



60
2ej.

**UNIVERSIDAD NACIONAL AUTONOMA
DE MEXICO**

FACULTAD DE QUIMICA

EXAMENES PROFESIONALES
FAC. DE QUIMICA

**“SIMULACION DINAMICA DEL PROCESO DE
ADSORCION-DESORCION DE CLORO SOBRE
Pt/AL₂O₃ EN REACTOR DE ISOMERIZACION
DE PARAFINAS”**

T E S I S

QUE PARA OBTENER EL TITULO DE
INGENIERO QUIMICO
P R E S E N T A
RODOLFO GARCIA FLORES



MEXICO, D. F.

**TESIS CON
FALLA DE ORIGEN**

1997

1997



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

Jurado asignado:

Presidente: Prof. Jorge Fernando Ramirez Solis
Vocal: Prof. Carlos Guzmán De las Casas
Secretario: Prof. Luis Cedeño Caero
1er. Suplente: Prof. Rodolfo Ruiz Trejo
2o. Suplente: Prof. Eric Mauricio Rivera Muñoz

Sitio donde se desarrolló el tema

Unidad de Investigación en Catálisis, Facultad de Química UNAM

Asesor:

Dr. Jorge Fernando Ramirez Solis

Sustentante:

Rodolfo Garcia Flores

Firma:

**TRABAJA PARA SER REVISADO
POR EL JURADO**



Firma:

Rodolfo Garcia Flores

29 OCT. 1997

A mis tíos David y Engracia, a mi hermana Déborah y muy especialmente a mi madre, María Teresa Flores.

Agradecimientos:

Quiero agradecer al Dr. Jorge Ramírez Solís por su apoyo como asesor y por su paciencia en todos estos semestres, y a la Dra. Lena Ruiz Azuara por su ayuda y orientación.

A mi familia en Celaya por la confianza y cariño que siempre me han demostrado y a todos los que fueron mis compañeros y maestros mientras estuve en la Facultad de Química de la Universidad de Guanajuato por todo lo que compartimos y seguimos compartiendo, especialmente a los piojos.

A J. Ángel Ojeda Nava por la motivación y orientación en el desarrollo del programa y a todos los miembros de la Unidad de Investigación en Catálisis.

A mis profesores por haberme enseñado una forma diferente de comprender el mundo.

A los miembros del jurado: Dr. Jorge Ramírez Solís, Dr. Carlos Guzmán de las Casas, M.C. Luis Cedeño Caero, profesor Eric Mauricio Rivera Muñoz e Ing. Rodolfo Ruiz Trejo por su valiosa contribución en este trabajo de tesis.

A la memoria del Dr. Adrián Benítez Patricio. Jamás te olvidaremos.

"Dios se manifiesta en la armonía de lo que existe"

B. Spinoza

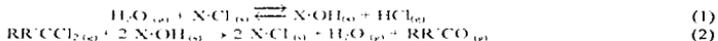
CONTENIDO

Resumen	i
Introducción	ii
Capítulo 1 Generalidades	1
1.1 Introducción.....	1
1.2 Generalidades sobre catalizadores sólidos.....	4
1.3 Metodología del diseño de catalizadores sólidos.....	5
1.4 Modelos matemáticos.....	7
1.5 Estrategia de solución con programación orientada a objetos.....	9
Referencias.....	11
Capítulo 2 Planteamiento matemático del modelo del reactor	13
2.1 Introducción.....	13
2.2 Descripción de los mecanismos de transporte.....	14
2.2.1 Fenómenos de transporte.....	18
2.2.2 Parámetros operacionales.....	19
2.2.3 Condiciones de frontera.....	20
2.3 Descripción de las expresiones cinéticas.....	23
2.3.1 Reacciones de isomerización.....	24
2.3.2 Reacciones de cloración de catalizadores de metales del grupo VIII B en alúmina.....	25
2.3.3 Desactivación del catalizador durante su tiempo de operación.....	28
2.3.4 Modelo cinético empleado en la simulación.....	31
2.4 Problemas numéricos.....	33
Referencias.....	33
Capítulo 3 Descripción y aplicación del modelo del reactor tubular	35
3.1 Ecuaciones a resolver.....	46
3.2 Modelos de prueba.....	32
3.2.1 Discusión sobre los métodos numéricos.....	40
Referencias.....	40
Capítulo 4 Análisis y discusión de resultados	42
4.1 Parámetros de referencia.....	42
4.1.1 Justificación.....	43
4.1.2 Metodología de solución del problema.....	44
4.2 Solución asintótica con parámetros de referencia.....	44
4.3 Análisis de variación de parámetros.....	51
4.3.1 Constantes de reacción.....	53
4.3.2 Difusividad.....	55
4.3.3 Longitud del reactor.....	60
4.3.4 Diámetro de partícula.....	61
4.3.5 Flujo.....	63
4.3.6 Composición de la corriente de alimentación.....	64

4.3.7 Discusión sobre el procedimiento de solución.....	66
Referencias.....	67
Capítulo 5 Conclusiones.....	68
Nomenclatura.....	70
Apéndice A Conceptos sobre ecuaciones diferenciales parciales.....	72
A.1 Clasificación.....	72
A.2 Condiciones de frontera.....	74
A.3 Métodos de residuos ponderados.....	75
A.4 Algoritmo MOLCH de IMSL empleado en la solución.....	76
Referencias.....	79
Apéndice B Tablas y listados.....	80
B.1 Descripción.....	80
B.2 Entrada de datos.....	81
Referencias.....	81
Apéndice C Modelo de cloración de alúmina para la obtención de $AlCl_3$.....	82
Referencias.....	86

RESUMEN

El tema central de este trabajo es la solución de un sistema de ecuaciones diferenciales parciales que representa el comportamiento de un reactor de lecho empacado con partículas de $\gamma\text{-Al}_2\text{O}_3$ a través del tiempo dentro del cual se llevan a cabo las siguientes reacciones



Los iones cloro e hidroxilo se encuentran adsorbidos en la superficie del soporte catalítico dentro del reactor. Aunque el propósito de este reactor sea la isomerización de parafinas, no se simula aquí ninguna reacción de isomerización. Sin embargo, desde el punto de vista matemático y numérico, la simulación de estas reacciones no representa ninguna dificultad adicional pues sólo habría que añadir una ecuación diferencial por cada componente de las reacciones de isomerización. La única limitación es la memoria de la computadora, pues mientras más componentes haya, más memoria se emplea.

Esta tesis está estructurada de la siguiente manera: en el capítulo 1 se dan las generalidades sobre los procesos de isomerización de parafinas, especialmente el proceso *Pemex* de UOP, se explican las características fisicoquímicas que confieren propiedades catalíticas y se da un sencillo procedimiento para el diseño de catalizadores sólidos, se hace una clasificación de los tipos de modelos matemáticos que pueden plantearse para describir un fenómeno físico y se hace una breve reseña del enfoque de programación orientado a objetos. El capítulo 2 está dedicado a la descripción en términos matemáticos del reactor de lecho empacado dividiéndolo en mecanismos de transporte y expresiones cinéticas, haciendo en cada caso una revisión de la información disponible. También en este capítulo se considera el problema de la estabilidad numérica de su solución. El capítulo 3 incluye el planteamiento completo de este problema en particular, menciona los sistemas de ecuaciones diferenciales parciales que se usaron para probar las rutinas de solución y establece las condiciones de frontera. En el capítulo 4 se justifica un juego de parámetros y se propone una serie de variaciones para estos parámetros. Se presentan y discuten las gráficas de resultados según las variaciones propuestas.

En los resultados obtenidos se pueden apreciar los efectos de los mecanismos de transporte (como el flujo convectivo y la difusividad) así como de los parámetros cinéticos a través del tiempo. También los efectos de variar características físicas del reactor como el diámetro de partícula, longitud del reactor y alimentación de reactivos. Se discute sobre el mecanismo de solución y se proponen variaciones adicionales en las que se aprovechan las características dinámicas de este modelo. Finalmente, en las conclusiones se señalan los parámetros que resultan ser de mayor importancia para mantener un perfil de cloro uniforme.

Los métodos de diferencias finitas resultaron mucho más lentos que la colocación ortogonal, tal como lo menciona la literatura. Los paquetes de programación usados en este trabajo permiten la programación con mezcla de lenguajes. Existen muchas rutinas programadas en FORTRAN, pero este lenguaje ya no es popular (excepto, tal vez, la versión 90). Sin embargo, no es necesario volver a codificar los algoritmos en lenguajes más actuales, pues podemos ahora tener acceso a todas las bibliotecas numéricas como la IMSL, usando las opciones de compilación y enlazado correctas, que se explican en el apéndice B.

INTRODUCCIÓN

Los componentes de la gasolina que frecuentemente ofrecen buenas posibilidades para mejorar su calidad son las fracciones de pentanos y hexanos. El corte de naftas ligeras en la refinación, que es la fuente de estos compuestos, se caracteriza por un bajo número de octano, por lo regular entre 60 y 70. La fracción de pentanos y hexanos representa un 10 % de una gasolina típica que anteriormente se mezclaba directamente sin ningún procesamiento adicional, excepto la hidrosulfuración (HDS) para remover compuestos de azufre. El bajo índice de octano se compensaba con la adición de compuestos de plomo, que incrementaban el octanaje entre 16 y 18 números. La fracción $C_5 - C_6$, era entonces el segmento de la gasolina al que menos procesamiento se le daba y que más se beneficiaba por la adición de plomo. Hoy que la industria del petróleo tiende a producir combustibles con niveles de plomo cero, se vuelve más complicado integrar la fracción de naftas ligeras a las gasolinas. La conversión de estas parafinas de cinco y seis carbonos a los isómeros ramificados correspondientes para incrementar el octanaje se ha vuelto un paso lógico y necesario.

Este es uno de tantos procesos catalíticos que se efectúan en gran escala en la industria de refinación del petróleo. La reformación catalítica, el craqueo catalítico, el hidrocrqueo y la hidrosulfuración se aplican a grandes volúmenes de hidrocarburos y el hidrotratamiento catalítico se usa para mejorar la calidad de varios intermediarios y productos, incluyendo los combustibles. La eficiencia de estos procesos de producción depende de las propiedades de los catalizadores, y para mejorar estas propiedades es necesario conocer los principios de su funcionamiento. Para ello es indispensable la determinación de los parámetros físicos y fisicoquímicos y las relaciones entre ellos por medios experimentales.

Sin embargo, gracias al constante incremento en el poder de cálculo de las computadoras y la disminución en su costo, cada vez en más áreas de la ingeniería se prefiere diseñar y evaluar equipos por medio de simuladores. La razón es sencilla cuando ya se conocen los principios que regulan el funcionamiento de un equipo, resulta mucho más económico programar y poner a funcionar una computadora que construir un modelo de un aparato (que puede ser de por sí muy costoso) y hacerlo trabajar en condiciones extremas para verificar la validez de su diseño. El tiempo de cálculo es además mucho menor al que requiere una prueba real. Un ejemplo conocido es la dinámica computacional de fluidos (CFD), que es un problema semejante al de este trabajo, pues se representa como un sistema de ecuaciones diferenciales parciales (ecuaciones de Navier-Stokes). La CFD tiene aplicaciones tan variadas como el diseño y prueba de redes de tuberías, equipos de mezclado, turbinas y aviones.

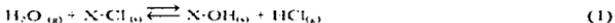
En los últimos años ha habido una intensa interacción entre la dinámica no lineal y la Ingeniería Química. La relación de la cinética química con la no linealidad y el comportamiento cíclico de algunas reacciones fue establecido a finales de los años 40, se sabe desde mediados de los 60 que el desempeño de los equipos (en nuestro caso los reactores catalíticos) puede mejorarse si se operan en condiciones cambiantes a través del tiempo; la mecánica de fluidos no newtonianos y la ciencia de materiales se han desarrollado bastante desde los años 80, muchos ingenieros químicos han hecho importantes aportaciones a los métodos de solución.

Un acercamiento desde el punto de vista del estado estacionario no siempre resulta satisfactorio para explicar el comportamiento de un reactor catalítico por las siguientes razones: 1) no puede sostenerse la existencia de especies intermediarias en estado adsorbido, 2) cuando se

tienen sistemas de reacción complejos, el gran número de reacciones rivales y posibles reacciones dominantes hace difícil discriminar la o las reacciones adecuadas para modelar. Sin embargo, un modelo dinámico presenta mayores complicaciones en su solución y requiere un buen análisis de estabilidad, sobre todo si el modelo es no adiabático y no isotérmico. Que el comportamiento dinámico de los reactores afecte su capacidad de producción tiene gran importancia para la industria y por ello este fenómeno ha sido ampliamente estudiado de forma teórica y experimental. El incremento en el poder de cálculo y la disponibilidad de las computadoras ha permitido un análisis más completo y cercano a la realidad de estos equipos.

El modelo propuesto en esta tesis describe el cambio de concentración de cada especie a lo largo de la coordenada axial a través del tiempo. El problema se plantea en forma de un sistema de ecuaciones diferenciales parciales en términos de variables adimensionales que se resuelven por método de líneas y colocación ortogonal en elemento finito y/o diferencias finitas (métodos de Runge-Kutta y predictor-corrector), en donde el número de ecuaciones es igual al número de especies reaccionantes. Se pretende utilizar código anteriormente escrito en FORTRAN y combinarlo con el nuevo enfoque orientado a objetos (OOP) en lenguaje C++. Las ventajas de este enfoque son mayor claridad en el código, facilidad para el crecimiento de los programas y sencillez en su estructura. Los resultados en cada corrida se presentarán en forma de una tabla de texto ASCII donde, a un tiempo y a una posición dadas, se podrá leer la composición de cada especie. Este formato de la tabla de resultados facilita su visualización gráfica en paquetes como Excel. El usuario especifica el intervalo de tiempo en el que desea el cálculo, los coeficientes de cada término en las ecuaciones diferenciales y los parámetros de cálculo de cada método numérico. El tiempo de cálculo dependerá del método y de la cantidad de ecuaciones a resolver.

Por todo lo mencionado anteriormente, el objetivo de este trabajo será analizar el efecto de las variables cinéticas, de transporte y operacionales en el comportamiento dinámico del balance agua-cloro dentro de un reactor de isomerización de parafinas según un modelo isotérmico que considera las siguientes reacciones



donde X es el sitio activo

GENERALIDADES

1.1 Introducción:

Este trabajo está dedicado al estudio teórico del proceso de adsorción/desorción de cloro sobre gamma-alúmina dentro de un reactor de lecho empacado, el cual es importante en procesos de refinación tales como la isomerización de parafinas y la reformación catalítica de naftas. Para comprender la utilidad de dicho estudio, en este capítulo se explica como ejemplo un proceso industrial para el cual este balance de cloro es importante. Después se describen las características fisicoquímicas que confieren a ciertos sólidos propiedades catalíticas poniendo especial atención a la acidez superficial, pues es esta la propiedad que confiere el cloro a la gamma-alúmina y que favorece la isomerización. Finalmente analizamos las herramientas matemáticas que se usan para la solución del problema.

Los procesos catalíticos que se efectúan en mayor escala se encuentran en la industria de refinación del petróleo. La reformación catalítica, el craqueo catalítico, el hidrocrqueo y la hidrosulfuración se aplican a grandes volúmenes de hidrocarburos y el hidrot ratamiento catalítico se usa para mejorar la calidad de varios intermediarios y productos. De estos productos, los más importantes económicamente son los combustibles.

El petróleo consiste principalmente de hidrocarburos pero también contiene pequeñas cantidades de azufre, nitrógeno y oxígeno en forma de compuestos orgánicos y vanadio y níquel en forma de porfirinas, además de azufre elemental. Las concentraciones de cada tipo de compuesto y la distribución de pesos moleculares dependen mucho del lugar de extracción y de la profundidad del yacimiento.

Los principales hidrocarburos componentes son parafinas, nafténicos y aromáticos y las impurezas más importantes desde el punto de vista del procesamiento catalítico de las fracciones del petróleo son los compuestos del azufre y del nitrógeno.

De forma representativa, las fracciones del petróleo son las siguientes:

1 - GAS	Hasta C_4	Combustible, LPG
2 - GASOLINA	$C_4 - C_6$	Mezcla para gasolina
3 - NAFTA VIRGEN (Destilado ligero)	C_6 $T < 150^\circ C$	Reformación catalítica o gasolina
4 - NAFTA PESADA	$C_6 - C_{15}$ $120^\circ C < T < 200^\circ C$	Gasavión, keroseno, Reformación catalítica
5 - GASÓLEO LIGERO	$C_{15} - C_{26}$ $200^\circ C < T < 310^\circ C$	gasavión o diesel
6 - GASÓLEO PESADO	$C_{21} - C_{25}$ $310^\circ C < T < 350^\circ C$	craqueo catalítico o combustibles pesados
7 - RESIDUO ATMOSFÉRICO	$350^\circ C < T < 560^\circ C$	varios
8 - RESIDUO DE VACIO	$T > 560^\circ C$	---

Los requerimientos principales que debe cubrir una gasolina son

- I *Volatilidad adecuada.* Una volatilidad baja disminuye la eficiencia del motor, alta volatilidad puede producir burbujas en partes críticas del motor y dañarlo
- II *Exceder un cierto octanaje mínimo.* El octanaje es una medida cuantitativa aunque imprecisa de la compresión máxima a la que puede someterse una gasolina en un motor de combustión interna sin que la mezcla aire-gasolina sufra una autoignición. Si la autoignición sucede antes de que la buja produzca la chispa, se produce un aumento súbito de presión llamado golpeteo que puede dañar el motor

La escala para medir el octanaje es la siguiente: se asigna de manera arbitraria 100 al 2, 2, 4-trimetilpentano (iso-octano) y 0 al n-heptano, de modo que a una mezcla de 10 partes de n-heptano y 90 de iso-octano corresponde un octanaje de 90. Existen dos maneras de medir el octanaje de una mezcla. Cuando se mide en condiciones suaves se usa el *Research Octane Method* (RON), y cuando se mide en condiciones severas como alta velocidad se llama *Motor Octane Method* (MON). En todos los casos el RON es mayor al MON y la diferencia entre ambos recibe el nombre de *sensibilidad* [1].

Las gasolinas se obtienen de mezclar corrientes provenientes de muchas fuentes en la refinería, siendo las principales

- Craqueo catalítico
- Reformación catalítica
- Isomerización
- Oxigenados (FAME y MTBE)
- Hidrocraqueo
- Alquilación
- Fracción del petróleo correspondiente a la gasolina

La reformación catalítica tiene como objetivo procesar una fracción del petróleo con una volatilidad adecuada para usarse como gasolina incrementando su octanaje y sin modificar sustancialmente su peso molecular. El catalizador de la unidad de reformación debe favorecer la cinética de las reacciones que producen un efluente rico en aromáticos y parafinas ramificadas (selectividad) a una temperatura de alrededor de 500 °C (actividad) durante el mayor tiempo posible (estabilidad) [4]. Cuando se incrementa la temperatura en los procesos de reformación el rendimiento de aromáticos es mayor, aunque el catalizador se desactiva más rápido. Disminuir la presión también incrementa el octanaje del reformado, pero se favorece la carbonización de reactivos sobre el catalizador. La alimentación de hidrógeno recirculado es necesaria en estos procesos para mantener la estabilidad del catalizador [2].

Para prolongar la vida del equipo debe evitarse la humedad ya que ésta produce la eliminación del cloro del catalizador. También deben mantenerse bajos los niveles de otros contaminantes en el afluente, como azufre, nitrógeno, arsénico, metales pesados y compuestos oxigenados y fluorados.

Las reacciones que se llevan a cabo en el proceso de reformación son [1,2] :

- 1) Deshidrogenación de naftenos a aromáticos
- 2) Deshidrogenación de parafinas a olefinas

- 3) Isomerización de alquileclopentanos
- 4) Deshidrociclización de parafinas aromáticas
- 5) Isomerización de n-parafinas a isoparafinas

Para comprender la importancia de las reacciones de isomerización, es ilustrativo saber que el RON del n-pentano es de 62, mientras que el del isopentano es de 92. El proceso de isomerización se emplea para ramificar los pentanos y hexanos lineales, así como para producir isobutano que es intermediario del metil t-butil éter. En condiciones de equilibrio las concentraciones mayores de parafinas ramificadas se dan a bajas temperaturas, por lo que se requiere un catalizador muy activo, de cierta acidez superficial. Condiciones típicas de reacción son de 200 a 500 °C, 2 a 3.5 MPa con alta presión parcial de hidrógeno para minimizar la formación de coque.

Existen varios procesos específicos para la isomerización de parafinas, algunos de los cuales son [3]

- Proceso Penex de UOP
- Isomerización total (TIP) de UCC
- UCC Hysomer de Shell
- Isomerización de British Petroleum
- Proceso Butamer de UOP

A continuación se describe el proceso *Penex* de UOP:

El proceso está específicamente diseñado para la isomerización catalítica de pentanos, hexanos y mezclas y utiliza un catalizador comercial que está compuesto de Pt/Al_2O_3 con cloro adsorbido en su superficie. Las condiciones de operación no son severas: 480 a 530 °C. Como se muestra en la figura 1.1, la alimentación pasa a uno o dos secadores cuyo fin es proteger el catalizador de la humedad presente en la corriente de alimentación para evitar la descloración del mismo. La corriente pasa, después de mezclarse con hidrógeno de recirculación, a intercambiar calor con el efluente del reactor y a un precalentador antes de entrar a los reactores (dos en serie). Debe tenerse cuidado de impedir la entrada de contaminantes a la alimentación a los reactores, estos contaminantes son azufre, nitrógeno, arsénico, metales pesados y compuestos oxigenados, además del agua, cuya concentración debe reducirse al mínimo para evitar que desplace al cloro. Como se justifica más adelante, la cinética de desorción de cloro domina sobre la adsorción y es por eso que, para evitar la pérdida de la acidez en el catalizador, debe evitarse la presencia de agua. También debe evitarse la presencia de hidrocarburos indeseables como olefinas, C_7 , butano y compuestos cíclicos. El efluente del reactor es enfriado, parcialmente condensado y separadas sus corrientes líquida y vapor. La corriente en fase vapor, rica en hidrógeno, se combina con hidrógeno seco de reposición y mezclada con la alimentación al reactor. Usualmente este hidrógeno viene de algún reformador catalítico. El líquido va a un estabilizador, cuyos domos van a un neutralizador donde se remueve el HCl formado por los cloruros orgánicos agregados en pocas partes por millón a la alimentación del reactor. Los fondos del estabilizador pasan a mezclarse con la gasolina, aunque a veces se fraccionan en parafinas ramificadas y normales para recircular estas últimas.

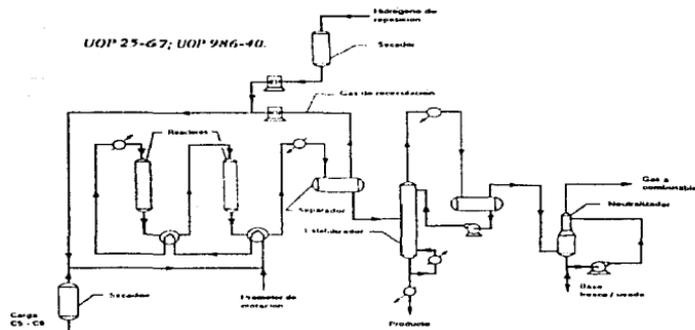


FIGURA 1.1 Proceso PLENEX de UOP

Los reactores de lecho empacado son el corazón del proceso y se colocan de tal manera que cualquiera de los dos pueda ser utilizado como el primero o segundo en el tren de procesamiento. Cada reactor está equipado con una línea de purga de hidrógeno ubicada en la entrada del equipo para vaciarlo o enfriarlo durante una emergencia, así como líneas para despresurizarlo. Ambos equipos tienen termopares insertados en el lecho de catalizador a lo largo del reactor para medir las diferencias de temperatura y supervisar la actividad del catalizador. En la corriente de alimentación a los reactores se introduce, además de los reactivos secos, un compuesto clorado que sirve para regenerar el catalizador. Esto se hace de forma intermitente según las necesidades del proceso, y es por eso que es importante conocer el comportamiento del reactor a través del tiempo. Una discusión más completa sobre los reactores de lecho empacado se encuentra en el siguiente capítulo.

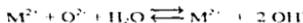
El empaque de los reactores consiste de partículas catalíticas a través de las cuales pasa la corriente de proceso. Estas partículas deben tener ciertas propiedades fisicoquímicas, que a continuación se explican.

1.2 Generalidades sobre los catalizadores sólidos:

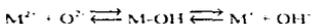
Un catalizador sólido puede consistir de uno o varios componentes, cada uno de ellos cumple uno de tres papeles: *fase activa*, *promotor* o *sopORTE* [5].

1. La *fase activa* es la responsable principal de la actividad catalítica, usualmente metales del grupo VIII
2. Los *promotores* son sustancias que no tienen una actividad catalítica considerable por sí mismas pero que hacen que la fase activa trabaje a su máxima capacidad. Un catalizador puede tener varios promotores. Aunque la manera en que trabajan no siempre se la ha determinado, se clasifican en tres grupos:
 - *Promotores de textura* - Mantienen las propiedades de área y porosidad del catalizador
 - *Promotores estructurales* o *electrónicos* - se disuelven en la fase activa y alteran sus propiedades electrónicas
 - *Promotores resistentes al envenenamiento* - protegen la fase activa contra el envenenamiento por impurezas de reactivos o generadas por reacciones paralelas. E.g. el renio de los catalizadores de platino/alúmina para reformación del petróleo protegen al platino de la deposición de carbono
3. El *soporte* es el vehículo de la fase activa y los promotores. Entre sus muchas funciones está la de proporcionar un área superficial adecuada en la que se distribuye la fase activa y los promotores, y constituye de 0.1% a 0.2% en peso en la mayoría de los casos. Por sí mismo es inactivo aunque, en ocasiones, combinado con la fase activa puede participar de forma muy importante en las reacciones. Cumple tres funciones principales:
 - *Proporciona soporte mecánico* - ya que tienen una temperatura de fusión mucho mayor que la fase activa
 - *Proporciona área superficial* - la rapidez de reacción es proporcional al área superficial de la fase activa. El área superficial de la gamma-alúmina, que es uno de los soportes más utilizados, está entre 180 y 220 m²/g
 - *Facilita la difusión* - aporta una estructura porosa para facilitar la difusión de reactivos y productos al interior de las partículas catalíticas.
 - *Propicia una baja caída de presión* - la forma de las partículas del catalizador produce una baja caída de presión

Los óxidos metálicos usados como soportes están cubiertos por una capa de grupos hidroxilo. Esta capa se forma de la disociación de agua sobre una molécula de óxido expuesta en la superficie



Tales grupos pueden disociarse como ácidos o como bases, según la naturaleza de M



Si M es un elemento electropositivo, como el magnesio, el óxido es básico y M - OH se disocia para dar iones hidroxilo. Si M es un elemento electronegativo, como el silicio, M - OH se disocia y produce un proton. Para catálisis heterogénea son de gran interés los soportes que presentan *acidez superficial* [6]. Si su acidez se debe a la presencia de protones, se dice que son *ácidos de Brønsted*. Si en cambio se debe a su capacidad de recibir electrones, se les llama

ácidos de Lewis Raras veces el óxido de un mismo metal tiene acidez suficiente para catalizar una reacción que requiera de sitios ácidos fuertes; las mezclas de óxidos si la tienen. Un ejemplo claro son las mezclas de óxidos de silicio y aluminio, como lo explica la siguiente figura:

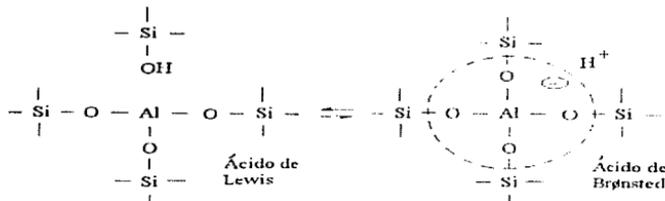


Fig. 1.2 Representación de sitio ácido en sílica - alúmina

Otra característica importante de los catalizadores es su *función metálica*. Los elementos más adecuados para catalizar reacciones de hidrotratamiento son los metales del grupo VIII B. Se llama *dispersión* a la fracción de metal que está expuesta sobre la superficie del soporte, por lo tanto para obtener una mejora en la función metálica es conveniente tener una dispersión alta [7].

Un catalizador que presenta ambas características (*acidez superficial y función metálica*) se llama *catalizador bifuncional*.

1.3 Metodología del diseño de catalizadores sólidos:

Ahora que conocemos las principales características de los catalizadores sólidos, revisaremos el siguiente paso para su aplicación a gran escala. Los catalizadores heterogéneos son los componentes clave de muchos procesos industriales. A pesar de su importancia en una amplia variedad de industrias (entre ellas la que nos ocupa, secc. 1.1), su desarrollo y mejora hasta ahora han sido en buena parte empíricos. El conocimiento que hoy tenemos sobre los procesos químicos en superficies sólidas no es suficiente como para diseñar catalizadores sin ayuda de estudios experimentales, sin embargo, esta búsqueda de nuevos catalizadores debe estar fundada en un estudio profundo de los fenómenos fisicoquímicos que ocurren en la superficie del catalizador durante la reacción.

A continuación se describe a grandes rasgos una metodología para la evaluación del desempeño de un catalizador (por James Dumesic et al. [19]), que integra el trabajo experimental con los catalizadores (secc. 1.2) y el conocimiento sobre los modelos (secc. 1.4). Esta metodología consta de tres tareas principales.

a) **Postulación de un mecanismo de reacción.**

Debe incluir una descripción de las reacciones principales, intermediarios de reacción y pasos de los mecanismos, fundamentándose en estudios experimentales.

Para el problema que se estudia en este trabajo se propone un mecanismo simple para las reacciones.

b) Construcción y calibración de un modelo cinético.

b.1) Construcción.

Un paso importante es el ajuste de las constantes de rapidez de cada paso elemental mediante una expresión del tipo Arrhenius y la estimación de las energías de activación, aunque este es un proceso más complicado.

Con los datos disponibles para el problema que nos ocupa, sólo podemos obtener una constante de Arrhenius para la reacción de adsorción de cloro. No se tienen datos para la reacción de desorción ni entalpías de reacción.

Los pasos siguientes en esta metodología son la calibración del modelo y la aplicación al diseño de catalizadores sólidos. Estos pasos no se aplican en este trabajo, pero para mayor información puede verse [19].

En este punto se ha concluido la discusión sobre los principios de los catalizadores sólidos y de la forma en que se pasa de la escala experimental a la industrial. En el siguiente capítulo se discutirán a fondo las reacciones de adsorción y desorción de cloro sobre alúmina, que son el tema principal de este trabajo. A continuación se define lo que es un modelo matemático, se clasifican los problemas por su complejidad y por su tipo y se discute su utilidad. En particular se explica el modelamiento orientado a objetos.

1.4 Modelos matemáticos.

“Un objeto M (proceso o sistema físico o ideal) es un modelo si existen analogías entre M y otro objeto O que permitan sacar conclusiones acerca de O” [8]. Tales analogías pueden establecerse en forma de ecuaciones algebraicas y/o diferenciales y/o integrales que describen el comportamiento del objeto O. Pero tal representación no es perfecta y presenta 4 limitaciones principales [9].

1. Tipo, cantidad y exactitud de los datos disponibles
2. Herramientas numéricas disponibles (e.g. [10])
3. Extrapolabilidad
4. Sensibilidad a parámetros de los modelos

Los modelos matemáticos pueden usarse para describir una amplia variedad de fenómenos, y aun para una misma clase de modelo el método numérico debe ser específico para el problema [11].

Por su tipo, podemos clasificarlos en dos grandes grupos

- *Físicoquímicos* - Se desarrollan a partir de los fenómenos de transporte. Requieren que el proceso pueda subdividirse en elementos que puedan ser descritos por los principios de conservación de masa, momentum y energía. Pueden ser *determinísticos* o *estadísticos*. A los

determinísticos corresponde un valor o conjunto de valores fijo a las variables si se define un cierto conjunto de parámetros del modelo. En cambio los modelos estadísticos pueden dar probabilidades o funciones de densidad de probabilidad. Sin embargo esto no significa que las leyes de conservación no apliquen a sistemas en los que es conveniente el uso de modelos estadísticos. Un ejemplo claro es la turbulencia. Puede tenerse un modelo que combine ambos enfoques, como la Teoría Cinética de los gases.

- *Empíricos* - Se emplean cuando desarrollarlos a partir de los principios de conservación es muy complicado. Se obtienen a partir de datos experimentales, ajustando polinomios entre las variables a relacionar. Son en principio modelos estadísticos, pues el ajuste implica un intervalo de valores posibles.

En este trabajo nos enfocaremos al modelado básico del reactor catalítico gas-sólido del proceso explicado en la sección anterior, que corresponde a un sistema diferencial, aunque tales sistemas son equivalentes a los sistemas en diferencia para su resolución (por ejemplo, los métodos de diferencias finitas [11]). El siguiente diagrama nos da una idea de lo complejo que puede ser un modelo basado en principios fisicoquímicos (la complejidad de su resolución se incrementa de izquierda a derecha).

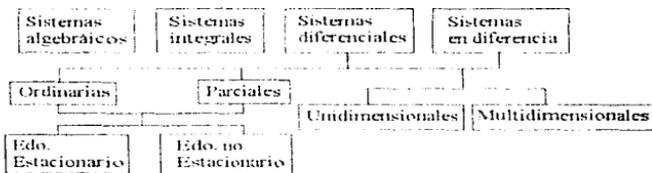


Fig. 1.3 Modelos basados en principios fisicoquímicos.

Puede tenerse además una combinación de diferentes tipos de ecuaciones para representar un sistema físico en particular.

Una de las maneras comunes de representar resultados en Ingeniería Química es, junto con las ecuaciones empíricas y los nomogramas [12], el uso de grupos o razones adimensionales. Estos grupos vienen del análisis dimensional de las ecuaciones diferenciales que representan los fenómenos físicos. Escribir las ecuaciones en forma adimensional facilita su tratamiento y su solución, además de que los grupos adimensionales son útiles por sí solos para comprender el funcionamiento de los equipos. Muchas ecuaciones empíricas y reglas de dedo se basan en el uso de grupos adimensionales sin que sea necesario el uso de ecuaciones diferenciales complejas o detalladas (e.g. [13, 14, 15, 16]).

Puede decirse que a partir de los sesentas el avance en la tecnología de computadoras y matemáticas computacionales se ha vuelto tan importante como la teoría y la experimentación, tradicionalmente los pilares de la investigación científica [17]. Estos recursos están revolucionando las áreas de optimización, materiales, procesos, control, sistemas y diseño y favoreciendo el trabajo multidisciplinario. Con ellos puede ahora modelarse a diferentes escalas. Hasta ahora, el medio continuo ha sido el más empleado por ser de utilización más inmediata para la ingeniería, aunque hoy puede modelarse a partir de la escala molecular y aun combinar ambos enfoques. El enfoque molecular complementa a la descripción continua, pues a partir de ella pueden [18]

- deducirse las ecuaciones de variación,
- obtenerse expresiones de las propiedades de transporte,
- explicarse los fenómenos con más detalle

Las manipulaciones matemáticas no pueden más que descubrir posibles consecuencias de una hipótesis y señalar las variables decisivas para el modelo. La validez del modelo solo puede probarse por medio de la experimentación. No obstante, la planeación inteligente de experimentos requiere cierto conocimiento de los modelos matemáticos y de las teorías básicas relacionadas.

Como se ha dicho en esta sección, la solución de sistemas diferenciales multivariados es bastante compleja. Para resolverlos es necesario el uso de métodos numéricos que requieren una gran cantidad de cálculos y, por lo tanto, una máquina de cálculo rápida con algoritmos eficientes. La eficiencia se incrementa con la estrategia de programación que se describe en la siguiente sección.

1.5 Estrategia de solución con programación orientada a objetos (OOP):

A finales de los años sesenta se creó la *programación estructurada* [20]. Este es un estilo de trabajo en el que un programa se separa sistemáticamente en sus componentes, los que se vuelven a separar en subcomponentes y así sucesivamente, hasta llegar a subrutinas individuales. Este estilo fue muy útil hasta finales de los ochenta, pero en los últimos años sus limitaciones se han vuelto aparentes. Uno de los últimos avances en programación estructurada son los sistemas CASE (Computer-Aided Software Engineering). Con un sistema CASE la computadora administra el proceso de descomposición funcional, produciendo diagramas en los que se pueden comprobar que las interacciones entre subrutinas son las correctas.

Otra mejora a la programación estructurada son los lenguajes de cuarta generación. Estos lenguajes tienen incluidas muchas herramientas para automatizar la generación de subrutinas. Ofrecen muchas ventajas sobre todo a quienes no son programadores. Pero su principal desventaja es que no permiten crecer a programas terminados. Este es un problema fundamental: la mayoría de los grandes programas de ingeniería y administración requieren crecer continuamente, expandirse y reutilizar datos. Para el manejo de datos se han desarrollado varios conceptos (como modelos jerárquicos, de redes o relacionales).

Puede decirse que a partir de los sesentas el avance en la tecnología de computadoras y matemáticas computacionales se ha vuelto tan importante como la teoría y la experimentación, tradicionalmente los pilares de la investigación científica [17]. Estos recursos están revolucionando las áreas de optimización, materiales, procesos, control, sistemas y diseño y favoreciendo el trabajo multidisciplinario. Con ellos puede ahora modelarse a diferentes escalas. Hasta ahora, el medio continuo ha sido el más empleado por ser de utilización más inmediata para la ingeniería, aunque hoy puede modelarse a partir de la escala molecular y aun combinar ambos enfoques. El enfoque molecular complementa a la descripción continua, pues a partir de ella pueden [18]

- deducirse las ecuaciones de variación,
- obtenerse expresiones de las propiedades de transporte,
- explicarse los fenómenos con más detalle

Las manipulaciones matemáticas no pueden más que describir posibles consecuencias de una hipótesis y señalar las variables decisivas para el modelo. La validez del modelo solo puede probarse por medio de la experimentación. No obstante, la planeación inteligente de experimentos requiere cierto conocimiento de los modelos matemáticos y de las teorías básicas relacionadas.

Como se ha dicho en esta sección, la solución de sistemas diferenciales multivariables es bastante compleja. Para resolverlos es necesario el uso de métodos numéricos que requieren una gran cantidad de cálculos y, por lo tanto, una máquina de cálculo rápida con algoritmos eficientes. La eficiencia se incrementa con la estrategia de programación que se describe en la siguiente sección.

1.5 Estrategia de solución con programación orientada a objetos (OOP):

A finales de los años sesenta se creó la *programación estructurada* [20]. Este es un estilo de trabajo en el que un programa se separa sistemáticamente en sus componentes, los que se vuelven a separar en subcomponentes y así sucesivamente, hasta llegar a subrutinas individuales. Este estilo fue muy útil hasta finales de los ochenta, pero en los últimos años sus limitaciones se han vuelto aparentes. Uno de los últimos avances en programación estructurada son los sistemas CASE (Computer-Aided Software Engineering). Con un sistema CASE la computadora administra el proceso de descomposición funcional, produciendo diagramas en los que se pueden comprobar que las interacciones entre subrutinas son las correctas.

Otra mejora a la programación estructurada son los lenguajes de cuarta generación. Estos lenguajes tienen incluidas muchas herramientas para automatizar la generación de subrutinas. Ofrecen muchas ventajas sobre todo a quienes no son programadores. Pero su principal desventaja es que no permiten crecer a programas terminados. Este es un problema fundamental. La mayoría de los grandes programas de ingeniería y administración requieren crecer continuamente, expandirse y reutilizar datos. Para el manejo de datos se han desarrollado varios conceptos (como modelos jerárquicos, de redes o relacionales).

A pesar de todos los esfuerzos para mejorar la producción de programas, los problemas siguen surgiendo cada año. Desde hace cuarenta años se construyen rutinas a mano, una instrucción a la vez. Se ha mejorado el proceso de construcción, pero mientras más grande es el programa más ineficiente es el enfoque.

El nuevo enfoque que permite mantener y expandir programas terminados es la *tecnología orientada a objetos*. Aunque este enfoque se ha popularizado últimamente, tiene en realidad más de 25 años. Todos los conceptos básicos de la programación orientada a objetos se introdujeron en el lenguaje *Simula* (acrónimo de Simulation language) desarrollado en Noruega por O. J. Dahl y K. Nygaard a finales de los sesenta. En Simula los módulos no están basados en procedimientos, sino en los objetos físicos que se modelan. La ventaja es la descomposición del problema de forma natural en las partes del objeto que se modela y no en un conjunto de instrucciones. Aunque Simula nunca fue ampliamente aceptado como un lenguaje de uso general, sus características se han adoptado en muchos lenguajes como Smalltalk (por Alan Kay de Xerox Co.) y C++, desarrollado por Bjarne Stroustrup en los laboratorios Bell de AT&T en 1986. Smalltalk está pensado para trabajar con una gran pureza en el enfoque orientado a objetos. C++ tiene lo que se ha llamado el *enfoque híbrido*, pues en este lenguaje coexisten características de la programación estructurada con orientación a objetos.

Las ventajas de los objetos son:

- Son ladrillos que modelan entidades del mundo real más que realizar tareas específicas.
- Son fáciles de reutilizar en proyectos posteriores.
- No necesitan crearse nuevos objetos para expandir un proyecto existente, y por ello ahorran tiempo a los desarrolladores de software.

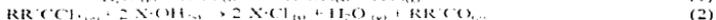
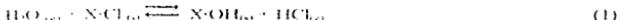
Para la programación, un *objeto* es un paquete que contiene *variables* y *métodos* relacionados entre sí. Los objetos son módulos ideales para programar, pues pueden definirse y mantenerse independientemente de los demás. Las variables representan lo que el objeto sabe, los métodos, lo que el objeto hace. Una *clase* es una plantilla que define los métodos y variables que se incluyen en un tipo determinado de objeto.

La programación orientada a objetos tiene tres propiedades fundamentales: la *herencia*, el *polimorfismo* y el *encapsulamiento*.

- **La herencia** es un mecanismo mediante el cual una clase de objetos puede definirse como un caso especial de una clase más general. Automáticamente la clase derivada hereda las definiciones de métodos y variables de la clase general. Las clases derivadas se llaman *subclases* y las clases generales se llaman *superclases*. Las subclases pueden adicionalmente definir sus propios métodos y variables. La estructura que forma la relación entre todas las clases se llama *jerarquía de clases*. Las clases pueden contener como miembros a otras clases.
- **El polimorfismo** es un proceso mediante el cual se efectúan procedimientos diferentes con el mismo nombre. Hace a los objetos más independientes entre sí y simplifica los programas de modo que se vuelve más sencillo coordinar las acciones.
- **El encapsulamiento** es el proceso de asociar datos y procedimientos relacionados y restringir el acceso a ellos a otros objetos. Así se evita la pérdida accidental de información por algún mal manejo de las variables por otro objeto.

Aunque hay trabajos publicados sobre la simulación dinámica de reactores tubulares [21, 22], muy pocos toman en cuenta las especies adsorbidas sobre el sólido. Sobre la cinética de la adsorción/desorción de cloro hay algunos artículos (ver siguiente capítulo) pero ninguno la aplica a la simulación de un reactor a través del tiempo.

Por lo mencionado anteriormente, el objetivo de este trabajo será analizar el efecto de las variables cinéticas, de transporte y operacionales en el comportamiento dinámico del balance agua-cloro dentro de un reactor tubular de isomerización de parafinas según un modelo isotérmico que considera las siguientes reacciones:



donde X es el sitio activo

Referencias:

1. Satterfield, *Heterogeneous Catalysis in Practice*, Mc Graw-Hill, Boston, 1980
2. Instituto Mexicano del Petróleo, *Manual de operación catalizadores IAP-RNA*, segunda edición, 1992
3. Meyers, *Handbook of Petroleum Refining Processes*, Mc Graw-Hill, 1986
4. Froment, Bischoff, *Chemical reactor Analysis and design*, 2nd ed., Wiley Series in Chemical Engineering, New York, 1990
5. Bond, G.C., *Heterogeneous Catalysis Principles and Applications*, 2nd Ed., Oxford University Press, 1987
6. Campbell, I., *Catalysis at Surfaces*, Chapman and Hall, 1988.
7. Smart, L., Moore, E., *Química del estado sólido*, Addison-Wesley Iberoamericana, Delaware, 1995.
8. Himmelblau, Bischoff, *Process Analysis and Simulation: Deterministic Systems*, John Wiley & sons, New York, 1968
9. Elvers, B., Hawkins, S., Schulz, G. (ed.), *Ullmann's Encyclopedia of Industrial Chemistry*, vol. B1, 1985
10. Shanley, A., Hamill, N., *Computational Fluid Dynamics. I - Pushing the limits, II - comes of age in the CPI*, Chem. Eng., 103, 66-72, 1996.
11. Ames, *Numerical Methods for Partial Differential Equations*, Academic Press, San Diego, 1977
12. Rise, H., *The Philosophy and Logic of Chemical Engineering*, Gulf Publishing Co., Houston, 1961
13. Anaya, A., *Fouling, Viscosity and the Heat Transfer Coefficient*, Chem. Eng., 103, 153-156, 1996
14. Bakker, A., Smith, J., Myers, K., *How to Disperse Gases in Liquids*, Chem. Eng., 101, 98-104, 1994
15. Cochran, P. *Simplifying Piping Network Analysis*, Chem. Eng., 102, 104-106, 1995.
16. Darby, R., *Determining Settling Rates of Particles*, Chem. Eng., 103, 109-112, 1996
17. Kevrekedis, I., *Matrices Are Forever: On Applied Mathematics and Computing in Chemical Engineering*, Chem. Eng. Sci., 24, 4005-4025, 1995

18. Bird, B., Stewart, W., Lightfoot, E., *Fenómenos de Transporte*, Ed Reverté S.A., Barcelona, 1982
19. Dumesic, J., Milligan, B., et al *A Kinetic Modeling Approach to the Design of Catalysts: Formulation of a Catalyst Advisory Program*, Ind Eng Chem Res., 26, 1399-1407, 1987
20. Taylor, D., *Object-Oriented Technology: a Manager's Guide*, Addison-Wesley, 1990.
21. Gawdzik, A., Rakowski, L., *The methods of Analysis of the Dynamic Properties of the Adiabatic Tubular Reactor*, Computers Chem. Engng., 13, 1165-1173, 1989
22. Kharabaf, H., Yonios, Y., *Solution of Hyperbolic Equations Involving Chemical Reactions*, Ind Eng Chem Res., 34, 2728-2732, 1995

PLANTEAMIENTO MATEMÁTICO DEL MODELO DEL REACTOR

A lo largo de este capítulo se hará la descripción en términos de ecuaciones diferenciales del reactor heterogéneo del proceso descrito en la sección 1.1. Se tratan primero los mecanismos de transporte. Después se desarrollan y justifican las expresiones para la velocidad de cloración del catalizador y posteriormente se discute la estabilidad numérica de la solución de los reactores de lecho empacado. Para concluir se discuten los problemas numéricos encontrados.

2.1 Introducción:

En principio, el comportamiento de un reactor catalítico heterogéneo puede simularse cuando se cuenta con coeficientes cinéticos y de transporte razonablemente precisos y adecuadamente incorporados en las ecuaciones de continuidad. Cuando se tiene un modelo que describe con suficiente aproximación el comportamiento del reactor, el análisis de sensibilidad de parámetros es de gran ayuda en el diseño.

Pero determinar el grado de sofisticación requerido para modelar un problema determinado no es sencillo. En mucho depende del *proceso* mismo, es decir, del esquema de reacción y lo sensible que éste sea al cambio en las condiciones de operación. También del grado de exactitud con que conocemos las *constantes físicas y de transporte*.

Froment y Bischoff [1] clasifican los modelos en dos categorías:

- Modelos *pseudohomogéneos*, no consideran de forma explícita la presencia del catalizador. No distinguen entre la fase fluida y la fase sólida.
- Modelos *heterogéneos*, proponen ecuaciones de conservación separadas para el fluido y el sólido.

Unidimensionales	básico, ideal	con gradiente interfacial
	con mezclado axial	con gradiente intraparticular
Bidimensionales	con mezclado axial	con mezclado radial

Tabla 2.1. Tipos de modelos de reactores tubulares

Los modelos más sencillos y de más amplio uso son los modelos pseudohomogéneos *unidimensionales*, que consideran el mezclado solo en dirección axial. Cuando se considera la coordenada radial, los modelos son *bidimensionales*.

Se ha comprobado que las predicciones de estos modelos fallan [2] si las condiciones de operación se alejan del intervalo para el cual se calcularon los parámetros. La ventaja que posee el modelo pseudohomogeneo unidimensional es la facilidad de su resolución. Pero para cuestiones de diseño se prefiere el modelo bidimensional, que permite un mejor estudio de variación de parámetros y es más cercano a la realidad.

Sin embargo hay ciertas condiciones de operación que pueden generar inestabilidad en el reactor (véase por ejemplo [3, 4]). A pesar de esta circunstancia se puede simular esta inestabilidad, aunque debe reconocerse que la predicción exacta de situaciones críticas requiere un conocimiento de los parámetros experimentales que raras veces se alcanza.

En resumen, un modelo dinámico es útil para describir un reactor de lecho fijo si cumple los siguientes requisitos [5]

1. Debe representar un sistema *conservativo*.
2. Debe describir de la forma más cercana a la realidad las variables del sistema.
3. Debe producir la solución asintótica correcta (*estado estacionario*).

Pero debemos observar que, según Schmitzlein, los modelos pseudohomogéneos no satisfacen todas estas condiciones [6]. Este autor señala que las ecuaciones diferenciales de segundo orden implican un retromezclado en gran escala que contradice los resultados experimentales y los estudios teóricos de modelos heterogéneos. Cita trabajos en los que experimentalmente se ha hallado que a altas velocidades de flujo no se observa retromezclado. Menciona también que en los modelos heterogéneos el retromezclado está limitado a un diámetro de partícula.

2.2 Descripción de los mecanismos de transporte:

Los reactores tubulares de lecho fijo se emplean comúnmente para llevar a cabo reacciones exotérmicas o endotérmicas en fase gaseosa sobre un catalizador poroso. Su diseño y/o simulación requiere consideraciones sobre las concentraciones, temperaturas y presiones en el espacio de reacción. El objetivo de estas tareas es determinar las condiciones óptimas para la reacción en términos del régimen de temperaturas y demás magnitudes asociadas, como diámetro del reactor, tamaño de las partículas catalíticas, velocidad de flujo, longitud del reactor y demás variables. Si la reacción que se simula y/o diseña es muy exotérmica, uno debe asegurarse de que el reactor tenga baja sensibilidad térmica. El resultado de estos estudios debe quedar bien registrado en el manual de operación del equipo para evitar caer en condiciones de operación que produzcan inestabilidad. (Véanse por ejemplo [1, 4]).

Para llevar a cabo reacciones muy exotérmicas se usan frecuentemente reactores del tipo de intercambiadores de calor, donde los tubos se empaquetan con el catalizador poroso y el medio de enfriamiento se hace pasar por el lado de la coraza. En estos reactores hay perfiles radiales de concentraciones y temperatura muy importantes.

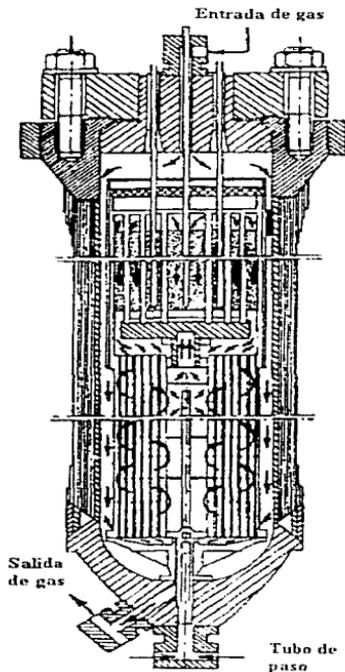


Fig. 2.1. Reactor tubular de síntesis de amoníaco con intercambiador de calor.

El modelo que se empleará en el desarrollo de este trabajo es, según la clasificación de Froment (secc 2.1), el modelo pseudohomogéneo unidimensional en

estado no estacionario (dinámico) Para llegar a una descripción de este tipo, debemos hacer una suposición básica Debemos asumir, en contraposición a la estructura física real del sistema, que trabajamos en un **medio isotrópico continuo** Esto significa que las propiedades del reactor son las mismas en cualquier dirección del espacio y en cualquier momento y que no hay distinción entre entre las fases sólida y fluida

A partir de esta suposición planteamos el balance de materia Empleando la ley de Fick de la difusión tenemos [7]

$$\frac{\partial}{\partial r} \left(r(D_r) \cdot \frac{\partial}{\partial r} C_i \right) + r \frac{\partial}{\partial z} \left(-u C_i + (D_r)_i \cdot \frac{\partial}{\partial z} C_i \right) \pm r_f \rho_B r_i = r \frac{\partial}{\partial t} C_i$$

Ecuación 2.1

La concentración tiene un subíndice porque se plantea un balance de materia por cada componente, y el signo de la expresión cinética depende de si el componente es reactivo o producto A partir de la ley de Fourier y el balance de energía se obtiene la ecuación 2.2, donde i va desde 1 hasta el número de reacciones consideradas

$$\frac{\partial}{\partial r} \left(r(k_r) \cdot \frac{\partial}{\partial r} T \right) + r \frac{\partial}{\partial z} \left(-u r C_p T + (k_r)_i \cdot \frac{\partial}{\partial z} T \right) \pm \sum_j (N I_{r,j}) \rho_B r_i = r C_p \frac{\partial}{\partial t} T$$

Ecuación 2.2

que son las ecuaciones de conservación en su forma más general (Ver nomenclatura la final del trabajo) Se han publicado varios criterios según los cuales se pueden eliminar términos para simplificar las ecuaciones Algunos son

- En reactores en estado estacionario el término de mezclado axial no es importante, siempre que las variaciones de concentración en la coordenada axial no sean muy abruptas [8] (La derivada en el tiempo y el término de difusividad axial se eliminan).
- En reactores isotérmicos matemáticamente no se considera el balance de energía. Físicamente es necesario resolver el balance de energía para que un reactor isotérmico pueda operar, aunque en la industria este caso no es común
- En reactores adiabáticos el perfil radial no es importante, a menos que la reacción sea muy exotérmica o el perfil de velocidades no sea aproximadamente plano [1,3]
- Si el reactor no es adiabático ni isotérmico, no se puede despreciar el perfil radial, a menos que la conductividad térmica radial del catalizador sea muy grande

En el presente trabajo consideraremos el **caso isotérmico de un sistema dinámico** y despreciaremos el término radial y el balance de energía, se considera un perfil plano de velocidades en r (e d. u. $\neq u(r)$) No se dispone de información sobre las expresiones de velocidad en términos de la temperatura, los valores de la entalpía ni del C_p En otros trabajos de simulación de reactores gas-sólido para reacciones de isomerización se ha encontrado que la resistencia a la transferencia de calor es poco importante y que para nuestro caso el calor de reacción no es significativo (e.g. [9] estudia la isomerización de

xilenos en fase vapor mediante un modelo heterogéneo). A continuación se plantean los balances de materia y el balance de energía considerando las simplificaciones de nuestro modelo.

El balance de materia queda

$$(D_c)_i \frac{\partial^2}{\partial z^2} C_k - u \frac{\partial}{\partial z} C_k \pm r_{p,i} \rho_B = \frac{\partial}{\partial t} C_k$$

Ecuación 2.3

y el balance de energía queda

$$(k_c)_i \frac{\partial^2}{\partial z^2} T - u \rho C_p \frac{\partial}{\partial z} T \pm \sum_i r_{p,i} \rho_B (-\Delta H_{r,i}) = \rho C_p \frac{\partial}{\partial t} T$$

Ecuación 2.4

Donde i va desde 1 hasta el número de reacciones que se consideren. Las ecuaciones 2.3 y 2.4 son el modelo a resolver. Para su resolución es conveniente ponerlas en términos de variables adimensionales

Sean nuestras variables adimensionales

$$\Phi_k = \frac{C_k}{\rho_{ref}} \quad S = \frac{z}{L} \quad \tau = \frac{t}{(L/u)}$$

Ecuaciones 2.5, 2.6 y 2.7

$$Pe_{M,k} = \frac{u \cdot dp}{(D_c \cdot k)_i} \quad Pe_{H,k} = \frac{u \cdot dp}{(k_c)_i} \quad \Theta = \frac{T}{T_0}$$

Ecuaciones 2.8, 2.9 y 2.10

El balance de materia adimensionalizado queda

$$\frac{1}{Pe_{M,k}} \frac{d_p}{L} \frac{\partial^2}{\partial S^2} \Phi_k - \frac{\partial}{\partial S} \Phi_k \pm \frac{L}{u \rho_{ref}} r_{p,i} \rho_B = \frac{\partial}{\partial \tau} \Phi_k$$

Ecuación 2.11

y el balance de energía es

$$\frac{1}{Pe_{H,k}} \frac{d_p}{L} \frac{\partial^2}{\partial S^2} \Theta - \frac{\partial}{\partial S} \Theta + \frac{L}{T_0 u \rho C_p} \sum_i r_{p,i} (-\Delta H_{r,i}) = \frac{\partial}{\partial \tau} \Theta$$

Ecuación 2.12

En la forma adimensional el valor de las variables $\Phi_i(s, \tau)$ está acotados entre cero y uno, y el valor de $\Theta(s, \tau)$ puede ser mayor o menor a uno, para detalles en el procedimiento de adimensionalización véanse por ejemplo [10] y [11]. Cabe hacer notar que el *pecllet* empleado en este planteamiento es u_d / D_1 . Así, para n componentes tendremos n balances de materia más un balance de energía global. Tendremos entonces $n + 1$ variables con $n + 1$ ecuaciones diferenciales parciales de tipo parabólico (Ver apéndice A). Vale la pena notar que las $n + 1$ ecuaciones tienen la misma forma. Cada ecuación requiere dos condiciones de frontera espaciales y una temporal, por lo que necesitamos en total $3 \cdot (n + 1)$ condiciones de frontera.

Los parámetros que emplean las ecuaciones diferenciales pueden clasificarse a grandes rasgos en *cinéticos, de transporte y de operación*. A continuación se discuten los diferentes fenómenos de transporte que ocurren en la realidad (sección 2.2.1) y los parámetros operacionales (sección 2.2.2) [12]. Luego se discuten las condiciones de frontera que se consideran correctas para un sistema de flujo de este tipo (sección 2.2.3).

2.2.1 Fenómenos de transporte

Transferencia de masa y calor.

a) Dentro y fuera de la partícula catalítica.

Para un análisis riguroso se debe considerar el transporte en la estructura catalítica. Si fuésemos a considerar un modelo más completo (heterogéneo no isotérmico), habría cinco factores importantes a considerar para la fase sólida: módulo de Thiele (ϕ), aumento adimensional de temperatura dentro de la partícula (β), energía de activación adimensional (γ), número de Sherwood (Sh) y número de Nusselt (Nu) [12]. Para las definiciones formales de estas magnitudes puede verse la Nomenclatura al final del trabajo.

En el caso pseudohomogéneo isotérmico unidimensional que nos ocupa, sólo emplearemos el número de Peclet axial de masa ($Pe_{m,z}$).

b) En el lecho empacado.

Las condiciones hidrodinámicas dentro del lecho empacado son tan complejas que es imposible describirlas en otros términos que no sean los estadísticos. Esto se debe a la turbulencia que genera la presencia de las partículas catalíticas y los efectos convectivos de la transferencia de calor sobre el fluido.

Mezclado axial.

Físicamente el mezclado axial produce un retromezclado de especies que ya han reaccionado y un mezclado hacia adelante de reactivos.

El coeficiente determinante para el mezclado es el número de Peclet de masa (Pe_m), pues incluye la difusividad y la velocidad de flujo. Froment [1] recomienda valores de Peclet para la coordenada axial entre 1 y 2, y Carberry [8, 17] ha demostrado que variaciones en Pe son irrelevantes para razones entre la longitud del reactor y diámetro de partícula mayores a 100, como es nuestro caso.

En reactores industriales, el tamaño de las partículas catalíticas es crítico. Partículas pequeñas mejoran la efectividad, pero raras veces se justifica económicamente su uso debido a la gran caída de presión.

Mezclado radial.

El valor de Peclet recomendado para el mezclado radial está entre 8 y 10. Su valor es mayor porque la difusividad en sentido radial es menor. La importancia de estos fenómenos es muy grande para los reactores operados industrialmente, como el de la figura 2.1. Se ha encontrado que los perfiles no muestran gran sensibilidad a esta constante.

Conductividad térmica.

La transferencia de calor en reactores de lecho empacado es un fenómeno compuesto por diferentes mecanismos que pueden estudiarse separadamente. Estudios de Yagi y Kunii [13] señalan que el flux radial de calor se debe a la conducción y radiación entre las fases sólida y fluida (factores estáticos) y factores dinámicos que pueden relacionarse con el número de Peclet.

Coefficiente de transferencia de calor en la pared.

Si el reactor no es isotérmico ni adiabático, se hace necesario establecer una temperatura o un perfil de temperaturas como condición de frontera. Pero hay mucha más incertidumbre en la estimación del coeficiente de transferencia de calor en la pared del reactor que en la estimación de la conductividad térmica en el lecho empacado [12]. En esta misma referencia se dan criterios para estimar este coeficiente, aunque los datos están muy dispersos.

Radiación.

Los efectos de la radiación se vuelven importantes a altas temperaturas (mayores a 400 °C), especialmente si las partículas catalíticas son grandes.

Distribución de velocidades y porosidad en el lecho empacado.

Un modelo realista puede incorporar también estas variables. Se ha encontrado (referencia citada por Hlavacek [12]) que el valor mínimo de la fracción hueca se encuentra cerca del eje del reactor, y una mayor fracción hueca se observa cerca de la pared. Esto afecta también el perfil de velocidades, pues la misma referencia indica que la velocidad alcanza un máximo cerca de la pared del reactor y un mínimo cerca del eje.

2.2.2. Parámetros operacionales.

Los parámetros operacionales son d_p (diámetro de partícula), número de Damköhler (Da), número de Biot (Bi), temperatura de enfriamiento si el reactor no es isotérmico ni adiabático (T_c), longitud del reactor y flujo volumétrico. Los criterios usuales para estas cantidades son [12]

$$0.04 \leq d_p / R_0 \leq 0.25$$

Ecuación 2.13

$$-0.5 < \Theta_0 < 1$$

Ecuación 2. 14

$$0.02 < Da < 1$$

Ecuación 2. 15

donde R_0 es el radio del reactor tubular, $\Theta_0 = (T - T_c) / (R \cdot T_c^2)$, T_c una temperatura de referencia, y el número de Damkohler es $Da = k_0 \theta_0 C_0^{n-1}$ (n es el orden de reacción, θ_0 es el tiempo de contacto y k_0 es la constante de reacción)

2.2.3. Condiciones de frontera

Para el estudio de un fenómeno físico (sistema), este se separa del resto del universo (alrededores). Las *fronteras*, que son superficies matemáticas idealizadas, separan la región en estudio del resto del universo. En un sentido puramente matemático, las condiciones de frontera son necesarias para obtener una solución particular de las ecuaciones de conservación en forma diferencial. Pero desde el punto de vista físico deben representar a todo el universo externo al sistema y expresar la interacción entre el sistema y los alrededores.

Para el problema del flujo a través de un lecho empacado se ha discutido bastante sobre cuáles deben ser las condiciones de frontera adecuadas. Se discuten a continuación los principales trabajos al respecto.

Danckwerts resolvió en 1953 [14] la ecuación diferencial ordinaria que describe el transporte por convección y mezclado axial en estado estacionario de una especie que sufre una transformación química con cinética de primer orden. La forma adimensional de esta ecuación es

$$\frac{1}{Pe} \frac{d^2 f}{ds^2} + f \frac{d f}{ds} - R f = 0 \quad \text{Ecuación 2. 16}$$

donde f es la fracción de reactivo restante, s es la coordenada axial adimensionalizada (ecuación 2.6) y R es el número de Damkohler para una reacción de primer orden. Notar la similitud con la ecuación 2.11. Notar que en este caso el número de Peclet es $(L_c / D_c) u$, y no $(u d_p / D_c)$ como en el planteamiento de nuestro problema. Si la longitud característica es d_p , el valor del Peclet es alrededor de 2, si la longitud característica es L_c , el valor de Pe es mucho más grande [11].

Para derivar sus famosas condiciones de frontera, Danckwerts planteó un balance de materia para un plano perpendicular al flujo

$$f(-) - \frac{1}{Pe(-)} \frac{df}{ds}(-) = f(+), \quad \frac{1}{Pe(+)} \frac{df}{ds}(+) \quad \text{Ecuación 2. 17}$$

donde $f(-)$ es el término convectivo y $(1/Pe)f(+)$ es debido a la difusión. los signos $(-)$ y $(+)$ significan que el término corresponde a la posición inmediatamente anterior e inmediatamente posterior al plano

La suposición crucial de Danckwerts es que el mezclado sucede sólo dentro del reactor, es decir, el coeficiente de difusividad pasa de cero (antes de entrar) a un valor constante (dentro del reactor) y otra vez a cero (al salir). Entonces a la entrada el balance queda:

$$f(+)-\frac{1}{Pe(-)}\frac{df}{ds}(-)=f(-)-1 \quad \text{Ecuación 2. 18}$$

y a la salida, puesto que el gradiente no puede ser ni positivo ni negativo, debe ser cero:

$$\frac{df}{ds}(1)=0 \quad \text{Ecuación 2. 19}$$

Las ecuaciones 2.18 y 2.19 son conocidas como las *condiciones de frontera de Danckwerts*, aunque la solución a este problema ya la había encontrado Langmuir en 1908 [15].

Wehner y Wilhelm [16] resolvieron el mismo problema pero consideraron regiones infinitas las de entrada y salida en las que se produce dispersión axial y una región finita de reacción (Fig. 2.2). Su modelo consistía en un sistema de tres ecuaciones diferenciales ordinarias, una por región. La ecuación que representa la sección de reacción ($0 \leq z \leq 1$) es igual a la de Danckwerts, ecuación 2.10. Las ecuaciones que representan las secciones infinitas de entrada y salida son:

$$\frac{1}{Pe_e} \frac{d^2}{ds^2} f - \frac{d}{ds} f = 0 \Leftrightarrow s \leq 0$$

$$\frac{1}{Pe_s} \frac{d^2}{ds^2} f - \frac{d}{ds} f = 0 \Leftrightarrow s \geq 1$$

Ecuaciones 2. 20 y 2. 21

para las que necesitaban seis condiciones de frontera (dos para cada una de las ecuaciones 2.16, 2.20 y 2.21).

- Fracciones continuas de reactivo en las fronteras de entrada y salida del reactor.
- Fluxes continuos de reactivo en las fronteras de entrada y salida del reactor.
- Fracciones finitas de reactivo al inicio de la sección de entrada y al final de la sección de salida.

La solución que obtuvieron para la sección de reacción es idéntica a la de Danckwerts, y su principal conclusión es que la distribución de reactivo dentro del reactor es completamente independiente del grado de mezclado axial en las secciones de entrada y salida, siempre que las secciones anterior y posterior al reactor tengan longitud infinita.

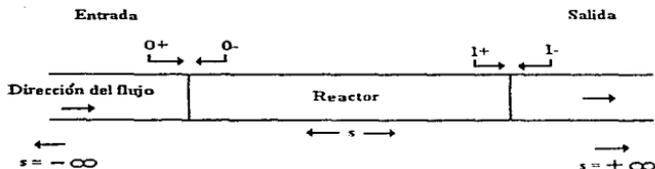


Fig. 2.2. Diagrama del sistema de flujo propuesto por Wehner y Wilhelm .

Pearson demostró rigurosamente en 1959 la validez de la ec 2.12 y Bischoff halló en 1961 que la solución de la sección de reacción es independiente del término cinético (citas hechas por Novy et al [15]). Usando el mismo modelo de Wehner y Wilhelm, excepto que las longitudes de las secciones de entrada y salida son finitas, Novy et. al. demostraron que la solución obtenida por Danckerts es una situación límite. Esta solución no es válida cuando las secciones anterior y posterior no son infinitas. Estos autores también estudiaron el modelo dinámico de las tres secciones, usando ecuaciones del tipo de la ecuación 2.5 utilizando una supercomputadora CRAY-2.

A continuación discutiremos una gráfica muy interesante del trabajo de Novy et al (fig 2.3)

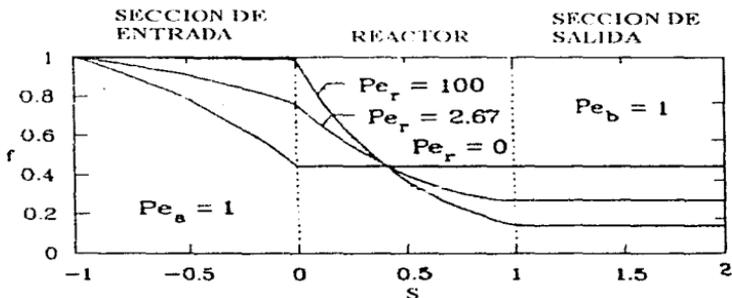


Fig. 2.3. Distribuciones de reactivo en estado estacionario con longitudes de las secciones de entrada y salida finitas (línea sólida) e infinitas (línea punteada).

De la figura podemos notar que los sistemas con valores del número de Peclet muy grandes (sistemas donde casi no hay mezclado axial) son menos sensibles a cambios en la longitud de la sección de entrada. Observar la poca pendiente en la primera sección para $Pe_{reactor} = 100$. El otro extremo ($Pe_{reactor} = 0$) representa el comportamiento de un reactor de tanque agitado por la importancia de su mezclado axial. En este caso es evidente la gran dependencia de la fracción de reactivo respecto a la longitud de la sección de entrada. Las líneas punteadas son las soluciones asintóticas cuando $s \rightarrow +\infty$.

Las conclusiones de Novy et al. son las siguientes. Para modelos en estado estacionario y no estacionario, la localización de la frontera de entrada de la primera sección y de la frontera de salida de la tercera sección tienen un efecto muy grande en las predicciones del perfil en la sección de reacción. Mientras mayores sean las longitudes de las secciones de entrada y salida, mayor será la aproximación a las condiciones de Danckwerts. La elección de condiciones de frontera no es algo que deba tomarse a la ligera en modelos con fuerte carácter elíptico (véase apéndice A).

Las condiciones de frontera de nuestro modelo se basan en las condiciones de Danckwerts. Para justificar que la concentración a la entrada del reactor es igual a la de la corriente de alimentación empleamos la solución analítica de Wehner y Wilhelm para la primera de las tres secciones [16]

$$\frac{1 - \Phi_k}{1 - \Phi_k(0)} = \exp(Pe_{entrada} s) \quad \text{Ecuación 2. 22}$$

Como se aprecia en la figura 2.3, mientras mayor sea el Peclet en la sección de entrada (menor difusividad), el perfil se acerca más a una función escalón. La ecuación 2.22 nos dice que $\Phi \rightarrow \Phi(0)$ (donde Φ es la concentración adimensional) si $Pe_{entrada} \rightarrow 0$ ó $s \rightarrow 0$. No podemos justificar un valor diferente que el de la corriente de alimentación para $\Phi(0)$ porque tendríamos que simular las tres secciones de modo dinámico.

Para la salida consideramos que la derivada de las concentraciones respecto a la longitud a cualquier tiempo es cero:

$$\left(\frac{\partial \Phi_k}{\partial s} \right)_{s=1} = 0 \quad \text{Ecuación 2. 23}$$

Y la temperatura inicial del reactor será la temperatura ambiente para la simulación adiabática

2.3 Descripción de las expresiones cinéticas:

Además de las reacciones de isomerización de n-alcanos, que no se modelan en este trabajo, en nuestro sistema se lleva a cabo la regeneración continua del cloro que se

encuentra en la superficie del catalizador. Con la corriente de reactivo se mezcla un compuesto que regenera continuamente los sitios clorados, como puede observarse en el diagrama del proceso Penex de UOP (sección 1.1)

En esta sección se discute principalmente la expresión cinética de la cloración de catalizadores de metales del grupo VIII B en alumina y, la cinética de las reacciones de isomerización y la desactivación del catalizador durante su operación.

2.3.1 Reacciones de isomerización

Aunque el objetivo del presente trabajo no es modelar las reacciones de isomerización de parafinas, a continuación se discute a grandes rasgos el mecanismo utilizando n-pentano [24].

El sitio ácido produce sobre la parafina (que es un nucleófilo) una carga positiva. Esta carga positiva es más estable sobre un carbono terciario que sobre uno secundario, y la misma carga es más estable sobre un carbono secundario que sobre uno primario.

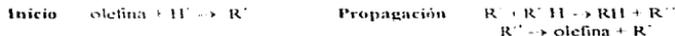


En el segundo paso de esta secuencia, el hidrógeno recibe la carga positiva, en este mecanismo se supone que la isomerización del n-pentano ocurre por un intermediario que contiene un anillo de ciclopropano. Numeremos los enlaces de este ciclo en el intermediario



Al romper el enlace 1 la reacción se regresa. Si se rompen los enlaces 2 ó 3 se produce el carbocatión que produce el esqueleto del isopentano.

Al ser las olefinas nucleófilos más fuertes que las parafinas, las reacciones de isomerización de las olefinas son también más rápidas. Una versión simplificada del mecanismo es la siguiente [24]



La distribución de productos de la isomerización del n-hexano de acuerdo a la temperatura según Goble [22] es la siguiente

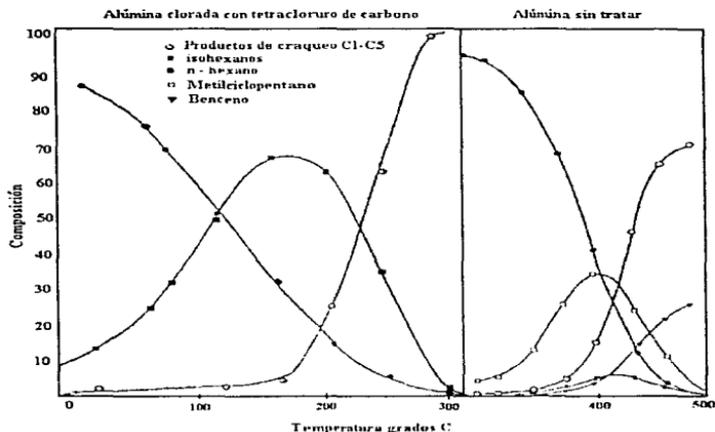


Fig. 2.4 Efecto de la temperatura sobre los productos de la reacción de n-hexano sobre alúmina clorada. Distribución de productos.

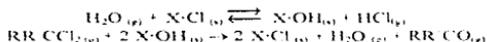
2.3.2 Reacciones de cloración de catalizadores de metales del grupo VIII en alúmina:

Goble [22] estudió la cloración de alúmina para su uso específico en reactores de isomerización. Anteriormente estos reactores empleaban un catalizador preparado haciendo reaccionar vapor de tricloruro de aluminio anhidro con catalizadores comerciales de reformación de platino-alumina, pero este sistema tiene graves desventajas: el $AlCl_3$ es un químico difícil de manejar, se hidroliza al contacto con el agua y tiende a tapar los poros de la alumina. Posteriormente se descubrió que podía clorarse la alumina de catalizadores comerciales haciendo pasar por ella tetracloruro de carbono en fase gasosa. Esta reacción es muy exotérmica, pero si se controla la temperatura dentro del sistema y la concentración del agente clorante entonces es posible obtener catalizadores bifuncionales de alta porosidad que conserven la estructura de la alumina y que contengan hasta 14 % en peso de cloro.

Goble observó que las aluminas activas para la isomerización eran aquellas que se cloraban con derivados del metano que contienen al menos dos átomos de cloro por

molécula [22], obteniendo muy buena selectividad cuando empleo tetracloruro de carbono. El agente clorador debe ser tal que pueda donar dos átomos de cloro y recibir un átomo de oxígeno de modo que el producto oxigenado resultante pueda ser fácilmente removido sin dejar residuos que puedan envenenar el catalizador. Por ejemplo, con CCl_4 se obtiene fósgeno en el efluente, lo que además indica que el fósgeno debe ser un intermediario en el mecanismo. Cornet et al. fueron un paso más allá en su estudio sobre la alquilación de isobutano por etileno [23] al comparar las velocidades de desactivación de catalizadores clorados con diferentes compuestos. Sus resultados son, en orden creciente de velocidad de desactivación: hexacloropropanona ($\text{C}_2\text{Cl}_6\text{O}$), hexacloroetano (C_2Cl_6), tetracloruro de carbono (CCl_4) y cloruro de tionilo (SOCl_2). La actividad de sus catalizadores se incrementó después de inyectar HCl, aunque el HCl por sí solo no clora la alúmina. Sin embargo, encontraron que la acidez, que es la propiedad que confiere el cloro a la alúmina, no depende del agente clorante.

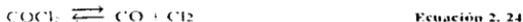
Para el presente estudio teórico y sólo para ilustrar el comportamiento de un compuesto clorante según las reacciones



se elige el fósgeno como caso particular de $\text{RR}\cdot\text{CCl}_2$. Las razones son las siguientes:

- Es un compuesto que tiene dos átomos de cloro unidos a un átomo de carbono
- Puede sustituir estos átomos por un átomo de oxígeno produciendo CO_2 , que no envenena al catalizador
- Es un compuesto que se considera intermediario en la cloración con CCl_4

De acuerdo a las referencias citadas al inicio de esta sección, cuando se lleva a cabo la cloración de alúmina se introducen CO , Cl_2 , y/o COCl_2 en un gas inerte como el nitrógeno. El fósgeno es un compuesto que se produce aunque la corriente de alimentación contenga exclusivamente monóxido de carbono y cloro:



Los primeros estudios sobre la cloración en óxidos metálicos tuvieron como objetivo determinar la temperatura mínima a la que aparecen productos de reacción, tales como cloruros metálicos (referencia citada por Goble). Alder et al. [19] descubrieron que el papel del área interfacial del óxido de aluminio es muy importante para la cloración. Ellos propusieron la siguiente ecuación:

$$r_s = (1-S)(dM/dt) = k_{12}P_{\text{CO}}^2 P_{\text{Cl}_2} \quad \text{Ecuación 2.25}$$

Encontraron que a 400 °C, la reacción química es el paso dominante de la reacción. Entre 500 y 800 °C, la transferencia de masa es el paso lento. Milne (Referencia

citada en Szabó) supuso la existencia de un complejo quimisorbido del tipo del fósgeno como un primer paso de la reacción. Esta suposición la justifican los resultados de Adelman y Burnet [20, 21], quienes encontraron que las velocidades de reacción obtenidas usando fósgeno como corriente de alimentación son mayores que cuando se alimentan monóxido de carbono y cloro, al alimentar fósgeno se eliminan pasos del mecanismo. Milne propuso a su vez la ecuación:

$$d C_{Cl_2} / dt = -k C_{Cl_2} C_{Cl_2}$$

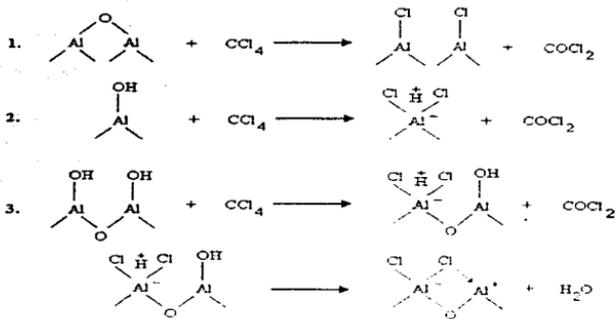
Ecuación 2. 26

observó que la mayor rapidez de cloración se producía con una alimentación equimolar de cloro y monóxido de carbono y también concluyó que a 700 °C el paso dominante es la difusión. Bertóti et al (1980, citada también en Szabó et. al.) determinó las temperaturas límite de adsorción física de las reacciones de $\gamma - Al_2O_3$ con $COCl_2$ (470 K) y $CO + Cl_2$ (430 K), es decir, las temperaturas a las que comienza la quimisorción. También estudió las temperaturas en las que se inicia la cloración (550 y 560 K respectivamente). Bertóti encontró una dependencia anómala entre la rapidez de la reacción y la temperatura (no se cumple la ecuación de Arrhenius) que adjudicó a la descomposición térmica del fósgeno. Hauthal et al obtuvieron resultados de los que se infiere que la concentración de hidroxilos superficiales tiene gran importancia para el mecanismo de cloración.

Ya Goble observó por medio del estudio de infrarrojo que la cloración procede no con la alúmina como un todo, sino que se debe a reacciones selectivas con las especies superficiales. Puede considerarse que estas especies son:

1. Iones óxido
2. Grupos hidroxilo aislados
3. Grupos hidroxilo en parejas que no pueden deshidratarse térmicamente sin producir defectos energéticamente inestables

Cuando se trabaja con CCl_4 , las reacciones con cada uno de estos grupos son:



La reacción 3 se inicia como la reacción 2, pero la configuración intermedia se deshidrata de tal manera que se produce una estructura deficiente de electrones estabilizada por un átomo de cloro. El producto de la reacción 2 es un *sitio ácido de Bronsted*. La reacción 3 produce también un sitio ácido de Bronsted, pero el protón se combina con el hidroxilo adyacente y produce un *sitio ácido de Lewis*. Es así como la cloración confiere acidez a la alúmina.

El modelo reportado más completo de cloración de alúmina se debe a Szabó et al. [25,26] y se explica en el apéndice C. A pesar de que este trabajo se desarrolló para la producción de $AlCl_3$, y no para la cloración superficial de la alúmina, el modelo nos será de utilidad porque discute la cinética de adsorción del fosgeno sobre el sólido y de la cinética de la descomposición del fosgeno en fase gaseosa.

2.3.3 Desactivación del catalizador durante su tiempo de operación

La desactivación de los catalizadores es un fenómeno muy importante para la industria y para la investigación básica de la cinética química. Para que un proceso sea viable industrialmente la desactivación debe mantenerse en un mínimo. Hay dos maneras de conseguir este objetivo:

- Mediante un desarrollo adecuado del catalizador (véase por ejemplo Dumesic et al. [18] o sección 1.3)

- Mediante un control cuidadoso de la alimentación y condiciones de operación del reactor.

Cuando el catalizador se desactiva es necesario someterlo a procesos de *regeneración*. La regeneración se usa frecuentemente cuando la desactivación se debe a deposición de carbón o por sinterizado. Los periodos de regeneración pueden variar desde pocos minutos (regeneración continua del catalizador de craqueo de hidrocarburos con zeolitas sintéticas) hasta muchos meses (por ejemplo en catalizadores de metales de transición para reformación, que es el caso que nos ocupa). En una regeneración controlada cuidadosamente se restablece casi completamente la actividad del catalizador. En modelos muy rigurosos se vuelve necesario ajustar el modelo cinético para compensar los efectos de la desactivación. Estudios experimentales demuestran que la rapidez de la reacción puede decaer linealmente respecto a la fracción de superficie desactivada o más rápidamente. El segundo caso se presenta si la sustancia envenenante se adsorbe selectivamente sobre los sitios catalíticos activos de la reacción principal, o si la reacción ocurre en la parte exterior de la partícula catalítica y el veneno se adsorbe en el mismo lugar.

En algunos estudios ([27] y otros citados por Lynch [28]) se ha supuesto que la actividad α puede separarse de la actividad cinética r , expresando la rapidez total r de la forma

$$r = \alpha (\text{historia pasada}) \cdot r, (\text{condiciones presentes}) \quad \text{Ecuación 2. 27}$$

en vez de la forma general no separable

$$r = r (\text{historia pasada, condiciones presentes}) \quad \text{Ecuación 2. 28}$$

En los casos en que esta suposición es correcta, entonces es posible estudiar la desactivación separada de la cinética. Sin embargo, esta suposición no siempre es válida. Lynch demostró que no se cumple cuando la desactivación se explica mediante un mecanismo acoplado a la reacción principal (desada), sin importar que la reacción se lleve a cabo en una superficie no ideal en el sentido de Langmuir. (Una superficie ideal es energéticamente uniforme y las moléculas adsorbidas no interactúan con sus vecinas.)

Un modelo sencillo de desactivación para una sola partícula catalítica que cumple la suposición anterior es el siguiente. Supongamos que los pasos de adsorción y reacción son rápidos respecto a la difusión intraparticular. Entonces la superficie de la partícula se envenenará uniformemente. Si α es la fracción de la superficie envenenada, la constante de reacción de la reacción principal será $k_1(1-\alpha)$. La rapidez de reacción de la partícula catalítica será [7]

$$r_p = \eta \cdot (1-\alpha) \cdot k_1 \cdot C_s \quad \text{Ecuación 2. 29}$$

donde η es el factor de efectividad para la partícula catalítica y $(1-\alpha)$ corresponde a α (historia pasada).

También se han realizado estudios en los que simultáneamente se plantea la activación y la desactivación del catalizador. Estos modelos son más complicados y se emplean para el análisis cinético riguroso de sistemas en los que la activación y la desactivación suceden al mismo tiempo. Incluso proponen que de cierta manera las especies (soporte, promotor, fase activa, reactivos, intermediarios y productos) interactúan para producir un comportamiento en el que la desactivación no es solo un descenso de actividad contra el tiempo, sino que la actividad alcanza un máximo antes de comenzar a disminuir [29]. Reacciones catalíticas reportadas con este comportamiento son la polimerización de etileno, polimerización de alcohol bencílico, procesos de hidrodesulfuración [30], reacciones de CO con hidrógeno y síntesis de Fischer-Tropsch (véanse referencias citadas por Agorreta et al [29]).

Para el proceso descrito en la sección 1.1, los *contaminantes* más frecuentes son *azufre, nitrógeno, metales pesados y arsénico* [32]. También compuestos oxigenados y cloruros contenidos en la carga pueden ser considerados contaminantes debido a que cambios en los mismos ocasionan el descontrol de la relación agua-cloro en el catalizador (ver sección 2.3.2).

El azufre se encuentra en proporciones que pueden ir del 0.05 % hasta el 5 % en peso. En las fracciones pesadas el contenido de azufre es mayor que en las ligeras y se encuentra en forma de derivados del tiofeno, mientras que en las ligeras se halla en forma de sulfuros o de disulfuros orgánicos, mercaptanos, tiofenos y sulfuro de hidrógeno. Si su concentración es mayor a 0.2 ppm en la carga, la operación será deficiente. El azufre es un veneno de la función metálica en los catalizadores bifuncionales, pero es temporal. Su efecto negativo se debe a que promueve el depósito de carbón.

Usualmente el nitrógeno está en concentraciones entre 0.1 % y 0.9 % en peso. Al igual que el azufre, la proporción de nitrógeno se incrementa en las fracciones pesadas, aunque sus compuestos no se han identificado tan bien como los del azufre. La mayoría son heterocíclicos, básicos (piridinas, quinolinas) y no básicos (derivados del indol, pirrol y carbazol). Los compuestos del nitrógeno producen remoción de cloruros del catalizador y el depósito de sales de cloruro de amonio en los equipos corriente abajo de los reactores.

El oxígeno está en las fracciones ligeras y medias principalmente como ácidos carboxílicos y, en pequeñas cantidades, como fenoles. Provocan el arrastre de cloruros en el catalizador y como consecuencia producen un desequilibrio en el balance agua-cloro.

La contaminación con metales pesados y arsénico es irreversible, por lo que debe localizarse y eliminarse su fuente. Los compuestos de metales pesados y arsénico se hallan en las fracciones pesadas del petróleo.

CONTAMINANTES	INTERVALO DE CONCENTRACION (ppm)
Cobre	150 - 250
Plomo	300 - 500
Cobalto	400 - 600
Molibdeno	400 - 600
Sodio	400 - 600
Fósforo	400 - 600
Silicio	> 1000
Arsénico	100 - 200

Tabla 2.2. Concentraciones máximas de venenos para catalizadores de reformación de naftas.

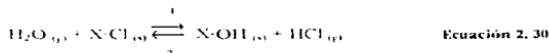
Una sobrecloración también produce efectos negativos como pérdida del gradiente de temperatura en los reactivos y disminución drástica de la duración del ciclo de operación, aunque al inicio se incrementa el número de octano del reformado. Del mismo modo, el exceso de humedad arrastra los cloruros y baja la actividad del catalizador; muy poca humedad producirá desintegración en metano y etano [31].

En la simulación desarrollada en este trabajo no se toma en cuenta la desactivación del catalizador.

2.3.4 Modelo cinético empleado en la simulación.

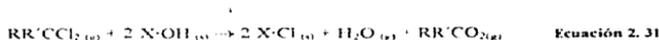
Para el *modelo cinético* de este trabajo se hacen las siguientes suposiciones.

1. La alumina está cubierta de una monocapa de iones hidroxilo que pueden sustituirse por cloruros. No se considera la participación de iones oxido en la superficie.
2. La suma de las concentraciones de los iones hidroxilo y cloruro es constante.
3. El paso determinante es el de la reacción superficial. La adsorción y desorción son relativamente rápidos.
4. La reacción de desorción es



que es reversible y sucede en un solo paso.

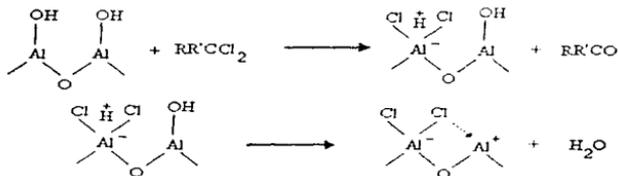
5. La reacción de adsorción sucede entre hidroxilos vecinos y es



que se considera irreversible porque:

- S-Cl + RR'CO no produce nada
- S-Cl + H₂O se considera en la reacción de desorción

- Es muy improbable el orden de reacción -4 en la reacción inversa
- Aunque para esta simulación la consideremos así, no es una reacción en un solo paso, como puede verse en el siguiente mecanismo para compuestos biclorados, análogo a los de la página 27



Las expresiones cinéticas quedan, considerando al fosgeno como ejemplo particular:

$$r_d = k_1 C_{H_2O} \bar{C}_{Cl} - k_2 C_{HCl} \bar{C}_{OH}$$

Ecuación 2. 32

$$r_a = k_3 C_{COCl_2} \bar{C}_{OH}^2$$

Ecuación 2. 33

Sean nuestras variables adimensionales

$$\Phi_0 = \frac{C_{H_2O}}{\rho_{f,ref}} \quad \Phi_1 = \frac{C_{HCl}}{\rho_{f,ref}} \quad \Phi_2 = \frac{C_{COCl_2}}{\rho_{f,ref}}$$

$$\Phi_3 = \frac{\bar{C}_{CO_2}}{\rho_{s,ref}} \quad \Phi_4 = \frac{\bar{C}_{Cl}}{\rho_{s,ref}} \quad \Phi_5 = \frac{\bar{C}_{OH}}{\rho_{s,ref}}$$

donde $\rho_{s,ref}$ y $\rho_{f,ref}$ son densidades molares en la superficie y en el fluido, respectivamente. Entonces las expresiones cinéticas en forma adimensional quedan:

$$r_d = \rho_{s,ref} \rho_{f,ref} (k_1 \Phi_0 \Phi_4 - k_2 \Phi_1 \Phi_5)$$

Ecuación 2. 34

$$r_a = k_3 \rho_{s,ref}^2 \rho_{f,ref} \Phi_2 \Phi_5^2$$

Ecuación 2.35

El buen funcionamiento de los reactores de isomerización de parafinas dependerá en mucho del adecuado balance agua-cloro en el catalizador.

2.4 Problemas numéricos:

La elección del tamaño de paso para las coordenadas espacial y temporal es crítica para los métodos de diferencias finitas. También el tiempo de cálculo es un inconveniente para resolver los problemas con este enfoque. Cuando se escoge un tamaño de paso grande pueden observarse oscilaciones que no tienen ningún significado físico.

El enlazado de las rutinas FORTRAN con el programa principal en C++ también representa gran dificultad.

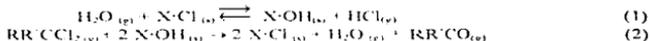
Referencias:

1. Froment, G.F., *Fixed Bed Catalytic Reactors. Current Design Status*, Industrial and Engineering Chemistry, 59, 18-27, 1967.
2. Froment, G.F. Bischoff, K.B., *Chemical Reactor Analysis and Design*, 2nd ed., John Wiley and Sons, 1990.
3. Liu, S., Amundson, N., *Stability of adiabatic Packed-Bed Reactors. An Elementary Treatment*, Ind Eng Chem Fund., 1, 201-208, 1962.
4. Liu, S., Amundson, N., *Stability of adiabatic Packed-Bed Reactors. Effect of Axial Mixing*, Ind Eng Chem Fundamentals, 2, 183-189, 1962.
5. Sundaresan, S., Amundson, N., Aris, R., *Observation of Fixed-Bed dispersion models: The Role of the Interstitial Fluid*, AIChE J., 26, 529-536, 1980.
6. Schmitzlein, K., Hoffmann, H., *An Alternative Model for Catalytic Fixed Bed Reactors*, Chem Eng Sci., 42, 2569-2577, 1987.
7. Smith, J.M., *Chemical Engineering Kinetics*, 3rd Ed., Mc. Graw-Hill, Singapore, 1981.
8. Carberry, J., Wendel, M., *A Computer Model of the Fixed Bed Catalytic Reactor: The Adiabatic and Quasi-Adiabatic Cases*, AIChE J., 9, 129-133, 1963.
9. González Rodríguez, H., *Aplicaciones de los Modelos Matemáticos para los Reactores Catalíticos Gas-Sólido y Gas-Líquido-Sólido*, tesis de Maestría en Ingeniería Química, DEPg EQ UNAM, 1996.
10. Gardini, L., Servida, A., *Use of orthogonal Collocation on finite elements with moving Boundaries for Fixed Bed Catalytic Reactor Simulation*, Computers and Chemical Engineering, 9, 1-17, 1985.
11. Carberry, J., *Chemical and Catalytic Reaction Engineering*, Mc Graw-Hill, 1976.
12. Hlavacek, V., *Aspects in Design of Packed Bed Catalytic Reactors*, Ind Eng Chem. Res., 62, 8-26, 1970.
13. Yagi, S., Kumi, D., *Studies on Heat Transfer Near Wall Surface in Packed Beds*, AIChE J., 6, 97-104, 1960.

14. Danckwerts, P. V., *Continuous Flow Systems: Distribution of Residence Times*, Chem Eng Sci., 2, 1-13, **1953**.
15. Novy, R., Davis, H., Scriven, L.E., *Upstream and Downstream Boundary Conditions for Continuous Flow Systems*, Chem Eng Sci., 45, 1515-1524, **1990**.
16. Wehner, J.F., Wilhelm, R.H., *Boundary Conditions of Flow Reactor*, Chem Eng Sci., 6, 89-93, **1956**.
17. Carberry, J., White, D., *On the Role of Transport Phenomena in Catalytic Reactor Behavior*, Ind Eng Chem Res., 61, 27-35, **1969**.
18. Dumesic, J. A., Miligan, B. A., et al., *A Kinetic Modeling Approach to the Design of Catalysts: Formulation of a Catalyst Design Advisory Program*, Ind Eng Chem Res., 26, 1300-1407, **1987**.
19. Alder, H.P., Muller, H.P., Richarz, W., *Kinetic Study of the Alumina Chlorination with Carbon Monoxide and Chlorine*, Light Met., N.Y., 1, 219-232, **1977**.
20. Adelman, D. J., Burnet, G., *Carbochlorination of Metal Oxides with Phosgene*, AIChE J., 33, 64-69, **1987**.
21. Dobbins, M. S., Burnet, G., *Carbochlorination of Dispersed Oxides in a Molten Salt Reactor*, AIChE J., 34, 1086-1093, **1988**.
22. Goble, A.G., Lawrence, P.A., *Proceedings of the 3rd International Congress on Catalysis*, North Holland Publ. Co., Amsterdam, **1964**.
23. Cornet, D., Goupil, J. M., Szabo, G., Poirier, J.L., Clet, G., *Alkylation of Isobutane by Ethylene Catalyzed by Chlorided Alumina: Influence of Experimental Conditions*, Applied Catalysis, 1-41, 193-205, **1996**.
24. Gates, B., *Catalytic Chemistry*, John Wiley and Sons, New York, **1992**.
25. Szabo, I., Blicke, T., Ujhidy, A., Jelinko, R., *Kinetics of Aluminum Oxide Chlorination. 1. The Mechanism and Mathematical Model*, Ind Eng Chem Res., 30, 292-298, **1991**.
26. Szabo, I., Blicke, T., Ujhidy, A., Jelinko, R., *Kinetics of Aluminum Oxide Chlorination. 2. Experiments and Evaluation of Results*, Ind Eng Chem Res., 30, 298-303, **1991**.
27. Agnew, J.B., Shankar, H. S., *Catalyst Deactivation in Acetylene Hydrochlorination*, Ind Eng Chem Prod. Res. Dev., 25, 19-22, **1986**.
28. Lynch, D., Emig, G., *On the Separability of Catalyst Activity and Kinetic Behavior*, Chem Eng Sci., 6, 1275-1280, **1989**.
29. Agorrieta, Peña, et al., *A Kinetic Model for Activation-Deactivation Processes in Solid Catalysts*, Ind Eng Chem Res., 30, 111-122, **1991**.
30. Laine, J., Brito, J., et al., *The role of water in the Initial Transformations of Hydrodesulfuration Catalysts*, J. Catal., 91, 64-68, **1985**.
31. Instituto Mexicano del Petroleo, *Manual de operación catalizadores IMP-RNA*, segunda edición, **1992**.
32. Perlmutter, D., *Stability of Chemical Reactors*, Prentice-Hall, Inc, New Jersey, **1973**.

DESCRIPCIÓN Y APLICACIÓN DEL MODELO DEL REACTOR TUBULAR DE ADSORCIÓN/DESORCIÓN DE CLORO

En este capítulo se resume el modelo del reactor tubular de lecho empacado con gamma-álumina sobre la que se realizan las reacciones de adsorción y desorción de cloro en forma isotérmica según las siguientes reacciones:



donde X es el sitio activo y empleando COCl_2 como compuesto clorante. Al inicio de este capítulo se escribe explícitamente el sistema de ecuaciones diferenciales parciales a resolver y se resumen las suposiciones que se asumieron cuando se planteó el modelo. Posteriormente se describen brevemente los modelos que se usaron para probar la efectividad de los métodos numéricos y se discuten sus propiedades, aplicaciones y exactitud. Finalmente se propone un juego de parámetros de referencia (para los resultados completos con estos valores ver la sección 4.2) y se plantea un análisis de variación partiendo de estos parámetros.

3.1 Ecuaciones a resolver:

Las ecuaciones se obtuvieron de acuerdo a las siguientes suposiciones:

- I. El medio dentro del reactor es *isotrópico y continuo*, es decir, sus propiedades son constantes en cualquier punto del sistema y en cualquier momento (modelo pseudohomogéneo).
- II. El reactor opera de manera isotérmica en toda su longitud.
- III. Hay ausencia de gradientes radiales de calor, masa y momentum.
- IV. El perfil de velocidades es plano, $e \cdot d \cdot u = u(r)$.
- V. La velocidad, densidad y difusividad de la corriente gaseosa no son función de la coordenada axial.
- VI. El paso limitante es el de reacción, las velocidades de adsorción y desorción se consideran en equilibrio. Los pasos de transporte de los reactivos desde el fluido a la interfase y de productos desde la superficie a la interfase al fluido se consideran rápidos (ver [1]).
- VII. No hay reacción entre las especies de la fase fluida.

Si definimos las variables adimensionales

$$\Phi_k = \frac{C_k}{\rho_{ref}}$$

$$s = \frac{z}{L}$$

$$\tau = \frac{t}{(L/u)}$$

Ecuaciones 3. 1, 3, 2 y 3. 3

$$Pe_{M,k} = \frac{u \cdot dp}{(D_{e,k})_L}$$

$$Pe_{H,k} = \frac{u \cdot dp}{(k_e)_L}$$

$$\Theta = \frac{T}{T_0}$$

Ecuaciones 3. 4, 3. 5 y 3. 6

entonces las expresiones cinéticas en forma adimensional son :

$$r_d = \rho_{s,ref} \rho_{f,ref} (k_1 \Phi_0 \Phi_4 - k_2 \Phi_1 \Phi_5)$$

Ecuación 3. 7

$$r_a = k_3 \rho_{s,ref}^2 \rho_{f,ref} \Phi_2 \Phi_3^2$$

Ecuación 3. 8

los balances de materia quedan

$$\frac{1}{Pe_{M,u_2,0}} \frac{d_p}{L} \frac{\partial^2}{\partial s^2} \Phi_{u_2,0} - \frac{\partial}{\partial s} \Phi_{u_2,0} - \frac{L}{u \rho_{ref}} (r_d - r_a) = \frac{\partial}{\partial \tau} \Phi_{u_2,0}$$

Ecuación 3. 9

$$\frac{1}{Pe_{M,uCl}} \frac{d_p}{L} \frac{\partial^2}{\partial s^2} \Phi_{uCl} - \frac{\partial}{\partial s} \Phi_{uCl} + \frac{L}{u \rho_{ref}} r_d = \frac{\partial}{\partial \tau} \Phi_{uCl}$$

Ecuación 3. 10

$$\frac{1}{Pe_{M,COCl_2}} \frac{d_p}{L} \frac{\partial^2}{\partial s^2} \Phi_{COCl_2} - \frac{\partial}{\partial s} \Phi_{COCl_2} - \frac{L}{u \rho_{ref}} r_a = \frac{\partial}{\partial \tau} \Phi_{COCl_2}$$

Ecuación 3. 11

$$\frac{1}{Pe_{M,CO_2}} \frac{d_p}{L} \frac{\partial^2}{\partial s^2} \Phi_{CO_2} - \frac{\partial}{\partial s} \Phi_{CO_2} + \frac{L}{u \rho_{ref}} r_a = \frac{\partial}{\partial \tau} \Phi_{CO_2}$$

Ecuación 3. 12

$$\frac{L_s}{u\rho_{ref}}(r_o - r_d) = \frac{\partial}{\partial \tau} \Phi_{\bar{c}} \quad \text{Ecuación 3. 13}$$

Ecuación 3. 13

$$\frac{L_s}{u\rho_{ref}}(r_d - r_o) = \frac{\partial}{\partial \tau} \Phi_{\bar{OH}} \quad \text{Ecuación 3. 14}$$

Ecuación 3. 14

ya el balance de energía es (aquí se plantea pero no se resuelve en el modelo; t va desde 1 hasta el número de ecuaciones a considerar)

$$\frac{1}{V\rho_{H_2O}} \frac{d_p}{L} \frac{\partial^2}{\partial \delta^2} \Theta - \frac{\partial}{\partial \delta} \Theta + \frac{L_s}{T_0 u \rho C_p} \left(r_o (-\Delta H_{r_o}) - \sum_j r_{d,j} (-\Delta H_{d,j}) \right) = \frac{\partial}{\partial \tau} \Theta \quad \text{Ecuación 3. 15}$$

Ecuación 3. 15

ya escrito en términos de variables adimensionales. Al inicio de la simulación hay una monocapa de cloro sobre la alúmina y no hay agua, fósgeno, ácido clorhídrico ni dióxido de carbono dentro del reactor. Esto se expresa en las condiciones iniciales.

Al tiempo cero

$$\Phi_{H_2O}|_{s=0} = 1 \quad \Phi_{H_2O}|_{s=L} = 0$$

Ecuaciones 3. 16 y 3. 17

$$\Phi_{HCl}|_{s=0} = 0 \quad \Phi_{HCl}|_{s=L} = 0$$

Ecuaciones 3. 18 y 3. 19

$$\Phi_{COCl_2}|_{s=0} = 1 \quad \Phi_{COCl_2}|_{s=L} = 0$$

Ecuaciones 3. 20 y 3. 21

$$\Phi_{CO_2}|_{s=0} = 0 \quad \Phi_{CO_2}|_{s=L} = 0$$

Ecuaciones 3. 22 y 3. 23

$$\Phi_{\bar{Cl}}|_{s=0} = 1 \quad \Phi_{\bar{Cl}}|_{s=L} = 1$$

Ecuaciones 3. 24 y 3. 25

$$\Phi_{OH}|_{s=0} = 0 \quad \Phi_{OH}|_{s=L} = 0$$

Ecuaciones 3. 26 y 3. 27

De acuerdo a Danckwerts (secc. 2.2.3) las condiciones de frontera del problema son:

A todo tiempo:

$$\Phi_{H_2O}|_{s=0} = 1 \quad \frac{d\Phi_{H_2O}}{ds} \Big|_{s=1} = 0$$

Ecuaciones 3. 28 y 3. 29

$$\Phi_{HCl}|_{s=0} = 0 \quad \frac{d\Phi_{HCl}}{ds} \Big|_{s=1} = 0$$

Ecuaciones 3. 30 y 3. 31

$$\Phi_{COCl_2}|_{s=0} = 1 \quad \frac{d\Phi_{COCl_2}}{ds} \Big|_{s=1} = 0$$

Ecuaciones 3. 32 y 3. 33

$$\Phi_{CO_2}|_{s=0} = 0 \quad \frac{d\Phi_{CO_2}}{ds} \Big|_{s=1} = 0$$

Ecuaciones 3. 34 y 3. 35

$$\frac{d\Phi_{Cl}}{ds} \Big|_{s=0} = 0 \quad \frac{d\Phi_{Cl}}{ds} \Big|_{s=1} = 0$$

Ecuaciones 3. 36 y 3. 37

$$\frac{d\Phi_{OH}}{ds} \Big|_{s=0} = 0 \quad \frac{d\Phi_{OH}}{ds} \Big|_{s=1} = 0$$

Ecuaciones 3. 38 y 3. 39

En las tablas de resultados se llama a cada componente por un número, comenzando por 0 para el agua, 1 para HCl, 2 para COCl₂, 3 para CO₂, 4 para Cl y 5 para OH.

3.2 Modelos de prueba:

Para comprobar la validez de los métodos numéricos usados se probó la solución de dos problemas diferentes

- La difusión y convección de vapor de agua en gas inerte a través del reactor. No se consideraron términos de reacción. Este problema tiene el mismo tipo de condiciones de frontera que el sistema descrito en la sección 3.1 con las ecuaciones 3.16 y 3.17 como condiciones iniciales y 3.28 y 3.29 como condiciones de frontera

$$\frac{1}{Pe} \frac{\partial}{\partial x} \frac{\partial \Phi_{H_2O}}{\partial x^2} - \frac{\partial}{\partial x} \Phi_{H_2O} = \frac{\partial}{\partial \tau} \Phi_{H_2O}$$

Ecuación 3.40

- Un sistema de ecuaciones diferenciales parciales que tiene la misma forma que el sistema de la sección 3.1 y que tiene solución analítica [2]. Las condiciones de frontera de este sistema son del tipo de Dirichlet (vease apéndice A). El sistema es

$$\frac{1}{Pe} \frac{\partial^2}{\partial x^2} \Phi_0 - \frac{\partial}{\partial x} \Phi_0 + 2 \Phi_1^{0.5} - \left(\frac{1}{Pe} + 1 \right) \Phi_0 = \frac{\partial}{\partial \tau} \Phi_0$$

Ecuación 3.41

$$\frac{1}{Pe} \frac{\partial^2}{\partial x^2} \Phi_1 - \frac{\partial}{\partial x} \Phi_1 + 2 \frac{\Phi_1^{1.5}}{\Phi_0} - \left(\frac{4}{Pe} + 2 \right) \Phi_1 = \frac{\partial}{\partial \tau} \Phi_1$$

Ecuación 3.42

$$\frac{1}{Pe} \frac{\partial^2}{\partial x^2} \Phi_2 - \frac{\partial}{\partial x} \Phi_2 + 3 \frac{\Phi_1^2}{\Phi_0} - \left(\frac{9}{Pe} + 3 \right) \Phi_2 = \frac{\partial}{\partial \tau} \Phi_2$$

Ecuación 3.43

cuya solución analítica es

$$\Phi_0(\tau, s) = (1 + \tau^2) e^{-s}$$

Ecuación 3.44

$$\Phi_1(\tau, s) = (1 + \tau^2) e^{-2s}$$

Ecuación 3.45

$$\Phi_0(\tau, s) = (1 + \tau^2)e^{-s}$$

Ecuación 3. 46

Esta prueba produce, con un tamaño de paso de $1 \cdot 10^{-3}$ para ambas variables independientes, un error respecto a la solución analítica de 0.5 %, sin problemas de convergencia para diferencias finitas. La solución se calculó hasta $\tau = 5$.

3.2.1. Discusión sobre los métodos numéricos

La simulación dinámica de reactores catalíticos es de gran importancia para la industria. Se ha demostrado [3] que la operación en un régimen cíclico puede mejorar el desempeño de los equipos comparada con la operación a régimen permanente. Esto ha provocado un gran desarrollo en el análisis numérico de las ecuaciones diferenciales "rigidas" (stiff). Aunque no existe una definición rigurosa de la "rigidez" de los sistemas de ecuaciones (ver por ejemplo [4]), un sistema puede considerarse rígido cuando el máximo valor propio (eigenvalor) es lo bastante grande para que la estabilidad o el error de la solución numérica sean inadecuados, a menos que se establezca un tamaño de paso exageradamente pequeño, tanto que el cálculo no se justifica [5]. La rigidez es un problema inherente a las ecuaciones. Significa que los valores de las variables para las que se resuelve tienen diferencias de varios órdenes de magnitud y esto provoca una pérdida en la exactitud de los cálculos o inestabilidad severa.

A grandes rasgos, podemos clasificar los métodos de solución en *métodos de diferencias finitas* y *métodos de residuos ponderados*. Ambos métodos tienen desventajas, que a continuación se mencionan. Una descripción general se puede hallar en el apéndice A.

- *Diferencias finitas* - Son los métodos que han sido tradicionalmente usados, pues son relativamente sencillos de implementar. Sin embargo presentan un problema importante: la mayoría sufren de oscilaciones numéricas (sin significado físico). Este problema es especialmente grave si en el sistema de ecuaciones domina la convección [6]. Existe una gran variedad de métodos de diferencias finitas. A su vez pueden clasificarse como *explícitos*, *semimplícitos* e *implícitos*. La definición formal de cada tipo puede hallarse en [5].
- *Residuos ponderados* - Los métodos de residuos ponderados más utilizados son, *subdominio*, *Bubnov-Galerkin*, *de los momentos*, *general* y *colocación*. Las desventajas de estos métodos son la mayor dificultad para programarse y la necesidad de resolver sistemas de ecuaciones diferenciales ordinarias y algebraicas mezcladas [7], aunque en principio su eficiencia es mayor.

Referencias:

1. Smith, J. M., *Chemical Engineering Kinetics*, 3rd Ed., Mc Graw-Hill, Singapore, 1981.

- 2 Wu, J C., Fan, T., Erickson, L. E., *Three-Point Backward Finite Difference Method for Solving a System of Mixed Hyperbolic-Parabolic Partial Differential Equations*, Computers & chem Engng, 16, 679-685, 1990
- 3 Kevrekedis, I., *Matrices Are Forever: On Applied Mathematics and Computing in Chemical Engineering*, Chem Eng Sci, 24, 4005-4025, 1995
- 4 Boyce, J., DiPrima, *Elementary Differential Equations*, John Wiley and Sons, 5th ed., 1995
- 5 Chan, Y N., Birnbaum, I., Lapidus, L., *Solution of Stiff Differential Equations and the use of Imbedding Techniques*, Ind Eng Chem Fundam, 17, 133-148, 1978
- 6 Finlayson, B A., *Nonlinear Analysis in Chemical Engineering*, Mc Graw-Hill, 1985.
- 7 Gardini, L., Servida, A., et al *Use of Orthogonal Collocation on Finite Elements with Moving Boundaries for Fixed-Bed Catalytic Reactor Simulation*, Computers & Chem Eng., 9, 1-17, 1985

ANÁLISIS Y DISCUSIÓN DE RESULTADOS

En este capítulo se justifica un juego de parámetros de referencia. Con estos parámetros se calcula la solución asintótica y después se propone un análisis de variación de parámetros en las sección 4.1 clasificándolos en parámetros cinéticos, de transporte y operacionales y se discute su significado físico.

4.1 Parámetros de referencia:

Se proponen los siguientes valores con el fin de establecer una referencia para hacer análisis de variación; estos parámetros son los usados por un reactor tubular de laboratorio. Los programas desarrollados producen directamente tablas de salida en formato de texto que pueden abrirse en cualquier editor.

Tabla 4. 1. Parámetros de referencia para análisis de variación.

Longitud del reactor	60 cm
Flujo total a la alimentación	50 cm ³ /min
Diámetro del reactor	1 cm
Diámetro de partícula	0.1 cm
Concentración de Cl a $t = 0$	2.3473 mmol/cm ³
Concentración de H ₂ O a $L = 0$	0.00035 mmol/cm ³
Concentración de HCl a $L = 0$	0 mmol/cm ³
Concentración de COCl ₂ a $L = 0$	0.00035 mmol/cm ³
Concentración de CO ₂ a $L = 0$	0 mmol/cm ³
Concentración de H ₂ O a $t = 0$ dentro del reactor	0 mmol/cm ³
Concentración de HCl a $t = 0$ dentro del reactor	0 mmol/cm ³
Concentración de COCl ₂ a $t = 0$ dentro del reactor	0 mmol/cm ³
Concentración de CO ₂ a $t = 0$ dentro del reactor	0 mmol/cm ³
Constante 1 (desorción)	0.0161 cm ³ /mmol min
Constante 2 (desorción reversible)	2E-05 cm ³ /mmol min
Constante 3 (adsorción)	0.0032 (cm ³ /mmol) ² /min
Peclet axial de H ₂ O	1.7
Peclet axial de HCl	1.7
Peclet axial de COCl ₂	1.7
Peclet axial de CO ₂	1.7

4.1.1. Justificación.

A continuación se explican las razones por las que se juzgaron aceptables los valores de la tabla anterior

1. *Longitud del reactor, Diámetro del reactor* - Son las características físicas del reactor que se emplea en la Unidad de Investigación en Catalisis de la Facultad de Química de la UNAM. Son tubos de vidrio en forma de "U" con una cuenca para introducir un termopar
2. *Flujo* - El flujo empleado experimentalmente es de aproximadamente 50 mL/min.
3. *Diámetro de partícula* - El catalizador industrial tiene forma de pequeños cilindros de alrededor de 1 mm de diámetro y 5 mm de longitud, pero para los estudios de laboratorio se cortan para obtener cilindros de 1 mm de diámetro y de longitud.
4. *Concentración de cloro a t = 0* - Al inicio de la simulación se considera que la alúmina tiene adsorbida sobre si una monocapa de cloro. Se sabe que la eta-alúmina utilizada contiene 10 % en peso de cloro y que su densidad es de 0.75 g/cm³. Entonces

$$\left(\frac{10\text{gCl}}{90\text{g}\eta\text{Al}_2\text{O}_3} \right) \left(\frac{1000\text{mmol}}{35.5\text{gCl}} \right) \left(\frac{0.75\text{g}\eta\text{Al}_2\text{O}_3}{\text{cm}^3} \right) = 2.3474 \frac{\text{mmolCl}}{\text{cm}^3}$$

5. *Concentración de agua a t = 0* - Es la concentración de agua en la corriente de entrada
6. *Concentración de cloruro de hidrógeno a t = 0* - Se considera que el HCl no es un agente clorador muy efectivo (como puede verse por el valor de la constante 2) y que por esa razón no es necesario introducirlo a la alimentación
7. *Concentración de fósgeno a t = 0* - El fósgeno es un mejor agente clorador que el HCl y por ello se alimenta junto con el agua y el inerte. Siendo su concentración idéntica a la del agua podemos comparar mejor el efecto de las cinéticas de adsorción y desorción en el perfil de las especies adsorbidas
8. *Concentración de dióxido de carbono a t = 0* - El CO₂ es solamente un producto de la adsorción
9. *Concentraciones de agua, cloruro de hidrógeno, fósgeno y dióxido de carbono a t = 0 dentro del reactor* - Como referencia, consideramos que al inicio no hay ninguno de estas especies dentro del reactor
10. *Constante 1* - Para esta reacción es para la que se tienen mejores datos cinéticos. Para calcular su valor se hicieron las siguientes suposiciones:

- es razonable suponer que solo mientras se establece el régimen de flujo la concentración de agua no es uno;
- el término convectivo es tan importante que la concentración de HCl es prácticamente cero dentro del reactor, es decir:

$$r_d = k_1 C_{1D} C_1 - k_2 C_{1D} C_{ClH} \quad \text{Ecuación 4.1}$$

$$r_d = k_1' C_1 \quad \text{Ecuación 4.2}$$

que es la cinética de la que se muestran los resultados en el apéndice B se considera que $k_1 \approx k'_1$ y se aplicó el siguiente tratamiento

- se calcularon los valores de las constantes que se tienen a cada temperatura (120 y 150 °C).
- Con estos puntos se hizo una regresión y se obtuvo una expresión del tipo Arrhenius para k_1 en función de la temperatura:

$$k_1 = 1.0544 \exp(-1768.9516 / T) \text{ [s]} \text{ min}^{-1} \quad \text{Ecuación 4.3}$$

y el cálculo se hace para $T = 423 \text{ K}$.

1. *Constante 2* - Se sabe que la constante de adsorción de cloro con HCl es tres órdenes menor a la de desorción, pero no se dispone de mayor información. El valor de la tabla es arbitrario.
2. *Constante 3* - Tampoco se dispone de información para esta constante. Se supuso arbitrariamente un valor intermedio entre las constantes 1 y 2.
3. *Peclet axial de agua, de cloruro de hidrógeno, de dióxido de carbono* - Se sabe [1, 2] el número de Peclet para gases está entre uno y dos. Mientras mayor sea el número de Reynolds para gases, mayor es la aproximación a 2.

4.1.2 Metodología de solución del problema.

Para resolver el problema, se emplearon dos aproximaciones diferentes:

- I. *Método de líneas y diferencias finitas*. Emplea un método de Runge-Kutta clásico de cuarto orden. En este trabajo lo usaremos para el análisis de los términos relacionados con la difusividad. Estos términos serán los números de Peclet de cada componente de la fase fluida, el diámetro de partícula y la longitud del reactor. Como la difusión es importante sólo al inicio de la simulación, las corridas con este método terminan en tiempos cortos (5 tiempos adimensionales).
- II. *Método de líneas y colocación ortogonal en elemento finito*. Emplea una subrutina en FORTRAN obtenida de *Transactions On Mathematical Software* (TOMS) [3, 4]. Usaremos este método para estudiar los términos relacionados con la reacción y la convección. Estos términos serán el flujo y las concentraciones iniciales de las especies de la fase fluida. Como los efectos de la cinética son los que requieren más tiempo para apreciarse, las corridas se harán a tiempos mayores (1600 tiempos adimensionales).

4.2 Solución asintótica con parámetros de referencia:

Todos los resultados en el presente trabajo se presentan en forma de gráficas. Las abscisas se muestran en **unidades de tiempo adimensional**. Una unidad de tiempo adimensional se define como la longitud del reactor entre la velocidad de flujo en la alimentación, que según la tabla 4.1 equivale a $60 \text{ cm} / 63.67 \text{ cm/min} = 0.94 \text{ minutos}$. Las

ordenadas representan las composiciones del compuesto químico correspondiente de acuerdo a las siguientes ecuaciones:

$$\Phi_{H_2O} = \frac{C_{H_2O}}{\rho_{f,ref}} \quad \Phi_{HCl} = \frac{C_{HCl}}{\rho_{f,ref}} \quad \Phi_{COCl_2} = \frac{C_{COCl_2}}{\rho_{f,ref}}$$

Ecuaciones 4. 4, 4. 5 y 4. 6

$$\Phi_{CO_2} = \frac{C_{CO_2}}{\rho_{s,ref}} \quad \Phi_{Cl} = \frac{C_{Cl}}{\rho_{s,ref}} \quad \Phi_{OH} = \frac{C_{OH}}{\rho_{s,ref}}$$

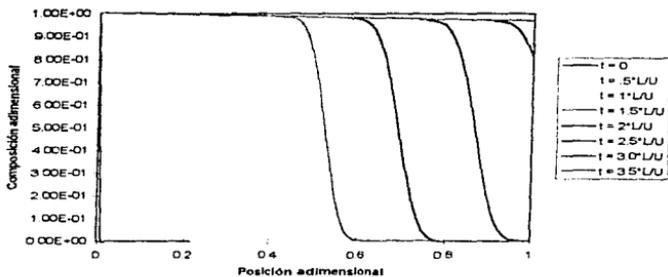
Ecuaciones 4. 7, 4. 8 y 4. 9

donde las cantidades $\rho_{f,ref}$, $\rho_{s,ref}$ son densidades molares de referencia en el fluido y sobre el catalizador sólido, respectivamente

Con los parámetros de la tabla 4.1 se obtiene la variación de los perfiles hasta un tiempo muy avanzado (1600 tiempos adimensionales por método de colocación en elemento finito) Cabe observar que, debido a los términos de reacción no lineales, no es posible obtener una solución analítica para el estado estacionario [1]

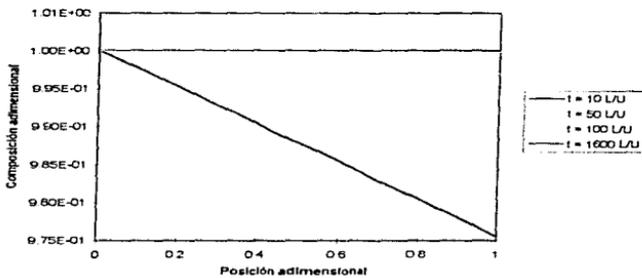
Para cada componente se presenta una gráfica de variación para los primeros tiempos adimensionales (hasta 3.5 L/u) con resultados obtenidos mediante diferencias finitas, y otra para tiempos mayores (hasta 1600 L/u) con resultados calculados con colocación ortogonal en elemento finito. Esta serie de resultados corresponde a las gráficas 4.1 a 4.10

Variación del Perfil de Agua con el Tiempo



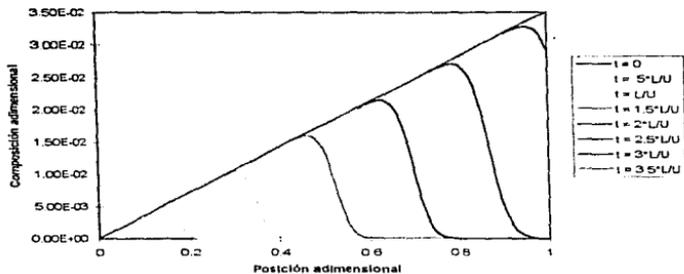
Gráfica 4.1

Variación del Perfil de Agua con el Tiempo



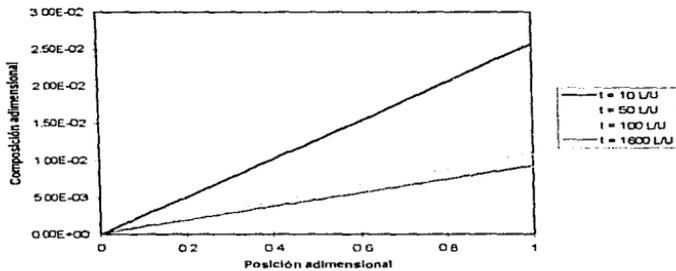
Gráfica 4.2

Perfil de HCl a través del tiempo



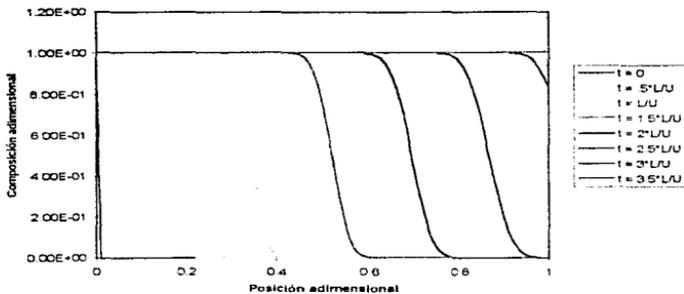
Gráfica 4.3

Variación del Perfil de HCl con el Tiempo



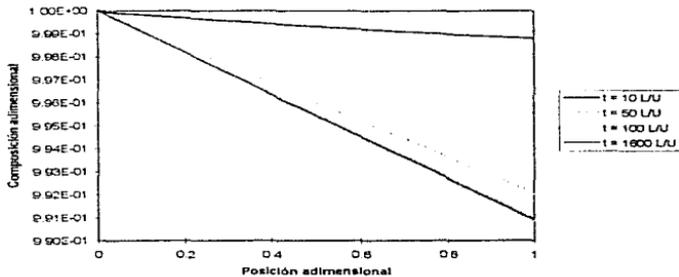
Gráfica 4.4

Variación del Perfil de COC12 a través del tiempo



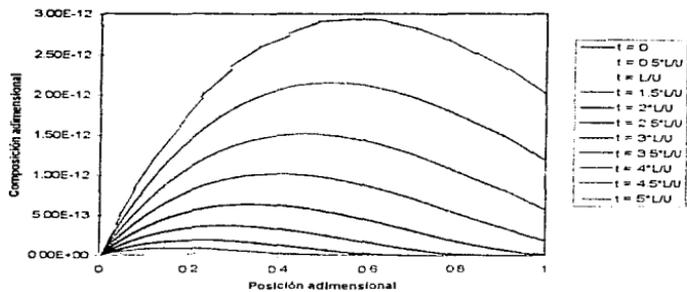
Gráfica 4. 5

Variación del Perfil de COC12 con el Tiempo



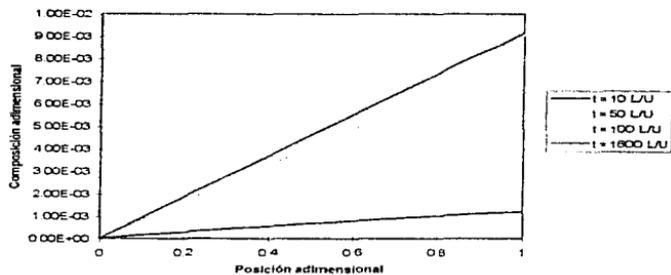
Gráfica 4. 6

Variación del Perfil de CO2 con el Tiempo



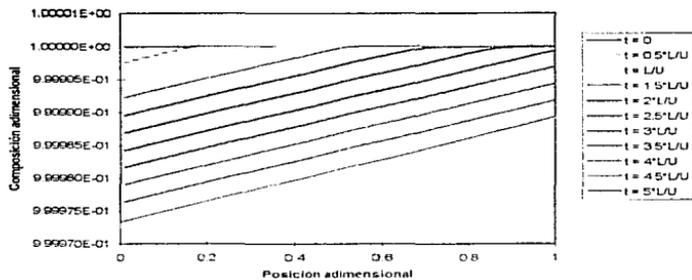
Gráfica 4. 7

Variación del Perfil de CO2 con el Tiempo



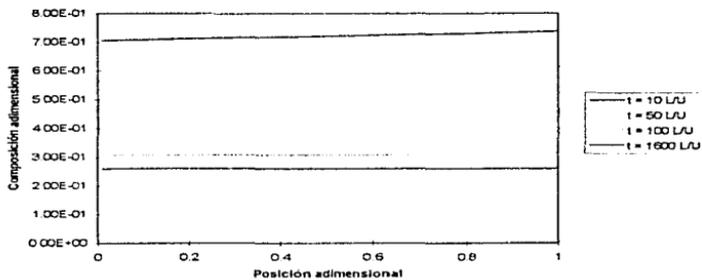
Gráfica 4. 8

Variación del Perfil de CI con el Tiempo



Gráfica 4. 9

Variación del Perfil de CI adsorbido con el Tiempo



Gráfica 4. 10

Los perfiles de reactivos (agua y fósgeno) se establecen casi completamente al cabo de 3.5 tiempos adimensionales (gráficas 4.1 y 4.5). Las gráficas 4.2 y 4.6 representan la variación del perfil de estos compuestos desde 10 hasta 1600 tiempos adimensionales; nótese el cambio de escala en las ordenadas en ambas gráficas. Los perfiles de agua y de fósgeno son lineales. En el caso del agua, la pendiente de esta recta se va incrementando hasta llegar a una concentración constante e igual a uno a mil seiscientos tiempos adimensionales. En el caso del fósgeno sucede lo contrario: la pendiente va disminuyendo. Esto se explica de la siguiente manera: al inicio de la simulación el equilibrio está desplazado hacia la desorción de cloro, lo que favorece un mayor consumo de agua y poco consumo del compuesto clorante; a medida que nos acercamos al equilibrio la velocidad de adsorción de cloro comienza a ser importante y el consumo del compuesto clorante aumenta, mientras que el consumo de agua disminuye.

El valor de la concentración de cloro a 1600 tiempos adimensionales en la gráfica 4.10 nos permite asegurar que, al menos con el juego de parámetros utilizado (tabla 4.1), la cinética de desorción domina sobre la de adsorción. Esto nos ayuda a explicar el comportamiento de las concentraciones de los demás compuestos, por ejemplo, los productos en fase gaseosa (HCl y CO₂). El HCl (producto de desorción) se produce más rápido que el CO₂, como puede verse por la diferencia de ordenes de magnitud entre las ordenadas de las gráficas 4.3 y 4.7. El HCl es arrastrado por la corriente al mismo tiempo que es producido, y a eso se debe la forma lineal del perfil antes de la caída. Las caídas bruscas en los perfiles de la gráfica 4.3 coinciden con el frente de los perfiles de agua y fósgeno en las gráficas 4.1 y 4.5. En el equilibrio los perfiles de HCl y CO₂ coinciden.

En la gráfica 4.9 (en la que se muestran los resultados en los primeros tiempos adimensionales) podemos ver que en las primeras secciones del reactor, que han estado en contacto con el agua por un mayor tiempo, la concentración de cloro es menor que en las secciones finales. Este perfil es lineal y tiene pendiente positiva hasta llegar a la concentración 1.0, y a partir de ahí es constante. Sin embargo, la pendiente de estas rectas es muy pequeña (alrededor de $1.3E-5$) y por eso no se aprecia en la gráfica 4.9.

4.3 Análisis de variación de parámetros:

Todas las constantes y coeficientes de las ecuaciones 3.9 a 3.14 se pueden agrupar en tres grupos: *cinéticas, operacionales* y de *transporte*. Los parámetros elegidos para hacer el análisis de variación son:

- constantes de reacción,
- longitud del reactor,
- flujo de alimentación,
- composición de reactivos,
- diámetro de partícula.

A continuación estudiaremos el efecto de variar estas cantidades.

a) cinéticos

Constantes de reacción.- Se usan los siguientes juegos de constantes (Método de colocación en elemento finito):

	k_1	k_2	k_3
juego 1	$1 \cdot 10^{-3}$	$1 \cdot 10^{-6}$	$3 \cdot 10^{-4}$
juego 2	0.0161	$2 \cdot 10^{-4}$	$3 \cdot 2 \cdot 10^{-1}$
juego 3	0.15	$5 \cdot 10^{-3}$	0.05

Tabla 4. 2

que corresponden a tres cálculos isotérmicos a temperaturas diferentes, donde la temperatura del juego de constantes 1 es una temperatura baja, la temperatura del juego de constantes dos es una temperatura media y el juego 3 corresponde a una temperatura alta. Notar que las constantes incrementan un orden en cada caso.

b) de transporte

Difusividad.- Se obtuvieron resultados con los siguientes valores (Método de Runge-Kutta)

P_{CH_4}	$P_{C_2H_6}$	$P_{C_3H_8}$	P_{CO_2}
1.3	1.3	1.3	1.3
1.5	1.5	1.5	1.5
1.7	1.7	1.7	1.7
2.0	2.0	2.0	2.0

Tabla 4. 3

c) operacionales

Longitud del reactor.- Se cambió la longitud del reactor a 45, 60 y 75 cm (método de Runge-Kutta)

Diámetro de partícula.- Se cambió el diámetro de partícula a 0.05, 0.1 y 0.15 cm (método de Runge-Kutta)

Flujo.- Se usan flujos de 35, 50 y 65 cm^3/min (método de colocación en elemento finito).

Composición.- Se varían las composiciones de fósgeno y de agua según las siguientes tablas (método de colocación en elemento finito)

C_{H_2O} (mM)	C_{COCl_2} (mM)
0.10	0.10
0.10	0.35
0.10	0.60

Tabla 4. 4

C_{H_2O} (mM)	C_{COCl_2} (mM)
0.35	0.10
0.35	0.35
0.35	0.60

Tabla 4. 5

C_{Cl_2O} (mM)	C_{COCl_2} (mM)
0.60	0.10
0.60	0.35
0.60	0.60

Tabla 4.6

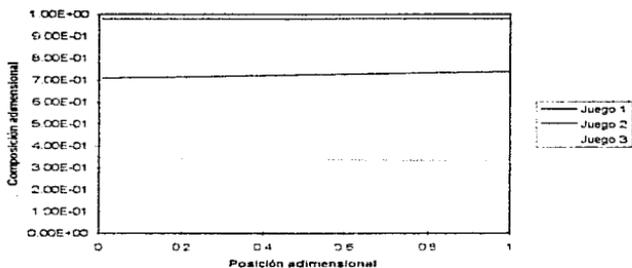
Para cada una de las tablas anteriores se muestra una gráfica en la sección correspondiente del capítulo 4 a 1600 tiempos adimensionales

4.3.1 Constantes de reacción

El efecto de las constantes de reacción en el perfil de cloro adsorbido (especie que confiere acidez a la alúmina) se muestra para los tres juegos de constantes (tres temperaturas) propuestos en la sección 3.4.1 en las gráficas 4.11 a los diez tiempos adimensionales y 4.12 para mil seiscientos tiempos adimensionales.

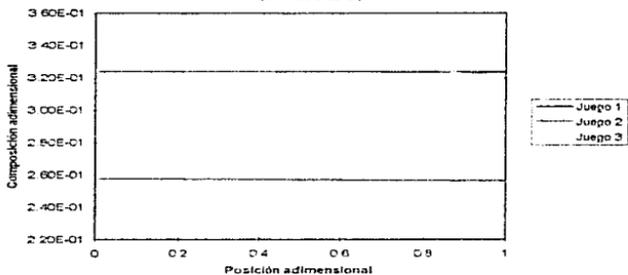
En estos parámetros observamos las siguientes variaciones. En el caso de una temperatura menor (juego 1), la concentración adimensional de cloro adsorbido pasa de ≈ 0.98 ($t = 10$ L/u) a ≈ 0.32 ($t = 1600$ L/u), lo que significa que domina la desorción. En el caso de una temperatura media (juego 2), pasamos de ≈ 0.71 ($t = 10$ L/u) a ≈ 0.26 ($t = 1600$ L/u). Domina nuevamente la desorción, pero la rapidez con que se desorbe el cloro es mayor a temperatura media que a la temperatura menor. Cuando la temperatura es mayor (juego 3) tenemos una desorción más rápida que a temperatura media (≈ 0.35 a $t = 10$ L/u), y a 1600 tiempos adimensionales tenemos casi la misma concentración que a los 10 L/u. Podemos notar en las gráficas 4.11 y 4.12 cierta pendiente en el perfil de Cl, positiva para la temperatura media en 4.11, negativa para la misma temperatura en 4.12 y negativa para la temperatura mayor en ambas gráficas. Esta pendiente es casi cero para la temperatura menor (juego 1) en ambos casos. Esto se explica de la siguiente manera: las secciones iniciales del reactor han estado expuestas a la corriente de reactivos (agua y fósgeno) durante un tiempo un poco mayor que las finales. El avance de la reacción y el consumo de productos han sido un poco mayores en las partes cercanas a la entrada del reactor que a la salida. Cuando la pendiente del perfil de cloro adsorbido es positiva, domina la desorción, cuando es negativa, domina la adsorción. Mientras mayor sea la pendiente, menos uniforme es la monocapa de cloro dentro del reactor.

Efecto de las Constantes de Reacción en el Perfil de CI
($t = 10 \text{ L/U}$)



Gráfica 4. 11

Efecto de las Constantes de Reacción en el Perfil de CI
($t = 1600 \text{ L/U}$)



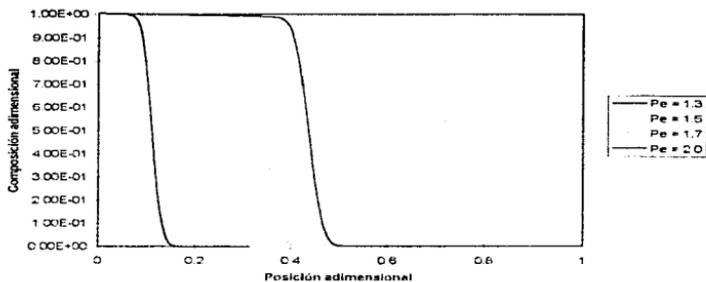
Gráfica 4. 12

4.3.2 Difusividad.

Los resultados en la variación del número de Peclet para los componentes de la fase fluida se presentan en las gráficas 4.13 a 4.20. Recordemos que $Pe_i = (u d_p) / D_{i,c}$. En todos los casos el incremento de Pe_i produce un incremento en la velocidad con que se establece el perfil de la especie i en la fase fluida, mientras que en los perfiles de las especies diferentes de i esta rapidez no cambia. En la variación de parámetros de esta sección este efecto se debe al cambio en $D_{i,c}$, por tanto es de esperar que el efecto de incrementar u sea un aumento parejo en la velocidad con que se establece el perfil de todas las especies en la fase fluida. Las concentraciones de las especies alimentadas (reactivos: agua y fósgeno) se asemejan a funciones escalón, aunque en composiciones muy altas o muy bajas el perfil se suaviza un poco. Esto puede explicarse fácilmente en términos de la difusividad. Como se dijo en el capítulo 2, la difusividad produce un efecto de mezclado de reactivos hacia adelante y de productos hacia atrás. Este mezclado corresponde a los bordes suavizados de las curvas, aunque parecen ser mayores a los que reporta la literatura. Por otro lado, si comparamos las gráficas 4.2 y 4.6 notamos que la concentración máxima de agua es menor a la de fósgeno al mismo tiempo y en la misma posición. Esto se debe a que el valor de la constante de desorción de cloro es mayor a la de adsorción y por eso el agua se consume en mayor cantidad.

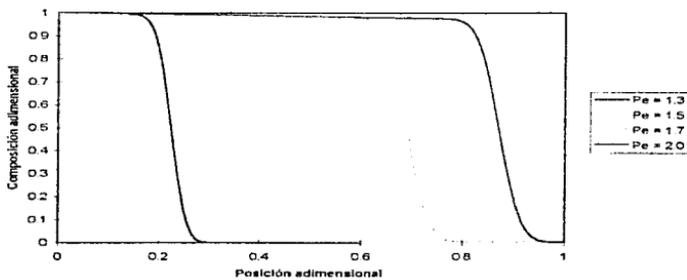
Las curvas de los productos tienen mayores diferencias entre sí: las de HCl comienzan con la misma pendiente para todos los valores de Pe y disminuyen bruscamente en un punto que coincide con la posición del perfil de reactivos. Mientras mayor es Pe_{HCl} , más bruscamente baja el perfil. Esto significa que el HCl es producido al mismo tiempo que es arrastrado rápidamente por la corriente; un Pe_{HCl} pequeño representa una difusividad grande y por tanto un mezclado mayor, que se traduce en una pendiente menos pronunciada (ver el perfil de HCl en la gráfica 4.4 con $Pe_{HCl} = 1.3$). En cambio, como la constante de producción de CO_2 es menor a la de HCl, las curvas son bastante más suaves. Por esta razón y porque la expresión cinética de la adsorción incluye la concentración de OH elevada al cuadrado, la diferencia en órdenes de magnitud entre las concentraciones de HCl y CO_2 es muy grande.

Efecto de Pe en el perfil de agua a $t = L/U$



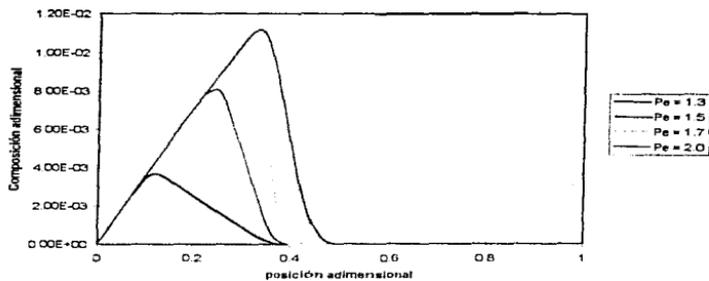
Gráfica 4.13

Perfil de agua a $t = 2^*$ (L/U)



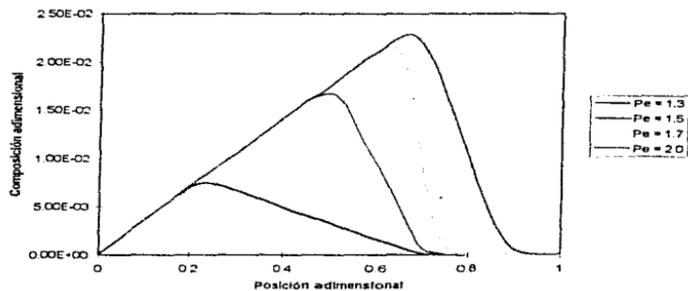
Gráfica 4.14

Efecto de Pe en el perfil de HCl a $t = L/U$



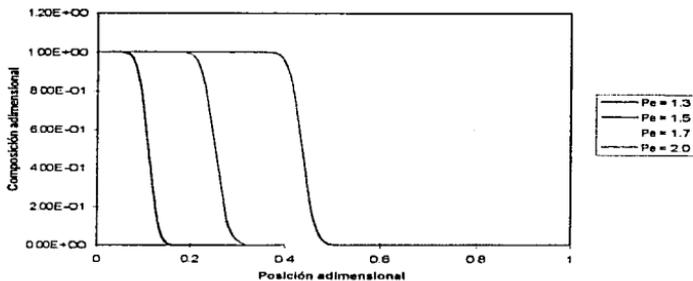
Gráfica 4.15

Efecto de Pe en el Perfil de HCl a $t = 2*(L/U)$



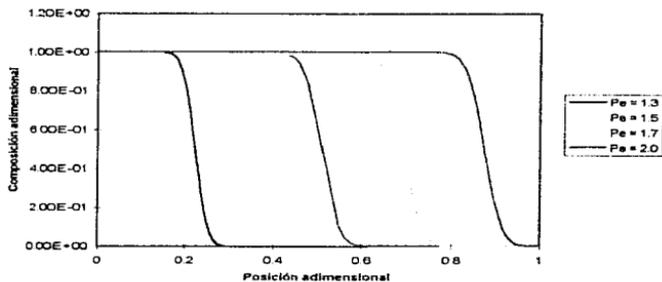
Gráfica 4.16

Efecto de Pe en el perfil de COC12 a $t = L/U$

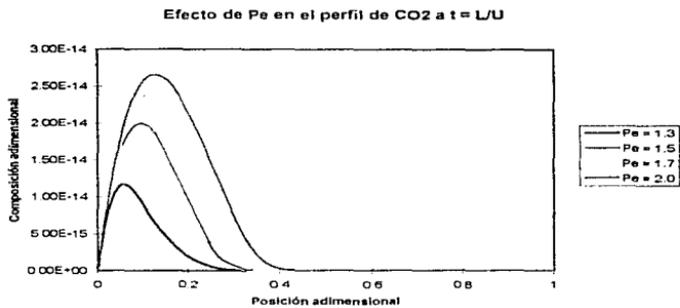


Gráfica 4.17

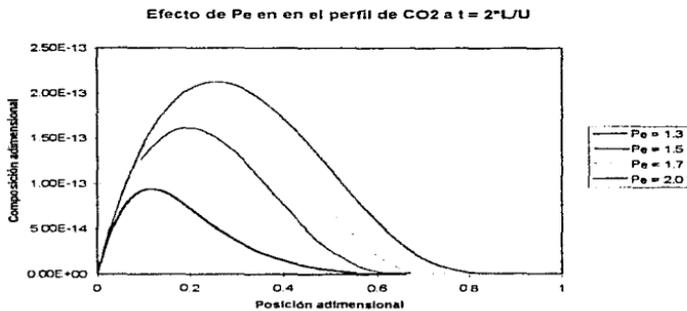
Efecto de Pe en Perfil de COC12 a $t = 2 \cdot L/U$



Gráfica 4.18



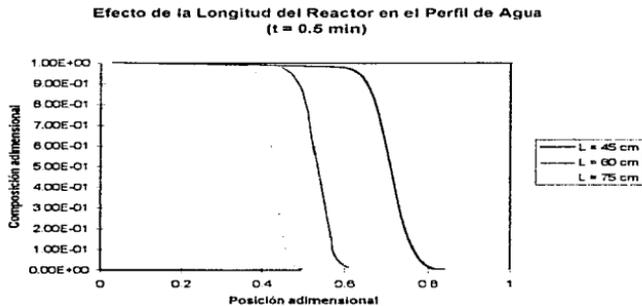
Gráfica 4.19



Gráfica 4.20

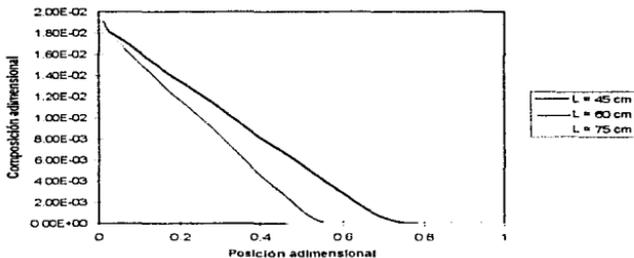
4.3.3 Longitud del reactor:

Se eligió un componente de la fase fluida (agua) y un componente adsorbido en el sólido (hidroxilo) para el análisis del efecto del cambio en la longitud del reactor. Las gráficas 4.21 y 4.22 se presentan a un tiempo dimensional igual a 0.5 minutos. La razón es la siguiente: la unidad adimensional de tiempo depende de la longitud del reactor y de la velocidad de flujo a la entrada, si se altera la longitud del reactor o la velocidad de flujo, no se puede establecer comparación en términos adimensionales. Los resultados a 0.5 minutos son los siguientes:



Gráfica 4.21

Efecto de la Longitud del Reactor sobre el Perfil de OH
($\tau=0.6$ min)



Gráfica 4.22

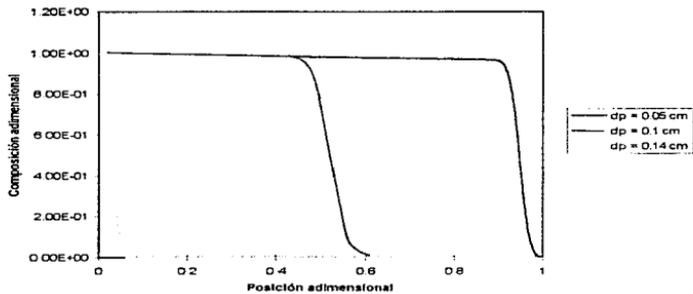
4.3.4. Diámetro de partícula:

Se eligió un componente de la fase fluida (agua) y un componente adsorbido en el sólido (hidroxilo) para el análisis del efecto del cambio en el diámetro de partícula catalítica. De las gráficas 4.11 y 4.12 observamos una sensibilidad muy grande para este parámetro; con cambios muy pequeños en el valor de d_p obtenemos una variación muy grande en las curvas de cada especie. Vale la pena mencionar que el método se vuelve inestable con valores de diámetro de partícula mayores a 0.14 cm o menores a 0.05 cm.

Es interesante notar que el efecto de incrementar este parámetro es el mismo que se esperaría de acuerdo al análisis del número de Peclet. El aumento de d_p produce un aumento en Pe y por consiguiente una disminución del término difusivo (el de la segunda derivada en el espacio, y que favorece al convectivo, el de la primera derivada). Esto puede verse en las gráficas 4.13 a 4.20, donde al aumentar Pe , aumenta la rapidez con que se establece el perfil.

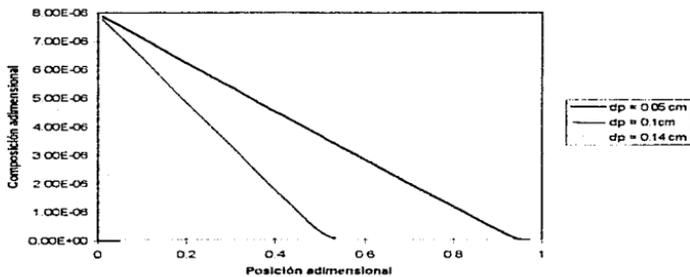
Los resultados son los siguientes:

Efecto del diámetro de partícula en el perfil de agua a $t = 1.5 L/U$



Gráfica 4.23

Efecto de dp en el perfil de OH a $t = 1.5 L/U$



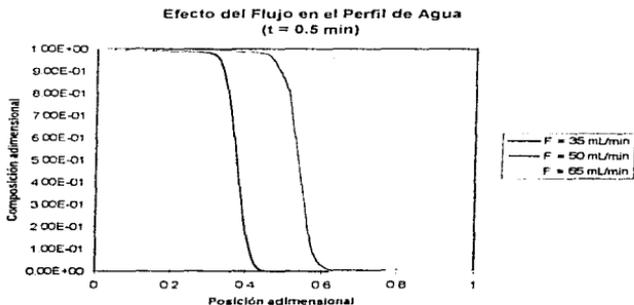
Gráfica 4.24

El aumento en el diámetro de partícula retrasa el establecimiento del perfil de todos los componentes, ya sean fluidos o adsorbidos en el sólido. Para los compuestos en el fluido un diámetro de partícula mayor representa mayor dificultad para moverse a lo largo del reactor. Para las especies adsorbidas esta dificultad en el transporte de masa en la dirección axial de los reactivos se traduce en una pendiente más pronunciada. Existe OH adsorbido solamente hasta la posición a la que haya llegado agua en ese momento. En otras palabras, las rectas de la gráfica 4 12 intersectan las abscisas aproximadamente en el lugar que corresponde al perfil de agua para el mismo diámetro de partícula.

4.3.5 Flujo:

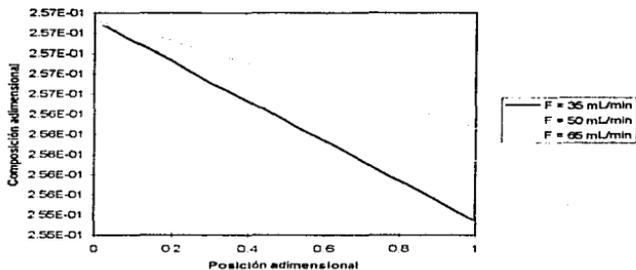
El incremento del flujo produce un aumento en la velocidad con que se arrastran los compuestos de la fase fluida. Como el tiempo de referencia es L/u , nuevamente debemos tomar como referencia un tiempo real. Para el agua, que se encuentra en la fase fluida, este tiempo será 0.5 minutos, para el cloro, 1440 minutos (24 horas).

De la gráfica 4.28 podemos observar que a flujos (velocidades) bajos, el tiempo de contacto de los reactivos con las partículas es mayor, por lo tanto se consume más reactivo a la entrada y llega menos reactivo a la sección final del reactor tubular, por eso la pendiente del perfil de 35 mL/min es mayor a un tiempo dado.



Gráfica 4.25

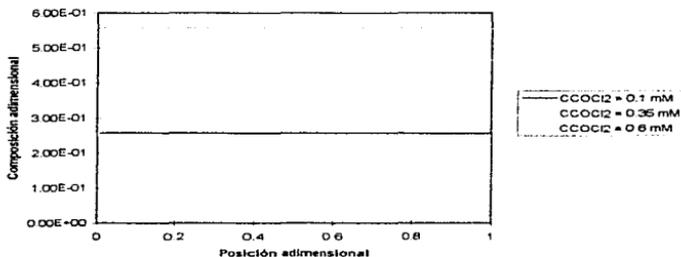
Efecto del Flujo en el Perfil de Cl (t = 1440 min)



Gráfica 4.26

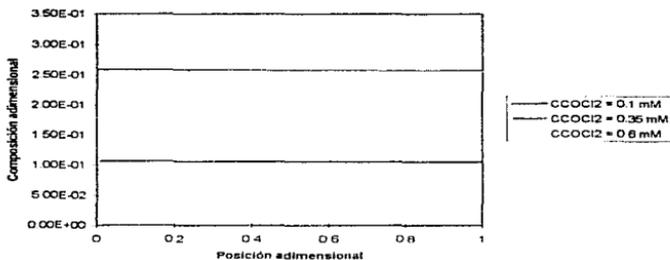
4.3.6 Composición de la corriente de alimentación:

Efecto de la Concentración de COCl₂ en el Perfil de Cl
(T = 1600 L/U, CH₂O = 0.1 mM)



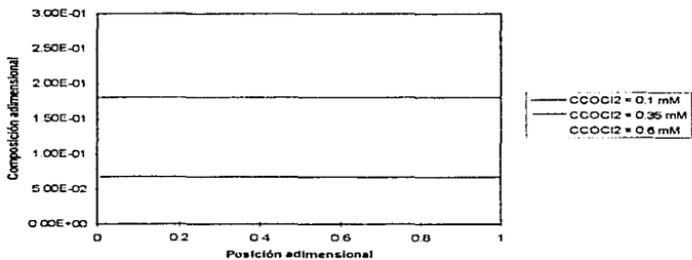
Gráfica 4.27

Efecto de la Concentración de COCl_2 en el Perfil de Cl^-
($t = 1600 \text{ L/U}$, $\text{CH}_2\text{O} = 0.35 \text{ mM}$)



Gráfica 4.28

Efecto de la Concentración de COCl_2 en el Perfil de Cl^-
($t = 1600 \text{ L/U}$, $\text{CH}_2\text{O} = 0.6 \text{ mM}$)



Gráfica 4.29

De las gráficas 4.29 a 4.31 podemos comprobar que la concentración de cloro adsorbido depende de la concentración de agua y del agente clorador. La máxima concentración superficial de cloro la tenemos en la gráfica 4.29 con 0.55 mmol / cm³ cuando hay 0.1 mM de agua y 0.6 mM de fósgeno y la mínima es 0.065 mmol / cm³ cuando hay 0.6 mM de agua y 0.1 mM de fósgeno. De acuerdo a la cinética con que estamos representando el sistema, estos resultados son consistentes. Vemos que la desorción resulta mas importante pues todos los perfiles de cloro excepto uno están por debajo de 0.5 mM por cm³, aun cuando tengamos concentraciones de fósgeno mucho mayores que las de agua.

Para comprobar la validez del modelo, se hace el balance de materia global del reactor en el régimen permanente. En el régimen estacionario las velocidades de las reacciones de adsorción y desorción son iguales, por lo que la cantidad de masa que entra al reactor debe ser la misma que sale. Entonces

$$[M_{H_2O}VC_{H_2O} + M_{CO_2}VC_{CO_2}]_{t=0} = [M_{H_2O}VC_{H_2O} + M_{CO_2}VC_{CO_2} + M_{HCl}VC_{HCl} + M_{CO_2}VC_{CO_2}]_{t=1}$$

V se elimina. Como todos los componentes están en la fase fluida, $C_i = \rho_i \cdot \nu_i / \Phi_i$ y $\rho_i \cdot \nu_i$ también se elimina. Para los componentes de la alimentación, $\Phi_i = 1$. El balance queda, utilizando las concentraciones que se obtienen en el cálculo de 2000 tiempos adimensionales:

$$[M_{H_2O} \Phi_{H_2O} + M_{CO_2} \Phi_{CO_2}]_{t=0} = [M_{H_2O} \Phi_{H_2O} + M_{CO_2} \Phi_{CO_2} + M_{HCl} \Phi_{HCl} + M_{CO_2} \Phi_{CO_2}]_{t=1}$$

$$18 \text{ g} + 99 \text{ g} = 18 (1.0) \text{ g} + 99 (0.99087) \text{ g} + 36.5 (9.1316E-3) \text{ g} + 44 (9.1316E-3) \text{ g}$$

$$117 \text{ g} = 116.83 \text{ g}$$

4.3.7 Discusión sobre el procedimiento de solución:

A lo largo del presente capítulo se ha efectuado la simulación dinámica de la adsorción-desorción de cloro en un reactor de lecho empacado empleando dos aproximaciones: la primera mediante el uso de método de líneas y diferencias finitas para el análisis de variación en tiempos cortos (hasta cinco tiempos adimensionales) y la segunda mediante método de líneas y colocación ortogonal en elemento finito para tiempos mayores (hasta 1600 tiempos adimensionales). De la comparación entre ambos métodos de solución podemos notar lo siguiente:

- El método de colocación es más eficiente que los métodos de diferencias finitas dado que, para la misma tolerancia de 10^{-3} , las diferencias finitas requieren alrededor de 10 minutos para llegar a 5 tiempos adimensionales, mientras que la colocación necesita 1.5 minutos para alcanzar 1600 tiempos adimensionales. Independientemente de que esta rutina está probada y es parte de una biblioteca comercial, la mayor eficiencia del

algoritmo de colocación se debe a la menor cantidad de cálculos por punto y a la rutina de ajuste de tamaño de paso que tiene incluida

- Cuando se resuelve por colocación ortogonal, se emplea más tiempo de cálculo en los primeros tiempos adimensionales, pues es entonces cuando se dan mayores variaciones en las derivadas espaciales de los perfiles. Aquí el método de colocación requiere tamaños de paso muy pequeños. La velocidad del método de diferencias finitas es la misma cada vez, pues su tamaño de paso es constante
- Los métodos de diferencias finitas son mucho más sencillos de programar, sobre todo si son del tipo predictor-corrector.
- Existe una gran cantidad de algoritmos en la literatura, especialmente los publicados durante los últimos 20 años. Muchos de ellos fueron publicados por primera vez en revistas técnicas de Ingeniería Química.

La dureza (stiffness) de una ecuación diferencial puede interpretarse como una diferencia de muchos órdenes de magnitud entre los valores de las variables dependientes. Esta propiedad suele presentarse en problemas que no son isotérmicos. Aunque la dureza de las ecuaciones resueltas en esta tesis es poca, sobre todo cerca del estado estacionario, el algoritmo de colocación empleado y algunos otros publicados pueden enfrentarse a esta dificultad.

Acercas del procedimiento de programación vale la pena notar lo siguiente:

- Existe una gran variedad de rutinas de solución que vienen incluidas en los paquetes de desarrollo comerciales. Dichas rutinas son muy eficientes, están bien probadas y vienen bien documentadas. Aunque todas ellas fueron escritas originalmente en FORTRAN, los compiladores actuales de Microsoft permiten mezclar lenguajes sin mayor dificultad para aprovechar el código existente. En la biblioteca numérica de IMSL empleada hay rutinas para resolver problemas de aritmética, teoría de números, funciones especiales, álgebra lineal, interpolación, solución de ecuaciones no lineales, optimización, ecuaciones diferenciales e integrales, transformadas y probabilidad y estadística.
- El algoritmo MOLCH es el mismo que utiliza la supercomputadora CRAY para resolver sistemas de ecuaciones parciales con una coordenada espacial.

Referencias:

1. Froment, G.F., Bischoff, K.B., *Chemical Reactor Analysis and Design*, 2nd. ed., John Wiley and Sons, 1990
2. Carberry, J., *Chemical and Catalytic Reaction Engineering*, Mc. Graw-Hill, 1976.
3. Collected algorithms from ACM, Transactions on Mathematical Software, 18, 343-344, 1992. Algoritmos 540 y 688, de <http://math.nist.gov/gams/Classes.html>
4. MOLCH, IMSL FORTRAN Library, Visual Numerics Inc., 1993

CONCLUSIONES

De los resultados obtenidos en el capítulo 4 las conclusiones son:

- La solución asintótica se alcanza rápidamente, pues para todos los componentes el perfil a 100 tiempos adimensionales, que corresponden a 106.1 minutos = 1 hora 46 minutos 6 segundos, es casi igual a 1600 tiempos adimensionales (1697.6 minutos = 28 horas 17 minutos 36 segundos).
- Los perfiles de los componentes HCl y CO₂, que son productos de reacción y que no son parte de la alimentación, presentan a régimen estacionario un perfil lineal con pendiente positiva que pasa por el origen. Esto se debe a que estos compuestos se producen al mismo tiempo que son arrastrados por la corriente. En las secciones finales del reactor la concentración de productos es mayor porque se concentra lo que se produce ahí mismo más lo que se ha arrastrado de las secciones iniciales.
- Dado que se ha justificado para la desorción de cloro que la reacción es el paso dominante y que se ha supuesto lo mismo para la adsorción, la forma de las expresiones cinéticas indica que la adsorción será más lenta a bajas concentraciones de S-OH. Como al inicio de la simulación se tiene dentro del reactor un perfil de cloro constante e igual a 1, la adsorción de cloro nunca puede dominar y por ello al régimen estacionario la máxima concentración de cloro que llegamos a tener es 0.34.
- Disminuir la difusividad se traduce en un incremento en la velocidad con que se establece el perfil. Recordemos que la difusión y la convección operan en sentidos opuestos. Esto es evidente cuando notamos que en cada ecuación, el término la segunda derivada en el espacio es positivo y el de la primera derivada es negativo. Por lo tanto, reducir el efecto de la difusión hace más evidente el efecto de la convección. Incrementar el diámetro de partícula o aumentar el flujo produce los mismos efectos que disminuir el coeficiente de difusividad.
- La literatura reporta que el efecto del retromezclado debido a la difusión se manifiesta en distancias de alrededor de un diámetro de partícula. Este efecto no se aprecia claramente en la gráfica 4.1. En esta gráfica se observa que el perfil de agua cambia de casi uno a cero en ≈ 0.2 longitudes adimensionales, lo que representa 12 cm, una distancia mucho mayor a un diámetro de partícula. Lo que sí se nota es que el perfil es menos empinado entre 2 y 3.5 tiempos adimensionales que entre 0 y 1.5 tiempos adimensionales. La mayor curvatura en tiempos de alrededor de 3 L/u se debe a que, cuando el frente de concentración llega a esa distancia, el agua ha tenido más tiempo para difundirse.
- Un diámetro de partícula pequeño favorece un perfil de cloro con poca pendiente dentro del reactor, o sea, una distribución de especies adsorbidas más uniforme.
- Si se incrementa la longitud del reactor se requiere entonces un mayor tiempo real para lograr el régimen permanente. Cuando el perfil de cloro tenga una pendiente muy grande en el estado estacionario como en el juego de constantes 3 de la gráfica 4.12, será necesario un estudio muy detallado si se quiere tener un reactor muy largo, pues

en las partes finales podría haber una concentración de cloro demasiado baja como para que se produzca la isomerización

- Las concentraciones son el parámetro más importante para la operación del reactor. Parece ser que para tener gran cantidad de cloro sobre la alúmina es necesario que la concentración de fósgeno sea mucho mayor a la de agua. Esto concuerda con el modelo cinético propuesto, aunque para hacer una sugerencia de utilidad para un reactor real será necesario simular las reacciones de isomerización de parafinas. Con estas reacciones se puede determinar una concentración óptima de cloro en el interior del reactor para obtener el mejor rendimiento de compuestos ramificados

Finalmente, la mejor combinación para obtener una alta concentración de cloro dentro del reactor es una alta fracción de fósgeno a la entrada, baja concentración de agua, diámetro de partícula pequeño, velocidades bajas o medias de flujo y altas temperaturas (constantes grandes). El valor de la constante de difusividad no puede cambiarse, pues depende de la alúmina y de la sustancia que se difunde, pero se hizo la variación para conocer su efecto a través del tiempo

Si se simulara la isomerización, sería de esperar que los hidrocarburos ramificados, al ser los productos deseados, tuvieran el mismo patrón de comportamiento que el HCl y el CO₂, es decir, se encontrarían en mayor cantidad al final del reactor. Pero su mayor concentración en este punto alteraría el rendimiento en las secciones finales. A partir de este razonamiento la longitud del reactor conveniente para la isomerización no puede determinarse con los resultados obtenidos en este trabajo

Con base en todo lo señalado anteriormente, el presente estudio puede ampliarse considerando las reacciones de isomerización, la desactivación del catalizador durante su tiempo de operación, explotar la característica dinámica del modelo introduciendo pulsos de reactivos y, en una etapa más avanzada, proponer un modelo heterogéneo

NOMENCLATURA

C	Concentración
C_p	Capacidad calorífica
D, D_c	Coefficiente de difusividad
d_p	Diámetro de partícula
f	Fracción de reactivo sin reaccionar
ΔH	Cambio de entalpía
k	Constante de reacción
k_c	Coefficiente de conductividad térmica
L	Longitud total del reactor
p	Presión parcial
R	Radio del reactor
r	Rapidez de reacción
S	Sitio activo; superficie total en partícula catalítica
s	Coordenada axial adimensional
T	Temperatura
t	Tiempo
u	Velocidad de flujo
z	Coordenada axial

Subíndices:

a	Anterior a la sección de reacción; relativo a la adsorción
B	De referencia
b	Posterior a la sección de reacción
d	Relativo a la desorción
k	Del componente k
L	Relativo a la coordenada axial
M	De masa
m	Relativo al consumo de la partícula
H	De calor
o	inicial o de referencia
r	De reacción
ref	De referencia
s	Relativo a la superficie
f	Relativo a la fase fluida

Superíndices:

f	Relativo a la superficie de la partícula catalítica
g	Relativo a la fase gaseosa de bulbo

Grupos adimensionales:

Da	Número de Damkholer = $k_n D_p C_0^{n-1}$, n es el orden de la reacción
Pe _l	Número de Peclet axial = $(u d_p) / D_L$, ó $(u L) / D_L$ = convección axial / dispersión axial
Pe _r	Número de Peclet radial = $(u d_p) / D_r$, ó $(u L) / D_r$ = convección axial / dispersión radial
Nu	Número de Nusselt = $(h d_p) / k_p$ = gradiente de temperatura adimensional / superficie de transmisión de calor, k_p es la conductividad térmica en la partícula
Sh	Número de Sherwood = $(\beta d_p) / D_L$ = incremento de temperatura / dispersión axial
ϕ	Módulo de Thiele = $L (k C_0^{n-1} / D_p)^{1/2}$ = rapidez de reacción en la superficie / rapidez de la difusión interfacial

Letras griegas:

α	Fracción de superficie envenenada
β	Relativo a la película gaseosa que rodea la partícula catalítica; incremento adimensional de temperatura adiabática
η	Factor de efectividad de la partícula catalítica
Θ	Temperatura adimensional
Φ	Composición adimensional
φ	Fracción de la superficie cubierta
ρ	Densidad
τ	Tiempo adimensional

CONCEPTOS SOBRE ECUACIONES DIFERENCIALES PARCIALES

A.1 Clasificación:

Las ecuaciones diferenciales parciales (EDP) se clasifican de acuerdo a su orden, linealidad y condiciones de frontera.

El orden de una ecuación diferencial parcial se determina por la derivada parcial de orden mayor presente en la ecuación. A continuación se dan ejemplos de EDP de primero, segundo y tercer orden.

Primer orden

$$\frac{\partial u}{\partial x} - \alpha \frac{\partial u}{\partial y} = 0 \quad \text{Ecuación A1.1}$$

Segundo orden

$$\frac{\partial^2 u}{\partial x^2} + u \frac{\partial u}{\partial y} = 0 \quad \text{Ecuación A1.2}$$

Tercer orden:

$$\left(\frac{\partial^3 u}{\partial x^3} \right)^2 + \frac{\partial^2 u}{\partial x \partial y} + \frac{\partial u}{\partial y} = 0 \quad \text{Ecuación A1.3}$$

Por su linealidad se clasifican como lineales, cuasilineales y no lineales. Para la siguiente ecuación de segundo orden (ec. A1.4)

$$a(\bullet) \frac{\partial^2 u}{\partial y^2} + 2b(\bullet) \frac{\partial^2 u}{\partial x \partial y} + c(\bullet) \frac{\partial^2 u}{\partial x^2} + d(\bullet) = 0$$

Ecuación A1.4

donde (\bullet) denota la funcionalidad del coeficiente. Si los coeficientes son constantes, o función de variables independientes solamente $[(\bullet) = (x,y)]$, entonces la ecuación es lineal. Si los coeficientes son funciones de la variable dependiente y/o cualquiera de sus derivadas de orden menor al de la ecuación diferencial entonces ésta es cuasilineal. En cambio, si los coeficientes fuesen función de derivadas del mismo orden que el de la ecuación, entonces la EDP es no lineal. Según este criterio la ecuación A1.1 es lineal, la A1.2 es cuasilineal y la A1.3 es no lineal.

Las EDP de segundo orden con dos variables independientes se clasifican a su vez en tres formas canónicas: elípticas, parabólicas e hiperbólicas. La forma general de esta clase de ecuaciones es

$$a \frac{\partial^2 u}{\partial x^2} + 2b \frac{\partial^2 u}{\partial x \partial y} + c \frac{\partial^2 u}{\partial y^2} + d \frac{\partial u}{\partial x} + e \frac{\partial u}{\partial y} + fu + g = 0$$

Ecuación A1.5

donde los coeficientes son constantes o función de las variables independientes solamente.

Se determina si una ecuación es de alguna de las formas canónicas según el siguiente criterio

$$\begin{aligned} b^2 - ac < 0 &\rightarrow \text{elíptica} \\ b^2 - ac = 0 &\rightarrow \text{parabólica} \\ b^2 - ac > 0 &\rightarrow \text{hiperbólica} \end{aligned}$$

y si $g = 0$, entonces la ecuación A1.5 es *homogénea*

Ejemplos clásicos de ecuaciones diferenciales parciales que se ajustan a las formas canónicas son:

Ecuación de Laplace (elíptica):

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0 \quad \text{Ecuación A1.6}$$

Ecuación de difusión (parabólica):

$$\alpha \frac{\partial^2 u}{\partial x^2} = \frac{\partial u}{\partial t} \quad \text{Ecuación A1.7}$$

Ecuación de onda (hiperbólica)

$$a^2 \frac{\partial^2 u}{\partial x^2} = \frac{\partial^2 u}{\partial t^2} \quad \text{Ecuación A1.8}$$

Una clasificación similar existe para ecuaciones diferenciales parciales de segundo orden con tres variables independientes. Esta clasificación incluye formas canónicas elíptica, parabólica, hiperbólica y ultrahiperbólica. La mayoría de las EDP que se presentan en ciencia e ingeniería son de segundo orden con hasta cuatro variables independientes. Casi todas tienen formas canónicas.

El método de solución de una ecuación dada depende de su forma canónica, pero como los coeficientes de la ecuación general de segundo grado (A1.5) pueden variar en función de las variables dependientes, una ecuación dada puede cambiar su forma canónica a lo largo del proceso de integración.

A.2 Condiciones de frontera:

El significado físico de las condiciones de frontera puede interpretarse como las interacciones del sistema que estudiamos con el resto del universo. Son además una necesidad matemática para la resolución del modelo, aunque los alrededores no sean descritos por éste.

Existen tres tipos de condiciones de frontera.

- **Condiciones de Dirichlet (primer tipo)** - Los valores de la variable dependiente se dan como valores fijos o en función de la variable independiente.

$$\begin{array}{ll} u = u(x) \text{ ó } u = u_0 & \text{a } x = 0 \text{ y } 0 \leq x \leq 1 \\ u = u(t) & \text{a } x = 0 \text{ y } t > 0 \\ u = u_1 & \text{a } x = 1 \text{ y } t > 0 \end{array}$$

- **Condiciones de Neumann (segundo tipo)** - La derivada de la variable dependiente se da como constante o como función de la variable independiente.

$$\text{a } x = 1 \text{ y } t > 0:$$

$$\frac{\partial T}{\partial x} = 0$$

Se dice que un problema tiene **condiciones de Cauchy** si combina condiciones de Neumann y de Dirichlet.

- **Condiciones de Robbins (tercer tipo)** - La derivada de la variable dependiente se da como función de la variable dependiente.

$$\text{a } x = 0 \text{ y } t > 0:$$

$$k \frac{\partial T}{\partial x} = h(T - T_0)$$

Ecuación A1.9

Los problemas pueden también clasificarse como *de valor inicial* o *de valor en la frontera*. En los problemas de valor inicial, al menos para una de las variables independientes, no se tiene un valor especificado en, por ejemplo, cuando t tiende a infinito (ejemplo: conducción del calor). Cuando la región es cerrada para todas las variables

independientes y se tienen condiciones en todas las fronteras, el problema es de valor en la frontera

A.3 Métodos de residuos ponderados:

Sea el problema de valor inicial

$$\begin{aligned} U_t &= L U & x \in V & \quad t > 0 \\ U(x, 0) &= U_0(x) & x \in V \\ U(x, t) &= f(x, t) & x \in S \end{aligned}$$

para $U = U(x, t)$, donde L denota un operador diferencial en las derivadas espaciales de U , x es un vector de variables espaciales y V es el dominio espacial con frontera S .

Para explicar de manera general los métodos de residuos ponderados propongamos una solución U_T de la forma.

$$U_T(x, t) = U_0(x, t) + \sum_{i=1}^N C_i(t) U_i(x)$$

Ecuación A1. 10

donde las $U_i(x)$ son funciones base conocidas (funciones trigonométricas, de Legendre, etc.) elegidas de un conjunto completo de funciones base (tal vez ortogonales) que satisfacen

$$U_i = f_i \quad U_i = 0 \quad x \in S$$

Para resolver se define el residuo de la ecuación

$$R_T(U_T) = L U_T - (U_T)_t$$

Ecuación A1. 11

y el residuo inicial

$$R_i(U_T) = U_0(x) - U_0(x, 0) - \sum_{i=1}^N C_i(0) U_i(x)$$

Ecuación A1. 12

Estas cantidades son medidas de qué tan bien la función de prueba satisface la ecuación y las condiciones iniciales, respectivamente. Si la función de prueba es la solución exacta, ambos residuos son cero. En los métodos de residuos ponderados las constantes $C_j(t)$ se escogen de tal forma que los residuos se vuelven cero mediante el siguiente procedimiento.

Elegimos N funciones de ponderación w_j , $j = 1, 2, \dots, N$, y definimos al producto interior o integral ponderada.

$$(w, v) = \int_I (wv) dx$$

Ecuación A1. 13

luego igualamos a cero las N integrales ponderadas de los residuos de las ecuaciones:

$$(w_n, R_1(U_1)) = 0$$

Ecuación A1. 14

expresión que representa un sistema de N ecuaciones algebraicas a resolver para C_n , siempre que C_j sea constante. Si, en cambio, $C_j = C_j(x)$, se tiene que resolver un sistema de ecuaciones diferenciales ordinarias. Lo mismo hay que hacer con los residuos iniciales para encontrar $C_j(0)$

Cuando se emplea **colocación ortogonal** las funciones de prueba U_1 son conjuntos de polinomios ortogonales que satisfacen las condiciones de frontera y las raíces de los polinomios son los puntos de colocación. Sea

$$P_n(x) = \sum_{j=0}^n C_j x^j$$

Ecuación A1. 15

un polinomio de orden n cuyos coeficientes C_j se determinan con el requerimiento de que P_n sea ortogonal en $a \leq x \leq b$ a todos los polinomios de orden menor a n, relativo a la función de ponderación $w(x) \geq 0$

$$\int_a^b (w(x) P_n(x) P_m(x)) dx = 0$$

Ecuación A1. 16

Los polinomios mas empleados son los de *Legendre*, *Chebyshev*, *Laguerre* y *Hermite* [2, 3]

A.4 Algoritmo MOLCH de IMSL, empleado en la solución:

La rutina MOLCH usa el método de líneas para resolver el sistema de ecuaciones diferenciales parciales.

$$\frac{\partial u_k}{\partial t} = f_k \left(x, t, u_1, \dots, u_M, \frac{\partial u_1}{\partial x}, \dots, \frac{\partial u_M}{\partial x}, \frac{\partial^2 u_1}{\partial x^2}, \dots, \frac{\partial^2 u_M}{\partial x^2} \right)$$

Ecuación A1. 17

con condiciones iniciales

$$u_k = u_k(x, t) \quad \text{a } t = t_0$$

y condiciones de frontera:

$$\alpha_k u_k + \beta_k \frac{\partial u_k}{\partial x} = \gamma_k \quad \text{a } x = x_1 \quad \forall \text{ a } x = x_M$$

para $k = 1, \dots, M$.

Se usan polinomios cúbicos de Hermite para aproximar la variable x de tal forma que la solución de prueba se expande en la serie

$$u_k(x, t) = \sum_{i=1}^{M'} (a_{i,k}(t) \phi_i(x) + b_{i,k}(t) \psi_i(x))$$

Ecuación A1. 18

donde $\phi_i(x)$ y $\psi_i(x)$ son las funciones de base comunes para los polinomios cúbicos de Hermite con los puntos $x_1 < x_2 < \dots < x_N$. Los polinomios cúbicos son los elementos finitos con primeras derivadas continuas. En los puntos de unión satisfacen

$$\begin{aligned} \phi_i(x_j) &= \delta_{ij} & \psi_i(x_j) &= 0 \\ \frac{d\phi_i}{dx}(x_j) &= 0 & \frac{d\psi_i}{dx}(x_j) &= \delta_{ij} \end{aligned}$$

Ecuaciones A1. 19, A1. 20, A1. 21, A1. 22

De acuerdo al método de colocación, los coeficientes de la aproximación se obtienen de tal forma que la solución de prueba satisfice la ecuación diferencial en los dos puntos gaussianos de cada subintervalo.

$$\begin{aligned} p_{2j-1} &= x_j + \frac{3-\sqrt{3}}{6} (x_{j+1} - x_j) \\ p_{2j} &= x_j + \frac{3+\sqrt{3}}{6} (x_{j+1} - x_j) \end{aligned}$$

Ecuaciones A1. 23, A1. 24

para $j = 1, \dots, N$. La aproximación por colocación a la ecuación diferencial es:

$$\frac{d\hat{u}_{i,k}}{dt} \phi_i(p_j) + \frac{d\hat{u}_{i,k}}{dt} \psi_i(p_j) = f_k(p_j, t, \hat{u}_1(p_j), \dots, \hat{u}_M(p_j), \dots, (\hat{u}_M)_\omega(p_j), \dots, (\hat{u}_M)_\omega(p_j))$$

Ecuación A1. 25

para $k = 1, \dots, M$ y $j = 1, \dots, 2(N-1)$

Este es un sistema de $2M(N-1)$ ecuaciones diferenciales ordinarias con $2M$ N unknown funciones-coeficiente desconocidas, $u_{i,k}$ y $b_{i,k}$. Este sistema puede escribirse en términos de vectores y matrices como $M \frac{d\hat{u}}{dt} = F(t, \hat{u})$ con $\hat{u}(t_0) = \hat{u}_0$ donde \hat{u} es un vector de coeficientes de tamaño $2M$ N y \hat{u}_0 contiene los valores iniciales de los

Referencias:

1. Constantinides, A., *Applied Numerical Methods with Personal Computers*, Mc Graw-Hill, 1987.
2. Ames, *Numerical Methods for Partial Differential Equations*, Academic Press, San Diego, 1977.
3. Do, D., Rice, J., *Applied Numerical Methods for Chemical Engineers*, Addison-Wesley, 1994.
4. Bibliotecas numéricas IMSL de Microsoft, archivo de ayuda, 1995.

**ESTA TESIS NO DEBE
SALIR DE LA BIBLIOTECA**

TABLAS Y LISTADOS

En las siguientes páginas se presentan las tablas ASCII para los parámetros de referencia de la tabla 3.1 tal como las produce el programa desarrollado.

El problema fue resuelto con el siguiente equipo operando en Windows 95:

Hardware:

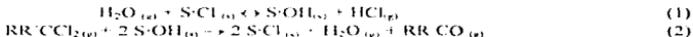
- Computadora Pentium 200 Mhz 32 MB RAM

Software:

- Microsoft Developer Studio, 1995
- Fortran PowerStation 4.0, 1995
- Visual C++ 4.0, 1995
- International Mathematical Software Libraries (IMSL) por Visual Numerics Inc.
- Borland C++ 4.02, 1993

B.1 Descripción:

Este programa fue desarrollado para simular las variaciones en la concentración de cloro adsorbido a través del tiempo dentro de un reactor tubular de lecho empacado con gamma-alumina según las siguientes reacciones



donde S es el sitio activo. Los componentes H_2O , HCl , $\text{RR}'\text{CCl}_2$ y $\text{RR}'\text{CO}$ se encuentran en la fase gaseosa, mientras que OH y Cl se encuentran adsorbidos en el soporte.

El problema se representa en forma matemática como un sistema de ecuaciones diferenciales parciales que se resuelven mediante colocación ortogonal en elemento finito y/o diferencias finitas. Para su resolución se requiere conocer las **condiciones iniciales** y **condiciones de frontera**. Estas condiciones se encuentran en el código que se incluye en este apéndice como las funciones FNBC (para colocación ortogonal) y 100, 101, ..., 800, s01, ... para diferencias finitas en el programa principal. Los listados tienen comentarios sobre la función que realiza cada parte del código.

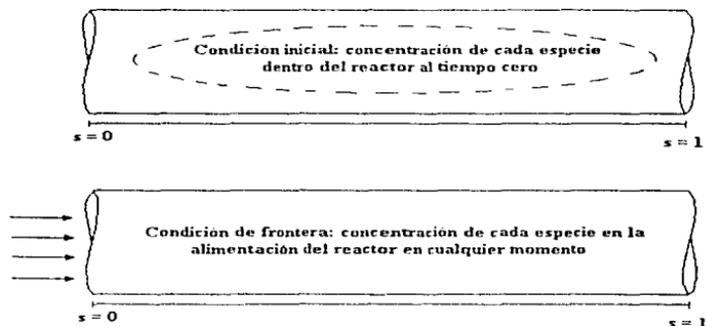
Las variables de programación que representan los parámetros físicos se incluyen en una estructura llamada **ProblemData** que se define en el listado **Reactub.h**. El programa indica al usuario las unidades que deben utilizarse. Para su compilación debe crearse un proyecto de consola en Visual C++ y elegir, en la opción **Build/Settings**, la lengüeta **Link**, donde se debe escribir en el cuadro de diálogo **Object/library modules** el texto **mathd_s.lib math_s.lib**

libf.lib Esto da acceso al compilador a las rutinas IMSL (escritas en FORTRAN) que realizan la integración numérica.

Además de las rutinas de colocación ortogonal, esta biblioteca permite resolver ecuaciones algebraicas no lineales, sistemas de ecuaciones diferenciales ordinarias, análisis estadísticos, rutinas de optimización, operaciones con vectores, matrices y números complejos, funciones especiales y otras rutinas numéricas. Desafortunadamente, esta biblioteca no funciona bajo ambiente Windows.

B.2 Entrada de datos:

Los siguientes diagramas explican el significado de las condiciones de frontera e iniciales. Las entradas que se recomiendan para realizar los cálculos están enumeradas en la tabla 4.1. Se pueden realizar las variaciones que se juzguen convenientes a partir de estos datos. Después de los diagramas se incluyen todos los listados.



Referencias:

1. Swan, Tom, *Mastering Borland C++ 4.5*, 2nd ed., Sams Publishing, Indiana, USA, 1995.
2. Visual C++ 4.0 Books On Line, 1995
3. Collected algorithms from ACM, Transactions on Mathematical Software, 18, 343-344, 1992. Algoritmos 540 y 688, de <http://math.nist.gov/gams/Classes.html>
4. *ATOLL*, IMSL FORTRAN Library, Visual Numerics Inc., 1993.

Además de que no se dispone de datos sobre las constantes de transferencia de masa entre el bulto y la fase gaseosa ni de las constantes cinéticas de cada paso individual

Referencias:

1. Szabó, I., Bliele, T., Ujhidy, A., Jelinko, R., *Kinetics of Aluminum Oxide Chlorination. 1. The Mechanism and Mathematical Model*, Ind. Eng. Chem. Res., 30, 292-298, 1991.
2. Szabó, I., Bliele, T., Ujhidy, A., Jelinko, R., *Kinetics of Aluminum Oxide Chlorination. 2. Experiments and Evaluation of Results*, Ind. Eng. Chem. Res., 30, 298-303, 1991.

```

1 #include "reactub.h"
2 #include "numlib.h"
3 #include <ctype.h>
4 #include <math.h>
5 #include <cr:\msl\fortran\include\mathd.h>
6 #include <cr:\msl\fortran\include\matha.h>
7
8 /* Definición de constante simbólica para colocación ortogonal. */
9 /* Corresponde a MAXDIV, definida en numlib.h */
10 #define COLDIV 100
11
12
13
14 /****** Declaración de funciones que usa la rutina de *****/
15 /****** Colocación ortogonal. *****/
16 /****** *****/
17
18 void __stdcall FCROT (long*, double*, double*, double*,
19 double*, double*, double*);
20 void __stdcall FCRES (long*, double*, double*, double*,
21 double*, double*);
22
23 /****** Declaración de funciones que usa la rutina de *****/
24 /****** Diferencias finitas *****/
25 /****** *****/
26
27
28 double rd (double *C[COMPONENTS][PREVIOUSPOINTS+1], int i, int j);
29 double ra (double *C[COMPONENTS][PREVIOUSPOINTS+1], int i, int j);
30 double dx0 (double *C[COMPONENTS][PREVIOUSPOINTS+1], int i, int j);
31 double dx1 (double *C[COMPONENTS][PREVIOUSPOINTS+1], int i, int j);
32 double dx2 (double *C[COMPONENTS][PREVIOUSPOINTS+1], int i, int j);
33 double dx3 (double *C[COMPONENTS][PREVIOUSPOINTS+1], int i, int j);
34 double dx4 (double *C[COMPONENTS][PREVIOUSPOINTS+1], int i, int j);
35 double dx5 (double *C[COMPONENTS][PREVIOUSPOINTS+1], int i, int j);
36
37 double t00 (double t);
38 double t01 (double t);
39 double t02 (double t);
40 double t03 (double t);
41 double t04 (double t);
42 double t05 (double t);
43 double s00 (double t);
44 double s01 (double t);
45 double s02 (double t);
46 double s03 (double t);
47 double s04 (double t);
48 double s05 (double t);
49 // double s10 (double t); // sBoundaryConditions las incluye
50 // double s11 (double t);
51 // double s12 (double t);
52 // double s13 (double t);
53 // double s14 (double t);
54 // double s15 (double t);
55
56 // Bcptr Conditions[3*COMPONENTS];
57 Bcptr Conditions[2*COMPONENTS]; // Ojo: por condiciones de frontera
58
59 ProblemData St;
60 ProblemData* Globals = &St;
61
62 /* Declaración de un objeto clase Tools */
63 Tools Reactor;

```

```

64 double ds;
65
66 void ShowMenu ();
67 void AskForLines();
68 void AskForCallout();
69 void CalculateLines(char reportfile[MAXLEN], double FinalTime, double dt);
70 void CalculateCellloc();
71
72 /*****
73 *****/
74 *****/
75 int main ()
76 {
77     ShowMenu ();
78     return 0;
79 }
80
81 void ShowMenu()
82 {
83     char option;
84     int done = FALSE;
85
86
87
88
89
90 while (!done)
91     printf("
92             UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO");
93     printf("\n
94             FACULTAD DE QUIMICA");
95     printf("\n
96             UNIDAD DE INVESTIGACION EN CATALISIS\n\n");
97     printf("\n
98             Programa para la solucion del modelo dinamico de un reactor tubular"
99 );
100     printf("\n
101             de adsorcion/desorcion de clore.");
102     printf("\n\n Elige una opcion:\n");
103     printf("\n\n1.- Crear archivo de parametros fisicos.\n");
104     printf("\n\n2.- Abrir/Editar archivo de parametros fisicos.\n");
105     printf("\n\n3.- Calcular mediante diferencias finitas.");
106     printf("\n\n4.- Metodo de Runge-Kutta de 4o. orden\n");
107     printf("\n\n5.- Calcular mediante colocacion ortogonal en elemento finito.");
108     printf("\n\n6.- Mas eficiente\n");
109     printf("\n\n7.- Ayuda sobre el programa.\n");
110     printf("\n\n8.- Salir.\n");
111
112     option = getch();
113     switch (option)
114     {
115     case '1':
116         {
117             Reactor.CreatePD();
118             printf("\nPresione cualquier tecla.\n\n");
119             getch();
120         }
121     case '2':
122         {
123             Reactor.EditPD();
124             printf("\nPresione cualquier tecla.\n\n");
125             getch();
126         }
127     case '3':
128         {
129             AskForLines();
130         }
131     }
132     break;
133 }

```

```

126     case '4':
127     {
128         AskForColloc();
129         break;
130     case '5':
131     {
132         Reactor.Help();
133         getch();
134         break;
135     case '6':
136     {
137         printf("\n\nGracias por usar este programa !!!\n\n");
138         done = TRUE;
139         break;
140     default:
141     {
142         printf("\n\nOpcion no valida!!!\n\n");
143         getch();
144         break;
145     }
146 }
147 }
148
149 void Calculatelines (char reportfile[MAXLEN],
150                    double FinalTime, double dt)
151 {
152     FILE *outf;
153     int TimeToSave = FALSE;
154     double SaveInterval = 0.01; // Salva cada 0.01 longitudes adimensionales
155     double ipart;
156
157     Conditions[0] = t00;
158     Conditions[1] = t01;
159     Conditions[2] = t02;
160     Conditions[3] = t03;
161     Conditions[4] = t04;
162     Conditions[5] = t05;
163     Conditions[6] = t06;
164     Conditions[7] = t01;
165     Conditions[8] = t02;
166     Conditions[9] = t03;
167     Conditions[10] = t04;
168     Conditions[11] = t05;
169     // Conditions[12] = s10;
170     // Conditions[13] = s11;
171     // Conditions[14] = s12;
172     // Conditions[15] = s13;
173     // Conditions[16] = s14;
174     // Conditions[17] = s15;
175
176     /* Declaracion de un objeto clase Lines */
177     Lines PhSystem (Conditions);
178     PhSystem.DESystem[0] = DX0;
179     PhSystem.DESystem[1] = DX1;
180     PhSystem.DESystem[2] = DX2;
181     PhSystem.DESystem[3] = DX3;
182     PhSystem.DESystem[4] = DX4;
183     PhSystem.DESystem[5] = DX5;
184
185     Lines *PhSPtr;
186     PhSPtr = &PhSystem;
187
188     ds = PhSystem.ds;

```

```

189 FinalTime = FinalTime + PhSystem.dt;
190
191 outf = fopen ("DATA.BIN", "wb");
192 if (!outf)
193 {
194     puts ("No se puede crear DATA.BIN");
195     exit (1);
196 }
197
198 double PointsPerPlot;
199 PointsPerPlot = 1.0/SaveIntervals + 1;
200 fwrite (&PointsPerPlot, sizeof (double), 1, outf);
201 Reactor.SaveLines (PhSPtr, SaveIntervals, outf, PREVIOUSPOINTS);
202
203 while (PhSystem.t[0] <= FinalTime)
204 {
205     TimeToSave = 1 - (modf (PhSystem.t[0] / dt, &ipart) < 0 + PhSystem.dt) ||
206               (modf (PhSystem.t[0] / dt, &ipart) > 1 - PhSystem.dt) ;
207     if (TimeToSave)
208         Reactor.SaveLines (PhSPtr, SaveIntervals, outf);
209
210     // if (PhSystem.t[0] == dummy)
211     //     for (int j = PREVIOUSPOINTS - 1; j >= 0; j--)
212     //     {
213     //         PhSystem.initialConditions (j);
214     //         PhSystem.PredictorCorrectorIp (j);
215     //         RNCI.MakeTable (PhSPtr);
216     //         PhSystem.boundaryConditions (j);
217     //     }
218     PhSystem.initialConditions ();
219     PhSystem.RungeKutta ();
220     PhSystem.boundaryConditions ();
221
222     PhSystem++;
223 }
224
225 fclose (outf);
226
227 Reactor.MakeReport (reportfile, FinalTime, dt);
228 }
229
230
231 void AskForLines()
232 {
233     double dt, FinalTime;
234
235     FILE* inpf;
236     char* stringR = (char*) malloc(MAXLEN);
237     char* stringW = (char*) malloc(MAXLEN);
238
239     printf("\n\n Se recomienda usar tiempos finales menores a 5 tiempos adimensionales.");
240     printf("\n\nPues los metodos de diferencias finitas son muy lentos. Se puede salvar");
241     printf("\nen el tiempo, por ejemplo, cada 0.5 o 1.0 tiempos adimensionales.\n");
242     printf("\nNombre del archivo a abrir para calcular (sin extension) = ");
243     scanf ("%s", stringR);
244     strcat (stringR, ".DAT");
245
246     inpf = fopen (stringR, "rb");
247     if (!inpf)
248     {
249         printf ("\nNo se puede abrir %s", stringR);
250         exit (1);
251     }

```

```

252 fread (&Reactor.Parameter, sizeof (ProblemData), 1, inpF);
253 fclose (inpF);
254 free (stringR);
255
256 printf ("Nombre del reporte (sin extension): ");
257 scanf ("%s", stringW);
258 printf ("Calvar en el tiempo cada: (tiempos dimensionales, t/(L/u)) = ");
259 scanf ("%i", &dt);
260 printf ("Tiempo final = ");
261 scanf ("%i", &FinalTime);
262
263 Global->L = <Reactor.Parameter.L;
264 Global->dp = <Reactor.Parameter.dp;
265 Global->Flow = <Reactor.Parameter.Flow;
266 Global->R1 = <Reactor.Parameter.R1;
267 Global->Rk1 = <Reactor.Parameter.Rk1;
268 Global->Rk2 = <Reactor.Parameter.Rk2;
269 Global->Rk3 = <Reactor.Parameter.Rk3;
270 Global->PEH2O = <Reactor.Parameter.PEH2O;
271 Global->PEHCl = <Reactor.Parameter.PEHCl;
272 Global->PECOCl2 = <Reactor.Parameter.PECOCl2;
273 Global->PECO2 = <Reactor.Parameter.PECO2;
274 Global->CLOH2O = <Reactor.Parameter.CLOH2O;
275 Global->CLOHCl = <Reactor.Parameter.CLOHCl;
276 Global->CLOCOCl2 = <Reactor.Parameter.CLOCOCl2;
277 Global->CLOCO2 = <Reactor.Parameter.CLOCO2;
278 Global->CEGH2O = <Reactor.Parameter.CEGH2O;
279 Global->CEGHCl = <Reactor.Parameter.CEGHCl;
280 Global->CECOCl2 = <Reactor.Parameter.CECOCl2;
281 Global->CECO2 = <Reactor.Parameter.CECO2;
282 Global->su = <Reactor.Parameter.su;
283 Global->prorf = <Reactor.Parameter.prorf;
284 Global->prorf = <Reactor.Parameter.prorf;
285 Global->prorf = <Reactor.Parameter.prorf;
286
287 Calculatelines (stringW, FinalTime, dt);
288
289
290
291 void Calculatelines (char reportfile[MAXLEN],
292                    double FinalTime, double dt)
293 {
294     FILE *outF;
295     // int TimeToSave = FALSE;
296     // double SaveInterval = 0.5; // ahora es dt argumento
297     double ds = 1.0 / COLDIV; // 0.01 longitudes adimensionales
298
299     // Punteros.
300     long Index, Leady, Numpar, Nint;
301     long * IDO, *LDY, *NPDES, *NX;
302     double dto, Tint, Tfin, Tolerr;
303     double * HINT, *T, *TEND, *TOL;
304     double *XBREAK[COLDIV + 1];
305     double *Line[COLDIV + 1][COMPONENTS];
306     double * Y;
307
308     Reactor.F = FCNUT;
309     Reactor.BC = FCNBC;
310
311     Numpar = COMPONENTS;
312     Leady = Numpar;
313     Nint = COLDIV + 1;
314     for (int i = 0; i < COLDIV + 1; i++)

```

```

315 {
316   XBREAK[] = dt*1;
317   /* Condiciones iniciales */
318   if (i == 0) Line[i][0] = St.CLOH2O /St.rofref; else Line[i][0] = 0.0;
319   // H2O
320   if (i == 0) Line[i][1] = St.CLOHCl /St.rofref; else Line[i][1] = 0.0;
321   // HCl
322   if (i == 0) Line[i][2] = St.CLOCOCl2/St.rofref; else Line[i][2] = 0.0;
323   // COCl2
324   if (i == 0) Line[i][3] = St.CLOCO2 /St.rofref; else Line[i][3] = 0.0;
325   // CO2
326   if (i == 0) Line[i][4] = St.CtOC1 /St.roaref; else Line[i][4] = St.CtOC1/S
327   t.roaref; // Cl
328   if (i == 0) Line[i][5] = (St.rofref-St.CtOC1)/St.roaref; else Line[i][5] = 0.0;
329   // OH
330 }
331
332 Toler = 1.0e-5;
333 dt0 = 1.0e-7;
334 Tfin = 0.0;
335 Tfin = 0.0;
336 Index = (long) 1.0;
337 // double NSTEP = 11.0;
338 Y = &Line[0][0];
339
340 IDO = &Index;
341 LDY = &Lead;
342 NPDES = &Numpar;
343 NX = &Nint;
344 HINIT = &dt0;
345 T = &Tfin;
346 TEND = &Tfin;
347 TOL = &Toler;
348
349 outf = fopen("DATA.BIN", "wb");
350 if (!outf)
351 {
352   puts("No se puede crear DATA.BIN");
353   exit(1);
354 }
355
356 double PointsPerPlot = COLDIV + 1;
357 fwrite(&PointsPerPlot, sizeof(double), 1, outf);
358
359 // Escribe en el archivo de resultados las condiciones
360 // iniciales
361 fwrite(&Tfin, sizeof(double), 1, outf);
362 for (i = 0; i < PointsPerPlot; i++)
363   for (int k = 0; k < COMPONENTS; k++)
364     fwrite(&Line[i][k], sizeof(double), 1, outf);
365
366 // Escribe en el archivo de resultados el resto del
367 // calculo.
368 do
369 {
370   Tfin += dt;
371   printf("\nCalculando t = %.21f (L/U)\n", Tfin);
372   /* --- Rutina de solucion por colocacion ortogonal --- */
373   DMOLCH (IDO, Reactor.F, Reactor.BC, NPDES, T, TEND,
374     NX, XBREAK, TOL, HINIT, Y, LDY);
375
376   fwrite(&Tfin, sizeof(double), 1, outf);
377   for (i = 0; i < PointsPerPlot; i++)

```

```

372     for (int k = 0; k < COMPONENTS; k++)
373         [write (&Line[1][k], sizeof (double), 1, outf);
374         if (Tfin == FinalTime) Index = (long) 3.0;
375         | while (Tfin < FinalTime);
376
377 fclose (outf);
378
379 Reactor.MakeReport (reportfile, FinalTime, dt);
380 |
381
382 void AskForCollor ()
383 |
384 double dt, FinalTime;
385
386 FILE *inpf;
387 char *stringR = (char *) malloc(MAXLEN);
388 char *stringW = (char *) malloc(MAXLEN);
389
390 printf ("\n\n Se puede usar el tiempo final que se desee, pues el metodo de");
391 printf ("Ancoleccion ortogonal en elemento finito es muy eficiente. Se puede");
392 printf ("Ansalvar en el tiempo, por ejemplo, cada 10 tiempos dimensionales");
393 printf ("Anbasta un tiempo final dimensional de 100.\n");
394 printf ("Nombre del archivo a abrir para calcular (sin extension) = ");
395 scanf ("%s", stringR);
396 strcat (stringR, ".DAT");
397
398 inpf = fopen (stringR, "rb");
399 if (!inpf)
400     |
401     printf ("Anno se puede abrir =s", stringR);
402     |
403     |
404
405 fread (&Reactor.Parameter, sizeof (ProblemData), 1, inpf);
406 fclose (inpf);
407 freopen (stringR,
408
409 printf ("Nombre del reporte (sin extension) = ");
410 scanf ("%s", stringW);
411 printf ("Anl", stringW);
412 printf ("Tiempo cada: (tiempos dimensionales, t/(L/u)) = ");
413 printf ("Tiempo final = ");
414 scanf ("%lf", &FinalTime);
415
416 Global->L = Reactor.Parameter.L;
417 Global->dp = Reactor.Parameter.dp;
418 Global->Flow = Reactor.Parameter.Flow;
419 Global->RD = Reactor.Parameter.RD;
420 Global->k1 = Reactor.Parameter.k1;
421 Global->k2 = Reactor.Parameter.k2;
422 Global->k3 = Reactor.Parameter.k3;
423 Global->PeH2O = Reactor.Parameter.PeH2O;
424 Global->PeHCl = Reactor.Parameter.PeHCl;
425 Global->PeCOCl2 = Reactor.Parameter.PeCOCl2;
426 Global->PeCO2 = Reactor.Parameter.PeCO2;
427 Global->CLOH2O = Reactor.Parameter.CLOH2O;
428 Global->CLOHCl = Reactor.Parameter.CLOHCl;
429 Global->CLOCOC12 = Reactor.Parameter.CLOCOC12;
430 Global->CLOCOC2 = Reactor.Parameter.CLOCOC2;
431 Global->CtOH2O = Reactor.Parameter.CtOH2O;
432 Global->CtOHCl = Reactor.Parameter.CtOHCl;
433 Global->CtOCOC12 = Reactor.Parameter.CtOCOC12;
434 Global->CtOCOC2 = Reactor.Parameter.CtOCOC2;

```

```

435 Globals->CtOC1 = Reactor.Parameter.CtOC1;
436 Globals->Su = Reactor.Parameter.u;
437 Globals->roref = Reactor.Parameter.roref;
438 Globals->rofreq = Reactor.Parameter.rofreq;
439
440 CalculateColloc (stringW, FinalTime, dt);
441 }
442
443
444 //.....
445 *** Las siguientes funciones se usan en la rutina de diferencias ***
446 *** finitas. ***
447 //.....
448
449 double rd (double *C[COMPONENTS][PREVIOUSPOINTS+1], int i, int j)
450 {
451     return St.rofreq * St.roref * ( St.k1*C[0][j][i]*C[4][j][i]
452                                     + St.k2*C[1][j][i]*C[5][j][i]);
453 }
454
455 double ra (double *C[COMPONENTS][PREVIOUSPOINTS+1], int i, int j)
456 {
457     return St.k3 * pow (St.rofreq, 2) * St.roref * (C[2][j][i] * pow (C[5][j][i], 2));
458 }
459
460 double DX0 (double *C[COMPONENTS][PREVIOUSPOINTS+1], int i, int j) // H
461 {
462     return ( St.dp/(St.L*St.PeH2O*pow (ds, 2)) * (C[0][j][i+1] - 2*C[0][j][i] + C[0][j][i-1])
463             - (0.5/ds) * (C[0][j][i+1] - C[0][j][i-1])
464             + St.L/(St.u*St.rofreq) * rd(C, i, j) + ra(C, i, j));
465 }
466
467 double DX1 (double *C[COMPONENTS][PREVIOUSPOINTS-1], int i, int j) // H
468 {
469     return ( St.dp/(St.L*St.PeHCl*pow (ds, 2)) * (C[1][j][i+1] - 2*C[1][j][i] + C[1][j][i-1])
470             + (0.5/ds) * (C[1][j][i+1] - C[1][j][i-1])
471             + St.L/(St.u*St.rofreq) * rd(C, i, j));
472 }
473
474 double DX2 (double *C[COMPONENTS][PREVIOUSPOINTS+1], int i, int j) // C
475 {
476     return ( St.dp/(St.L*St.PeCO2*pow (ds, 2)) * (C[2][j][i+1] - 2*C[2][j][i] + C[2][j][i-1])
477             - (0.5/ds) * (C[2][j][i+1] - C[2][j][i-1])
478             + St.L/(St.u*St.rofreq) * ra (C, i, j));
479 }
480
481 double DX3 (double *C[COMPONENTS][PREVIOUSPOINTS+1], int i, int j) // O
482 {
483     return ( St.dp/(St.L*St.PeCO2*pow (ds, 2)) * (C[3][j][i+1] - 2*C[3][j][i] + C[3][j][i-1])
484             + (0.5/ds) * (C[3][j][i+1] - C[3][j][i-1])
485             + St.L/(St.u*St.rofreq) * rd(C, i, j));
486 }
487
488
489 }

```

```

490      = (0.5/ds) * (C[3][j][i+1] - C[3][j][i-1])
491      + St.L/(St.u*St.rofref) * ra (C, i, j));
492 }
493
494 double DX4 (double *C[COMPONENTS][PREVIOUSPOINTS-1], int i, int j) // 0
495 {
496 return
497     ( + St.L/(St.u*St.rofref) * (ra (C, i, j) - rd (C, i, j)));
498 }
499
500 double DX5 (double *C[COMPONENTS][PREVIOUSPOINTS-1], int i, int j) // 0
501 {
502 return
503     ( + St.L/(St.u*St.rofref) * (ra (C, i, j) - ra (C, i, j)));
504 }
505
506 double t00 (double s) { if (s > 0.0) return St.CLOH20/St.rofref;
507                       } else return St.COH20/St.rofref; }
508 double t01 (double s) { if (s > 0.0) return St.CLOHCl/St.rofref;
509                       } else return St.COHCl/St.rofref; }
510 double t02 (double s) { if (s > 0.0) return St.CLOCC12/St.rofref;
511                       } else return St.COCC12/St.rofref; }
512 double t03 (double s) { if (s > 0.0) return St.CLOCO2/St.rofref;
513                       } else return St.COCCO2/St.rofref; }
514 double t04 (double s) { if (s > 0.0) return 0.0/St.rofref;
515                       } else return St.COC1/St.rofref; }
516 double t05 (double s) { if (s > 0.0) return St.COC1/St.rofref;
517                       } else return 0.0/St.rofref; }
518 double s00 (double t) { return St.CLOH20/St.rofref; }
519 double s01 (double t) { return St.CLOHCl/St.rofref; }
520 double s02 (double t) { return St.CLOCC12/St.rofref; }
521 double s03 (double t) { return St.CLOCCO2/St.rofref; }
522 double s04 (double t) { return 0.0/St.rofref; }
523 double s05 (double t) { return St.COC1/St.rofref; }
524 // double s10 (double t) { return PSystem.X[0][j][MAXDIV - 1]; }
525 // double s11 (double t) { return PSystem.X[1][j][MAXDIV - 1]; }
526 // double s17 (double t) { return PSystem.X[2][j][MAXDIV - 1]; }
527 // double s13 (double t) { return PSystem.X[3][j][MAXDIV - 1]; }
528 // double s14 (double t) { return PSystem.X[3][j][MAXDIV - 1]; }
529 // double s15 (double t) { return PSystem.X[3][j][MAXDIV - 1]; }
530
531 //*****
532 *** Las siguientes funciones se usan en la rutina de colocacion ***
533 *** ortogonal en elemento finito. ***
534
535 //*****
536 void _stdcall FCNUT (long* NPDES, double* X, double* T, double* U, double* UX, double*
537 UXK, double* UT)
538 {
539 UT[0] = (1.0/St.PeH2O) *(St.dp/St.L)*UXK[0] - UX[0] - St.L/(St.u*St.rofref)*(St.rofref*
540 St.rofref*(St.k1*U[0]*U[4]-St.k2*U[1]*U[5]) - St.rofref*St.rofref*St.rofref*St.k3*U[2]*U
541 (5)*U[3]);
542 UT[1] = (1.0/St.PeHC1) *(St.dp/St.L)*UXK[1] - UX[1] + St.L/(St.u*St.rofref)*(St.rofref*
543 St.rofref*(St.k1*U[0]*U[4]-St.k2*U[1]*U[5]) -
544 St.rofref*St.rofref*St.k3*U[2]*U[3]*U[5]);
545 UT[2] = (1.0/St.PeCC12) *(St.dp/St.L)*UXK[2] - UX[2] - St.L/(St.u*St.rofref)*(St.rofref*
546 St.rofref*St.rofref*St.k3*U[2]*U[3]*U[5]);
547 UT[3] = (1.0/St.PeCO2) *(St.dp/St.L)*UXK[3] - UX[3] + St.L/(St.u*St.rofref)*(St.rofref*
548 St.rofref*St.rofref*St.k3*U[2]*U[3]*U[5]);
549 UT[4] =
550     St.L/(St.u*St.rofref)*(St.rofref*
551     St.rofref*St.rofref*St.k3*U[2]*U[3]*U[5]) - St.rofref*St.rofref*(St.k1*U[0]*U[4]-St.k2*U[1]

```

```

543  i)*U[5]);
544  UT[5] = St.LZ*(St.u*St.rofref)*(St.rofref*
545  St.rofref*(St.k1*U[0]+U[4]-St.k2*U[1])*U[5]+St.rofref*St.rofref*St.k1*U[2])*U
546  [5]*U[5]);
547  }
548  /*-----
549  **** Las condiciones de frontera en colocacion ortogonal ****
550  **** se expresan de acuerdo a ALPHA,U + BETA,Uk = GAMP(U), ****
551  **** donde GAMP(t) = (d/dt) GAMMA(t), GAMMA(t) = ****
552  **** condicion de frontera. ****
553  ****-----
554  void __stdcall FONDE (long * NPDES, double * X, double * T, double * ALPHA, double * BETA, do
555  uble * GAMP)
556  {
557  if (*X != 0.0)
558  for (int i = 0; i < *NPDES; i++)
559  {
560  ALPHA[i] = 1.0; BETA[i] = 0.0; GAMP[i] = 0.0;
561  }
562  else
563  for (int i = 0; i < *NPDES; i++)
564  {
565  ALPHA[i] = 0.0; BETA[i] = 1.0; GAMP[i] = 0.0;
566  }
567  }
568  }

```

```

1 // reactub.h -- Declaracion de clases para problema reactor tubular
2 // UNISOFT (R) 1997
3
4 #ifndef __REACTUB_H
5 #define __REACTUB_H // Previene multiples #include
6
7 #include <conio.h>
8 #include <stdio.h>
9 #include <string.h>
10 #include <ctype.h>
11 #include <stdlib.h>
12 #include "numlib.h"
13
14 #define TRUE 1
15 #define FALSE 0
16 #define MAXLEN 50
17
18 // Pi no esta definida en las bibliotecas estandar de Microsoft.
19 #define M_PI 3.14159265358979323846
20
21 /* void __stdcall PLOTCH(long*, void*, __stdcall *(long*, double*, double*, double*, doub
22 le*,
23 double*, double*), void*, __stdcall *(long*, double*, double*, double*,
24 double*, double*), long*, double*, double*, long*, double*, double*,
25 double*, double*, long*); */
26
27 typedef void (__stdcall * FCNUTptr) (long*, double*, double*, double*, double*, double*,
28 double*);
29
30 typedef void (__stdcall * FCNBCptr) (long*, double*, double*, double*, double*, double*)
31 ;
32
33 /* La estructura que aloja los parametros de calculo */
34 typedef struct
35 {
36     double L, dpr;
37     double Flow, RD;
38     double kl, k2, k3;
39     double PeH2O, PeHCl, PeCOCl2, PeCO2;
40     double ClOH2O, ClOHCl, ClCOCl2, ClCO2;
41     double ClOCl;
42     double u, rofrat, rosret;
43 } ProblemaData;
44
45 class Tools
46 {
47 public:
48     // Parametros de calculo.
49     ProblemaData Parameter;
50     double ds;
51
52     // Punteros a funciones requeridos para colocacion ortogonal.
53     FCNUTptr F;
54     FCNBCptr BC;
55
56     // Punteros a funciones requeridos para diferencias finitas.
57     DEPtr DESystem[COMPONENTS];
58     BCPtr Conditions[2*COMPONENTS];
59
60     // Funciones de entrada / salida a pantalla.
61     void CreatePD();
62     void EditPD();
63     void Help ();

```

```
61
62 // Funciones de entrada / salida a archivos.
63 void MakeReport (char* stringW, double FinalTime, double dt);
64 void SaveLines (Lines *C, double SaveInterval, FILE *outf, int j = 0);
65
66 // Constructor/Destructor.
67 Tools ();
68 ~Tools () {}
69
70 };
71 #endif // __REACTUB_H
72
73
```

```

1 // Reactub.cpp Implementacion del header Reactub.h (Modelo)
2 // UNISOFT (R) 1997
3
4 #include <stdio.h>
5 #include <stdlib.h>
6 #include <conio.h>
7 #include <math.h>
8 #include <time.h>
9 #include <string.h>
10 #include "reactub.h"
11
12 /* Constructor - Inicializa la estructura de parametros */
13 Tools::Tools ()
14 {
15     Parameter.L = 0;
16     Parameter.dp = 0;
17     Parameter.Flow = 0;
18     Parameter.RD = 0;
19     Parameter.k1 = 0;
20     Parameter.k2 = 0;
21     Parameter.k3 = 0;
22     Parameter.PeHCO = 0;
23     Parameter.PeHCl = 0;
24     Parameter.PeCOCl2 = 0;
25     Parameter.PeCO2 = 0;
26     Parameter.CLOH2O = 0;
27     Parameter.CLOHCl = 0;
28     Parameter.CLOCOCl2 = 0;
29     Parameter.CLOCO2 = 0;
30     Parameter.CLOH2O = 0;
31     Parameter.CLOHCl = 0;
32     Parameter.CtOCOCl2 = 0;
33     Parameter.CtOCO2 = 0;
34     Parameter.CtUCl = 0;
35     Parameter.u = 0;
36     Parameter.rotrei = 0;
37     Parameter.rotrei = 0;
38 }
39
40
41 /* Menu */
42 void Tools::CreatePD()
43 {
44     FILE* outf;
45     char* stringC = (char*)malloc(MAXLEN);
46
47     printf ("\nNombre del archivo (sin extension) = ");
48     scanf ("%s", stringC);
49     strcat (stringC, ".DAT");
50
51     outf = fopen (stringC, "wb");
52     if (!outf)
53     {
54         printf ("No se puede crear %s", stringC);
55         exit (1);
56     }
57     printf ("\n Longitud del reactor (cm) = ");
58     scanf ("%f", &Parameter.L);
59     printf ("\n Diametro de particula catalitica (cm) = ");
60     scanf ("%f", &Parameter.dp);
61     printf ("\n Flujo de alimentacion (cm3/min) = ");
62     scanf ("%f", &Parameter.Flow);
63     printf ("\n Diametro del reactor tubular (cm) = ");

```

```

64 scanf ("%d", &Parameter.RD);
65 printf (" Constante de desorcion (cm3/mmol min) = ");
66 scanf ("%d", &Parameter.k1);
67 printf (" Constante de desorcion inversa (cm3/mmol min) = ");
68 scanf ("%d", &Parameter.k2);
69 printf (" Constante de adsorcion (cm3/mmol2/min) = ");
70 scanf ("%d", &Parameter.k3);
71 printf (" Porcentaje axial de agua = ");
72 scanf ("%d", &Parameter.PaH2O);
73 printf (" Porcentaje axial de HCl = ");
74 scanf ("%d", &Parameter.PaHCl);
75 printf (" Porcentaje axial de RR'CCl2 = ");
76 scanf ("%d", &Parameter.PaCCl2);
77 printf (" Porcentaje axial de RR'CO = ");
78 scanf ("%d", &Parameter.PaCO);
79 printf (" Conc. de H2O a la entrada del reactor (mmol/cm3) = ");
80 scanf ("%d", &Parameter.CLOH2O);
81 printf (" Conc. de HCl a la entrada del reactor (mmol/cm3) = ");
82 scanf ("%d", &Parameter.CLOHCl);
83 printf (" Conc. de RR'CCl2 a la entrada del reactor (mmol/cm3) = ");
84 scanf ("%d", &Parameter.CLOCCl2);
85 printf (" Conc. de RR'CO a la entrada del reactor (mmol/cm3) = ");
86 scanf ("%d", &Parameter.CLOCO);
87 printf (" Conc. de H2O dentro del reactor (t=0) (mmol/cm3) = ");
88 scanf ("%d", &Parameter.CLOH2O);
89 printf (" Conc. de HCl dentro del reactor (t=0) (mmol/cm3) = ");
90 scanf ("%d", &Parameter.CLOHCl);
91 printf (" Conc. de RR'CCl2 dentro del reactor (t=0) (mmol/cm3) = ");
92 scanf ("%d", &Parameter.CLOCCl2);
93 printf (" Conc. de RR'CO dentro del reactor (t=0) (mmol/cm3) = ");
94 scanf ("%d", &Parameter.CLOCO);
95 printf (" Conc. de Cl dentro del reactor (t=0) (mmol/cm3) = ");
96 scanf ("%d", &Parameter.CLOCl);
97
98 Parameter.u = 4.0 * Parameter.Flow / (M_PI * pow (Parameter.PD, 2));
99 Parameter.totref = Parameter.CLOH2O;
100 Parameter.totref = Parameter.CLOCl;
101
102 fwrite (&Parameter, sizeof (ProblemData), 1, outf);
103 fclose (outf);
104 free (stringR);
105 }
106
107 /* Menu */
108 void Tools::EditPD ()
109 {
110 int done = FALSE;
111
112 FILE * inpf;
113 FILE * outf;
114 char selection;
115 char *stringR = (char *) malloc (MAXLEN);
116 char *stringW = (char *) malloc (MAXLEN);
117
118 printf ("\nNombre del archivo a abrir (sin extension) = ");
119 scanf ("%s", stringR);
120 strcat (stringR, ".DAT");
121
122 inpf = fopen (stringR, "rb");
123 if (!inpf)
124 {
125 printf ("\nNo se puede abrir %s", stringR);
126 exit (1);

```

```

127     |
128     fread (&Parameter, sizeof (ProblembData), 1, inpf);
129     fclose (inpf);
130     free(stringR);
131
132     while (!done)
133     |
134     printf("\tOpcion a editar:\n");
135     printf("\n A. - longitud del reactor                = 4.4G cm", Parameter.L);
136     printf("\n B. - Diametro de partucula catalitica      = 4.4G cm", Parameter.dp);
137     printf("\n C. - Flujo de alimentacion                    = 4.4G cm3/min", Parameter
.Flow);
138     printf("\n D. - Diametro de reactor tubular            = 4.4G cm", Parameter.RD);
139     printf("\n E. - Constante de desorcion                    = 4.4G cm3/mmol min", Para
meter.k1);
140     printf("\n F. - Constante de desorcion inversa        = 4.4G cm3/mmol min", Para
meter.k2);
141     printf("\n G. - Constante de adsorcion                = 4.4G (cm3/mmol)2/min", P
arameter.k3);
142     printf("\n H. - Peclat de agua                      = 4.4G", Parameter.PeH2O);
143     printf("\n I. - Peclat de HCl                        = 4.4G", Parameter.PeHCl);
144     printf("\n J. - Peclat de RR'CC12                    = 4.4G", Parameter.PeCC12
);
145     printf("\n K. - Peclat de RR'CO                      = 4.4G", Parameter.PeCO);
146     printf("\n L. - Conc. de H2O a la entrada del reactor  = 4.4G mmol/cm3", Paramete
r.CLOH2O);
147     printf("\n M. - Conc. de HCl a la entrada del reactor = 4.4G mmol/cm3", Paramete
r.CLOHCl);
148     printf("\n N. - Conc. de RR'CC12 a la entrada del reactor = 4.4G mmol/cm3", Paramete
r.CLOCC12);
149     printf("\n O. - Conc. de RR'CO a la entrada del reactor = 4.4G mmol/cm3", Paramete
r.CLOCO);
150     printf("\n P. - Conc. de H2O dentro del reactor (t=0) = 4.4G mmol/cm3", Paramete
r.CDOH2O);
151     printf("\n Q. - Conc. de HCl dentro del reactor (t=0) = 4.4G mmol/cm3", Paramete
r.CDOHCl);
152     printf("\n R. - Conc. de RR'CC12 dentro del reactor (t=0) = 4.4G mmol/cm3", Paramete
r.CDOCC12);
153     printf("\n S. - Conc. de RR'CO dentro del reactor (t=0) = 4.4G mmol/cm3", Paramete
r.CDOCO);
154     printf("\n T. - Conc. de Cl Dentro del reactor (t=0) = 4.4G mmol/cm3", Paramete
r.CDOCl);
155
156
157     selection = toupper (getch ());
158     switch (selection)
159     |
160     case 'A':
161         (printf("\n longitud del reactor (cm) = ");
162          scanf ("%lf", &Parameter.L); done = TRUE;) break;
163     case 'B':
164         (printf("\n Diametro de partucula catalitica (cm) = ");
165          scanf ("%lf", &Parameter.dp); done = TRUE;) break;
166     case 'C':
167         (printf("\n Flujo de alimentacion (cm3/min) = ");
168          scanf ("%lf", &Parameter.Flow); done = TRUE;) break;
169     case 'D':
170         (printf ("\n Diametro del reactor (cm) = ");
171          scanf ("%lf", &Parameter.RD); done = TRUE;) break;
172     case 'E':
173         (printf ("\n Constante de desorcion (cm3/mmol min) = ");
174          scanf ("%lf", &Parameter.k1); done = TRUE;) break;
175     case 'F':

```

```

176     printf ("\n Constante de velocidad inversa (cm3/mmol min) = ");
177     scanf ("%lf", &Parameter.k1); done = TRUE; break;
178   case 'G':
179     printf ("\n Constante de absorcion (cm3/mmol12/min) = ");
180     scanf ("%lf", &Parameter.k2); done = TRUE; break;
181   case 'H':
182     printf ("\n Pelet axial de agua = ");
183     scanf ("%lf", &Parameter.PEAG); done = TRUE; break;
184   case 'I':
185     printf ("\n Pelet axial de HCl = ");
186     scanf ("%lf", &Parameter.PEHC); done = TRUE; break;
187   case 'J':
188     printf ("\n Pelet axial de PRACO1 = ");
189     scanf ("%lf", &Parameter.PEON1); done = TRUE; break;
190   case 'K':
191     printf ("\n Pelet axial de PRACO2 = ");
192     scanf ("%lf", &Parameter.PEON2); done = TRUE; break;
193   case 'L':
194     printf ("\n Conc. de H2O a la entrada del reactor (mmol/cm3) = ");
195     scanf ("%lf", &Parameter.CEHO); done = TRUE; break;
196   case 'M':
197     printf ("\n Conc. de HCl a la entrada del reactor (mmol/cm3) = ");
198     scanf ("%lf", &Parameter.CEHC); done = TRUE; break;
199   case 'N':
200     printf ("\n Conc. de PRACO1 a la entrada del reactor (mmol/cm3) = ");
201     scanf ("%lf", &Parameter.CEON1); done = TRUE; break;
202   case 'O':
203     printf ("\n Conc. de PRACO2 a la entrada del reactor (mmol/cm3) = ");
204     scanf ("%lf", &Parameter.CEON2); done = TRUE; break;
205   case 'P':
206     printf ("\n Conc. de H2O dentro del reactor (t=0) (mmol/cm3) = ");
207     scanf ("%lf", &Parameter.CDHO); done = TRUE; break;
208   case 'Q':
209     printf ("\n Conc. de HCl dentro del reactor (t=0) (mmol/cm3) = ");
210     scanf ("%lf", &Parameter.CDHC); done = TRUE; break;
211   case 'R':
212     printf ("\n Conc. de PRACO1 dentro del reactor (t=0) (mmol/cm3) = ");
213     scanf ("%lf", &Parameter.CDON1); done = TRUE; break;
214   case 'S':
215     printf ("\n Conc. de PRACO2 dentro del reactor (t=0) (mmol/cm3) = ");
216     scanf ("%lf", &Parameter.CDON2); done = TRUE; break;
217   case 'T':
218     printf ("\n Conc. de Cl dentro del reactor (t=0) (mmol/cm3) = ");
219     scanf ("%lf", &Parameter.CDC); done = TRUE; break;
220   default:
221     printf ("\n\t--Selección no válida-- Presione cualquier tecla.\n");
222     getch (); done = FALSE; break;
223   }
224 }
225
226 Parameter.u = 4*Parameter.Flow / (M_PI*pow (Parameter.RD, 2));
227 Parameter.c0ref = Parameter.CDHO;
228 Parameter.c0ref = Parameter.CDHC;
229
230 printf ("\nNombre del archivo a guardar (sin extension) = ");
231 scanf ("%s", stringW);
232 strcat (stringW, ".DAT");
233 outf << fopen (stringW, "wb");
234 if (!outf)
235 {
236     printf ("\nNo se puede crear '%s', stringW);
237     exit (1);

```

```

239     }
240
241     fwrite (&Parameter, sizeof (ProblemData), 1, infile);
242     fclose (outfile);
243     free(stringW);
244     }
245
246     void Tools::Help()
247     {
248         printf("\n --- Programa para la solucion del modelo dinamico de un reactor tubular ---\n");
249         printf("\n          --- de adsorción/desorción de cloro. ---\n");
250         printf("\n          UNISOFT (R) - 1997 / Rodolfo Garcia Flores\n");
251         printf("\n Este programa resuelve el modelo dinamico de un reactor tubular de:\n");
252         printf("\n lecho empacado dentro del cual se llevan a cabo las siguientes:\n");
253         printf("\n reacciones:\n");
254         printf("\n  $H_2O(g) + S.Cl(a) \rightleftharpoons S.OH(s) + HCl(g)$ ;\n");
255         printf("\n  $RR'CO_2(g) + 2 S.OH \rightleftharpoons 2 S.Cl(a) + H_2O(g) + RR'CO(g)$ ;\n");
256         printf("\n donde S representa un sitio activo. El modelo representa el cambio en el\n");
257         printf("\n perfil de concentraciones de cada especie quimica en la coordenada axial\n");
258         printf("\n a traves del tiempo, considerando los efectos de la difusion, conveccion\n");
259         printf("\n y suponiendo reacciones simples.\n");
260         printf("\n ¿Las ecuaciones a resolver son:\n");
261         printf("\n  $Dt(X_0) = dp/(PeH_2O*L) Da_0(X_0) - Da_1(X_0) - L/(U \text{rofrref}) (da-ra)$ ;\n");
262         printf("\n  $Dt(X_1) = dp/(PeHCl*L) Da_1(X_1) - Da(X_2) + L/(U \text{rofrref}) (ra)$ ;\n");
263         printf("\n  $Dt(X_2) = dp/(PeCO_2*L) Da_2(X_2) - Da(X_2) - L/(U \text{rofrref}) (ra)$ ;\n");
264         printf("\n  $Dt(X_3) = dp/(PeCO_2*L) Da_3(X_3) - Da(X_3) + L/(U \text{rofrref}) (ra)$ ;\n");
265         printf("\n  $Dt(X_4) = L/(u \text{rofrref}) (da-ra)$ ;\n");
266         printf("\n  $Dt(X_5) = L/(u \text{rofrref}) (da-ra)$ ;\n");
267         printf("\n Atiende cada  $X_i$  es uno de los compuestos quimicos, de la primera derivada\n");
268         printf("\n parcial en el tiempo, de la primera derivada parcial en la coordenada\n");
269         printf("\n axial y  $Da_0$  la segunda derivada parcial en la coordenada axial. El resto\n");
270         printf("\n de los parametros se introduce en forma dimensional en el archivo de datos\n");
271         printf("\n\n");
272         printf("\n --- Presione cualquier tecla ---\n");
273
274         // Escribe el reporte de texto ASCII
275         void Tools::MakeReport (char* stringW, double FinalTime, double dt)
276         {
277             FILE * resultsf;
278             time_t theTime;
279             FILE *textf;
280
281             double p, t, ipart;
282             double X[COMPONENTS];
283
284             time (&theTime);
285
286             resultsf = fopen ("DATA.BIN", "rb");
287             if (!resultsf)
288                 {
289                     puts ("No puede abrirse DATA.BIN");
290                     exit (1);
291                 }
292
293             strcpy (stringW, ".TXT");
294             textf = fopen (stringW, "w");

```



```

341 fprintf (textf, "ANAL (X0) = op/(p*CO2*L) Dss(X0) - Ds(X0) - L/(U corref) (rd-tw)");
342 fprintf (textf, "ANAL (X1) = op/(p*CO2*L) Dss(X1) - Ds(X2) + L/(U corref) rd");
343 fprintf (textf, "ANAL (X2) = op/(p*CO2*L) Dss(X2) - Ds(X2) - L/(U corref) ra");
344 fprintf (textf, "ANAL (X3) = op/(p*CO2*L) Dss(X3) - Ds(X3) + L/(U corref) ra");
345 fprintf (textf, "ANAL (X4) = L/(U corref) (ca-rd)");
346 fprintf (textf, "ANAL (X5) = L/(U corref) (ca-ra)");
347
348 fread (&p, sizeof (double), 1, resultf);
349
350 fprintf (textf, "ANAL Puntos calculados a un tiempo dado = %G\n", p);
351
352 do
353 {
354     fread (&t, sizeof (double), 1, resultf);
355     if (modf (t/dt, &part) < 1e-3 || modf (t/dt, &part) > 1-1e-3)
356     {
357         fprintf (textf, "ANAL Tiempo = %.21f Tiempos de residencia (Tiempo / (L/U))\n", t);
358
359         fprintf (textf, "AN Variables dimensionales\n");
360         fprintf (textf, "ANAL");
361         for (int k = 0; k < COMPONENTS; k++)
362             fprintf (textf, "%e", X[k]);
363         fprintf (textf, "AN\n");
364         for (int j = 0; j < p; j++)
365         {
366             for (int k = 0; k < COMPONENTS; k++)
367                 fread (&X[k], sizeof (double), 1, resultf);
368             if ( (modf (t/dt, &part) < 1e-3 || (modf (t/dt, &part) > 1-1e-3))
369                 {
370                 fprintf (textf, "AN %.21f AN", j/(p-1));
371                 for (int k = 0; k < COMPONENTS; k++)
372                     fprintf (textf, "%e", X[k]);
373             }
374         }
375     } while (!feof (resultf));
376
377 fclose (resultf);
378 fclose (textf);
379 }
380
381
382 /* Escribe un archivo binario intermedio */
383 void Tools::SaveLines (Lines *C, double SaveIntervals, FILE *outf, int j)
384 {
385     double ipart;
386     double epsilon = C->ds;
387     double dummy = 1.0 / C->ds;
388
389     fwrite (&C->t[j], sizeof (double), 1, outf);
390     printf ("AN Calculando t = %.21f (L/U)\n", C->t[j]);
391     for (int i = 0; i < dummy + 1; i++)
392         if ( (modf (C->t[j][i] / SaveIntervals, &part) < 0 + epsilon) ||
393             (modf (C->t[j][i] / SaveIntervals, &part) > 1 - epsilon) )
394             for (int k = 0; k < COMPONENTS; k++)
395                 fwrite (&C->X[k][j][i], sizeof (double), 1, outf);
396 }
397
398

```

```

1 // numlib.h -- Declaracion de clases para biblioteca numerica. (Modelo4)
2 // UNISOF7 (R) 1997
3
4 #ifndef __NUMLIB_H
5 #define __NUMLIB_H // Previene multiples #include
6
7 #define MAXDIV 1000 // Divisiones en coordenada axial
8 #define COMPONENTS 6
9 #define PREVIOUSPOINTS 1 // Metodo de un solo paso
10
11 #include <stdarg.h>
12 #include <stdio.h>
13 #include <conio.h>
14
15 /* Alias condiciones de frontera */
16 typedef double (* BCptr)(double var);
17 /* Alias sistema de ecuaciones diferenciales */
18 typedef double (* DEptr)(double *COMPONENTS[[PREVIOUSPOINTS+1]], int i, int j);
19
20 class Lines
21 {
22 public:
23     double dt;
24     double ds;
25     double t[[PREVIOUSPOINTS+1]]; // Cuenta j (tiempo)
26     double *x[[PREVIOUSPOINTS+1]]; // Cuenta i (posicion)
27     double *X[[COMPONENTS]][[PREVIOUSPOINTS+1]]; // Cuenta k (componente)
28     DEptr DESys; // Cuenta k (componente)
29     BCptr Conditions[[COMPONENTS]];
30
31     /* Funciones miembro */
32     double sInitialConditions (int j = 0);
33     double sBoundaryConditions (int j = 0);
34
35     /* Constructor/Destructor */
36     Lines (BCptr Cond[[COMPONENTS]], double deltas = 1.0/MAXDIV,
37           double deltaT = 1e-3);
38     ~Lines () { delete [] t; delete [] x; delete [] X; delete [] DESys; }
39
40     /* Metodos de un solo paso */
41     double RungeKutta (int j = 0);
42     double Euler (int j = 0);
43     double Heun (int j = 0, double Tol = 1e-5);
44     double PredictorCorrector1p (int j = 0, double Tol = 1e-5);
45
46     /* Metodos multipaso */
47     double AddamsBashforth (int j);
48     double AddamsMoulton (double Tol = 1e-5);
49     double PredictorCorrector3p (double Tol = 1e-5);
50
51     /* Recorre la linea */
52     void operator ++ (int);
53     void operator = (const Lines &copy);
54 };
55
56 #endif // __NUMLIB_H
57

```

```

1 // numlib.cpp - Implementacion de clases del header numlib.hpp
2 // UNISCF7 (Rev. 1997)
3
4 #include "numlib.h"
5 #include "math.h"
6 #include "stdlib.h"
7 #include "numlib.h"
8
9 /* Constructor inicializa la clase Lines */
10 Lines::Lines (intptr Cond[3*COMPONENTS], double deltas, double deltat)
11 {
12     int i, j, k;
13     double dummy;
14
15     dt = deltat;
16     ds = deltas;
17     dummy = MAXDIV;
18
19     // for (i = 0; i < 3*COMPONENTS; i++)
20     for (i = 0; i < 3*COMPONENTS; i++)
21         Conditions[i] = Cond[i];
22
23     for (j = 0; j < PREVIOUSPOINTS + 1; j++)
24         x[j] = new double [MAXDIV + 1];
25     for (k = 0; k < COMPONENTS; k++)
26         for (i = 0; i < PREVIOUSPOINTS + 1; i++)
27             x[k][i] = new double [MAXDIV + 1];
28
29     for (j = 0; j < PREVIOUSPOINTS + 1; j++)
30         t[j] = (PREVIOUSPOINTS-j)*dt;
31
32     for (j = 0; j < PREVIOUSPOINTS + 1; j++)
33         for (i = 0; i < dummy + 1; i++)
34             x[i][j] = 1.0;
35     for (k = 0; k < COMPONENTS; k++)
36         for (i = 0; i < PREVIOUSPOINTS + 1; i++)
37             for (j = 0; j < dummy + 1; j++)
38                 if (i == 0 || i == dummy)
39                     x[k][j][i] = (Conditions[0*COMPONENTS+k])(x[j][i]);
40                 if (i == 0)
41                     x[k][j][i] = (Conditions[1*COMPONENTS+k])(t[j]);
42                 if (i == dummy)
43                     // x[k][j][i] = (Conditions[2*COMPONENTS+k])(t[j]);
44                     x[k][j][i] = x[k][j][i-1];
45             }
46     }
47 }
48
49 double Lines::InitialConditions (int j)
50 {
51     for (int k = 0; k < COMPONENTS; k++) // Asigna condiciones iniciales
52         x[k][j][0] = (Conditions[1*COMPONENTS+k])(t[j]);
53
54     return 0;
55 }
56
57 double Lines::BoundaryConditions (int j)
58 {
59     for (int k = 0; k < COMPONENTS; k++) // Asigna condiciones de fronteras
60         // x[k][j][MAXDIV] = (Conditions[2*COMPONENTS+k])(t[j]);
61         x[k][j][MAXDIV] = x[k][j][MAXDIV - 1];
62
63     return 0;

```

```

63 .....
64 .....
65 .....
66 *** El siguiente codigo implementa los metodos ***
67 *** de diferencias finitas. ***
68 .....
69 .....
70
71 double Lines::RungeKutta (int j)
72 {
73     int i, k;
74     double K[4][COMPONENTS];
75
76     for (i = 1; i < 1.0/ds; i++)
77     {
78         for (k = 0; k < COMPONENTS; k++)
79             K[0][k] = (DESystem[k])(X, i, j + 1);
80         t[i] += dt/2;
81         for (k = 0; k < COMPONENTS; k++)
82             X[k][i] += dt/2 * K[0][k]/2;
83         for (k = 0; k < COMPONENTS; k++)
84             K[1][k] = (DESystem[k])(X, i, j + 1);
85         for (k = 0; k < COMPONENTS; k++)
86             X[k][i] += dt * K[1][k] - K[0][k]/2;
87         for (k = 0; k < COMPONENTS; k++)
88             K[2][k] = (DESystem[k])(X, i, j + 1);
89         t[i] += dt/2;
90         for (k = 0; k < COMPONENTS; k++)
91             X[k][i] += dt * K[2][k] - 0.5 * K[1][k];
92         for (k = 0; k < COMPONENTS; k++)
93             K[3][k] = (DESystem[k])(X, i, j + 1);
94         for (k = 0; k < COMPONENTS; k++)
95             X[k][i] += dt * K[3][k];
96         t[i] = t[i] + dt;
97         t[i] += dt;
98     }
99     for (k = 0; k < COMPONENTS; k++)
100         X[k][i][i] = X[k][i] + t[i] + (dt/6) * (K[0][k] + 2*K[1][k] + 2*K[2][k] + K[3][k]);
101 }
102
103 return 0;
104 }
105
106 double Lines::Euler (int j)
107 {
108     for (int i = 1; i < 1.0/ds; i++)
109     {
110         printf("\nXB[2J]");
111         printf("ap = %i t = %i", i, t[i]);
112         for (int k = 0; k < COMPONENTS; k++)
113             X[k][i][i] = X[k][i-1][i] + dt/2 * (DESystem[k])(X, i, j + 1);
114     }
115
116     return 0;
117 }
118
119 double Lines::Heun (int j, double Tol)
120 {
121     int i, k;
122     double dummy, dif = 0;
123     double XTem[COMPONENTS][MAXDIV + 1];
124
125     for (i = 1; i < 1.0/ds; i++)
126     {

```

```

127 printf("\nX[2J]");
128 printf ("ic = %i t = %f dif = %1.0E", i, t[], dif);
129 dummy = 0;
130 for (k = 0; k < COMPONENTS; k++)
131 {
132     dummy = X[k][i][1];
133     XTem[k][i] = X[k][i][1] + dt/2 * ((DESystem[k] - (X, i, j + 1) + (DESystem[k]) (X
134     , i, j));
135     if (fabs (dummy - XTem[k][i]) > Tol)
136         dif = fabs (dummy - XTem[k][i]);
137 }
138
139 for (i = 1; i < MAXDIV; i++)
140     for (k = 0; k < COMPONENTS; k++)
141         X[k][i][1] = XTem[k][i];
142
143 return dif;
144 }
145
146 double Lines::PredictorCorrectorip (int j, double Tol)
147 {
148     double dif;
149
150     Euler (j);
151     do
152     {
153         dif = Heun (j, Tol);
154     } while (dif > Tol);
155
156     return 0;
157 }
158
159 double Lines::AddamsKishforth ()
160 {
161     for (int i = 1; i < 1.0/ds; i++)
162     {
163         printf("\nX[2J]");
164         printf ("ip = %i t = %f", i, t[0]);
165         for (int k = 0; k < COMPONENTS; k++)
166             X[k][0][i] = X[k][1][i] + dt/12 * (
167                 23 * (DESystem[k]) (X, i, 1)
168                 + 16 * (DESystem[k]) (X, i, 2)
169                 + 5 * (DESystem[k]) (X, i, 3));
170
171     }
172     return 0;
173 }
174
175 double Lines::AddamsMoulton (double Tol)
176 {
177     int i, k;
178     double dummy, dif = 0;
179     double XTem[COMPONENTS][MAXDIV + 1];
180
181     for (i = 1; i < 1.0/ds; i++)
182     {
183         printf("\nX[2J]");
184         printf ("ic = %i t = %f dif = %1.0E", i, t[0], dif);
185         dummy = 0;
186         for (k = 0; k < COMPONENTS; k++)
187             dummy = X[k][0][i];
188         XTem[k][i] = X[k][1][i] + dt/24 * (
189             9 * (DESystem[k]) (X, i, 0)

```

```

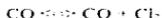
189             + 19 * DESystem[k] (X, 1, 1)
190             + 5 * DESystem[k] (X, 1, 2)
191             + DESystem[k] (X, 1, 3));
192     if (fabs (dummy - XTem[k][1]) > Tol)
193         dif = fabs (dummy - XTem[k][1]);
194     }
195 }
196
197 for (i = 1; i < MAXDIV; i++)
198     for (k = 0; k < COMPONENTS; k++)
199         X[k][0][i] = XTem[k][1];
200
201 return dif;
202 }
203
204 double Lines::PredictorCorrector3p (double Tol)
205 {
206     double dif;
207
208     AddamsBashforth ();
209     do
210     {
211         dif = AddamsMoulton (Tol);
212     } while (dif > Tol);
213
214     return 0;
215 }
216
217 void Lines::operator ++ (int)
218 {
219     double dummy = 1.0/ds;
220
221     for (int j = PREVIOUSPOINTS; j > 0; j--)
222         for (int k = 0; k < COMPONENTS; k++)
223             for (int i = 0; i <= dummy; i++) // Los valores iniciales de la linea 0 siguen
224                 X[k][i][1] = X[k][j-1][1]; // Ojo
225
226     for (j = PREVIOUSPOINTS; j > 0; j--)
227         t[j] = t[j-1];
228 }
229
230 void Lines::operator = (const Lines &copy)
231 {
232     int i, j, k;
233     if (this == &copy)
234         return;
235     ds = copy.ds;
236     for (j = 0; j < PREVIOUSPOINTS + 1; j++)
237         for (i = 0; i < 1/ds + 1; i++)
238             X[j][i] = copy.X[j][i];
239     for (k = 0; k < COMPONENTS; k++)
240         for (j = 0; j < PREVIOUSPOINTS + 1; j++)
241             for (i = 0; i < 1/ds + 1; i++)
242                 X[k][j][1] = copy.X[k][j][1];
243 }
244
245
246

```

MODELO DE CLORACIÓN DE ALÚMINA PARA LA OBTENCIÓN DE $AlCl_3$

Szabo et al. [1, 2] propusieron un modelo basado en el de Milne para la cloración de alúmina empleando monóxido de carbono, cloro y fosgeno diluidos en un gas inerte (nitrógeno). Su modelo hace las siguientes consideraciones:

- a) La mezcla de reactivos en el gas inerte se introducen al reactor clorador a presión atmosférica y temperatura ambiente a una presión parcial p^0 y a un flujo volumétrico v^0 .
- b) La corriente de gas entrante es calentada a la temperatura requerida y la mezcla gaseosa reacciona con el óxido de aluminio.
- c) Como resultado, la composición de la mezcla gaseosa cambia según la reacción:



Ecuación C. 1

El periodo de calentamiento es relativamente corto, por lo que no se espera que la mezcla gaseosa entre al clorador con composición de equilibrio.

- d) Los componentes de la mezcla gaseosa penetran a través de una delgada capa desde la fase gaseosa de hulto hasta la superficie de la partícula sólida. El coeficiente de transferencia de masa es β y la presión parcial de los componentes gaseosos sobre la superficie de la partícula es p^s (véase figura 2.5).
- e) El primer paso en la cloración es la *quimisorción de cloro*. El segundo es la *adsorción del monóxido de carbono* sobre la capa de cloro quimisorbido. Así se forma una monocapa de un *complejo tipo fosgeno*, según se aprecia en la figura 2.6.
- f) La configuración de la figura 2.6 se forma en un solo paso si el fosgeno está presente cerca de la superficie sólida.
- g) A esta fase de transición sigue una transformación para producir CO_2 y $AlCl_3$ que permanecen en estado adsorbido sobre la superficie.
- h) El último paso de la reacción es la desorción del tricloruro de aluminio y dióxido de carbono.

La reacción total es:



Ecuación C. 2

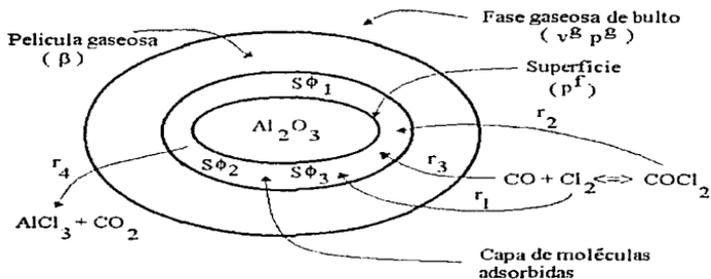


Fig. C.1. Mecanismo de la cloración de alúmina según Szabó.

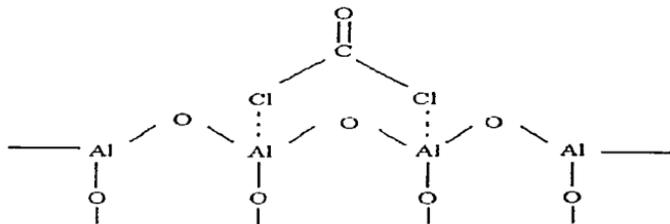


Fig. C.2. Complejo tipo fósgeno intermediario en la cloración de alúmina.

A continuación derivamos el modelo general para la reacción en pasos. Ver [1, 2] para detalles:

- I. **Reacciones sobre el sólido.** Sea S la superficie total de óxido de aluminio en la partícula. Sea la fracción no cubierta $S\phi_1$, la fracción cubierta solamente por cloro adsorbido $S\phi_2$, y la fracción cubierta por complejo tipo fósgeno o por productos $S\phi_3$ (Fig. C.1).

La rapidez con que se adsorbe químicamente el cloro sobre la superficie descubierta es:

$$r_1 = S k_1 p'_{Cl_2} \quad \text{Ecuación C. 3}$$

la rapidez de quimisorción de las moléculas de fósgeno es:

$$r_2 = S k_2 p'_{COCl_2} \quad \text{Ecuación C. 4}$$

la rapidez con que se adsorbe el CO sobre la capa ya formada de cloro es:

$$r_3 = S k_3 p'_{CO} \quad \text{Ecuación C. 5}$$

la rapidez de transformación del complejo tipo fósgeno incluyendo la remoción de Al_2O_3 y CO_2 es (si suponemos que la superficie completa está cubierta, es decir, concentración unitaria)

$$r_4 = S k_4 \quad \text{Ecuación C. 6}$$

Entonces, en un estado casi estacionario (porque en este modelo se supone que la partícula se consume)

- La rapidez con que el cloro se adsorbe sobre la fracción descubierta debe ser igual a la rapidez con que el monóxido de carbono se adsorbe sobre la monocapa de cloro.

$$\phi_1 r_1 = \phi_2 r_3 \quad \text{Ecuación C. 7}$$

- La rapidez con que se desorben los productos debe ser igual a la velocidad total con que se produce el complejo tipo fósgeno (ec. 2.6).

$$\phi_1 r_2 + \phi_2 r_3 = \phi_4 r_4 \quad \text{Ecuación C. 8}$$

- La suma de las fracciones de la superficie debe ser igual a uno.

$$\phi_1 + \phi_2 + \phi_3 + \phi_4 = 1 \quad \text{Ecuación C. 9}$$

Combinando las ecuaciones C.3 a C.9

$$\frac{S}{r_m} = \frac{1}{k_1 p'_{Cl_2} + k_2 p'_{COCl_2}} \left(1 + \frac{k_1 p'_{Cl_2}}{k_3 p'_{CO}} \right) + \frac{1}{k_4} \quad \text{Ecuación C. 10}$$

donde r_m es la rapidez con que se consume la partícula de Al_2O_3 en unidades de $\text{mol g}^{-1} \text{s}^{-1}$ (Ver Nomenclatura).

2. **Transferencia de masa desde la fase gas a la superficie sólida.** En un estado casi estacionario las siguientes ecuaciones son válidas:

$$k_1 S \phi_1 p_{Cl_2}^E = \beta_{Cl_2} S \phi_1 (p_{Cl_2}^E - \alpha_{Cl_2} p_{Cl_2}^G)$$

Ecuación C. 11

$$p_{Cl_2}^E = \beta_{Cl_2} p_{Cl_2}^G / (k_1 + \beta_{Cl_2} \alpha_{Cl_2})$$

Ecuación C. 12

$$k_2 S \phi_1 p_{COCl_2}^E = \beta_{COCl_2} S \phi_1 (p_{COCl_2}^E - \alpha_{COCl_2} p_{COCl_2}^G)$$

Ecuación C. 13

$$p_{COCl_2}^E = \beta_{COCl_2} p_{COCl_2}^G / (k_2 + \beta_{COCl_2} \alpha_{COCl_2})$$

Ecuación C. 14

$$k_3 S \phi_2 p_{CO}^E = \beta_{CO} S \phi_2 (p_{CO}^E - \alpha_{CO} p_{CO}^G)$$

Ecuación C. 15

$$p_{CO}^E = \beta_{CO} p_{CO}^G / (k_3 + \beta_{CO} \alpha_{CO})$$

Ecuación C. 16

si definimos las siguientes expresiones

$$A_1 = \frac{k_1 \beta_{Cl_2}}{k_1 + \beta_{Cl_2} \alpha_{Cl_2}}$$

Ecuación C. 17

$$A_2 = \frac{k_2 \beta_{COCl_2}}{k_2 + \beta_{COCl_2} \alpha_{COCl_2}}$$

Ecuación C. 18

$$A_3 = \frac{k_3 \beta_{CO}}{k_3 + \beta_{CO} \alpha_{CO}}$$

Ecuación C. 19

Combinando las ecuaciones anteriores, la ecuación C.10 queda:

$$\frac{S}{r_m} = \frac{1}{A_1 p_{Cl_2}^G + A_2 p_{COCl_2}^G} \left(1 + \frac{A_1 p_{Cl_2}^G}{A_3 p_{CO}^G} \right) + \frac{1}{k_4} \quad \text{Ecuación C. 20}$$

ya en terminos de las presiones parciales en la fase gaseosa de bulbo

Desafortunadamente, el artículo de Szabó no da valores para las constantes, su modelo considera que el tricloruro de aluminio se desprende de la partícula y se limita a estudiar los siguientes casos.

- Cuando sólo se introduce $COCl_2$ en inerte y hay reacción (ec. C.1) en fase gaseosa.
- Cuando sólo se introducen $Cl_2 + CO$ en inerte y hay reacción en fase gaseosa
- Cuando sólo se introduce $COCl_2$ en inerte y no hay reacción en fase gaseosa.
- Cuando sólo se introducen $Cl_2 + CO$ en inerte y no hay reacción en fase gaseosa.

Además de que no se dispone de datos sobre las constantes de transferencia de masa entre el bulto y la fase gaseosa ni de las constantes cinéticas de cada paso individual.

Referencias:

1. Szabó, I., Blicle, T., Ujhidy, A., Jelinko, R., *Kinetics of Aluminum Oxide Chlorination. 1. The Mechanism and Mathematical Model*, Ind. Eng. Chem. Res., 30, 292-298, 1991.
2. Szabó, I., Blicle, T., Ujhidy, A., Jelinko, R., *Kinetics of Aluminum Oxide Chlorination. 2. Experiments and Evaluation of Results*, Ind. Eng. Chem. Res., 30, 298-303, 1991.