del componente  $j$  en la corriente  $i$   
 y  $F_j$  = flujo total del componente  $j$ .

Los flujos por componente y total en las corrientes de salida se obtienen, en base a las fracciones  $\alpha_i$  del flujo total manejado por el módulo deseadas en cada una de ellas, aplicando:

$$f_{ij_{out}} = F_{j_{out}} \alpha_i \quad \text{y} \quad F_{i_{out}} = \sum_{j=1}^{N_c} f_{ij_{out}} = F_{out} \alpha_i$$

en donde  $\alpha_i = \frac{F_{i_{out}}}{F_{in}}$ ,  $i = 1, 2, \dots, N_{out}$ , y

$F$  = flujo total.

Todos los flujos deberán ser molares ( $\frac{\text{moles}}{\text{unidad de tiempo}}$ ).

El balance de energía, puesto que no hay acumulación, ni reacción, ni se efectúa ni se recibe trabajo alguno, ni se absorbe ni se desprende calor a los alrededores y considerando que tanto la energía potencial como la cinética son despreciables, correspondería al balance de entalpía:

$$\sum_{i=1}^{N_s} \sum_{j=1}^{N_c} f_{ij} h_j = 0 \quad . .$$

$$\sum_{i=1}^{N_{in}} \sum_{j=1}^{N_c} f_{ij_{in}} h_{j_{in}} = \sum_{i=1}^{N_{out}} \sum_{j=1}^{N_c} f_{ij_{out}} h_{j_{out}}$$

en donde  $h_j$  = entalpía molar para el componente  $j$ .

Evaluando las entalpías molares como una función polinomial de tercer grado de la temperatura,

$$\underline{h}_L = a + bT + cT^2 + dT^3 \quad \text{y} \quad \underline{H}_V = A + BT + CT^2 + DT^3$$

en donde  $\underline{h}_L$  = entalpía molar para las corrientes en fase líquida [=]  $\frac{\text{unidades de calor}}{\text{moles}}$ ,

$\underline{H}_V$  = entalpía molar para las corrientes en fase vapor [=]  $\frac{\text{unidades de calor}}{\text{moles}}$  y

a, b, c, d y A, B, C, D = constantes específicas para cada componente j;

las "entalpías" ("contenido calórico") de las corrientes de entrada y de salida se obtienen mediante:

$$h_i = \sum_{j=1}^{Nc} f_{ij} h_j \quad [=] \quad \frac{\text{unidades de calor}}{\text{unidad de tiempo}}$$

y la temperatura de salida se calcula a partir de:

$$\Delta T = \frac{\Delta h}{\frac{dh_{out}}{dT}} = \frac{h_{out} - h_{in}}{\frac{dh_{out}}{dT}} \quad [=] \quad \text{unidades de temperatura}$$

en donde  $h_{in}$  = "entalpía" total de entrada =  $\sum_{i=1}^{Nin} h_{i_{in}}$  y

$h_{out}$  = "entalpía" total de salida =  $\sum_{i=1}^{Nout} h_{i_{out}}$ ,

ambas expresadas en  $\frac{\text{unidades de calor}}{\text{unidad de tiempo}}$ , y

$\Delta T$  = diferencia entre la temperatura calculada y la temperatura supuesta;

$$\therefore T_c = T_s - \Delta T$$

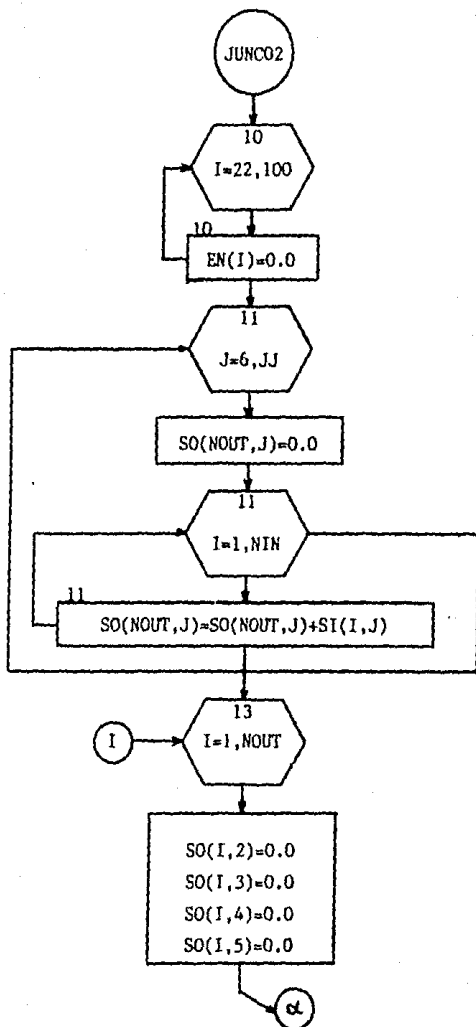
siendo  $T_s$  = temperatura supuesta y  $T_c$  = temp. calculada.

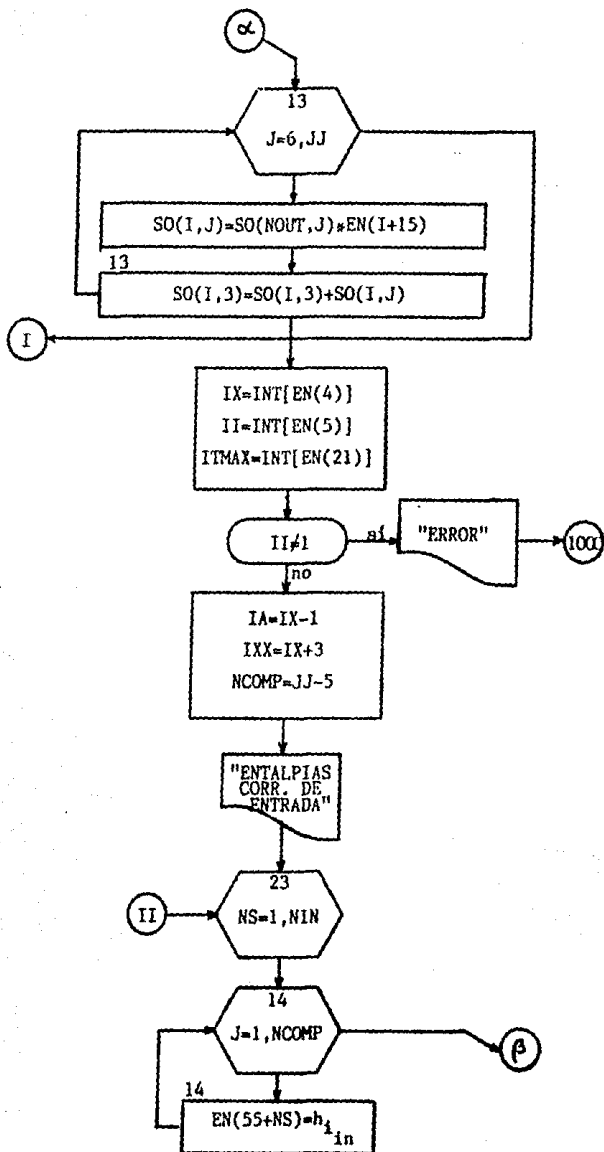
Como se trata de un procedimiento de tanteo y error, que se resuelve por el método de substituciones sucesivas, la primer suposición para la temperatura de salida corresponderá al promedio aritmético de las temperaturas de las corrientes de entrada y mientras no se cumpla con la tolerancia especificada, la temperatura calculada pasará a ser la temperatura supuesta de la siguiente iteración:

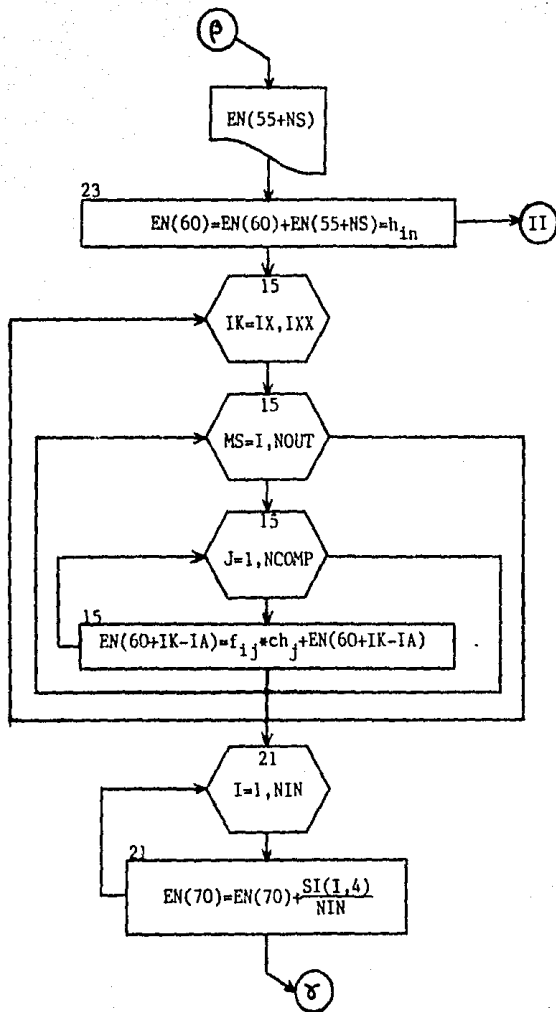
$$T_s^{n+1} = T_c^n$$

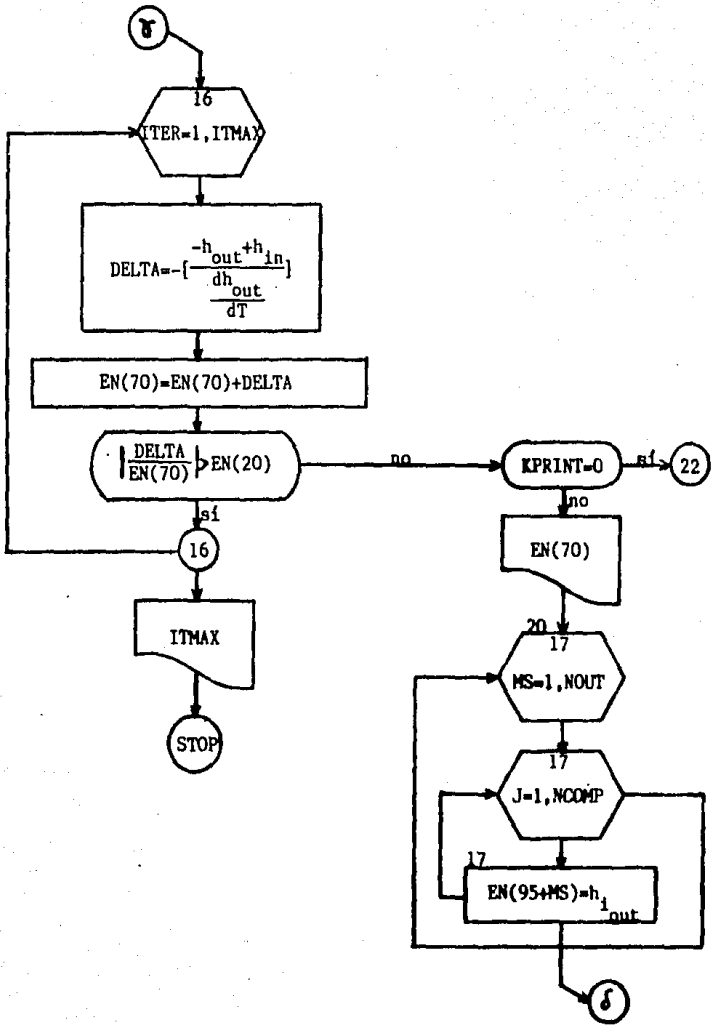
en donde n = cualquier iteración dentro del proceso de convergencia.

5.1.1 Diagrama de Flujo de la Subrutina JUNCO2

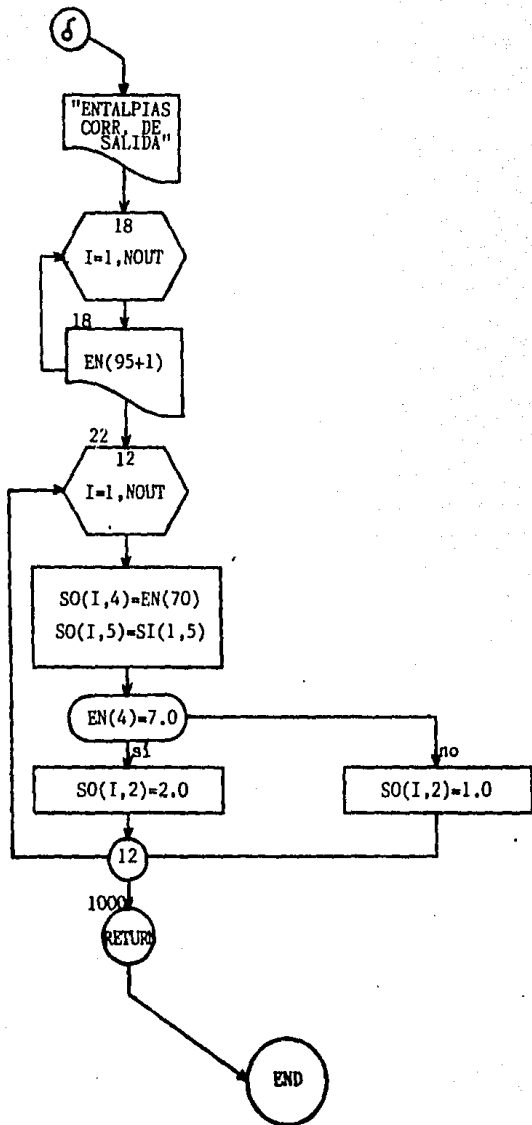












### Variables Particulares de la Subrutina JUNCO2

A continuación se listan, en el orden en que aparecen en el diagrama de flujo, las variables utilizadas por el módulo JUNCO2:

- EN(22) a EN(100) = elementos del vector de parámetros de equipo 'EN' disponibles
- SO(NOUT,J) = flujo total de entrada  
= flujo total de salida
- NIN = número de corrientes de entrada
- NOUT = número de corrientes de salida
- J = número de componente
- JJ = número de parámetros en un arreglo de corrientes [SI, SN o SO]
- I, NS, MS = número de corriente
- SO(I,2) = fase de la corriente I
- SO(I,3) = flujo total de salida de la corriente I
- SO(I,4) = temperatura de salida de la corriente I
- SO(I,5) = presión de salida de la corriente I
- SO(I,J) = flujo de salida del componente J en la corriente I
- EN(16) a EN(19) = proporción del flujo total en cada corriente de salida
- IX = valor entero de la fase de las corrientes manejadas por el separador, necesario para el cálculo de las entalpías

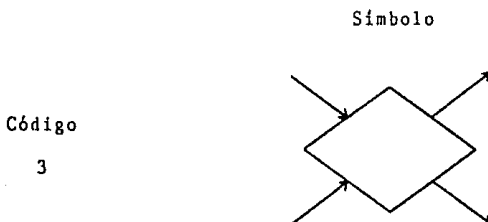
- II = valor entero del grado de la función para el cálculo de las entalpías
- ITMAX = número máximo de iteraciones
- NCOMP = número de componentes presentes
- I, NS, MS, IK e  
ITER = contadores para los ciclos
- EN(56) a EN(59) = "entalpías" de las corrientes de entrada
- EN(60) = "entalpía" de entrada, total
- EN(61) a EN(64) = flujo del componente J en la corriente I multiplicado por el coeficiente entálpico correspondiente
- $f_{ij}$  = flujo del componente J en la corriente I
- $ch_j$  = coeficiente para el cálculo de las entalpías para el componente J; a, b, c, y d para fase líquida y A, B, C y D para fase vapor
- EN(70) = temperatura de mezclado (salida)
- SI(I,4) = temperatura de entrada de la corriente I
- DELTA = decremento en la temperatura de mezclado
- EN(20) = tolerancia fraccional para la convergencia de la temperatura
- KPRINT = control de impresión
- EN(96) a EN(99) = "entalpías" de las corrientes

de salida

SI(1,5) = presión de la primer corriente  
de entrada

EN(4) = identificación de la fase de  
las corrientes manejadas por  
JUNCO2.

### 5.1.2 Módulo de Computación JUNC02.



Se trata de un mezclador sobre el que se realiza tanto un sencillo balance de energía como un balance de materia. El módulo permite combinar hasta cuatro corrientes, en la misma fase pero con diferentes temperaturas y composiciones, estimando la temperatura resultante y asignando como presión de salida el valor de la presión en la primer corriente de entrada. La mezcla fluida puede entonces dividirse, de acuerdo con las proporciones deseadas [EN(16) - EN(19)], hasta en cuatro corrientes de salida de idéntica composición.

Para el balance de energía, el módulo calcula la entalpía de cada corriente de entrada [EN(56) - EN(59)], como una función polinomial de la temperatura (por ser de tercer grado, EN(5) = 1), para usarlas como base para el cálculo de la temperatura de la mezcla resultante de acuerdo al balance de entalpía.

Puesto que dicha temperatura se evalúa por el método de substituciones sucesivas es necesario establecer una tolerancia para la prueba de convergencia [EN(20)] y un número máximo de iteraciones en caso de que ésta no se logre [EN(21)].

Una vez obtenida la temperatura de salida [EN(70)]

se calculan las entalpías de las corrientes que abandonan el mezclador-separador [EN(96) - EN(100)] y se les asigna también su presión de salida.

Para la correcta evaluación de las entalpías es necesario especificar si las corrientes se encuentran en fase líquida [EN(4) = 7.0  $\rightarrow$   $\underline{h}_L$ ] o en fase vapor [EN(4) = 12.0  $\rightarrow$   $\underline{H}_V$ ]. Los demás parámetros que definen a la subrutina siguen el formato standard (ver Tabla 4.1).

Listado de la Subrutina JUNC02

```
SUBROUTINE JUNC02
C
C *** LA SUBROUTINA JUNC02 EFECTUA TANTO EL BALANCE DE MATERIA COMO
C *** EL DE ENERGIA DE UN MEZCLADOR DE CORRIENTES EN LA MISMA FASE
C
C   *** VECTOR "EN" ***
C   EN( 1) A EN( 3) = FORMATO STANDARD
C   EN( 4)          = FASE DE LAS CORRIENTES
C   EN( 5)          = GRADO DE LA FUNCION PARA CALCULAR ENTALPIAS
C   EN( 6) A EN(15) = FORMATO STANDARD
C   EN(16) A EN(19) = PROPORCION DEL TOTAL EN C/CORRIENTE DE SALIDA
C   EN(20)          = TOLERANCIA FRACCIONAL PARA LA CONVERGENCIA
C   EN(21)          = NUMERO MAXIMO DE ITERACIONES
C
C *** BLOQUE COMMON ***
C
COMMON /SET1/EN(100), JJ, NIN, NOUT, SI(4, 30), SO(4, 30), PP(15, 25)
COMMON /SET2/KPRINT
C
200 FORMAT(///, ' ERROR LA FORMA DE LOS DATOS DEBIERA SER CUBICA
*POR LO TANTO EN(5) => 1 ')
201 FORMAT(///, 22X, ' TEMPERATURA DE SALIDA =', F10.4)
202 FORMAT(///, 20X, ' NO CONVERGE DESPUES DE ', I3, ' ITERACIONES ')
203 FORMAT(///, 20X, ' ENTALPIAS DE LAS CORRIENTES DE SALIDA ', /)
204 FORMAT(34X, 4F11.2)
205 FORMAT(///, 20X, ' ENTALPIAS DE LAS CORRIENTES DE ENTRADA ', /)
C
C *** INICIALIZANDO VALORES Y CALCULANDO LOS FLUJOS TOTALES DE
C   ENTRADA PARA CADA COMPONENTE ***
C
DO 10 I=22, 100
10 EN(I)=0.0
DO 11 J=6, JJ
SO(NOUT, J)=0.0
DO 11 I=1, NIN
11 SO(NOUT, J)=SO(NOUT, J)+SI(I, J)
C
C *** INICIALIZANDO LOS VALORES DE LAS CORRIENTES DE SALIDA Y
C   MULTIPLICANDO LOS FLUJOS TOTALES DE LOS COMPONENTES POR
C   SU FRACCION EN CADA UNA DE ELLAS ***
C
DO 13 I=1, NOUT
SO(I, 2)=0.0
SO(I, 3)=0.0
SO(I, 4)=0.0
SO(I, 5)=0.0
DO 13 J=6, JJ
```

```

SO(I, J)=SO(NOUT, J)*EN(I+15)
13 SO(I, 3)=SO(I, 3)+SO(I, J)
    IX=INT(EN(4))
    II=INT(EN(5))
    ITMAX=INT(EN(21))
    IF(II. NE. 1)GOTO 980
    IA=IX-1
    IXX=IX+3
    NCOMP=JJ-5
C
C *** CALCULANDO LAS ENTALPIAS DE LAS CORRIENTES DE ENTRADA ***
C
    WRITE(6, 205)
    DO 23 NS=1, NIN
    DO 14 J=1, NCOMP
14 EN(55+NS)=(PP(IX, J)+PP(IX+1, J))*SI(NS, 4)+(PP(IX+2, J))*SI(NS, 4)*
    *SI(NS, 4)+(PP(IX+3, J))*SI(NS, 4)*SI(NS, 4)*SI(NS, 4))*SI(NS, J+5)
    **EN(55+NS)
    WRITE(6, 204)EN(55+NS)
23 EN(60)=EN(60)+EN(55+NS)
C
C *** CALCULANDO LA TEMPERATURA DE SALIDA ***
C
    DO 15 IK=1, IXX
    DO 15 MS=1, NOUT
    DO 15 J=1, NCOMP
15 EN(60+IK-IA)=(SO(MS, J+5)+PP(IX, J))+EN(60+IK-IA)
    DO 21 I=1, NIN
21 EN(70)=EN(70)+(SI(I, 4)/NIN)
    DO 16 ITER=1, ITMAX
    DELTA=-((-EN(60)+EN(61)+(EN(62)*EN(70))+(EN(63)*EN(70)*EN(70))+
    *(EN(64)*EN(70)*EN(70)*EN(70)))/(EN(62)+(2. *EN(63)*EN(70))+
    *3. *EN(64)*EN(70)*EN(70)))
    EN(70)=EN(70)+DELTA
    IF(ABS(DELTA/EN(70)). GT. EN(20))GOTO 15
    IF (KPRINT. EQ. 0)GOTO 22
    WRITE(6, 201)EN(70)
    GOTO 20
16 CONTINUE
    WRITE(6, 202)ITMAX
    STOP
C
C *** CALCULANDO LAS ENTALPIAS DE LAS CORRIENTES DE SALIDA Y FIJANDO
C LOS DEMAS PARAMETROS DE LAS MISMAS ***
C
20 DO 17 MS=1, NOUT
    DO 17 J=1, NCOMP
17 EN(95+MS)=(PP(IX, J)+PP(IX+1, J)*EN(70)+(PP(IX+2, J)*EN(70)+EN(70))+
    *(PP(IX+3, J)*EN(70)*EN(70)+EN(70)))*SO(MS, J+5))+EN(95+MS)
    WRITE(6, 203)

```



```
DO 18 I=1, NOUT
18 WRITE(6,204)EN(95+I)
22 DO 12 I=1,NOUT
   SO(I,4)=EN(70)
   SO(I,5)=SI(1,5)
   IF(EN(4).EQ.7.0)GOTO 19
   SO(I,2)=1.0
   GOTO 12
19 SO(I,2)=2.0
12 CONTINUE
   GOTO 1000
900 WRITE(6,200)
1000 RETURN
END
```

## 5.2 FORMULACION MATEMATICA DE UNA SUBROUTINA CORRESPONDIENTE A UNA DESTILACION FLASH.

Dentro de los procesos de separación es importante destacar la vaporización flash o destilación en equilibrio, como en ocasiones se le llama, que corresponde a una operación de una sólo etapa en la que se evapora parcialmente una mezcla líquida permitiendo que el vapor alcance el equilibrio con el líquido residual; las fases vapor y líquido resultantes son separadas y, a continuación, abandonan el aparato.

Hay ocasiones en que, por las condiciones del proceso, la corriente de alimentación está constituida por una mezcla bifásica y lo que se persigue con el flasheo es generar una mayor cantidad de vapor y al mismo tiempo separarlo del líquido con el que se encuentra en equilibrio.

La destilación flash puede ser una operación "batch" o continua; el siguiente esquema muestra el diagrama de flujo típico de la segunda (Figura 5.2). En este caso, una corriente de alimentación líquida conteniendo dos o más componentes es precalentada en un intercambiador de calor tubular convencional o pasándola a través de los tubos calientes de un horno. Al reducirse la presión, se forma vapor adiabáticamente a expensas del líquido por lo que, debido a esta evaporación parcial, la mezcla se enfría conforme entra al tanque de separación. Los baffles o mamparas en la cámara remueven las gotas de líquido de la corriente de vapor que asciende por la chimenea central abandonando la unidad por su parte superior mientras que la corriente líquida lo hace por el fondo. El vapor puede pasar entonces a un condensador (no mostrado en la figura).

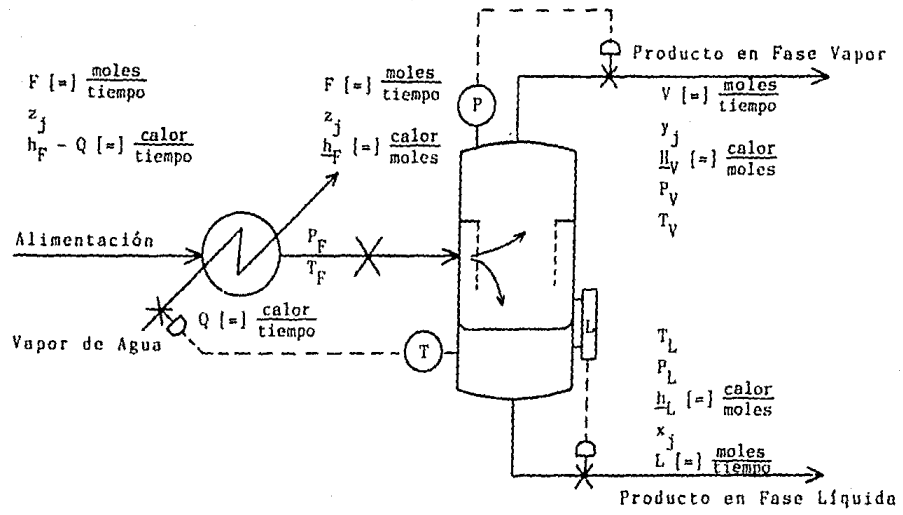


Figura 5.2 VAPORIZACION FLASH CONTINUA

El objetivo de la vaporización flash es separar los componentes ligeros, que se concentran en el vapor, de los pesados, que se concentran en el líquido. Para mantener el régimen permanente, las corrientes de vapor y líquido son continuamente retiradas del separador.

La cantidad de alimentación evaporada depende de las condiciones de la misma (temperatura y presión), de la caída de presión conforme es flasheada y de la cantidad de calor removido o añadido en el vaporizador (cuando la destilación es isotérmica). Como por lo general el vapor flasheado de la alimentación contiene casi to dos los componentes de la mezcla, la solución del balance de materia dependerá de la distribución de cada uno de ellos dentro del líquido y el vapor.

A veces conviene llevar a cabo destilaciones flash sucesivas sobre el líquido residual, como una serie de operaciones de una etapa, para lograr una mejor separación que la que se obtendría si la misma cantidad de vapor se formara en una operación solamente. Conforme la cantidad de vapor formado en cada etapa se vuelve menor, el número total de vaporizaciones asciende y, en el límite, la operación se aproxima a una destilación diferencial.

Puesto que los vaporizadores suelen diseñarse de manera que el tiempo de residencia en la cámara sea suficiente para lograr un equilibrio entre las corrientes de vapor y líquido que abandonan la unidad (es decir, existe suficiente tiempo para que la transferencia de masa y calor entre líquido y vapor alcance la condición de equilibrio), la distribución de cada componente estará dada por su constante de equilibrio  $k$  que se define para el componente  $j$  como

$$k_j = \frac{y_j}{x_j}.$$

Así,  $k$  es la razón de la fracción mol en el vapor entre la fracción mol en el líquido. Las condiciones de equilibrio se alcanzan cuando cesa la transferencia neta de masa entre las fases. En una mezcla binaria, el componente con  $k > 1$  es más volátil y se le llama componente ligero mientras que el otro tiene una  $k < 1$  y se le conoce como componente pesado.

A consecuencia del equilibrio, los productos, tanto líquido como vapor, abandonan el tanque flash a la misma temperatura y presión. Aunque en general el valor de  $k$  depende de la temperatura, presión total y composición de los productos, existen algunos casos especiales para los que el cálculo de la constante de equilibrio es particularmente simple.

A continuación se expone el desarrollo matemático de los balances de materia y energía resultado del flasheo adiabático, a una presión de vaporización conocida, de una corriente de alimentación líquida conteniendo  $N_c$  componentes haciendo referencia a la nomenclatura de la Figura 5.2 y a la literatura relativa publicada (3) (4) (5).

Balance total:

$$F = L + V \quad (1)$$

Balance por componente:

$$Fz_j = Lx_j + Vy_j \quad j = 1, 2, 3, \dots, N_c \quad (2)$$

en donde  $F$  = moles totales alimentados al vaporizador por unidad de tiempo;

$L$  = moles en la fase líquida que salen del separador por unidad de tiempo;

$V$  = moles en la fase vapor que abandonan el

tanque flash por unidad de tiempo;

$z_j$  = fracción mol del componente  $j$  en la alimentación;

$x_j$  = fracción mol del componente  $j$  en la fase líquida;

$y_j$  = fracción mol del componente  $j$  en la fase vapor.

Restricciones de las fracciones mol

$$\sum_{j=1}^{N_c} x_j = 1 \quad (3.a)$$

$$\sum_{j=1}^{N_c} y_j = 1 \quad (3.b)$$

En cuanto a las ecuaciones del equilibrio entre fases, dado que las corrientes de producto están en equilibrio,

$$T_V = T_L = T_{eq.} \quad (4)$$

$$P_V = P_L = P_{vap.} \quad (5)$$

y de la relación de equilibrio en función de la temperatura, la presión total y las composiciones,

$$y_j = k_j x_j \quad j = 1, 2, 3, \dots, N_c \quad (6)$$

en donde  $k_j$  se puede obtener como una función polinomial de la temperatura ( $y$ , por supuesto de la presión total):

$$k_j = \alpha_j + \beta_j T + \gamma_j T^2 + \delta_j T^3 \quad (7)$$

La ecuación representando el balance de energía es:

$$F \underline{h}_F = V \underline{H}_V + L \underline{h}_L + \frac{dQ}{dt}$$

en donde  $\underline{h}_F$  = entalpía molar de la corriente de alimentación;

$\underline{H}_V$  = entalpía molar de la corriente en fase vapor;

$\underline{h}_L$  = entalpía molar de la corriente en fase líquida;

$\frac{dQ}{dt}$  = razón de enfriamiento de la cámara de vaporización; para el caso adiabático que se está considerando  $\frac{dQ}{dt} = 0$ .

Por lo tanto,

$$F\underline{h}_F = V\underline{H}_V + L\underline{h}_L \quad (8)$$

La evaluación de la entalpías es función de la temperatura y la composición de cada corriente,

$$\underline{h}_i = \underline{h}_i[T_i, x_{i1}, x_{i2}, \dots, x_{iNc}] \quad i = F, V, L \quad (9)$$

y suponiendo que las entalpías molares parciales de las mezclas de los componentes bajo consideración se encuentran disponibles como funciones cúbicas de la temperatura, en forma semejante a lo que sucede con la constante de equilibrio,

$$\underline{h}_j = a_j + b_j T + c_j T^2 + d_j T^3 \quad j = 1, 2, 3, \dots, Nc \quad (10.a)$$

$$\underline{H}_j = A_j + B_j T + C_j T^2 + D_j T^3 \quad j = 1, 2, 3, \dots, Nc \quad (10.b)$$

entonces

$$\underline{h}_i = \sum_{j=1}^{Nc} \underline{h}_j \quad i = F, L \quad (11.a) \quad \text{y} \quad \underline{H}_V = \sum_{j=1}^{Nc} \underline{H}_j \quad (11.b)$$

en donde  $h_j$  = entalpía molar parcial del componente  $j$   
en corrientes en fase líquida;

$H_j$  = entalpía molar parcial del componente  $j$   
en la corriente de vapor;

$\alpha, \beta, \gamma, \delta$ ;  $a, b, c, d$  y  $A, B, C, D$  = constantes  
específicas para cada componente  $j$ .

El balance de materia se encuentra relacionado con el de energía a través de la variable  $T_{eq.}$  y, en consecuencia, deberán resolverse simultáneamente.

El procedimiento de solución implica el suponer un valor de la temperatura de equilibrio, llamado  $T_{eq.}^*$ , que permita la solución de las ecuaciones (2), (3) y (6) para después resolver las (10) y (11) obteniendo  $h_L$  y  $H_V$ . La solución será aquel valor de la temperatura de equilibrio supuesta que satisfaga el balance de energía, ecuación (8).

Ya que los productos consisten en líquido y vapor en equilibrio, los valores solución para  $T_{eq.}$  a la presión de flasheo se encuentran delimitados entre las temperaturas del punto de burbuja [ $T_b$ ] y del punto de rocío [ $T_r$ ] de la alimentación cuya evaluación se expondrá posteriormente:

$$T_b < T_{eq.} < T_r.$$

Incluso el primer valor que se supone para la temperatura de equilibrio corresponde a:

$$T_{eq.}^* = \frac{1}{2} (T_b + T_r).$$

Ahondando en el tema, si se substituye el valor de la fracción mol del componente  $j$  en el vapor obtenido de la ecuación (6) en el balance de materia por componente, ecuación (2),



$$Fz_j = Lx_j + k_j V x_j$$

$$Fz_j = x_j (L + k_j V)$$

y despejando  $x_j$ ,

$$x_j = \frac{Fz_j}{L + k_j V} = \frac{z_j}{\frac{1}{F} (L + k_j V)} = \frac{z_j}{\frac{L}{F} + \frac{k_j V}{F}}$$

del balance de materia total, ecuación (1),

$$V = F - L \quad (12)$$

y al sustituir en la ecuación anterior,

$$x_j = \frac{z_j}{\frac{L}{F} + \frac{k_j (F - L)}{F}} = \frac{z_j}{\frac{L}{F} + k_j - \frac{k_j L}{F}} = \frac{z_j}{\frac{L}{F} \left[ \frac{k_j}{L} - k_j + 1 \right]}$$

finalmente,

$$x_j = \frac{F}{L} \cdot \frac{z_j}{k_j \left[ \frac{F}{L} - 1 \right] + 1} \quad (13)$$

que también puede expresarse como

$$x_j = \frac{z_j}{\left[ 1 - \frac{L}{F} \right] k_j + \frac{L}{F}} \quad (14)$$

En forma análoga, si se substituye la ecuación (6) despejada en función de la fracción mol del componente  $j$  en el líquido en la ecuación (2), balance de materia por componente, se obtiene

$$Fz_j = \frac{L y_j}{k_j} + V y_j$$

$$Fz_j = y_j \left[ \frac{L}{k_j} + V \right]$$

y despejando  $y_j$ ,

$$y_j = \frac{Fz_j}{\frac{L}{k_j} + V} = \frac{z_j}{\frac{1}{F} \left[ \frac{L}{k_j} + V \right]} = \frac{z_j}{\frac{L}{Fk_j} + \frac{V}{F}}$$

substituyendo V por su valor en la ecuación (12),

$$y_j = \frac{z_j}{\frac{L}{Fk_j} + \frac{F-L}{F}} = \frac{z_j}{\frac{L}{Fk_j} + 1 - \frac{L}{F}}$$

y finalmente,

$$y_j = \frac{z_j}{1 - \frac{L}{F} \left[ 1 - \frac{1}{k_j} \right]} \quad (15)$$

por lo que la ecuación (3.b) puede reescribirse como

$$\sum_{j=1}^{Nc} \frac{z_j}{1 - \frac{L}{F} \left[ 1 - \frac{1}{k_j} \right]} = 1$$

ecuación que se designará como función  $g_1$

$$g_1 = \sum_{j=1}^{Nc} \frac{z_j}{1 - \frac{L}{F} \left[ 1 - \frac{1}{k_j} \right]} - 1 \quad (16)$$

La función  $g_1$  es una ecuación claramente dependiente del valor de la relación  $\frac{L}{F}$  y también, a través de las constantes de equilibrio de cada componente, implícitamente dependiente de la temperatura; por lo tanto, para resolverla se requiere de una segunda ecuación en función de las mismas variables.

Así, replanteando el balance de entalpía global a través de la unidad,

$$F \sum_{j=1}^{Nc} z_j h_{jF} = L \sum_{j=1}^{Nc} x_j h_{jL} + V \sum_{j=1}^{Nc} y_j h_{jY} \quad (8')$$

en donde  $\underline{h}_{jF}$  = entalpía molar del componente  $j$  en la alimentación (fase líquida);

$\underline{h}_{jL}$  = entalpía molar del componente  $j$  en la corriente de productos líquida;

$\underline{H}_{jV}$  = entalpía molar del componente  $j$  en la corriente de productos en fase vapor.

Dividiendo entre  $F$  y substituyendo el valor de  $V$  obtenido de (1), ecuación (12),

$$\sum_{j=1}^{Nc} z_j \underline{h}_{jF} = \frac{L}{F} \sum_{j=1}^{Nc} x_j \underline{h}_{jL} + \frac{F-L}{F} \sum_{j=1}^{Nc} y_j \underline{H}_{jV}$$

$$\sum_{j=1}^{Nc} z_j \underline{h}_{jF} = \frac{L}{F} \sum_{j=1}^{Nc} x_j \underline{h}_{jL} + \sum_{j=1}^{Nc} y_j \underline{H}_{jV} - \frac{L}{F} \sum_{j=1}^{Nc} y_j \underline{H}_{jV}$$

reemplazando  $x_j$  y  $y_j$  por los valores encontrados en las ecuaciones (13) y (15),

$$\begin{aligned} \sum_{j=1}^{Nc} z_j \underline{h}_{jF} &= \frac{L}{F} \sum_{j=1}^{Nc} \frac{F}{L} \cdot \frac{z_j \underline{h}_{jL}}{k_j \left[ \frac{F}{L} - 1 \right] + 1} + \\ &+ \sum_{j=1}^{Nc} \frac{z_j \underline{H}_{jV}}{1 - \frac{L}{F} \left[ 1 - \frac{1}{k_j} \right]} - \frac{L}{F} \sum_{j=1}^{Nc} \frac{z_j \underline{H}_{jV}}{1 - \frac{L}{F} \left[ 1 - \frac{1}{k_j} \right]} \end{aligned}$$

simplificando y reorganizando,

$$\begin{aligned} \sum_{j=1}^{Nc} z_j \underline{h}_{jF} &= \sum_{j=1}^{Nc} \frac{\left[ \frac{1}{k_j} \right] z_j \underline{h}_{jL}}{\left[ \frac{F}{L} - 1 \right] + \frac{1}{k_j}} + \\ &+ \sum_{j=1}^{Nc} \frac{z_j \underline{H}_{jV}}{1 + \frac{L}{F} \left[ \frac{1}{k_j} - 1 \right]} - \sum_{j=1}^{Nc} \frac{z_j \underline{H}_{jV}}{\frac{F}{L} - 1 + \frac{1}{k_j}} \end{aligned}$$

ecuación que se designará como función  $g_2$

$$g_2 = \sum_{j=1}^{Nc} \frac{z_j H_{jV}}{1 + \frac{L}{F} \left[ \frac{1}{k_j} - 1 \right]} - \sum_{j=1}^{Nc} \frac{z_j H_{jV}}{\frac{F}{L} - 1 + \frac{1}{k_j}} +$$

$$+ \sum_{j=1}^{Nc} \frac{z_j h_{jL}}{\frac{F}{L} - 1 + \frac{1}{k_j}} - \sum_{j=1}^{Nc} z_j h_{jF} \quad (17)$$

Las funciones  $g_1$  y  $g_2$  constituyen un sistema de dos ecuaciones no lineales con dos incógnitas: la relación  $\frac{L}{F}$  y la temperatura pues los coeficientes involucrados en los cálculos de la constante de equilibrio y de las entalpías pueden encontrarse en libros y revistas (7) (8) (9).

Si un sistema de ecuaciones es no lineal, no se puede resolver mediante los métodos del álgebra de matrices sino que deben emplearse métodos basados en sustituciones sucesivas (método de Newton) u otras técnicas entre las que están las iterativas sin el uso de derivadas parciales (Gauss-Jacobi y Wegstein).

Para resolver el sistema de ecuaciones planteado se escogió el algoritmo de convergencia de Newton-Raphson por encontrarse entre los métodos iterativos con derivadas parciales que reducen el número de iteraciones necesario para la convergencia y por ser de los más conocidos (10).

Para aplicarlo se requieren las primeras derivadas de cada ecuación con respecto a cada variable involucrada; en este caso, la matriz resultante puede resolverse en forma directa por ser de segundo orden como se ejemplifica a continuación:

$$\frac{\partial g_1}{\partial T} \Delta T + \frac{\partial g_1}{\partial \frac{L}{F}} \Delta \frac{L}{F} = -g_1$$

$$\frac{\partial g_2}{\partial T} \Delta T + \frac{\partial g_2}{\partial \frac{L}{F}} \Delta \frac{L}{F} = -g_2$$

en donde el determinante es

$$D = \frac{\partial g_1}{\partial T} \frac{\partial g_2}{\partial \frac{L}{F}} - \frac{\partial g_1}{\partial \frac{L}{F}} \frac{\partial g_2}{\partial T}$$

por lo que, para cada variable, el incremento en la siguiente iteración estará dado por

$$\Delta T = \frac{g_2 \frac{\partial g_1}{\partial \frac{L}{F}} - g_1 \frac{\partial g_2}{\partial \frac{L}{F}}}{D}$$

$$\Delta \frac{L}{F} = \frac{g_1 \frac{\partial g_2}{\partial T} - g_2 \frac{\partial g_1}{\partial T}}{D}$$

Este tipo de solución plantea la necesidad de establecer valores iniciales para las variables desconocidas. En lo que se refiere a la relación del producto líquido con respecto a la alimentación,  $\frac{L}{F}$ , el valor más adecuado es considerar el caso extremo en el cual la presión de operación no fue suficiente para vaporizar nada de la alimentación, es decir,  $\frac{L}{F}^* = 1$ ; la suposición inicial de la temperatura se estimará entre las temperaturas del punto de burbuja y del punto de rocío.

La temperatura en el punto de burbuja, momento en que la mezcla líquida empieza a evaporarse, se evalúa de la siguiente manera:

la restricción de las fracciones mol en la fase vapor indica que

$$\sum_{j=1}^{Nc} y_j = 1$$

y como

$$y_j = k_j x_j$$

entonces

$$\sum_{j=1}^{Nc} k_j x_j = 1$$

en donde los valores de  $x_j$  estarán representados por las fracciones mol en la corriente de entrada al tanque flash; por lo tanto,

$$\sum_{j=1}^{Nc} k_j z_j = 1$$

de donde se obtiene la función

$$f = \sum_{j=1}^{Nc} k_j z_j - 1$$

ecuación que depende exclusivamente de la temperatura para una presión dada y que se resuelve por el método de convergencia de Newton,

$$T_b^{n+1} = T_b^n - \frac{f^n}{\left[\frac{df}{dT}\right]^n}$$

en donde  $n$  corresponde a cualquier iteración dentro del proceso de convergencia.

De manera similar es posible estimar la temperatura en el punto de rocío, instante en el que una mezcla en fase vapor comienza a condensarse,

$$\sum_{j=1}^{Nc} x_j = 1 \quad \text{y} \quad x_j = \frac{y_j}{k_j} \quad \therefore \quad \sum_{j=1}^{Nc} \frac{y_j}{k_j} = 1$$

y considerando que

$$y_j = z_j$$

$$f = \sum_{j=1}^{Nc} \frac{z_j}{k_j} - 1$$

ecuación en función únicamente de la temperatura a una presión dada que también se resuelve por el método de

convergencia de Newton,

$$T_r^{n+1} = T_r^n - \frac{f^n}{\left[\frac{df}{dT}\right]^n}$$

Con estas dos temperaturas se enmarca la temperatura de equilibrio; el criterio de inicialización la supone como el promedio aritmético de ellas, utilizándose el método de sustituciones sucesivas para calcular los siguientes valores supuestos de las variables siempre y cuando no hayan convergido,

$$T_{eq.}^{n+1} = T_{eq.}^n + \Delta T \quad \text{y} \quad \left[\frac{L}{F}\right]^{n+1} = \left[\frac{L}{F}\right]^n + \Delta \frac{L}{F}$$

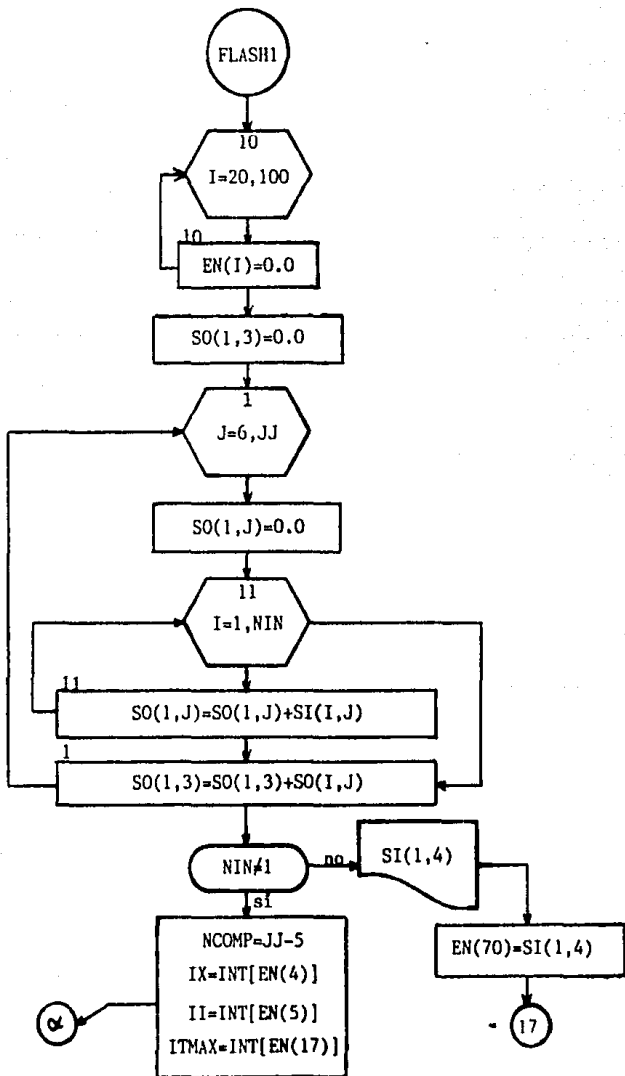
La convergencia de los valores solución se logra cuando

$$\frac{T_{eq.} - T_{eq.}^*}{T_{eq.}} < \epsilon_1 \quad \text{y} \quad \frac{\frac{L}{F} - \frac{L}{F}^*}{\frac{L}{F}} < \epsilon_2$$

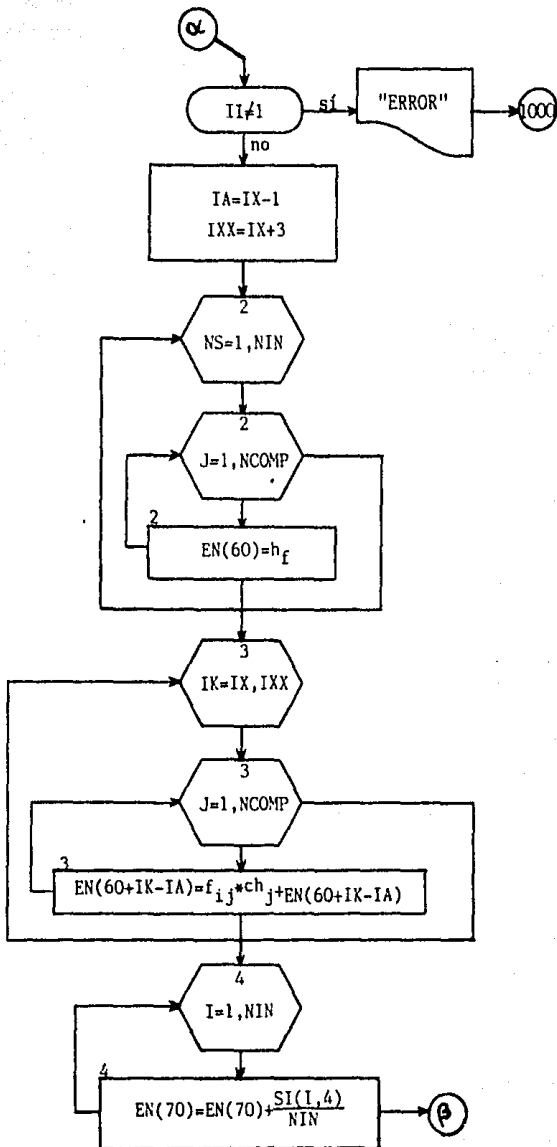
en donde  $\epsilon_1$  = tolerancia para la temperatura de equilibrio  
 y  $\epsilon_2$  = tolerancia para la relación  $\frac{L}{F}$ .

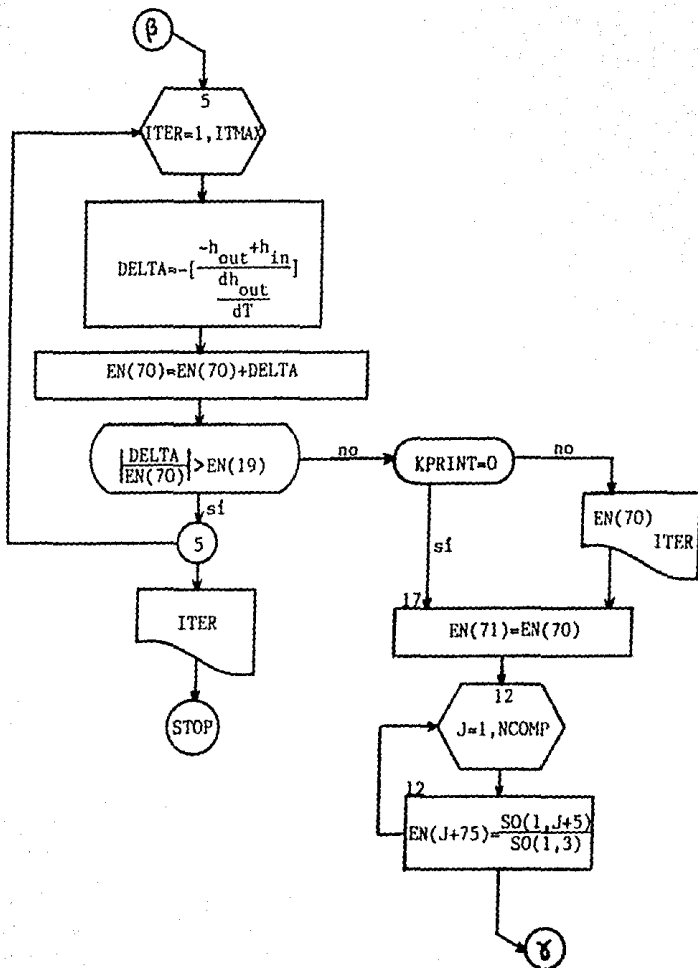
Al converger es posible calcular los flujos totales de vapor y líquido obtenidos mediante la destilación flash así como sus composiciones [ecuaciones (14) y (6)] y sus flujos por componente.

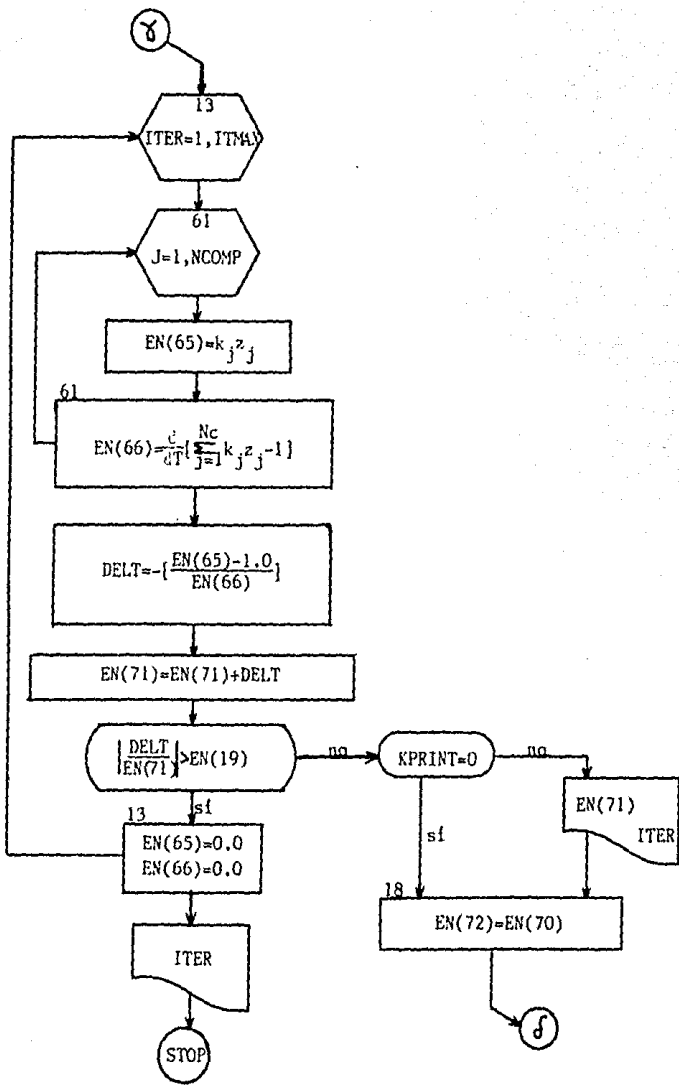
5.2.1 Diagrama de Flujo de la Subrutina FLASH1

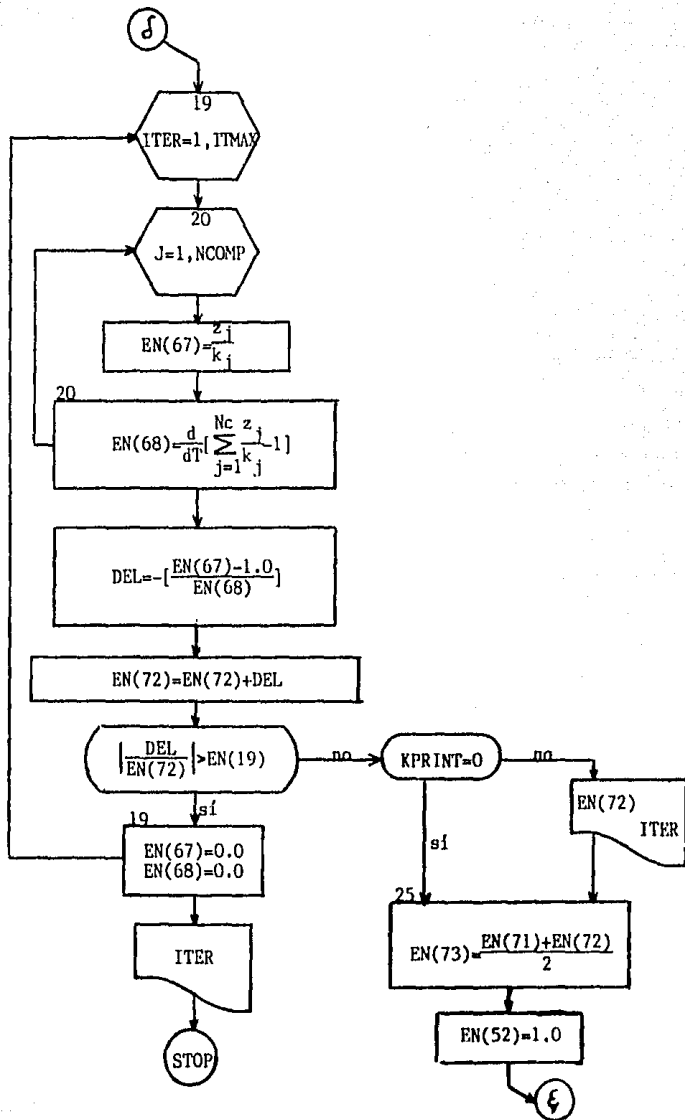


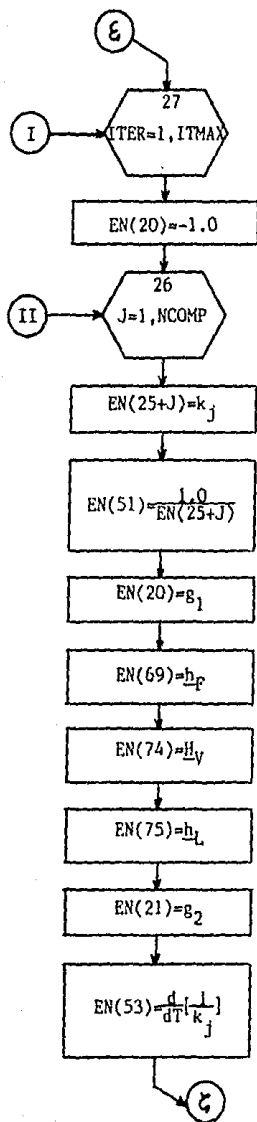


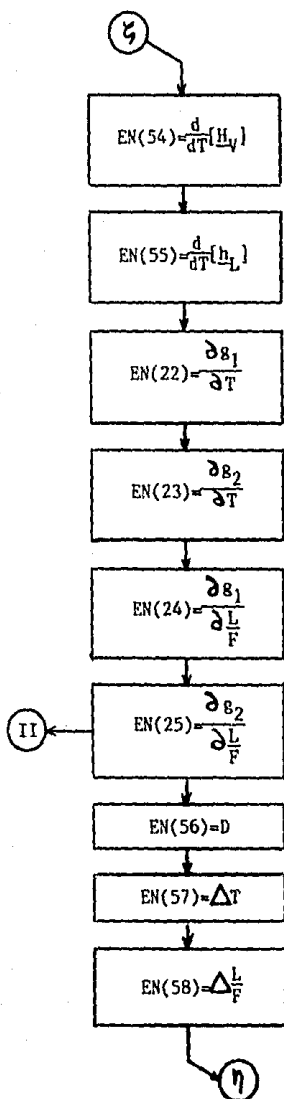


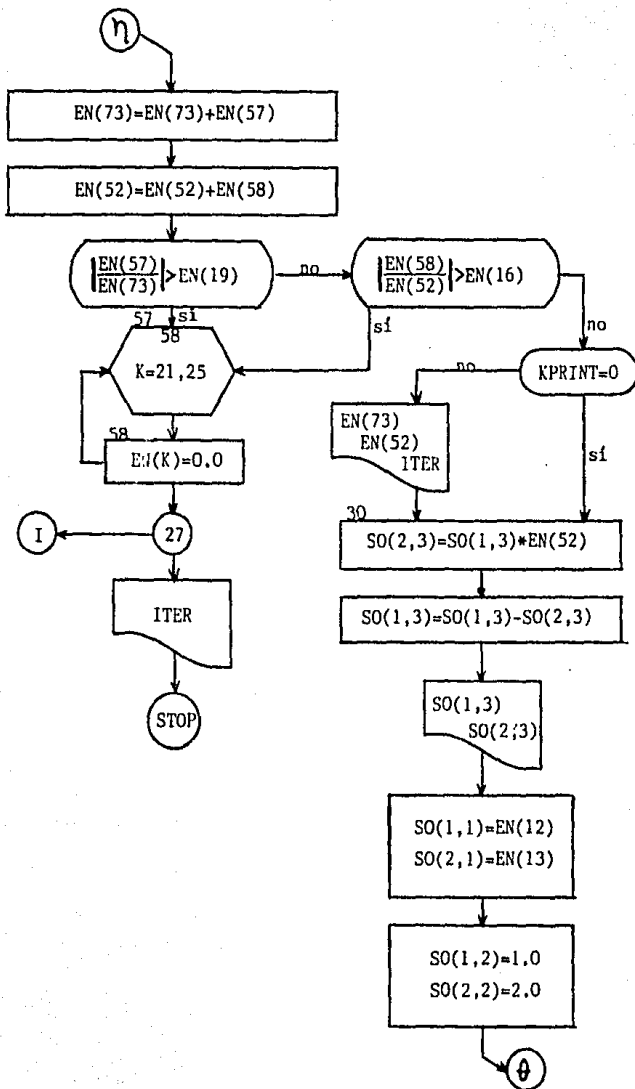


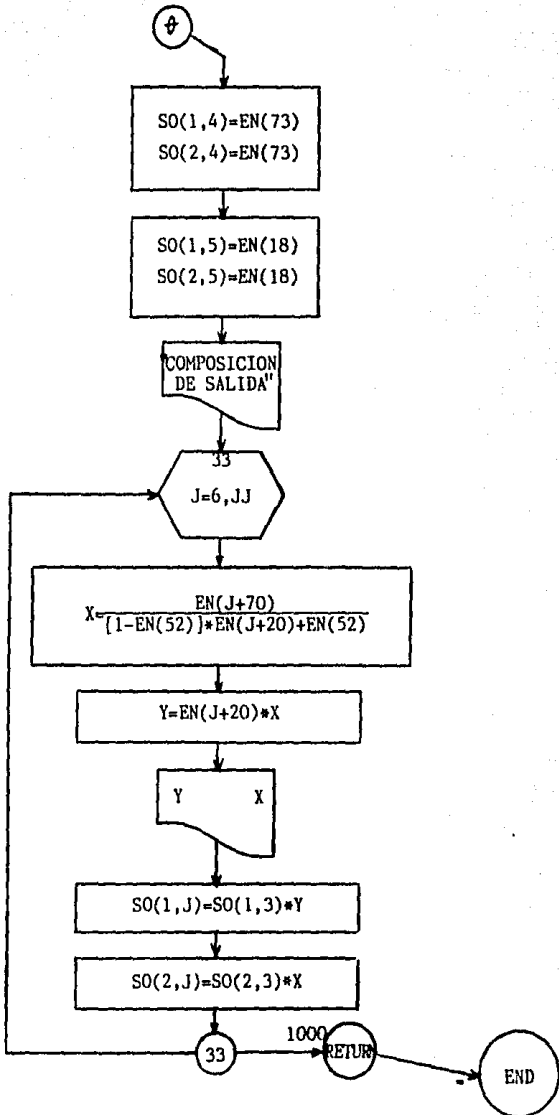














### VARIABLES PARTICULARES DE LA SUBROUTINA FLASH1

A continuación se listan, en el orden en que aparecen en el diagrama de flujo, las variables utilizadas por el módulo FLASH1:

EN(20) a EN(100)	= elementos del vector de parámetros de equipo 'EN' disponibles
SO(1,J)	= flujo total de entrada por componente (también de salida)
SI(1,J)	= flujo de entrada del componente J
SO(1,3)	= flujo total de salida
J	= número de componente
I, NS	= número de corriente
NIN	= número de corrientes de entrada
SI(1,4)	= temperatura de la corriente de alimentación única
EN(70)	= temperatura de entrada (alimentación única) o de mezclado (varias corrientes de alimentación)
NCOMP	= número de componentes presentes
IX	= valor entero de la fase de las corrientes de entrada, necesario para el cálculo de sus entalpías y, a través de un balance de energía, su temperatura de mezclado
II	= valor entero del grado de las funciones para la evaluación de las entalpías y las constantes de equilibrio
ITMAX	= número máximo de iteraciones

- EN(60) = "entalpía" de la alimentación  
I, J, NS, IK e  
ITER = contadores para los ciclos
- EN(61) a EN(64) = flujo del componente J en la corriente I multiplicado por el coeficiente entálpico correspondiente
- $f_{ij}$  = flujo del componente J en la corriente I
- $ch_j$  = coeficiente para el cálculo de las entalpías para el componente J; a, b, c y d para fase líquida y A, B, C, y D para fase vapor
- SI(I,4) = temperatura de la corriente de entrada I
- DELTA = decremento en la temperatura de mezclado (varias corrientes de alimentación)
- EN(19) = tolerancia para la temperatura
- KPRINT = control de impresión
- EN(J+75) = fracciones mol en la alimentación
- EN(65) = fracción mol en el vapor
- EN(66) = derivada con respecto a la temperatura de la sumatoria de las fracciones mol en el vapor menos uno
- EN(71) = temperatura de burbuja
- EN(72) = temperatura de rocío
- EN(67) = fracción mol en el líquido
- EN(68) = derivada con respecto a la tempe

- ratura de la sumatoria de las fracciones mol en el líquido menos uno
- DELT = decremento en la temperatura de burbuja
- DEL = decremento en la temperatura de rocío
- EN(73) = temperatura de equilibrio
- EN(52) = relación  $\frac{L}{F}$
- EN(20) = función  $g_1$  del desarrollo matemático
- EN(25+J) = constante de equilibrio para el componente J
- EN(51) = inverso de la constante de equilibrio
- EN(69) = entalpía de la alimentación
- EN(74) = entalpía del vapor
- EN(75) = entalpía del líquido
- EN(91) = función  $g_2$  del desarrollo matemático
- EN(53) = derivada con respecto a la temperatura del inverso de la constante de equilibrio
- EN(54) = derivada con respecto a la temperatura de la entalpía del vapor
- EN(55) = derivada con respecto a la temperatura de la entalpía del líquido
- EN(22) = derivada parcial con respecto a la temperatura de la función  $g_1$

- EN(23) = derivada parcial con respecto a la temperatura de la función  $g_2$
- EN(24) = derivada parcial con respecto a la relación  $\frac{L}{F}$  de la función  $g_1$
- EN(25) = derivada parcial con respecto a la relación  $\frac{L}{F}$  de la función  $g_2$
- EN(56) = determinante para la resolución del sistema de ecuaciones planteado en el desarrollo matemático
- EN(57) = incremento en la temperatura
- EN(58) = incremento en la relación  $\frac{L}{F}$
- EN(16) = tolerancia para la relación  $\frac{L}{F}$
- SO(2,3) = flujo total de líquido
- SO(1,3) = flujo total de vapor
- SO(1,1) = número de la corriente en fase vapor
- SO(2,1) = número de la corriente en fase líquida
- SO(I,2) = fase de la corriente; 1.0 para vapor y 2.0 para líquido
- SO(I,4) = temperatura de salida de la corriente I
- SO(I,5) = presión de salida de la corriente I
- EN(18) = presión de vaporización
- X = fracción mol en el líquido
- Y = fracción mol en el vapor
- SO(1,J) = flujo del componente J en la

corriente de vapor

$S_0(2,J)$

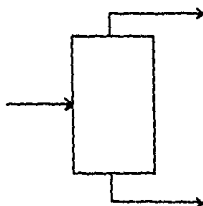
= flujo del componente J en la  
corriente de líquido

### 5.2.2 Módulo de Computación FLASH1.

Código

12

Símbolo



Esta subrutina simula el cálculo del flasheo adiabático de una solución multicomponente que entra al tanque en condiciones que permiten la generación de vapor; se considera que la primer corriente de salida corresponde al vapor [EN(12)] y la segunda, al producto líquido [EN(13)].

Es importante hacer énfasis en que la corriente de alimentación al vaporizador deber ser totalmente líquida [EN(4) = 7.0] y que los datos para el cálculo de las constantes de equilibrio y las entalpías son función de la temperatura [EN(5) = 1.0 por tratarse de polinomios de tercer grado] a una presión de operación dada [EN(18)].

Finalmente, debido al empleo de métodos de convergencia, deberán fijarse las tolerancias deseadas para la relación  $\frac{L}{F}$  [EN(16)] y para la temperatura de equilibrio [EN(19)] así como el número máximo de iteraciones a efectuar [EN(17)]. Los demás parámetros del módulo siguen el formato standard (ver Tabla 4.1).

Listado de la Subrutina FLASH1

## SUBROUTINE FLASH1

```

C
C *** LA SUBROUTINA FLASH1 REPRESENTA UN TANQUE DE DESTILACION
C *** FLASH ADIABATICA
C
C *** VECTOR "EN" ***
C EN( 1) A EN( 3) = FORMATO STANDARD
C EN( 4)          = FASE DE LA CORRIENTE DE ALIMENTACION
C EN( 5)          = GRADO DE LAS FUNCIONES PARA EL CALCULO
C                 DE ENTALPIAS Y CONSTANTES DE EQUILIBRIO
C EN( 6) A EN(11) = FORMATO STANDARD
C EN(12)          = NUMERO DE LA CORRIENTE EN FASE VAPOR
C EN(13)          = NUMERO DE LA CORRIENTE EN FASE LIQUIDA
C EN(14) Y EN(15) = CEROS
C EN(16)          = TOLERANCIA PARA LA RELACION L/F
C EN(17)          = NUMERO MAXIMO DE ITERACIONES
C EN(18)          = PRESION DE VAPORIZACION
C EN(19)          = TOLERANCIA PARA LA TEMPERATURA
C
C *** BLOQUE COMMON ***
C
COMMON /SET1/EN(100), JJ, NIN, NOUT, SI(4, 30), SO(4, 30), PP(15, 25)
COMMON /SET2/KFRINT
C
9 FORMAT(//, 12X, 'TEMPERATURA DE LA CORRIENTE DE ALIMENTACION =',
+ F10. 4)
6 FORMAT(//, 4X, 'TEMPERATURA OBTENIDA AL MEZCLARSE LAS
+ CORRIENTES DE ENTRADA =', F10. 4)
8 FORMAT(//, 11X, 'NUMERO DE ITERACIONES NECESARIAS PARA SU
+ CONVERGENCIA: ', I2)
14 FORMAT(//, 23X, 'TEMPERATURA DE BURBUJA =', F10. 4)
7 FORMAT(//, 7X, 'DESPUES DE ', I2, ' ITERACIONES, LA TEMPERATURA DE
+ MEZCLADO NO CONVERGE')
16 FORMAT(//, 8X, 'DESPUES DE ', I2, ' ITERACIONES, LA TEMPERATURA DE
+ BURBUJA NO CONVERGE')
22 FORMAT(//, 24X, 'TEMPERATURA DE ROCIO =', F10. 4)
24 FORMAT(//, 9X, 'DESPUES DE ', I2, ' ITERACIONES, LA TEMPERATURA DE
+ ROCIO NO CONVERGE')
28 FORMAT(//, 12X, 'TEMPERATURA DE EQUILIBRIO EN EL TANQUE FLASH =',
+ F10. 4)
29 FORMAT(//, 19X, 'RELACION L/F EN EL TANQUE FLASH =', F8. 5)
32 FORMAT(//, 18X, 'DESPUES DE ', I2, ' ITERACIONES, LA TEMPERATURA DE'
+ ', 14X, 'EQUILIBRIO Y LA RELACION L/F DEL FLASH, NO CONVERGEN')
34 FORMAT(//, 29X, 'COMPOSICION DE SALIDA', //, 24X, 'VAPOR', 19X,
+ 'LIQUIDO')
35 FORMAT(//, 24X, 'Y( ', I1, ') =', F8. 5, 3X, 'X( ', I1, ') =', F8. 5)
50 FORMAT(//, 25X, 'A LA SALIDA DEL TANQUE FLASH, //, 24X, 'FLUJO TOTAL
+ DE VAPOR =', F10. 3, //, 23X, 'FLUJO TOTAL DE LIQUIDO =', F10. 3)
51 FORMAT(//, 22X, 'ERROR, EL TIPO DE DATOS NO ES CUBICO')

```

```
C
C *** INICIALIZANDO VALORES Y SUMANDO TODOS LOS FLUJOS DE ENTRADA ***
C
  DO 10 I=20,100
10 EN(I)=0.0
   SO(1,3)=0.0
   DO 1 J=6, JJ
   SO(1,J)=0.0
   DO 11 I=1,NIN
11 SO(1,J)=SO(1,J)+SI(I,J)
   1 SO(1,3)=SO(1,3)+SO(1,J)
C
C *** DETERMINANDO LA TEMPERATURA DE ENTRADA ***
C
  IF(NIN.NE.1)GOTO 60
  WRITE(6,9)SI(1,4)
  EN(70)=SI(1,4)
  GOTO 17
60 NCOMP=JJ-5
   IX=INT(EN(4))
   II=INT(EN(5))
   ITMAX=INT(EN(17))
   IF(II.NE.1)GOTO 900
   IA=IX-1
   IXX=IX+3
   DO 2 NS=1,NIN
   DO 2 J=1,NCOMP
2 EN(60)=EN(60)+(PP(IX,J)+PP(IX+1,J)*SI(NS,4)+PP(IX+2,J)*SI(NS,4)
  **SI(NS,4)+PP(IX+3,J)*SI(NS,4)*SI(NS,4)*SI(NS,4)*SI(NS,4)+SI(NS,J+5)
   DO 3 IK=IX,IXX
   DO 3 J=1,NCOMP
3 EN(60+IK-IA)=SO(1,J+5)*PP(IX,J)+EN(60+IK-IA)
   DO 4 I=1,NIN
4 EN(70)=EN(70)+(SI(I,4)/NIN)
   DO 5 ITER=1,ITMAX
   DELTA=-((EN(60)+EN(61)+(EN(62)*EN(70))+(EN(63)*EN(70)*EN(70))+
  *(EN(64)*EN(70)+EN(70)*EN(70)))/(EN(62)+(2.0*EN(63)*EN(70))+3.0*
  *EN(64)+EN(70)+EN(70))
   EN(70)=EN(70)+DELTA
   IF(ABS(DELTA/EN(70)).GT.EN(19))GOTO 5
   IF(KPRINT.EQ.0)GOTO 17
   WRITE(6,6)EN(70)
   WRITE(6,8)ITER
   GOTO 17
5 CONTINUE
  WRITE(6,7)ITER
  STOP
C
C *** CALCULANDO EL PUNTO DE BURBUJA ***
C
17 EN(71)=EN(70)
```





```

DO 26 J=1, NCOMP
EN(25+J)=PP(2, J)+PP(3, J)*EN(73)+PP(4, J)*EN(73)+EN(73)+PP(5, J)*
*EN(73)+EN(73)+EN(73)
EN(51)=1.0/EN(25+J)
EN(20)=EN(20)+EN(J+75)/((1.0+EN(52))*(EN(51)-1.0))
EN(69)=PP(7, J)+PP(8, J)+EN(70)+PP(9, J)*EN(70)+EN(70)+PP(10, J)
**EN(70)+EN(70)+EN(70)
EN(74)=PP(12, J)+PP(13, J)+EN(73)+PP(14, J)*EN(73)+EN(73)+PP(15, J)
**EN(73)+EN(73)+EN(73)
EN(75)=PP(7, J)+PP(8, J)*EN(73)+PP(9, J)*EN(73)+EN(73)+PP(10, J)*
*EN(73)+EN(73)+EN(73)
EN(21)=EN(21)+EN(J+75)+EN(74)/((1.0+EN(52))*(EN(51)-1.0))-
*EN(J+75)+EN(74)/((1.0/EN(52)+EN(51)-1.0)+EN(51)+EN(J+75)+EN(75)/
*(1.0/EN(52)-1.0+EN(51))-EN(J+75)+EN(69)
EN(53)=-((PP(3, J)+2.0*PP(4, J)+EN(73)+3.0*PP(5, J)+EN(73)+EN(73))/
*(EN(50)+EN(50))
EN(54)=PP(13, J)+2.0*PP(14, J)+EN(73)+3.0*PP(15, J)+EN(73)+EN(73)
EN(55)=PP(8, J)+2.0*PP(9, J)+EN(73)+3.0*PP(10, J)+EN(73)+EN(73)
EN(22)=EN(22)-EN(52)+EN(J+75)+EN(53)/((1.0+EN(52))*(EN(51)-1.0))
***2)
EN(23)=EN(23)+((1.0+EN(52))*(EN(51)-1.0))*EN(J+75)+EN(54)-
*EN(J+75)+EN(74)+EN(52)+EN(53)/((1.0+EN(52))*(EN(51)-1.0)**2))-
*((1.0/EN(52)+EN(51)-1.0)*EN(J+75)+EN(54)-EN(J+75)+EN(74)*
*EN(53))/((1.0/EN(52)+EN(51)-1.0)**2))+((1.0/EN(52)-1.0+
*EN(51))*EN(J+75)+*(EN(51)+EN(55)+EN(55)+EN(75)+EN(53))-EN(51)*
*EN(J+75)+EN(42)+EN(53))/((1.0/EN(52)-1.0+EN(51)**2))
EN(24)=EN(24)-EN(J+75)*EN(51)-1.0)/((1.0+EN(52))*(EN(51)-1.0))
***2)
EN(25)=EN(25)-EN(J+75)+EN(74)*EN(51)-1.0)/((1.0+EN(52))*(EN(51)-
*1.0)**2)+EN(J+75)+EN(74)*(-1.0/EN(52)**2))/((1.0/EN(52)+
*EN(51)-1.0)**2)-EN(51)+EN(J+75)+EN(75)*(-1.0/EN(52)**2))/
*((1.0/EN(52)-1.0+EN(51)**2))
26 CONTINUE
EN(56)=EN(22)+EN(25)-EN(24)+EN(23)
EN(57)=(EN(21)+EN(24)-EN(20)+EN(25))/EN(56)
EN(58)=(EN(20)+EN(23)-EN(21)+EN(22))/EN(56)
EN(73)=EN(73)+EN(57)
EN(52)=EN(52)+EN(58)
IF(ABS(EN(57)/EN(73)).GT.EN(19))GOTO 57
IF(ABS(EN(58)/EN(52)).GT.EN(16))GOTO 57
IF(NPRINT.EQ.0)GOTO 30
WRITE(6, 28)EN(73)
WRITE(6, 29)EN(52)
WRITE(6, 8)ITER
GOTO 30
57 DO 58 K=21, 25
58 EN(K)=0.0
27 CONTINUE
WRITE(6, 32)ITER
STOP

```

```
C
C *** CALCULANDO LOS FLUJOS TOTALES DE VAPOR Y LIQUIDO Y FIJANDO
C   LOS DEMAS PARAMETROS DE LAS CORRIENTES DE SALIDA ***
C
30 SO(2,3)=SO(1,3)*EN(52)
   SO(1,3)=SO(1,3)-SO(2,3)
   WRITE(6,50)SO(1,3),SO(2,3)
   SO(1,1)=EN(12)
   SO(2,1)=EN(13)
   SO(1,2)=1.0
   SO(2,2)=2.0
   SO(1,4)=EN(73)
   SO(2,4)=EN(73)
   SO(1,5)=EN(18)
   SO(2,5)=EN(18)

C
C *** CALCULANDO LAS COMPOSICIONES DE SALIDA Y LOS FLUJOS DE CADA
C   COMPONENTE ***
C
WRITE(6,34)
DO 33 J=6,JJ
  I=J-5
  X=EN(J+20)/(1.0-EN(52))+EN(J+20)+EN(52)
  Y=EN(26+J)*X
  WRITE(6,35)I,Y,I,X
  SO(1,J)=SO(1,3)*Y
  SO(2,J)=SO(2,3)*X
33 CONTINUE
GOTO 1000
900 WRITE(6,51)
1000 RETURN
END
```

Referencias.

1. Balzhiser, R. E. et al.; "Termodinámica Química para Ingenieros", [Prentice-Hall Internacional, España (1979)].
2. Himmelblau, D. M.; "Principios y Cálculos Básicos de la Ingeniería Química", [C. E. C. S. A., México (1972)].
3. Hougen, O. A. et al.; "Chemical Process Principles Part 1 - Material & Energy Balances", [John Wiley & Sons Inc., U. S. A. (1954)].
4. Myers, A. L. & Seider, W. D.; "Introduction to Chemical Engineering and Computer Calculations", [Prentice Hall, U. S. A. (1976)].
5. Treybal, R. E.; "Mass-Transfer Operations", [McGraw-Hill Inc., U. S. A. (1980)].
6. King, J. C.; "Separation Processes", [McGraw-Hill Inc., U. S. A. (1980)].
7. Lyster, W. N. et al.; "Digital Computer Used to ... Figure Distillation This New Way", [Petroleum Refiner, 38 (1959), No 7)].
8. Maxwell, J. B. "Data Book on Hydrocarbons", [Van Nostrand, U. S. A. (1950)].
9. Robinson, C. S. & Gilliland, E. R. "Elements of Fractional Distillation", [McGraw-Hill Inc., U. S. A. (1951)].
10. Carnahan, B. et al.; "Applied Numerical Methods", [John Wiley & Sons Inc., U.S. A. (1960)].

## CAPITULO VI

### MANUAL DE OPERACION DEL PROGRAMA 'GEMCS'

El programa de simulación de procesos químicos 'GEMCS' se encuentra implementado en dos diskettes grabados por los dos lados. Dos de ellos, FORT1 y FORT2, contienen el sistema operativo para el lenguaje FORTRAN, el programa principal, las subrutinas compiladas y los archivos de datos para los distintos ejemplos. El ter ce ro, TEXT, contiene los textos tanto del programa maestro como de las subrutinas que fue necesario segmentar para ocupar menos memoria. Finalmente, el cuarto se encuentra escrito en BASIC y contiene este manual ('OPBAS').

#### Tutorial

##### Opciones:

- 6.1 Como correr el programa.
- 6.2 Archivos de datos.
- 6.3 Modificación de textos.\*
- 6.4 Adición de nuevas subrutinas.\*
- 6.5 Todas las anteriores.

##### \*Nota:

Para estos puntos es necesario saber FORTRAN.

##### IMPORTANTE:

Para cualquier cambio que desees hacer, copia los diskettes originales en un par de diskettes tuyos y efectúalo en ellos. Por favor conserva protegidos los dis ket tes originales — no efectúes modificaciones en ellos.

Muchas gracias.

### 6.1 Como Correr el Programa de Simulación de Procesos 'GEMCS'.

Tomar los diskettes e introducirlos en los impulsores de disco (drives) al lado de la microcomputadora de manera que el denominado FORT2 quede en el #4 y el FORT1, en el #5.

Encender la máquina y la pantalla.

Al conectarse el sistema, la máquina lee los discos y en la pantalla aparecerán los comandos del sistema operativo de FORTRAN.

\*Así, para iniciar cada corrida se utilizará la siguiente instrucción:

[ X ] #5:GEMCS

es decir, oprimir la tecla [ X ], la máquina preguntará:

EXECUTE WHAT FILE?

responder:

#5:GEMCS

enseguida aparecerá un letrero de introducción.

Después de esta bienvenida, oprimir la tecla [ RETURN ]; a continuación aparecerá en la pantalla la siguiente pregunta:

DESEA CORRER EL PROGRAMA...? (S/N)

si la respuesta es negativa, se regresará al sistema operativo del lenguaje FORTRAN; en nuestro caso, presionar la tecla [ S ] y después [ RETURN ].

El programa empezará a correr inquiriendo:

RESULTADOS EN LA IMPRESORA...? (S/N)

Con [ N ] y [ RETURN ] la corrida se podrá seguir en pantalla; usar [ CTRL A ] para tener acceso al lado derecho de ésta.

Con [ S ] y [ RETURN ] se obtiene la corrida impresa.

Los datos que se utilizarán en esta corrida ejemplo corresponden a la Ilustración I en la que intervienen los módulos del reactor (REACT1) y los tres separadores (SEPA01, SEPA02 y SEPA03).

Para poder familiarizarse con la forma de ejecución del programa 'GEMCS', se pidió la impresión incrementada, es decir, la que incluye tanto resultados parciales como finales.

Al terminar de correr el programa (END OF EXECUTION), la máquina imprimirá en la pantalla:

```
STOP  
PROGRAM TERMINATED.
```

apareciendo los comandos del sistema operativo en su parte superior.

Si se quiere efectuar una corrida con datos diferentes, hacer referencia a la opción A de la Sección 2.

## 6.2 Datos.

El archivo de datos que se use en las corridas del programa deberá ser un texto, que se encuentre en FORT1 (drive #5), bajo el nombre de 'DATOS'.

En este caso, 'DATOS' corresponde a los datos representativos del problema planteado en la Ilustración I.

En todo archivo 'DATOS' a utilizar deberán encontrarse almacenados los parámetros necesarios para la operación de las subrutinas del programa principal y las subrutinas adicionales de acuerdo a un formato específico:

valores enteros:       \* = I3  
valores reales:       \*\* = F10.3, alimentados de 5 en 5  
                      \*\*\* = F17.11, alimentados de 4 en 4

Para el programa maestro:

```
PROGRAM MASTER  
18A4 ITITLE  
* KPRINT  
* NCALC, NOCOMP  
* LLST(I), I=1, NCALC  
* MSN  
* NSR  
** SN(I, J), I=1, NSR; J=1, JJ  
* NOE  
** EN(N), N=1, EN(3)  
* NP  
*** PP(N, I), N=1, NP; I=1, NOCOMP
```

Para las subrutinas de equipo:  
(valores fuera del formato standard)



JUNCO1

\*\* EN( 2) = 1.0  
\*\* EN( 3) = 21.0  
\*\* EN(16) = temperatura de salida  
\*\* EN(17) = presión de salida  
\*\* EN(18) a EN(17+NOUT) = proporción del flujo total  
de entrada en cada corriente  
de salida, si no se requiere  
útese 0.0

CONTL1

\*\* EN( 2) = 2.0  
\*\* EN( 3) = 20.0  
\*\* EN( 4) = 0.0  
\*\* EN( 5) = control de impresión  
\*\* EN( 6) a EN(15) = 0.0  
\*\* EN(16) = número de módulos controlados  
por CONTL1  
\*\* EN(17) = número máximo de iteraciones  
\*\* EN(18) = número de la corriente para  
la prueba de convergencia  
\*\* EN(19) = tolerancia fraccional para  
dicha prueba  
\*\* EN(20) = número de identificación  
de la iteración

JUNCO2

\*\* EN( 2) = 3.0  
\*\* EN( 3) = 21.0  
\*\* EN( 4) = 7.0 para el cálculo de  
entalpías de corrientes  
en fase líquida  
\*\* EN( 4) = 12.0 para el cálculo de  
entalpías de corrientes  
en fase vapor

JUNCO2  
(continuación)

- \*\* EN( 5) = 1.0 tipo de datos del polinomio (función cúbica)
- \*\* EN(16) a EN(19) = proporción del flujo total de entrada en cada corriente de salida, si no se requiere úsese 0.0
- \*\* EN(20) = tolerancia fraccional para la prueba de convergencia
- \*\* EN(21) = número máximo de iteraciones

SEPA01

- \*\* EN( 2) = 5.0
- \*\* EN( 3) = 19.0+NOCOMP
- \*\* EN(16) = temperatura de salida de la corriente EN(12)
- \*\* EN(17) = temperatura de salida de la corriente EN(13)
- \*\* EN(18) = presión de salida en EN(12)
- \*\* EN(19) = presión de salida en EN(13)
- \*\* EN(20) a EN(19+NOCOMP) = fracción de la alimentación, para cada componente, en la corriente especificada en EN(12)

FLASH1

- \*\* EN( 2) = 12.0
- \*\* EN( 3) = 19.0
- \*\* EN( 4) = 7.0 alimentación líquida exclusivamente
- \*\* EN( 5) = 1.0 tipo de datos de los polinomios (funciones cúbicas)
- \*\* EN(12) = número de la corriente en fase vapor

FLASH1

(continuación)

- \*\* EN(13) = número de la corriente en fase líquida
- \*\* EN(16) = tolerancia para la relación  $\frac{L}{F}$
- \*\* EN(17) = número máximo de iteraciones
- \*\* EN(18) = presión de vaporización
- \*\* EN(19) = tolerancia para la temperatura de equilibrio

SEPA02

- \*\* EN( 2) = 21.0
- \*\* EN( 3) = 17.0+NOCOMP
- \*\* EN(16) = porcentaje de recuperación del componente especificado en en la corriente indicada por EN(12)
- \*\* EN(18) a EN(17+NOCOMP) = composición de todos los componentes en la corriente señalada en EN(12)

SEPA03

- \*\* EN( 2) = 22.0
- \*\* EN( 3) = 17.0+NOCOMP
- \*\* EN(16) = composición del componente especificado en en la corriente señalada en EN(13)
- \*\* EN(18) a EN(17+NOCOMP) = composición de todos los componentes en la corriente indicada por EN(12)

REACT1

\*\* EN( 2) = 23.0  
\*\* EN( 3) = 17.0+NOCOMP  
\*\* EN(16) = conversión del componente  
especificado en  
\*\* EN(17) (reactivo limitante)  
\*\* EN(18) a EN(17+NOCOMP) = factores estequiométricos  
de la reacción en base al  
reactivo limitante [EN(17)]

Comentarios:

1. Si se quiere seguir el desarrollo de la corrida de cerca, será necesario que a KPRINT se le asigne el valor de 1; si fuera 0, sólo aparecerán resultados finales.

2. No debe olvidarse que NCALC debe alimentarse como número negativo para problemas directos y como positivo para problemas con recirculación.

Es muy importante que al elaborar un nuevo conjunto de datos se tenga especial cuidado en que las unidades sean consistentes entre sí y compatibles con las subrutinas módulo o unidades computacionales.

En ocasiones el espacio destinado a la memoria no es suficiente para permitir que la corrida se complete; cuando ésto suceda, vuelve a cargar el sistema y al aparecer los comandos oprime [ S ], la máquina preguntará:

SWAPPING IS OFF  
TOGGLE SWAPPING?

responder:

Y

con esto se cambiará la partición y se podrá disponer de más memoria para las corridas extensas.

En esta sección existen dos opciones:

- A. Cambio del archivo de datos por:  
DATOS2 (JUNCO1 y CONTL1) o  
DATOS3 (JUNCO2 y FLASH1).
- B. Generación de un nuevo archivo de datos:  
DATOSU.TEXT

### 6.2.1 Cambio del Archivo de Datos.

Los comandos que aparecen en la parte superior de la pantalla una vez cargado el sistema operativo del lenguaje FORTRAN son:

COMMAND: E(DIT), R(UN), F(ILE), C(OMP), L(INK),  
X(ECUTE), A(SSEM), D(EBUG)?

\*De entre ellos, presionar [ F ] (FILER), sección que contiene el listado de los archivos, textos y programas compilados así como del sistema operativo y que tiene sus propios comandos:

FILER: G, S, N, L, R, C, T, D, Q

Por lo tanto,

presionar            respuesta computadora            respuesta usuario

[L] o [E]    DIRECTORY LISTING OF?            #DRIVE:

          Listado del contenido del diskette

(Objetivo: checar el estado del disco y su contenido; con [ L ] se obtiene un listado sencillo mientras que con [ E ], uno más extenso que incluye el número de bloques de cada programa indicando también el número de éstos que permanecen libres como { UNUSED }.)

[C]            CHANGE?                            #5:DATOS  
          CHANGE TO WHAT?                #5:DATOS1  
          FORT1:DATOS    →    DATOS1

(Objetivo: cambiar 'DATOS' a 'DATOS1', archivo de datos)

correspondiente a la Ilustración 1).

[C] CHANGE? #DRIVE:DATOSX  
X = 2, 3, U.TEXT

CHANGE TO WHAT? #5:DATOS

FORT1:DATOSX → DATOS

(si el archivo estaba en el drive #5) o

FORT2:DATOSX → DATOS

(si el archivo estaba en el drive #4)

(Objetivo: cambiar el archivo elegido a 'DATOS' para correr el programa con él.)

[L] o [E] DIRECTORY LISTING OF? #DRIVE:

Listado del contenido del diskette

(Objetivo: verificar cambios.)

Otras instrucciones que te pueden ser útiles son:

[K] CRUNCH? #DRIVE:

FROM END OF DISK, BLOCK 280? Y

NOMBRE DEL DISCO: CRUNCHED

(Objetivo: eliminar espacios entre programas y archivos dentro del disco.)

[R] REMOVE? #DRIVE:ARCHIVO

NOMBRE DISCO: ARCHIVO → REMOVED

UPDATE DIRECTORY? Y

(lo elimina)

N

(anula la instrucción)

(Objetivo: borrar un archivo del diskette para generar espacio para un nuevo archivo.)

[T] TRANSFER? #DRIVE:ARCHIVO

TO WHERE? #DRIVE:ARCHIVO

NOMBRE DISCO: ARCHIVO →

NOMBRE DISCO: ARCHIVO

(Objetivo: cambiar de un diskette a otro (drives #4

y #5), imprimirlo (#6) o listarlo en pantalla (#1).)

Finalmente,

[Q] Comandos del sistema operativo  
(Objetivo: Salir de FILER y regresar a los comandos principales que aparecerán en la parte superior de la pantalla.)

Las instrucciones mencionadas no sólo se aplican a los archivos sino también a textos y versiones compiladas.

Para correr el programa con los nuevos datos referirse a la Sección 1.

El procedimiento puede efectuarse varias veces por lo que no hay que olvidar cuál es el archivo contenido en 'DATOS' para poder reasignarle su nombre al llevar a cabo los cambios.

6.2.2 Generación, por Parte del Usuario, de un Nuevo Archivo de Datos que Represente un Problema que él Desea Resolver.

Como se mencionó en la introducción de esta sección, los parámetros necesarios para la operación de las subrutinas del programa principal y las subrutinas adicionales se encuentran almacenados en los archivos 'DATOS' de acuerdo al formato especificado.

Para mayor claridad se recomienda leer los capítulos IV y V y estudiar las 3 corridas presentadas en el capítulo VII de la tesis 'Estudio, Complementación y Aplicación de un Programa de Simulación de Procesos'.





[CTRL C] Comandos del editor Salir de cada comando.

[Q] QUIT: Salir del editor;  
de las opciones planteadas escoger:

W(RITE TO A FILE NAME AND RETURN) [W]  
NAME OF OUTPUT FILE [CTRL A] #5:DATOSU.TEXT

Es muy importante que se grabe como texto pues si lo hace como DATOSU.DATA, el programa no correrá.

WRITING...

YOUR FILE IS # BYTES LONG.

DO YOU WANT TO E(XIT FROM) OR

R(ETURN TO) THE EDITOR? [E]

Una vez que aparezcan los comandos del sistema operativo, referirse al punto marcado con asterisco dentro de la Sección 2.1 (Sección 2, opción A) para checar que el archivo se haya registrado adecuadamente en el disco y, después, transformarlo en 'DATOS'.

Para correr el programa con el nuevo archivo ver la Sección 1.

### 6.3 Modificación de Textos.

Entrar a FILER ([ F ]) y transferir ([ T ]) el texto de interés a FORT1 o FORT2 (donde tengas espacio), editar lo ([ E ]) y modificarlo.

Regresar al sistema operativo para compilarlo ([ C ]) y eslabonarlo ([ L ]) al programa principal que deberá quedar, bajo el título de 'GEMCS', en FORT1.

### 6.4 Adición de Nuevas Subrutinas.

Si se quiere agregar un nuevo módulo para la resolución de una operación unitaria no considerada en este trabajo se deberá elaborar su unidad computacional correspondiente, codificar la subrutina y teclear el texto teniendo en cuenta que la nueva subrutina deberá ser compatible con el resto del programa, es decir, deberá considerar las corrientes a tratar como SI(I,J) desde I = 1 hasta I = 4 corrientes de entrada y desde J = 1 hasta J = NOCOMP+5. Una vez resuelto el modelo matemático, se deberán dar instrucciones de salida mediante SO(I,J) corrientes, con los mismos límites que para las de entrada, de manera que queden definidas todas las corrientes que abandonan el equipo para poderlas utilizar como datos iniciales en otra sección del proceso o bien, como resultado de la simulación.

El texto tecleado deberá compilarse y anexarse al programa maestro mediante:

```
$USES UNOMBRE IN NOMBRE.CODE OVERLAY      (DRIVE #4)  
$USES UNOMBRE IN #5:NOMBRE.CODE OVERLAY    (DRIVE #5)
```

asignándole alguna de las etiquetas libres en la subrutina 'MODULE'.

Compilar ([ C ]), en el diskette correspondiente, y eslabonar ([ L ]) todo el programa, en FORT1, bajo el nombre de 'GEMCS'.

Universidad La Salle  
Manual de Operación del Programa 'GEMCS'  
Perteneiente a la Tesis:  
Estudio, Complementación y Aplicación de  
un Programa de Simulación de Procesos  
Leonor Doris Herreramoro Gómez    Ingeniería Química    1985

## CAPITULO VII

### APLICACION DEL PROGRAMA DE SIMULACION DE PROCESOS QUIMICOS 'GEMCS'

Los ejemplos subsecuentes corresponden a estudios de aplicación pequeños pero valiosos para la ilustración de la flexibilidad del sistema GEMCS para estudiantes universitarios y nuevos usuarios del acercamiento modular.

Así, su objetivo es presentar en forma clara y concisa el manejo del programa de modo que el usuario pueda emplearlo sin necesidad de conocer detalladamente su contenido sino bastándole con considerar correctamente las reglas para su aplicación a problemas particulares.

Conviene comentar que puesto que el cálculo de las recirculaciones de un proceso es la etapa que consume más tiempo de computación y memoria dentro del estudio de simulación y como la convergencia se logra mediante el método de substituciones sucesivas por su simplicidad y efectividad, sólo se incluye un sencillo ejemplo de la aplicación de la subrutina CONTL1.

Por otro lado, es importante que una vez elaborado el diagrama de flujo del proceso a simular, se numeren las corrientes que intervienen en el mismo así como los equipos que conectan para poder establecer adecuadamente la secuencia a seguir durante la ejecución del programa (generación del diagrama de flujo de información). No hay que olvidar que para que ésto sea posible el archivo de datos deberá estar completo y de acuerdo a los formatos correspondientes.

Ilustración 1.-

APLICACION DE LAS SUBROUTINAS REACT1, SEPA01, SEPA02 Y  
SEPA03 DEL PROGRAMA DE SIMULACION DE PROCESOS QUIMICOS  
'GEMCS'

Se desea simular, mediante balances de materia, un proceso en el que intervienen un reactor y tres separadores que operan simultáneamente; en el proceso, cuyo diagrama de flujo de información se muestra en la Figura 7.1, se manejan cuatro componentes denominados A, B, C y D.

La corriente de alimentación (líquida) al reactor, primer equipo del proceso en estudio, contiene  $30 \frac{\text{kg mol}}{\text{hr}}$  de B,  $40 \frac{\text{kg mol}}{\text{hr}}$  de C y  $30 \frac{\text{kg mol}}{\text{hr}}$  de D que reaccionarán de acuerdo a la siguiente ecuación:



La conversión de B en esta reacción es del 50% y no se presentan cambios de fase.

Del reactor el flujo se dirige al primer separador (equipo II) que está diseñado de manera que se espera que por el fondo se obtengan las siguientes fracciones, para cada componente, del flujo de entrada:

componente	A	B	C	D
fracción del flujo de entrada	0.90	0.80	0.80	0.05

Esta corriente del fondo se introduce a un segundo separador (equipo III) en donde se desea recuperar el 95% del componente C por el domo, corriente de productos finales cuya composición deberá ser:

componente	A	B	C	D
composición	0.01	0.02	0.90	0.07

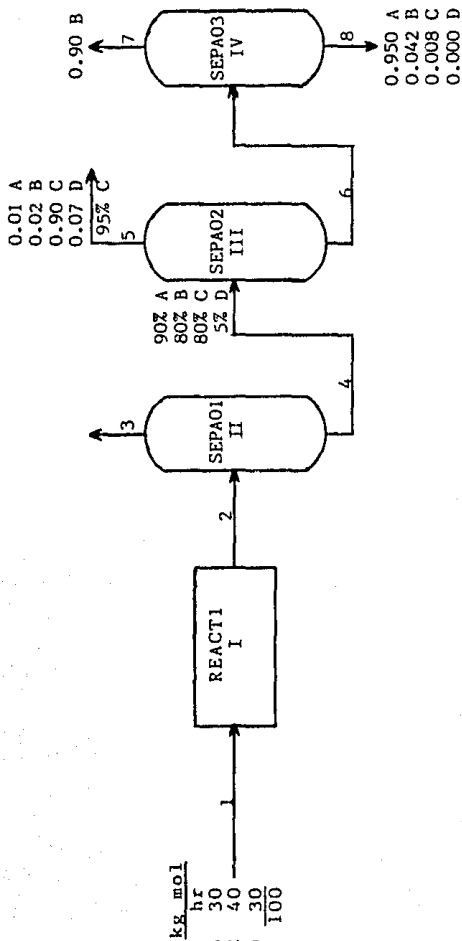
La corriente del fondo se alimenta al tercer tanque de separación (equipo IV) que permite obtener un 90% del componente B por el domo mientras que por la corriente de los fondos las composiciones esperadas son:

componente	A	B	C	D
composición	0.950	0.042	0.008	0.000

Con toda esta información, se desea evaluar los flujos totales y por componente y las características de todas las corrientes que intervienen en el proceso.

Figura 7.1

DIAGRAMA DE FLUJO DE INFORMACION DE LA ILUSTRACION I



Base de Datos Correspondiente a la Ilustración I

APLICACION DEL PROGRAMA "GENCS" USANDO REACT1, SEPA01, SEPA02, SEPA03

1					
-4	4				
1	2	3	4		
8					
1					
1.000	2.000	100.000	0.000	0.000	
0.000	30.000	40.000	30.000		
4					
1.000	23.000	21.000	0.000	0.000	
1.000	1.000	0.000	0.000	0.000	
1.000	2.000	0.000	0.000	0.000	
0.500	2.000	3.000	-1.000	-2.000	
0.000					
2.000	5.000	23.000	0.000	0.000	
1.000	2.000	0.000	0.000	0.000	
2.000	4.000	3.000	0.000	0.000	
0.000	0.000	0.000	0.000	0.900	
0.000	0.000	0.050			
3.000	21.000	21.000	0.000	0.000	
1.000	4.000	0.000	0.000	0.000	
2.000	5.000	6.000	0.000	0.000	
0.950	3.000	0.010	0.020	0.900	
0.070					
0.000	22.000	21.000	0.000	0.000	
1.000	6.000	0.000	0.000	0.000	
2.000	8.000	7.000	0.000	0.000	
0.900	2.000	0.950	0.042	0.000	
0.000					

0





	FINAL OUTPUT STREAMS FOR EQUIPMENT UNIT 1			
2.000	.000	100.000	.000	.000
45.000	15.000	10.000	30.000	
	EQUIPMENT PARAMETERS FOR UNIT 2			
2.000	5.000	20.000	.000	.000
1.000	2.000	.000	.000	.000
2.000	4.000	3.000	.000	.000
.000	.000	.000	.000	.900
.000	.000	.050		
	INPUT STREAMS FOR EQUIPMENT UNIT 2			
2.000	.000	100.000	.000	.000
45.000	15.000	10.000	30.000	
	FINAL OUTPUT STREAMS FOR EQUIPMENT UNIT 2			
4.000	.000	62.000	.000	.000
40.500	12.000	8.000	1.500	
3.000	.000	38.000	.000	.000
4.500	3.000	2.000	28.500	
	EQUIPMENT PARAMETERS FOR UNIT 3			
3.000	21.000	21.000	.000	.000
1.000	4.000	.000	.000	.000
2.000	5.000	6.000	.000	.000
.950	3.000	.010	.020	.900
.070				
	INPUT STREAMS FOR EQUIPMENT UNIT 3			
4.000	.000	62.000	.000	.000
40.500	12.000	8.000	1.500	
	FINAL OUTPUT STREAMS FOR EQUIPMENT UNIT 3			
5.000	.000	8.444	.000	.000
.084	.169	7.600	.591	
6.000	.000	53.556	.000	.000
40.416	11.831	.400	.909	
	EQUIPMENT PARAMETERS FOR UNIT 4			
4.000	22.000	21.000	.000	.000
1.000	6.000	.000	.000	.000
2.000	8.000	7.000	.000	.000
.900	2.000	.950	.042	.000
.000				
	INPUT STREAMS FOR EQUIPMENT UNIT 4			
6.000	.000	53.556	.000	.000
40.416	11.831	.400	.909	
	FINAL OUTPUT STREAMS FOR EQUIPMENT UNIT 4			
8.000	.000	42.388	.000	.000
40.269	1.790	.339	.000	
7.000	.000	11.168	.000	.000
147	10.051	.061	.909	

END OF EXECUTION

Análisis de los Resultados.-

En el reactor, a pesar de la baja conversión, se obtiene una concentración del 45% de A en el efluente debido a la favorable relación estequiométrica.

En el primer separador se tiende a eliminar gran parte del componente D de los fondos al concentrarse éste en la corriente del domo (75%).

La finalidad del segundo separador es concentrar el componente C en el domo (90%) y enriquecer la corriente de los fondos con el componente A (75%).

Similarmente, el tercer separador concentra al componente B en la corriente del domo (90%) de manera que por los fondos se obtiene el componente A con una pureza del 95%.

Ilustración II.-

APLICACION DE LAS SUBROUTINAS JUNCO1 Y CONTL1 DEL  
PROGRAMA DE SIMULACION DE PROCESOS QUIMICOS 'GENCS'

Debido a la importancia de las recirculaciones dentro de los procesos químicos se describirá la acción del programa GEMCS sobre una "planta" sencilla que está compuesta por los tres módulos mezcladores que se muestran en la Figura 7.2 que corresponde al diagrama de flujo de información de la planta e indica la numeración de los módulos y las corrientes. La alimentación a la planta consiste de una corriente que contiene dos componentes, cada uno con un flujo de  $50 \frac{\text{moles}}{\text{hr}}$ .

En este ejemplo también se ilustra la preparación de la base de datos para la ejecución de la simulación mostrando las hojas de codificación correspondientes. En ellas, al control de impresión (KPRINT) se le asignó el valor de cero para evitar una impresión muy extenso. De todas formas, se recomienda que al principio de cualquier corrida se le asigne el valor de uno para poder detectar si existen errores y en qué punto; una vez que las subrutinas y el programa se han probado a fondo, se le podrá fijar en cero.

Para su simulación, la planta contendrá cuatro unidades por calcular (NCALC) y dos componentes (NOCOMP); el orden de cálculo (LLST) es unidad 1, unidad 2, unidad 3 y, finalmente, unidad 4 que es la que controla la convergencia y hace regresar la secuencia de ejecución a la unidad 1 hasta que ésta se logra.

La planta contiene 7 corrientes (MSN) de las que se escogió la corriente 2 para checar la convergencia.

A continuación aparece el número de corrientes que se describen inicialmente (en este caso solamente una, la corriente de alimentación) seguido de las descripciones correspondientes.

Después se leerá el número de conjuntos de parámetros por alimentar, generalmente corresponde al número de módulos por lo que habrá uno para cada unidad del sistema incluyendo al controlador. Los conjuntos de parámetros se listan a continuación. Aunque el orden en que cada parámetro se alimenta dentro del conjunto es importante, el orden en que los diferentes conjuntos se colocan unos con respecto a otros no tiene importancia; se recomienda al usuario ponerlos en orden ascendente de acuerdo a su número de unidad según el diagrama de flujo de información.

Como en este caso no se requieren propiedades físicas, químicas ni termodinámicas, el archivo de datos se cerrará con  $NP = 0$ .

Las líneas que permanecen en blanco deberán ser ignoradas pues las hojas de codificación fueron hechas para problemas con conjuntos de datos más grandes.

Un listado de la base de datos generada así como la corrida correspondiente siguen a las hojas de codificación.

SIMULACION DE PLANTAS QUIMICAS

Fecha Marzo 27, 1985.

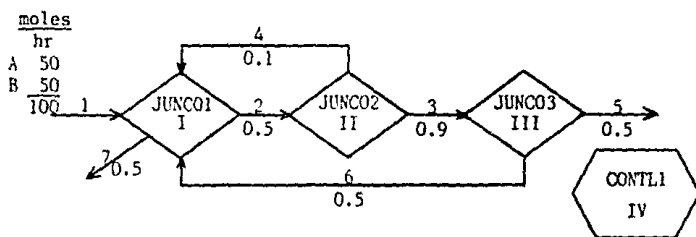
HOJA DE CODIFICACION A

Preparada por L. D. H. C.

Revisada por G. M. Z.

Diagrama de Flujo de Información

Figura 7.2



ITITLE				
APLICACION DEL PROGRAMA "GENCS" EMPLEANDO JUNCO1 Y CONTL1				
KPRINT	0			
NCALC	4	NOCOMP	2	
LLST	1	2	3	4
MSN	7			
NSR	1			
SN	1,000	0,000	100,000	0,000
	50,000	50,000		

Fecha Marzo 27, 1985.

HOJA DE CODIFICACION B

Preparada por L. D. H. G.

Revisada por G. M. Z.

NOE				
4				
EN				
1.000	1.000	19.000	0.000	0.000
3.000	1.000	4.000	6.000	0.000
2.000	2.000	7.000	0.000	0.000
0.000	0.000	0.500	0.500	
2.000	1.000	19.000	0.000	0.000
1.000	2.000	0.000	0.000	0.000
2.000	4.000	3.000	0.000	0.000
0.000	0.000	0.100	0.900	
3.000	1.000	19.000	0.000	0.000
1.000	3.000	0.000	0.000	0.000
2.000	5.000	6.000	0.000	0.000
0.000	0.000	0.500	0.500	
4.000	2.000	25.000	0.000	3.000
0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000
3.000	20.000	2.000	0.001	1.000
0.000	0.000	0.000	0.000	0.000
NP				
0				
PP				

Base de Datos Correspondiente a la Ilustración II

APLICACION DEL PROGRAMA "GEMCS" EMPLEANDO JUNC01 Y CONTL1

0					
4	2				
1	2	3	4		
7					
1					
	1.000	0.000	100.000	0.000	0.000
	50.000	50.000			
4					
	1.000	1.000	19.000	0.000	0.000
	3.000	1.000	4.000	6.000	0.000
	2.000	2.000	7.000	0.000	0.000
	0.000	0.000	0.500	0.500	
	2.000	1.000	19.000	0.000	0.000
	1.000	2.000	0.000	0.000	0.000
	2.000	4.000	3.000	0.000	0.000
	0.000	0.000	0.100	0.900	
	3.000	1.000	19.000	0.000	0.000
	1.000	3.000	0.000	0.000	0.000
	2.000	5.000	6.000	0.000	0.000
	0.000	0.000	0.500	0.500	
	4.000	2.000	25.000	0.000	3.000
	0.000	0.000	0.000	0.000	0.000
	0.000	0.000	0.000	0.000	0.000
	3.000	20.000	2.000	0.001	1.000
	0.000	0.000	0.000	0.000	0.000
0					



Ejecución de la Ilustración II

APLICACION DEL PROGRAMA "GENCS" EMPLEANDO JUNCO1 Y CONTLI

\*\*\*\*\*  
 CONTROLLER (UNIT NO 4) LOOP NO 3  
 FRACTIONAL CHANGE IN TOTAL FLOW IN STREAM 2 IS .05599

J	VALUE	FRACTIONAL CHANGE
3	67.5512	.05599
6	33.7656	.05599
7	33.7656	.05599

CONTROLLER (UNIT NO 4) LOOP NO 6  
 FRACTIONAL CHANGE IN TOTAL FLOW IN STREAM 2 IS .00114

J	VALUE	FRACTIONAL CHANGE
3	68.9357	.00114
6	34.4678	.00114
7	34.4678	.00114

\*\*\*\*\* CONVERGED \*\*\*\*\*  
 EQUIPMENT PARAMETERS FOR UNIT 1

1.000	1.000	19.000	.000	.000
3.000	1.000	4.000	6.000	.000
2.000	2.000	7.000	.000	.000
.000	.000	.500	.500	

INPUT STREAMS FOR EQUIPMENT UNIT 1

1.000	.000	100.000	.000	.000
50.000	50.000			
4.000	.000	6.896	.000	.000
3.448	3.448			
6.000	.000	31.031	.000	.000
15.515	15.515			

FINAL OUTPUT STREAMS FOR EQUIPMENT UNIT 1

2.000	.000	68.963	.000	.000
34.482	34.482			
7.000	.000	68.963	.000	.000
34.482	34.482			

EQUIPMENT PARAMETERS FOR UNIT 2

2.000	1.000	19.000	.000	.000
1.000	2.000	.000	.000	.000
2.000	4.000	3.000	.000	.000
.000	.000	.100	.900	

INPUT STREAMS FOR EQUIPMENT UNIT 2

2.000	.000	68.963	.000	.000
34.482	34.482			

FINAL OUTPUT STREAMS FOR EQUIPMENT UNIT 2

4.000	.000	6.896	.000	.000
3.448	3.448			
3.000	.000	62.067	.000	.000
31.033	31.033			

EQUIPMENT PARAMETERS FOR UNIT 3				
3.000	1.000	19.000	.000	.000
1.000	3.000	.000	.000	.000
2.000	5.000	6.000	.000	.000
.000	.000	.500	.500	
INPUT STREAMS FOR EQUIPMENT UNIT 3				
3.000	.000	62.067	.000	.000
31.033	31.033			
FINAL OUTPUT STREAMS FOR EQUIPMENT UNIT 3				
5.000	.000	31.033	.000	.000
15.517	15.517			
6.000	.000	31.033	.000	.000
15.517	15.517			

END OF EXECUTION

Análisis de Resultados.-

Este ejemplo tiene como objetivo principal mostrar la forma de actuar del módulo de control CONTL1. Así en la corrida anterior puede observarse que después de un cierto número de iteraciones (loops), fijado por el usuario y en este caso 3, se reporta el cambio fraccional en el flujo total de la corriente de prueba, corriente 2, y se lista el cambio fraccional en los flujos de todos los componentes de la corriente precedido por éstos. Aunque en esta ilustración dicho cambio fraccional es el mismo para todos los elementos, en ocasiones suele variar.

Al final de la séptima iteración, la corriente de prueba ha pasado la restricción de la tolerancia (0.001) especificada para la subrutina CONTL1 por lo que se imprime el mensaje "CONVERGED".

El programa maestro hace que los cálculos se efectúen por última vez, iteración en la que se imprimen las condiciones de las corrientes de salida de los módulos del proceso que se está simulando.

Ilustración III.-

APLICACION DE LAS SUBROUTINAS JUNCO2 Y FLASH1 DEL  
PROGRAMA DE SIMULACION DE PROCESOS QUIMICOS 'GEMCS'

En un proceso establecido, cuyo diagrama de flujo de información se presenta en la Figura 7.3, dos corrientes líquidas designadas como 1 y 2 se combinan en un mezclador (equipo I) sin que ocurran cambios de fase.

El 90% del flujo resultante (corriente líquida 3) es llevado junto con otra corriente externa (corriente líquida 5) a un tanque de destilación flash adiabática (equipo II) que opera a una presión de 250 p.s.i.a. (17 atmósferas absolutas).

Los flujos molares, condiciones de alimentación y datos para el cálculo de las propiedades termodinámicas de las corrientes 1, 2 y 5 se encuentran especificados en las Tablas 7.1, 7.2, 7.3 y 7.4.

Se desea simular el proceso de forma que se conozcan todas las condiciones de las corrientes de salida así como sus composiciones.

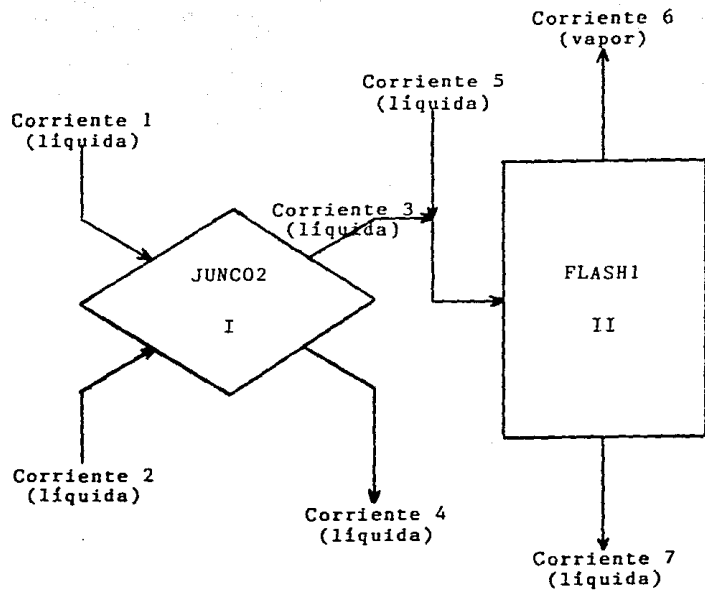


Figura 7.3

DIAGRAMA DE FLUJO DE INFORMACION DE LA ILUSTRACION III

Tabla 7.1

CONDICIONES DE ALIMENTACION

Característica		Corriente		
		1	2	5
Flujo A	[=] $\frac{\text{lb mol}}{\text{hr}}$	120	130	125
Flujo B	[=] $\frac{\text{lb mol}}{\text{hr}}$	100	90	105
Flujo C	[=] $\frac{\text{lb mol}}{\text{hr}}$	50	45	170
Flujo Total	[=] $\frac{\text{lb mol}}{\text{hr}}$	270	265	400
Temperatura	[=] °F	187	200	80
Presión	[=] $\frac{\text{lb}}{\text{in}^2}$ abs	1000	1000	1000

Tabla 7.2

COEFICIENTES PARA EL CALCULO DE ENTALPIAS EN FASE LIQUIDA

Componente	a [=] $\frac{\text{Btu}}{\text{lb mol}}$	b [=] $\frac{\text{Btu}}{\text{lb mol } ^\circ\text{F}}$	c [=] $\frac{\text{Btu}}{\text{lb mol } ^\circ\text{F}^2}$	d
Etano A	6.12E3	9.35	1.50E-2	0.0
Propano B	3.62E3	40.50	-1.32E-2	0.0
Butano C	6.28E3	25.00	4.65E-2	0.0

(1)

Tabla 7.3

COEFICIENTES PARA EL CALCULO DE ENTALPIAS EN FASE VAPOR

Componente	a [=] $\frac{\text{Btu}}{\text{lb mol}}$	b [=] $\frac{\text{Btu}}{\text{lb mol } ^\circ\text{F}}$	c [=] $\frac{\text{Btu}}{\text{lb mol } ^\circ\text{F}^2}$	d
Etano A	8.310E3	13.200	0.602E-2	0.0
Propano B	11.950E3	12.350	2.210E-2	0.0
Butano C	16.550E3	9.010	3.780E-2	0.0

(1)

Tabla 7.4

COEFICIENTES DE LA CONSTANTE DE EQUILIBRIO A 250 P.S.I.A.

Componente	$\alpha$	$\beta$ [=] $^\circ\text{F}^{-1}$	$\gamma$ [=] $^\circ\text{F}^{-2}$	$\delta$ [=] $^\circ\text{F}^{-3}$
Etano A	1.665	-1.500E-4	7.350E-5	-3.000E-8
Propano B	0.840	-4.660E-3	4.940E-5	-3.033E-8
Butano C	-0.177	4.950E-3	-4.150E-6	2.220E-8

(2)

Base de Datos Correspondiente a la Ilustración III

APLICACION DEL PROGRAMA "GENCS" UTILIZANDO JVIC02 Y FLASH1

1					
-2	3				
1	2				
7					
3					
	1. 000	2. 000	270. 000	187. 000	1000. 000
	120. 000	100. 000	50. 000		
	2. 000	2. 000	265. 000	200. 000	1000. 000
	130. 000	90. 000	45. 000		
	5. 000	2. 000	400. 000	80. 000	1000. 000
	125. 000	105. 000	170. 000		
2					
	1. 000	3. 000	21. 000	7. 000	1. 000
	2. 000	1. 000	2. 000	0. 000	0. 000
	2. 000	3. 000	4. 000	0. 000	0. 000
	0. 900	0. 100	0. 000	0. 000	0. 001
	50. 000				
	2. 000	12. 000	19. 000	7. 000	1. 000
	2. 000	3. 000	5. 000	0. 000	0. 000
	2. 000	6. 000	7. 000	0. 000	0. 000
	0. 001	50. 000	250. 000	0. 010	
15					
	1. 0000000000	1. 6650000000	-0. 000150000	0. 0000735000	
	-0. 0000000000	1. 0000000000	6120. 000000000	9. 3500000000	
	0. 0150000000	0. 0000000000	1. 000000000	8310. 000000000	
	13. 2000000000	0. 0000200000	0. 000000000		
	1. 0000000000	0. 8400000000	-0. 004660000	0. 0000494000	
	-0. 0000000000	1. 0000000000	3620. 000000000	40. 5000000000	
	-0. 0132000000	0. 0000000000	1. 000000000	11950. 000000000	
	12. 3500000000	0. 0221000000	0. 000000000		
	1. 0000000000	-0. 1770000000	0. 004950000	-0. 0000041500	
	0. 0000000000	1. 0000000000	6200. 000000000	25. 0000000000	
	0. 0465000000	0. 0000000000	1. 000000000	16550. 000000000	
	9. 0100000000	0. 0370000000	0. 000000000		



Ejecución de la Ilustración III

APLICACION DEL PROGRAMA "GEMCS" UTILIZANDO JUNC02 Y FLASH1

\*\*\*\*\*  
 \*\*\*\*\*

DATA

NUMBER OF COMPONENTS = 3  
 NUMBER OF UNITS IN THE PLANT = 2  
 CALCULATION ORDER

1

2

INITIAL VALUES FOR 3 STREAMS

1.000	2.000	270.000	187.000	1000.000
120.000	100.000	50.000		
2.000	2.000	265.000	200.000	1000.000
130.000	90.000	45.000		
5.000	2.000	400.000	80.000	1000.000
125.000	105.000	170.000		

INITIAL VALUES FOR EQUIPMENT UNIT 1

1.000	3.000	21.000	7.000	1.000
2.000	1.000	2.000	.000	.000
2.000	3.000	4.000	.000	.000
.900	.100	.000	.000	.001
50.000				

INITIAL VALUES FOR EQUIPMENT UNIT 2

2.000	12.000	19.000	7.000	1.000
2.000	3.000	5.000	.000	.000
2.000	6.000	7.000	.000	.000
.001	50.000	250.000	.010	

15 PHYSICAL PROPERTY PARAMETERS FOR EACH COMPONENT

.10000E+01	.16650E+01	-.15000E-03	.73500E-04	-.30000E-07
.10000E+01	.61200E+04	.93500E+01	.15000E-01	.00000E+00
.10000E+01	.83100E+04	.13200E+02	.60200E-02	.00000E+00
.10000E+01	.84000E+00	-.46600E-02	.49400E-04	-.30330E-07
.10000E+01	.36200E+04	.40500E+02	-.13200E-01	.00000E+00
.10000E+01	.11950E+05	.12350E+02	.22100E-01	.00000E+00
.10000E+01	-.17700E+00	.49500E-02	-.41500E-05	.22200E-07
.10000E+01	.62800E+04	.25000E+02	.46500E-01	.00000E+00
.10000E+01	.16550E+05	.90100E+01	.37800E-01	.00000E+00

\*\*\*\*\*  
 \*\*\*\*\*

EQUIPMENT PARAMETERS FOR UNIT 1

1.000	3.000	21.000	7.000	1.000
2.000	1.000	2.000	.000	.000
2.000	3.000	4.000	.000	.000
.900	.100	.000	.000	.001
50.000				

INPUT STREAMS FOR EQUIPMENT UNIT 1

1.000	2.000	270.000	187.000	1000.000
120.000	100.000	50.000		
2.000	2.000	265.000	200.000	1000.000
130.000	90.000	45.000		

ENTALPIAS DE LAS CORRIENTES DE ENTRADA

2709400.00

2715200.00

TEMPERATURA DE SALIDA = 193.3310

ENTALPIAS DE LAS CORRIENTES DE SALIDA

4882210.00

542468.00

FINAL OUTPUT STREAMS FOR EQUIPMENT UNIT 1

3.000	2.000	481.500	193.331	1000.000
225.000	171.000	85.500		
4.000	2.000	53.500	193.331	1000.000
25.000	19.000	9.500		

EQUIPMENT PARAMETERS FOR UNIT 2

2.000	12.000	19.000	7.000	1.000
2.000	3.000	5.000	.000	.000
2.000	6.000	7.000	.000	.000
.001	50.000	250.000	.010	

INPUT STREAMS FOR EQUIPMENT UNIT 2

3.000	2.000	481.500	193.331	1000.000
225.000	171.000	85.500		
5.000	2.000	400.000	80.000	1000.000
125.000	105.000	170.000		

TEMPERATURA DE ENTRADA AL MECLARSE LAS CORRIENTES DE ENTRADA = 139.7970

NUMERO DE ITERACIONES NECESARIAS PARA SU CONVERGENCIA: 2

TEMPERATURA DE BURBUJA = 55.6269

NUMERO DE ITERACIONES NECESARIAS PARA SU CONVERGENCIA: 4

TEMPERATURA DE ROCIO = 141.0170

NUMERO DE ITERACIONES NECESARIAS PARA SU CONVERGENCIA: 1

TEMPERATURA DE EQUILIBRIO EN EL TANQUE FLASH = 91.9013

RELACION L/F EN EL TANQUE FLASH = .66623

NUMERO DE ITERACIONES NECESARIAS PARA SU CONVERGENCIA: 29

A LA SALIDA DEL TANQUE FLASH.

FLUJO TOTAL DE VAPOR = 294.219

FLUJO TOTAL DE LIQUIDO = 587.281

COMPOSICION DE SALIDA

VAPOR LIQUIDO

Y(1) = .63030 X(1) = .28020

Y(2) = .26975 X(2) = .33482

Y(3) = .10021 X(3) = .38485

FINAL OUTPUT STREAMS FOR EQUIPMENT UNIT 2

6.000	1.000	294.219	91.901	250.000
185.446	79.365	29.484		
7.000	2.000	587.281	91.901	250.000
164.554	196.635	226.016		

END OF EXECUTION

### Análisis de los Resultados.-

A diferencia del mezclador utilizado en el ejemplo anterior, JUNCO2, a través del balance de energía, permite conocer la temperatura de salida resultante así como las "entalpías" de las corrientes de salida para complementar la información relativa a las mismas imprimiendo también las "entalpías" de las corrientes de entrada.

Por otro lado, la subrutina JUNCO2 podría emplearse para simular la unión de las dos corrientes de alimentación al tanque flash. En este caso, por presentar la unidad computacional correspondiente a FLASH1 la opción de evaluar dicha combinación, se optó por mantener sólo dos módulos en el proceso.

Como puede observarse, dentro de los procedimientos matemáticos iterativos desarrollados por FLASH1 el que más tiempo consume es el cálculo de la temperatura de equilibrio y la relación líquido-vapor en el tanque flash.

Las condiciones de operación del sistema son la causa de que la generación de vapor sea del 33.38% solamente pues la temperatura de equilibrio se encuentra más cerca de la temperatura de burbuja que de la de rocío.

Sin embargo, el tanque flash cumple con su objetivo ya que el etano (componente más ligero) se concentró en el vapor (63%), pobre en butano (componente más pesado y del que sólo contiene un 1%), mientras que, sin ser tan notable, la relación en el líquido residual es inversa.

Hasta aquí las constantes de equilibrio,  $K_j$ , se han usado asumiéndolas como función solamente de la tempe

ratura, la presión y el componente único designado como  $j$ . En el caso general,  $k_j$  también depende de los demás componentes pero mientras se cuente con un método factible para obtener dichos valores no deberán presentarse dificultades serias. Así, si los datos de equilibrio se encuentran disponibles a partir de consideraciones termodinámicas más fundamentales o de correlaciones empíricas, se podrían hacer los ajustes apropiados para el correcto funcionamiento del programa.

Algo parecido sucede con el cálculo de las entalpías pues, en vez de evaluarlas como función de la temperatura, podrían determinarse utilizando el método tradicional a través de las capacidades caloríficas (fase líquida) o de las capacidades caloríficas y los calores latentes (fase vapor) de los componentes involucrados.

Referencias.

1. Maxwell, J.B.; "Data Book on Hydrocarbons", [Van Nostrand, U. S. A. (1950)].
2. Robinson, C. S. & Gilliland, E. R.; "Elements of Fractional Distillation", [McGraw-Hill, U. S. A. (1951)].

## CAPITULO VIII

### C O N C L U S I O N E S

La realidad de la industria química mundial indica un claro desplazamiento de la simulación física (modelos a escala / plantas piloto) hacia la simulación matemática de los procesos con el fin de implementar más rápidamente las innovaciones y adaptaciones de los mismos a un menor costo. La importancia, cada día mayor, que ha ido adquiriendo la simulación de procesos en los países desarrollados nos hace visualizar la necesidad de no atrasarnos tecnológicamente en este ramo dentro de la Ingeniería Química.

No obstante, en el presente, existe una escasez de gente con experiencia en la simulación de procesos con ayuda de programas maestros. Esto se debe en parte a que los problemas de planta reales son muy difíciles de analizar y en parte a que los ejercicios matemáticos idealizados no dan la experiencia requerida.

Por lo mismo, el problema le ha sido heredado a la educación que, en conjunción con la industria, deberá encargarse de alentar y facilitar el uso de problemas de plantas reales en sus programas de universidad y de maestrías.

Así, como se planteó en los objetivos, esta tesis permitirá introducir al alumnado a la simulación de procesos a través del programa 'GEMCS', facilitando su empleo mediante un manual sencillo y comprensible, de manera que pueda empezar a obtener una valiosa experiencia para futuras aplicaciones. Además representa un medio para

facilitar la labor docente al constituir una herramienta más en el estudio de balances de materia y energía así como de las operaciones unitarias.

El programa de simulación de procesos químicos 'GEMCS' da al usuario varias U. C. representativas de equipos comunmente utilizados pero, por su estructura, es un programa abierto a modificaciones y mejoras lo que permite a los usuarios desarrollar sus propias subrutinas de módulos de proceso.

La posibilidad de elaborar e incorporar nuevas subrutinas para el caso en el que el proceso contenga una operación para la que no existe el módulo correspondiente en la librería del programa maestro hacen más atractivo aún a 'GEMCS' ya que se ampliaría su alcance haciéndolo accesible a un mayor número de problemas por resolver.

Se ha hablado de tres aspectos básicos que deben considerarse en el diseño de un sistema de simulación: procesamiento de datos, modelos de ingeniería y procedimientos matemáticos. Para facilitar la labor del usuario, resulta muy importante contar con un adecuado sistema de preparación y entrada de datos; de igual forma, se debe tener un sistema apropiado de salida de datos que haga sencilla la interpretación de los resultados y en su caso, el análisis de los problemas en una simulación. La preparación de modelos de ingeniería eficientes y exactos para las unidades de proceso (módulos) y las correlaciones de estimación de propiedades termodinámicas son trascendentales ya que es en estos programas donde realmente se lleva a cabo la simulación. Por último, debido a que la mayoría de las ecuaciones manejadas deben resolverse iterativamente, los procedimientos matemá

ticos son una herramienta básica para los cálculos de simulación.

Además hay que considerar que el comportamiento de los simuladores se ve fuertemente afectado por dos factores: los ciclos de recirculación (orden de cálculo) y las especificaciones de diseño. La secuencia de ejecución y la partición que se efectúe de los ciclos ocasionados por recirculaciones afectan notablemente la eficiencia del sistema de simulación de tal modo que una decisión equivocada en este aspecto puede provocar la imposibilidad del sistema para llegar a los resultados. Por otro lado, dado que los módulos evalúan las propiedades de las corrientes de salida a partir de las de entrada, no se puede especificar dentro de una simulación una corriente de salida; este problema se ha resuelto parcialmente mediante el procedimiento iterativo del controlador de corrientes.

El propósito de una simulación es generar información general y detallada de un proceso para luego usarla para optimizar dicho proceso tanto técnica como económicamente, por lo que gran variedad de nuevos procesos está siendo diseñada con los requerimientos de datos para simulación y mejoras en el proceso en mente. No obstante, a menudo se presenta la falta de datos exactos puesto que muchas plantas aún carecen de la instrumentación necesaria, más allá y por encima de la que se requiere para el control de la planta y su operación, para la generación de datos que permitan lograr resultados significativos durante la simulación.

La simulación de procesos requiere de la estimación de propiedades físicas y puesto que numerosos cálculos



involucrados en una simulación son iterativos, en muchos casos conviene recurrir a los paquetes de propiedades termodinámicas. Estos programas son los de más baja jerarquía y, dependiendo de su eficiencia, los cálculos relacionados con la estimación de propiedades termofísicas pueden representar más del 80% del tiempo de cómputo de una simulación. Por lo tanto, se debe procurar que dichos paquetes tengan una gran eficiencia computacional dado que el tiempo de cómputo es importante cuando se involucran costos.

El empleo de los paquetes de propiedades está en función del tipo de computadora que se utiliza; en el caso de las microcomputadoras, por contarse con espacio de memoria limitado, no se recomienda pero para computadoras grandes resultan excelentes pues además de dar resultados lo más exacto y consistente posible, el usuario cuenta con diversas opciones para evaluar dichas propiedades lo que le permite elegir el modelo que mejor se ajuste a las condiciones del proceso simulado dándole gran versatilidad al simulador. Esto es consecuencia de la no universalidad de los métodos de estimación de propiedades termodinámicas y de que la mayoría de las correlaciones tienen sus límites en las condiciones en que se aplican.

Obviamente, los paquetes de propiedades cuentan con un banco de datos para un gran número de sustancias amén de que generalmente plantean la opción de incorporar y/o permitir que el usuario proporcione durante la simulación los parámetros de las sustancias que no se encuentran en el mismo.

El creciente uso de las técnicas de simulación ha servido para ampliar la utilidad del acercamiento

científico a la ingeniería. Sin embargo, los adelantos presentes y futuros de la simulación no sólo dependen del desarrollo de nuevas capacidades en las computadoras sino también de nuestra habilidad para hacer uso eficiente de ellas pues la verdadera clave en la resolución de problemas es un ingeniero con imaginación, de pensamiento creativo, amplios conocimientos y buen entendimiento.

B I B L I O G R A F I A

- Amundson, Neal R. & Pontinen, Arlene J. "Multicomponent Distillation Calculations on a Large Digital Computer". Industrial and Engineering Chemistry. Vol. 50, No. 5, Mayo 1958. pp. 730-736.
- Balzhiser, Richard E.; Samuels, Michael R. & Eliassen, John D. Termodinámica Química para Ingenieros. Prentice Hall Internacional. España, 1979.
- Billingsley, D. S.; McLaughlin, W. S.; Welch, N. E. & Holland, C. D. "Use of Computers in Reactor Design". Industrial & Engineering Chemistry. Vol. 50, No. 5, Mayo 1958. pp. 741-752.
- Bridell, Talbot E. "Process Design by Computer, Part I". Chemical Engineering. 81[3], 60 (1974a).  
----- "Process Design by Computer, Part II". ibid. 81[5], 113 (1974b).  
----- "Process Design by Computer, Part III". ibid. 81[7], 77 (1974c).
- Bronson, Richard. "Computer Simulation. What It Is and How It's Done". BYTE. Marzo 1984. pp. 95-102.
- Carnahan, Brice; Luther, H. A. & Wilkes, James O. Applied Numerical Methods. John Wiley & Sons, Inc. U. S. A., 1969. p. 319.
- Coan, James S. Basic FORTRAN. Hayden Book Company, Inc. U. S. A., 1980.
- Crowe, C. M.; Hamielec, A. E.; Hoffman, T. W.; Johnson, A. I.; Shannon, P. T. & Woods, D. R. CHEMICAL PLANT SIMULATION: An Introduction to Computer-Aided Steady-State Process Analysis. Prentice-Hall, Inc. U. S. A., 1971.

- DiBiano, R.; Hales G. & Autenrieth, A. "Advantages of Third Generation Computer Control" Hydrocarbon Processing. Vol. 61, No. 6, Junio 1982. pp. 117-121.
- Evans, L. B.; Steward, D. G. & Sprague, C. R. "Computer Aided Chemical Process Design". Chemical Engineering Progress. Vol. 64, No. 4, Abril 1968. pp. 39-46.
- Franks, Roger E. Modeling and Simulation in Chemical Engineering. Wiley Interscience. U. S. A., 1972.
- Ganapathy, V.; Ravichandran, R.; Janakiraman, V. & Rajamohan, A. "Physical Properties of Selected Gas Streams". Chemical Engineering. Febrero 1977. pp. 195-199.
- Giral, José; Barnés, Francisco y Ramírez, Alejandro. Ingeniería de Procesos. Ed. Alhambra México, S. A.; Segunda Edición. México, 1979.
- Graybeal, Wayne J. & Pooch, Udo W. Simulation: Principles and Methods. Winthrop Publishers, Inc. U. S. A., 1980.
- Henley, Ernest J. & Seader, J. D. Equilibrium-Stage Operations in Chemical Engineering. John Wiley & Sons, Inc. U. S. A., 1981.
- Himmelblau, David M. Principios y Cálculos Básicos de la Ingeniería Química. Compañía Editorial Continental, S. A.; Cuarta Impresión en Español. México, 1972.
- Himmelblau, David M. & Bischoff, Kenneth R. PROCESS ANALYSIS and SIMULATION Deterministic Systems. John Wiley & Sons, Inc. U. S. A., 1968.
- Hougen, Olaf A.; Watson, Kenneth M. & Ragatz, Roland A. Chemical Process Principles Part 1 - Material and Energy Balances. John Wiley & Sons, Inc.; Second Edition. U. S. A., 1954.

- Johnson, A. I. "Computer Aided Process Analysis and Design — A Modular Approach". Process Design and Development. British Chemical Engineering and Process Technology. 17[1], 28 (1972b).
- Johnson, A. I.; Peters, N. & Associates. GEMCS General Engineering and Management Computation System (CDC CYBER 73 Version). Second Edition. The Systems Analysis, Control and Design Activity. Faculty of Engineering Science. The University of Western Ontario. Canada, 1979.
- King, Judson C. Separation Processes. McGraw-Hill Inc. Second Edition. U. S. A., 1980.
- Lapidus, Leon. Digital Computation for Chemical Engineers. McGraw-Hill Inc. U. S. A., 1962.
- Lee, Wooyoung; Christiansen, James H. & Rudd, Dale F. "Design Variable Selection to Simplify Process Calculations". A. I. Ch. E. Journal. 12[6], 1104 (1966).
- Lee, Wooyoung & Rudd, Dale F. "On the Ordering of Recycle Calculations". A. I. Ch. E. Journal. 12[6], 1184 (1966).
- Ludwig, Ernest. Applied Process Design for Chemical and Petrochemical Plants Vol. I. Gulf Publishing Co. U. S. A., 1977.
- Luyben, N. L. Process Modeling, Simulation and Control for Chemical Engineers. Kingsport Press, Inc. U. S. A., 1973.
- Maxwell, J. B. Data Book on Hydrocarbons. Van Nostrand. U. S. A., 1950.
- Motard, R. L.; Shacham, M. & Rosen, E. M. "Steady State Chemical Process Simulation". A. I. Ch. E. Journal. 21[3], 417 (1975).

- Myers, Alan L. & Seider, Warren D. Introduction to Chemical Engineering and Computer Calculations. Prentice-Hall, Inc. U. S. A., 1976.
- Ord-Smith, R. J. & Stephenson, J. Computer Simulation of Continuous Systems. Cambridge University Press. Great Britain, 1975.
- Organick, Elliot I. A FORTRAN IV Primer. Addison-Wesley Publishing Co. U. S. A., 1966.
- Peters, N. & Barker, P. E. "An Appraisal of the Use of PACER, GEMCS and CONCEPT for Chemical Plant Simulation and Design". The Chemical Engineer (London). 283, 149 (1974).
- Peterson, Jeffrey N.; Chen, Chau-Chyun & Evans, Lawrence B. "Computer Programs for Chemical Engineers: 1978 — Part. 1". Chemical Engineering. Junio 5, 1978. pp. 145-152.
- Ramirez, Fred W. Process Simulation. D. C. Heath and Company. U. S. A., 1976.
- Ravicz, A. E. & Norman, R. L. "Heat and Mass Balancing on a Digital Computer". Chemical Engineering Progress Vol. 60, No. 5, Mayo 1964. pp. 71-76.
- Reid, Robert C.; Prausnitz, John M. & Sherwood, Thomas K. The Properties of Gases and Liquids. McGraw-Hill Inc.; Third Edition. U. S. A., 1977.
- Reid, Robert C. "Engineering Properties of Fluids". Chemical Engineering Progress. Vol. 61, No. 12, Octubre 1965. pp. 58-64.
- Robinson, C. S. & Gilliland, E. R. Elements of Fractional Distillation. McGraw-Hill Inc. U. S. A., 1951.

- Rudd, Dale F.; Powers, Gary J. & Siirola, Jeffrey J. Process Synthesis. Prentice-Hall, Inc. U. S. A., 1973. pp. 58-60.
- Schmidt, J. W. & Taylor, R. E. Análisis y Simulación de Sistemas Industriales. Ed. Trillas. México, 1979.
- Shah, G. C. "Understanding Minicomputer Control Systems". Hydrocarbon Processing. Abril, 1980. pp. 153-157.
- Shannon, Robert E. SYSTEMS SIMULATION the art and science. Prentice-Hall, Inc. U. S. A., 1976.
- The Systems Analysis, Control and Design Activity. Process Engineering Calculation. Engineering Science Building. University of Western Ontario. Canada, 1981.
- Treybal, Robert E. Mass-Transfer Operations. McGraw-Hill Inc.; Third Edition. U. S. A., 1980.
- Westerberg, A. W.; Hutchison, H. P.; Motard, R. L. & Winter, P. Process Flowsheeting. Cambridge University Press. Great Britain, 1979.
- Yaws, Carl L. et al. "Correlation Constants for Chemical Compounds". Chemical Engineering. Agosto 16, 1976. pp. 79-87.  
----- "Correlation Constants for Liquids". ibid. Octubre 25, 1976. pp. 127-135.  
----- "Correlation Constants for Chemical Compounds", ibid. Noviembre 22, 1976. pp. 153-162.