UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES "ACATLAN"

GENERACION DE UN PROGRAMA PARA LA SIMULACION DEL PROCESO DE COPOLIMERIZACION EN EMULSION

T E S I S

Que para obtener el Título de

LICENCIADO EN MATEMATICAS APLICADAS Y COMPUTACION

p r e s e n t a n:

AIDA LOPEZ BLANCO

ALMA LOPEZ BLANCO

México, D. F. 1993

UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso

INDICE

CAPITULO III
DISEÑO E IMPLANTACION DEL PROGRAMA.

CAPITULO IV
RESULTADOS.

INTRODUCCION

Durante los últimos años, la investigación tecnológica de los polímeros se ha venido desarrollando a gran escala. Este desarrollo responde a la necesidad de satisfacer los requerimientos que presentan los consumidores, en cuanto a propiedades físicas, químicas y/o mecánicas, las cuales dependen de la aplicación a la que se destinará el producto. La investigación se ha dirigido a desarrollar nuevos productos que satisfagan estas necesidades, así como a mejorar los procesos de producción, con el objetivo de lanzar al mercado productos de alta calidad y a precios competitivos con los demás productos nacionales y extranjeros.

Para llevar a cabo el mejoramiento de estos productos, se cuenta con la experimentación en una planta piloto y en algunos casos con el apoyo de algún simulador. Se ha comprobado que haciendo uso simultáneo de un simulador y de la experimentación en planta piloto, se ahorran recursos, ya que generalmente los simuladores consumen menos tiempo que lo que llevaría en planta y por otro lado no se hace ningún consumo de materia prima y se eliminan los costos de producción que son mayores al costo del tiempo de cálculo del simulador del proceso. Existen en la práctica simuladores creados para casos específicos, los cuales son desarrollados por los investigadores que estan trabajando en tal caso, sin embargo, no existe un simulador que pueda ser usado para varios problemas que sea confiable y que se pueda usar fácilmente.

Por tal motivo decidimos crear una herramienta que permitiera de una manera fácil y confiable jugar con componentes del proceso a fin de reducir costos en la investigación y que fuera fácil de usar para diferentes investigaciones.

La simulación de procesos de polimerización que involucran varios componentes, es una herramienta muy valiosa para la creación de nuevos materiales que requieren de propiedades físicas a la medida, ya que permite utilizar con fines prácticos todos los fundamentos teóricos que se conocen respecto a los mecanismos de polimerización, además de poder explicar los complejos fenómenos que se observan experimentalmente. Aunque a la fecha no es posible justificar teóricamente todos los fenómenos observados, se ha llegado a un punto en el que un simulador puede predecir los efectos e iteracciones de las variables del proceso y de algunas variables directamente relacionadas con la calidad de los materiales.

Para iniciar este trabajo el primer paso es introducirse a lo que es el análisis y simulación de procesos químicos y a conocer el mecanismo cinético del proceso de

copolimerización en emulsión. Para esto el capítulo uno describe de la manera más sencilla como funciona y en que consiste el proceso a fin de familiarizarse con términos y entender el proceso a efectuar en el simulador y así llegar a la solución. Este capítulo también describe las ecuaciones que forman el modelo seleccionado para la simulación y su solución.

Una vez establecidos los conceptos básicos y planteado el modelo, se procede a hacer un estudio de las ecuaciones que se involucran en el modelo a fin de hacer la elección adecuada de los métodos numéricos que las resuelvan. El capítulo dos es una selección de los posibles métodos a utilizar de ecuaciones diferenciales ordinarias, ecuaciones no lineales y de métodos de optimización sin restricciones, que son las que se involucran en el modelo. Esto para garantizar que los métodos empleados son los adecuados y que el resultado sea confiable.

Además de crear un programa confiable se necesita que éste sea amigable y que sea fácil de mantener. Hay que señalar que hay dos posibles usuarios, uno que solamente hará experimentos con el modelo para luego experimentar en planta piloto y otro que tratará de encontrar características especiales en el producto. Así el siguiente paso es el de conocer cuales son los requerimientos de los usuarios a fin de llegar a diseñar un programa que utilice técnicas de programación permita fácil manejo tanto del código, como del programa mismo. Para esto el capítulo tres explica el concepto de ingeniería de software, los pasos que forman el ciclo de vida de un sistema, programación estructurada y las técnicas que se siguieron para llegar a un software confiable y amigable.

Una vez con el software listo e instalado es necesario hacer pruebas de que los resultados obtenidos son los esperados y para esto se realizan diferentes corridas de prueba.

Por último en el capítulo cuatro se presentan los resultados que se obtuvieron al hacer el estudio de las ecuaciones y de su solución. También se presentan los requerimientos de los usuarios que junto con la teoría del capítulo ya mencionada dieron origen al código realizado. Se presentan gráficas, tablas y los resultados que arroja el simulador. Además de esto se demuestra la capacidad del simulador comparando los resultados de éste con resultados publicados por otros simuladores en diferentes artículos.

CAPITULO UNO

MODELO MATEMATICO

DEL PROCESO

I.1 ANALISIS Y SIMULACION DE PROCESOS.

El rápido desarrollo del análisis y simulación de procesos ha puesto en uso gran número de términos y de conceptos, tales como modelo, sistema, variable, etc., que han sido utilizados en notaciones muy diferentes. El análisis de procesos se refiere a la aplicación de métodos científicos al reconocimiento y definición de problemas, así como al desarrollo de perocedimientos para su solución; en una forma más concreta esto quiere decir: especificación matemática del problema para la situación física dada, análisis detallado para obtener modelos matemáticos, síntesis y presentación de resultados para asegurar la total comprensión.

Desde un punto de vista muy general, al análisis y simulación de procesos presenta las ventajas que se señalan a continuación:

- Es posible estudiar procesos existentes de una forma más rápida, económica y completa que en la planta real.

- Se pueden ensayar intervalos extremos de las condiciones de operación, que pueden ser impracticables en una planta real.

- Se pueden introducir nuevos factores o elementos de un sistema y suprimir otros, lo que permite ensayar hipótesis antes de llevarlas a la práctica. Y también la repetición de experimentos.

- Se puede ensayar la sensibilidad de los costos.

- Se puede examinar la estabilidad de sistemas frente a diferentes perturbaciones.

Por estas razones, se puede mencionar que el análisis y simulación de procesos constituye un elemento muy importante para tomar una decisión.

La unidad de los conceptos: Modelo Matemático, Métodos Numéricos, Computación, se han convertido en una poderosa herramienta de cálculo y predicción denominada simulación.

En los problemas de simulación, la construcción del modelo matemático, es una de las etapas más complicadas y de mayor responsabilidad. La experiencia muestra que en muchos casos la elección correcta del modelo, significa resolver más de la mitad del problema. Para que la simulación sea exitosa, es de enorme importancia la comprensión del problema, es decir que se tengan los conocimientos suficientes sobre el fenómeno a estudiar y además alguna experiencia y aplicación de los métodos matemáticos en el

ámbito de investigación.

El modelo matemático nunca es idéntico al fenómeno que se considera, no transmite todas sus propiedades y particularidades. Basado en simplificaciones e idealizaciones, el modelo es la descripción aproximada del fenómeno. Por esta razón, los resultados que se obtienen al analizar el modelo siempre tienen respecto de éste un carácter aproximado. Su precisión está determinada por el grado de correspondencia semejanza adecuada entre el modelo y el fenómeno. La precisión y fiabilidad de los resultados es uno de los problemas más delicados de las matemáticas aplicadas. Este problema se resuelve de la forma más sencilla, cuando las leyes que determinan la conducta y propiedades del fenómeno son bien conocidas y existe una gran experiencia práctica de su aplicación. Entonces, a priori (antes del experimento, aquí antes de resolver el problema matemático) es posible apreciar la exactitud de los resultados que genera el modelo.

Una situación más complicada surge cuando nuestros conocimientos sobre el fenómeno que estudiamos son insuficientes. En tal caso, al construir el modelo matemático, es necesario hacer suposiciones adicionales que tienen carácter de hipótesis. Las deducciones obtenidas como resultado de la investigación de dicho modelo hipotético tienen carácter convencional con relación al fenómeno que se estudia. Para él, son ellas justas en el grado en que son correctas las suposiciones iniciales. Para la verificación es preciso comparar los resultados de la investigación del modelo con toda la información que se posee acerca del fenómeno que se estudia. El grado de aproximación de los datos calculados y experimentales permite juzgar sobre la calidad del modelo hipotético, acerca de si las suposiciones iniciales son correctas o erróneas.

Así, el problema de la posibilidad de aplicación de cierto modelo matemático para el estudio del fenómeno, no es puramente matemático y no puede ser resuelto solo por métodos matemáticos. El criterio fundamental de veracidad es el experimento, la práctica, en el sentido más amplio de la palabra. El criterio de la práctica permite comparar diversos modelos hipotéticos y de ellos elegir aquel que es más sencillo y que, al mismo tiempo, en los margenes de precisión requerida, reproduce correctamente las propiedades del fenómeno.

De esta forma, podemos decir que nuestro modelo sirve o no sirve. El análisis nos proporciona la posibilidad de establecer para el modelo las condiciones de aplicación ligándolas con la gama de variación de parámetros del problema y la precisión requerida. Por otra parte, es claro que los modelos matemáticos permiten reducir la investigación de un fenómeno real "no matemático" a la solución de un problema matemático, dando así la posibilidad de emplear para su estudio el aparato matemático, perfectamente elaborado, en conjunción con toda la maquinaria de cómputo.

La elaboración del modelo del fenómeno a estudiar permite plantear el pro-

blema de estudio como matemático. Después de ello, surge la segunda etapa de la investigación, esto es, la búsqueda del método de solución del problema matemático planteado. Y es aquí donde entra nuestro trabajo, solucionar el problema matemático. Con frecuencia en matemáticas se tropieza con problemas cuya solución no se logra obtener en forma de una fórmula que ligue las magnitudes buscadas. Al hablar de tales problemas decimos que no se resuelven en forma explícita. Para resolverlo se tiende a utilizar cierto proceso infinito que converja hacia la solución buscada. Si dicho proceso está indicado realizando cierto número de pasos y terminando después los cálculos (no pueden continuarse infinitamente), obtenemos la solución aproximada del problema. Este proceso está ligado con la realización de cálculos de acuerdo con un riguroso sistema de reglas que se prefija en función del carácter del proceso y recibe el nombre de algorítmo.

El problema de aplicación de los algorítmos en los que se emplea un proceso infinitamente convergente, no consiste en el carácter aproximado de la solución sino en el gran volumen de cálculos necesarios. No es casual que tales algorítmos suelan ser llamados de cálculo. Los métodos para resolver problemas matemáticos, basados en ellos se llaman métodos numéricos. La extensa aplicación de los algorítmos de cálculo ha sido posible gracias a las computadoras. Hasta su aparición, los métodos numéricos se empleaban rara vez y debido a la extremada laboriosidad de los cálculos manuales, se utilizaban solo en casos verdaderamente sencillos.

Existe una diversidad de métodos numéricos, estos en realidad dependen del problema al que se les aplique, así encontramos que hay métodos numéricos para la solución de ecuaciones algebráicas, ecuaciones diferenciales ordinarias, problemas de optimización con o sin restricciones, entre otros.

Una vez que se han establecido las reglas para la solución del modelo, es necesario llevar este proceso a la computadora. En este paso, es de vital importancia conocer o tener alguna experiencia en lenguajes de programación y sistemas operativos para las computadoras. Al realizar una simulación de algun proceso es necesario que esta sea económica (en el sentido de tiempo de cómputo) y eficiente. De este modo la elección del lenguaje de programación está basado en satisfacer estos requerimientos. Existe una variedad de lenguajes de programación desde los más simples como BASIC clasificados de alto nivel debido a su parecido con el diálogo humano, hasta los de bajo nivel como el lenguaje C y Ensamblador. Entre esta gama de lenguajes de programación se debe de seleccionar el más adecuado, sin embargo ya que una simulación tiene que ver necesariamente con cálculos numéricos es requisto elegir un lenguaje que considere rapidez de cálculo y precisión, en este sentido el lenguaje BASIC es de antemano descartado debido a su lentitud de ejecución y además a su excesiva generación de códigos, de este modo, los lenguajes FORTRAN-77, Pascal y C suelen ser los más adecuados.

Aunado a esta elección de lenguaje debe tenerse en mente la capacidad del sistema de cómputo con que se cuenta. A partir del auge de las computadoras personales en los años 70's, la generación de software para estos sistemas ha crecido en forma indiscriminada, de tal manera que al realizar una simulación casi siempre se está pensando en su compatibilidad con las computadoras personales. No obstante esto no es una regla general ya que existen problemas que requieren una gran capacidad de memoria y es necesario programar una versión para míni o macro computadoras.

Así como es importante lo anterior también hay que contemplar que el llevar el modelo a la computadora implica el confeccionar el algorítmo de solución del correspondiente problema matemático y de realizarlo en la computadora. Es importante señalar que se tendran que manejar conceptos de diseño de sistemas, llevar a cabo los pasos del ciclo de vida de un sistema y todo lo que esto significa, como es el análisis de requerimientos, la definición de estos, el diseño la codificación, instalación y pruebas y finalmente la documentación y mantenimiento de dicho sistema. Así, las computadoras han hecho variar la actitud ante la aplicación de las matemáticas como método de investigación. Hoy en día, las computadoras son uno de los factores determinantes del proceso científico, por lo cual transcurre un intenso proceso de matematización, no solo en las ciencias naturales, sino también en las ciencias sociales. Los modelos matemáticos empiezan a utilizarse extensamente en química, geología, medicina, etc.

I.2 DESCRIPCION DEL PROCESO QUIMICO DE COPOLIMERIZACION EN EMULSION

I.2.1. GENERALIDADES Y DEFINICIONES

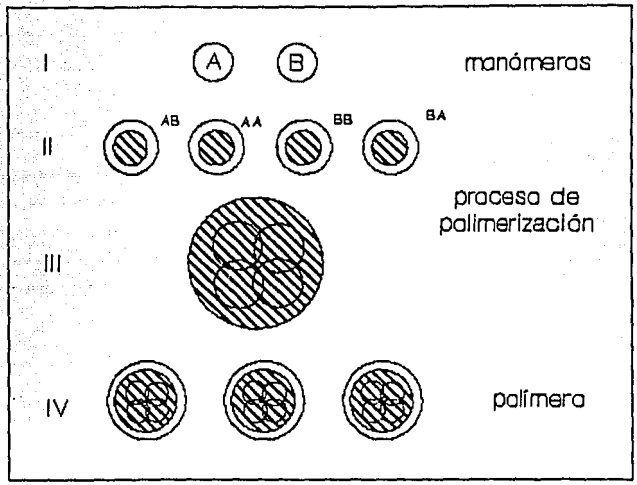
Hablamos de generar un programa simulador del proceso químico llamado copolimerización en emulsión utilizando técnicas y herramientas de matemáticas y computación adecuadas.

Para iniciar este estudio cabe señalar que partimos de un modelo matemático ya desarrollado, esto es porque el modelo debe ser diseñado por expertos en el área. La selección de éste modelo fué en base a la busqueda de ciertas características de los diferentes modelos que simulan este proceso.

Así, esta parte tiene como objeto dar un panorama amplio de lo que consiste físicamente el proceso modelado, para entender de que se hablará posteriormente, y describiremos el modelo que se seleccionó para la simulación. A lo largo de la descripción del proceso se explicarán diversas definiciones de términos específicos del área para así, poder comprender con más claridad el problema que se presenta.

La copolimerización en emulsión es uno de los muchos procesos químicos existentes para la producción de polímeros, un polímero es lo que comunmente conocemos como plástico. Los polímeros son macromoléculas construídas por una serie de eslabones de un gran número de pequeñas moléculas. Dichas moléculas se combinan unas con otras para formar moléculas de polímero, a esas pequeñas moléculas se les llama monómeros, y las relaciones por las cuales se combinan se llaman polimerizaciones.

Pueden haber cientos, miles, decenas de miles o más moléculas de monómero unidas juntas en una molécula de polímero. Frecuentemente las propiedades físicas de un polímero pueden ser mejoradas o modificadas por incorporación de diferentes grupos dentro de la cadena de polímero. Esto se puede efectuar durante la polimerización usando una mezcla de monómeros en vez de un monónomero, y cuando se usan dos monómeros el proceso se conoce como Copolimerización y el producto como copolímero. Nos referiremos indistintamente a polimerización y copolimerización.



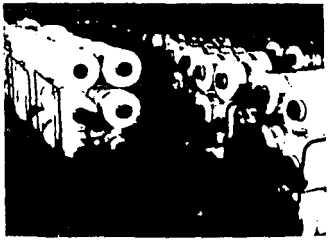
La polimerización en emulsión se empleo primero en gran escala en Estados Unidos para la producción de hule sintético de estireno-butadieno en los años 40's, cuando el aprovisionamiento de hule natural fué cortado durante la Segunda Guerra Mundial. Butadieno y estireno conjugados son actualmente polimerizados y copolimerizados en gran parte por emulsión.

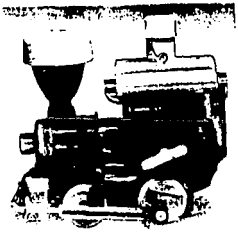
El producto final de la polimerización en emulsión se conoce como látex y con frecuencia se usa directamente como emulsión, es decir, no se separa el polímero del agua y otros componentes.

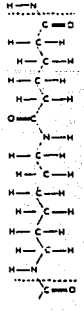
Por este medio también se obtienen infinidad de materias primas que se utilizan en varias ramas industriales, sirve para hacer productos tales como teléfonos, juguetes, partes de automóviles, pegamentos, muebles etc. Además como el producto final de este proceso se utiliza tanto sólido como líquido, puede usarse también como pintura. Es un proceso dominante para la polimerización comercial de acetato de vinilo, cloropreno, varios acrilatos y copolimerizaciones de butadieno con estireno y acrilonitrilo. Aunque

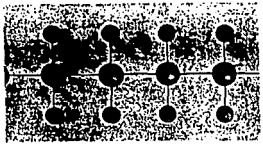
este proceso no es el único empleado, se usa también para metil metacrilato, cloruro de vinilo, cloruro de vinilideno y estireno.

En las siguientes figuras se muestran algunos de los usos que se le da al producto final del proceso químico, mencionadas anteriormente, se muestra una molecula típica de un plástico y el crecimiento en cadena que forma un plástico.









La polimerización puede hacerse por varias técnicas, una de ellas es la emulsión, pero cabe mencionar que también se puede polimerizar por suspensión o por masa. Cada técnica de polimerizar presenta ciertas ventajas, la emulsión presenta las ventajas que mencionamos a continuación :

- a) El producto de la polimerización puede utilizarse directamente o bien en forma sólida. Con apropiadas operaciones de mezclado, pueden ser utilizados como recubrimientos, acabados, ceras para pisos y pinturas principalmente.
- b) El estado físico del sistema de emulsión es una ventaja para controlar el proceso.
- c) Se obtienen altas conversiones.
- d) Se requieren bajas temperaturas de reacción.
- e) No se requiere del uso de solventes costosos, puesto que el medio de dispersión es el agua.
- f) Se facilita la transferencia de calor.
- g) Pueden obtenerse pesos moleculares altos a velocidades de polimerización elevadas.
- h) Las emulsiones se manejan con facilidad debido a su baja viscosidad.
- i) Además de la diferencia física entre polimerización en emulsión y otros procesos, hay una diferencia cinética notable.

Cabe mencionar que la polimerización en emulsión difiere fundamentalmente de la polimerización en suspensión y en masa, en éstas el iniciador debe ser soluble en el monómero, asimismo las velocidades de reacción y los grados de polimerización que se alcanzan son análogos, mientras que en los de emulsión suelen ser mucho mayores.

Una emulsión puede ser considerada como un sistema que contiene dos fases liquidas, una de las cuales está dispersa en un estado globular en la otra; uno de estos liquidos alrededor del cual estan los globulos es la fase continua, la otra es la dispersa.

Bajo un punto de vista práctico, las emulsiones se manejan mucho más fácilmente que las soluciones viscosas de polímeros obtenidas en la polimerización en suspensión o en masa y, por otro lado los grandes efectos térmicos que suelen aparecer a estos tipos de polimerizaciones y de muy difícil control a escala industrial, desaparecen en la polimerización en emulsión.

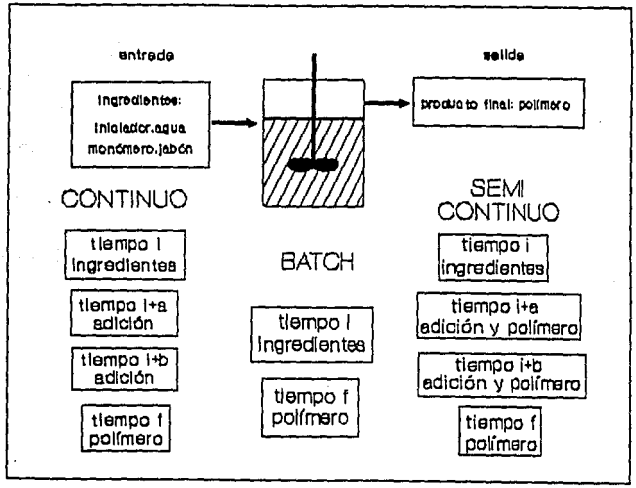
El producto de la reacción es un látex de polímero, esto es, una solución coloidal que posee esferas pequeñisimas de polímero (0.1 a 0.5 μ) suspendidas en agua. Este tipo de polimerización se puede efectuar mediante tres procesos diferentes:

a) Proceso por lotes, en el que se agregan al reactor todos los ingredientes desde

el principio. La mezcla se mantiene en agitación y se calienta hasta la temperatura en que se lleva a cabo la polimerización.

b) Proceso semicontinuo, en el que el monómero así como los demas ingredientes se agregan continuamente o a intervalos regulares durante el curso de la polimerización, para eliminar el calor de reacción que no puede ser transferido al sistema de enfriamiento del reactor, o bien para obtener una morfología específica en el polímero final.

c) Proceso continuo, en el cual los ingredientes se añaden continuamente a un reactor tabular o a un reactor de tanque agitado, utilizandose normalmente reactores en serie; en este caso el látex que puede estar parcial o totalmente polimerizado se extrae continuamente.



Los tres tipos de procesos mencionados se pueden efecuar induciendo la polimerización, esto es, añadiendo el monómero, el iniciador y el emulsificante a un poco de látex que se ha preparado previamemte, denominado semilla, con el fin de obtener mejor reproductibilidad y un látex estable del tamaño de partícula deseado.

Se han hecho diversas hipótesis acerca de la estabilidad del látex producido en este tipo de polimerización. Es necesario predecir el efecto de la adición de electrolítos, la velocidad de agitación del sistema y la temperatura entre otras cosa, para comprender las condiciones requeridas para desestabilizar el látex, esto es, para conseguir la coagulación y separar el polímero coagulado para su posterior tratamiento. Lo anterior implíca un problema industrial importante.

I.2.2. MECANISMO

COMPONENTES DE UNA FORMULACION

El proceso es como una receta la cual llamamos formulación, a cada ingrediente se le llama componente y el procedimiento a seguir es el mecanismo. La tabla siguiente muestra una formulación típica para un sistema de polimerización en emulsión :

monómero
medio de dispersión
emulsificante
agente de transferencia de cadena
iniciador

1. Monómeros que pueden ser entre otros: acetato de vinilocloropréno, acrilatos, butadieno-estireno, butadieno - acrilonitrilo, metilmetacrilato, cloruro de vinilo, cloruro de vinilideno, estireno. Pueden ser uno o dos monóneros.
2. Medio de dispersión , generalmente se usa el agua, todos los demás componentes se dispersan en ella en forma o estado emulsionado gracias al uso de un agente emulsificante. La relación entre agua y monómero es de 70/30 a 40/60.
3. Emulsificante, generador de micelas : generalmente surfactantes aniónicos - (jabones) como sales de sodio o postasio; sulfatos como lauril, sulfato de sodio, dodecil bencen. Las micelas son conjuntos de moléculas que se agrupan entre si, y se forman debido a que los jabones utilizados son poco solubles en agua.
4. Agente de tranferencia de cadena : para controlar el peso molecular, como los mercaptanos.
5. Iniciadores : solubles en agua como persulfato de potasio, persulfato de amonio, peróxido de hidrógeno.

UBICACION DEL LOS COMPONENTES DURANTE LA POLIMERIZACION

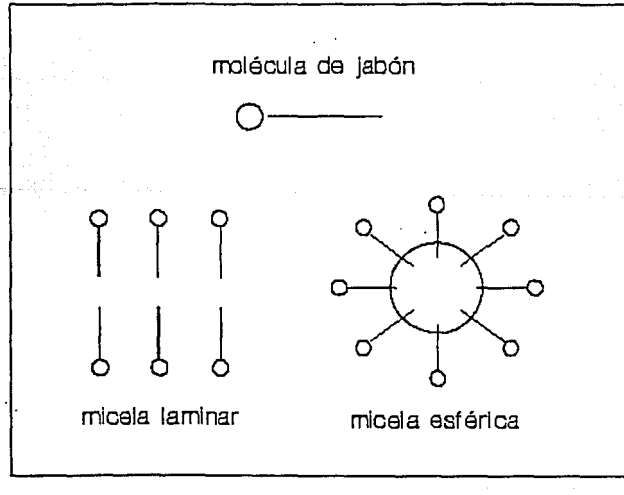
Cuando la concentración de un surfactante (jabón) sobrepasa su concentración micelar crítica, el exceso de moléculas de jabón se agregan (aglomeran) para formar pequeños sistemas coloidales llamados micelas.

A medida que la concentración de jabón sobrepasa la concentración micelar crítica, la solución inicial se transforma, o va adquiriendo, un estado coloidal.

Durante esta transformación se minimiza la energía libre de la solución (se libera calor) y se abate fuertemente la tensión superficial de la solución.

La forma de las micelas dependerá de la concentración de jabón, si la concentración de jabón es baja se producen micelas esféricas pequeñas y si la concentración de jabón es alta se producen micelas grandes y con forma de rodillos.

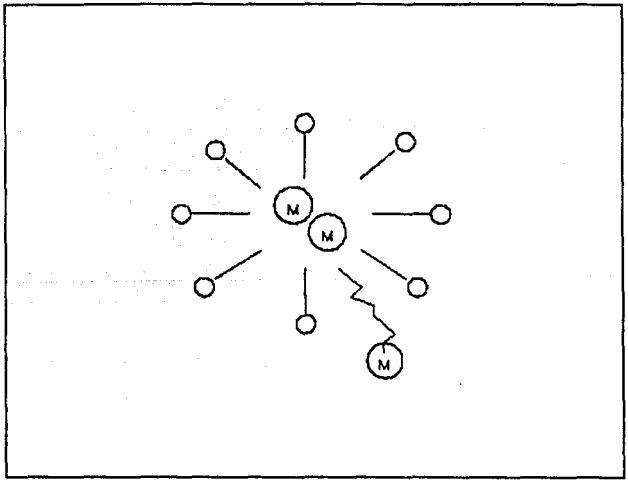
Las micelas generalmente tienen la siguiente forma :



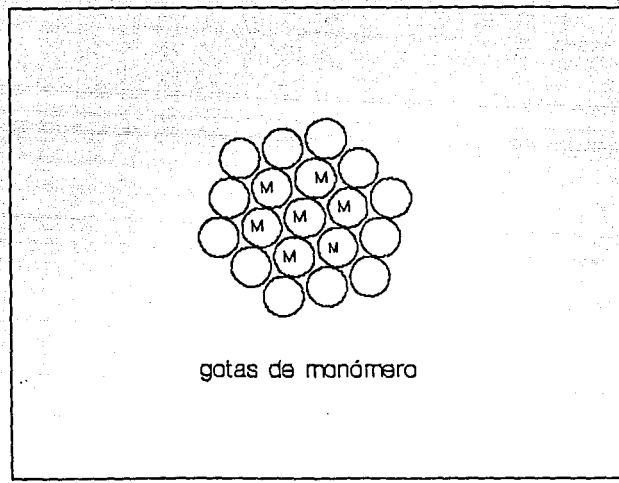
El número de micelas y su forma dependerán de la cantidad de jabón contra la cantidad de monómero : mayor cantidad de jabón implíca mayor cantidad de partículas pequeñas, esto es, aumenta el área superficial de las micelas (reducen su tamaño) al aumentar el jabón.

En este punto se tiene un sistema emulsionado y en primer lugar cuando se agrega un monómero insoluble, o escasamente soluble en agua, tan solo se disuelve una pequeña fracción y se va a la solución, o sea al agua. Generalmente la solubilidad en agua de los monómeros utilizados en polimerización en emulsión es baja.

Después de esto, otra pequeña fracción de monómero entra en las micelas :



La mayor parte del monónomero se dispersa y forma "gotas de monómero", cuyo tamaño depende de la intensidad de la agitación. Estas gotas de monómero probablemente se estabilecen al absorberse moléculas de jabón en su superficie.



Las gotas de monómero son un poco mayores que las de las micelas. Otra diferencia entre las micelas y las gotas de monómero es que las micelas tienen mayor área superficial.

SITIO DE LA POLIMERIZACION

En un principio el iniciador esta presente en la fase acuosa, que es donde se producen los radicales libres:

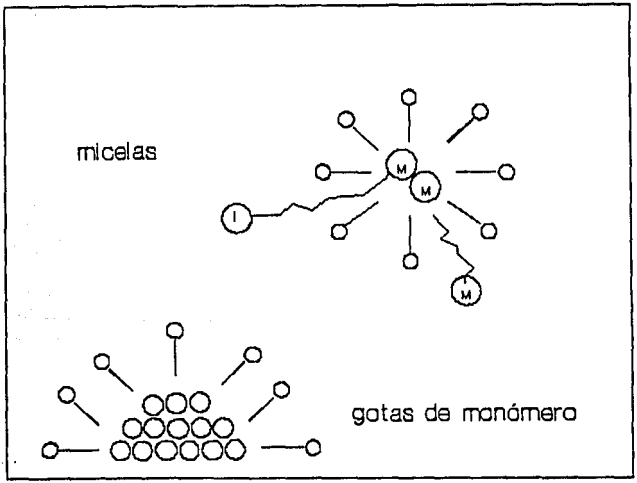
$$I \longrightarrow 2R^{\bullet}$$

El lugar de la polimerización no es en las gotas de monómero ya que los iniciadores utilizados son insolubles en los monómeros. Este tipo de iniciadores se denominan : oil-insoluble initiator. Esta es una diferencia de la polimerización en emulsión contra

la polimerización en suspensión, ya que en polimerización en suspensión la reacción ocurre en las gotas de monómero lo que no ocurre en la polimerización en emulsión.

La polimerización del monómero en solución, sin lugar a dudas se lleva a cabo, pero no contribuye significativamente, ya que la concentración de monómero es baja y los radicales propagantes (R^*)precipitarían fuera de la solución acuosa dando lugar a cadenas propagantes con tamaños muy pequeños.

De aquí que la polimerización se lleve a cabo casi exclusivamente en el interior de las micelas. Las micelas actuan como centro de encuentro entre los monómeros y el iniciador que es soluble en agua.

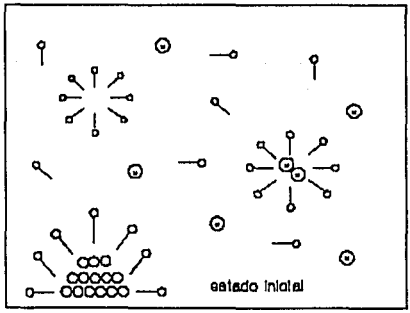


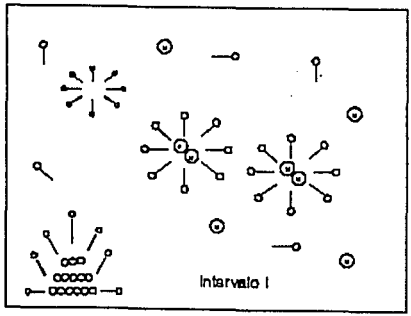
Las micelas también se ven favorecidas como sitio de polimerización debido a su alta concentración de monómero en comparación con el monómero que se encuentra en el agua.

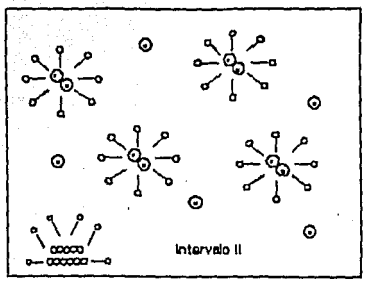
Las micelas tienen una relación superficie/volumen más alta en comparación con las gotas de monómero, esto es muy razonable ya que son más pequeñas. A medida

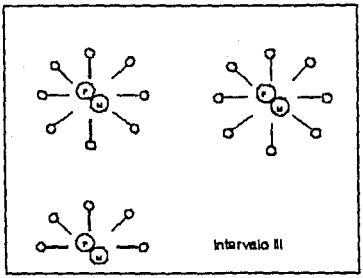
que avanza la polimerización, las micelas aumentan de tamaño debido a la continua adición de monómero al interior de ellas , desde la fase acuosa. Esta concentración de monómero en el agua se reestablece al diluirse monómero de las gotas de monómero.

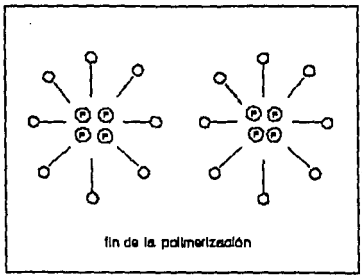
La siguiente es una representación típica de lo que ocurre en una polimerización en emulsión :






Intervalo II


Intervalo III


fin de la polimerización

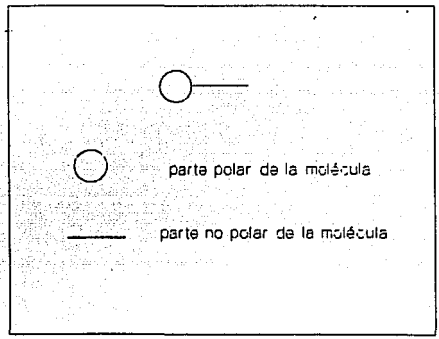
Se observa que el sistema está compuesto por tres tipos de partículas :

1) Gotas de Monómero

2) Micelas Inactivas (no hay polimerización)

3) Micelas Activas (hay polimerización)

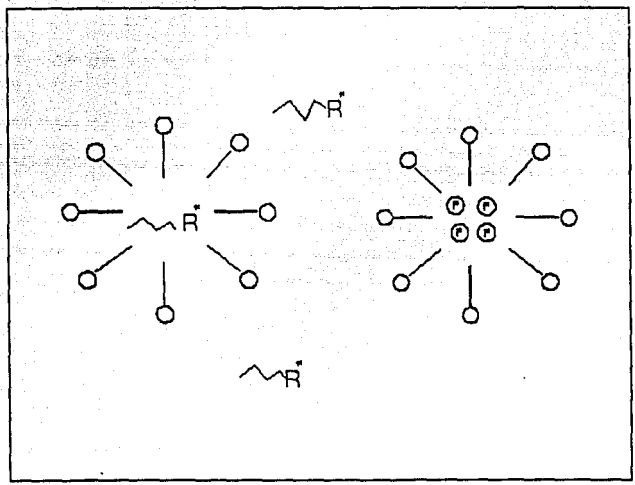
Las micelas inactivas pueden ser micelas donde no han entrado oligómeros o micelas sin monómero. Las últimas ya no son consideradas micelas, sino que pasan a ser partículas de polímero. La molécula de emulsificante se representa como:



El mecanismo por el cual se forman las partículas de polímero (nucleación de la partícula) se describe en dos procesos simultáneos:

1. Nucleación Micelar : Entrada de radicales R o radicales oligoméricos de la fase acuosa a las micelas.

2. Nucleación Homogénea : Los radicales oligoméricos se vuelven insolubles y precipitan en ellos mismos o en oligómeros muertos. Las especies precipitadas se estabilizan al absorber surfactante de la solución o de las gotas de monómero. Consecuentemente, absorben monómero y se convierten en un equivalente a las partículas de polímero formadas por nucleación micelar.



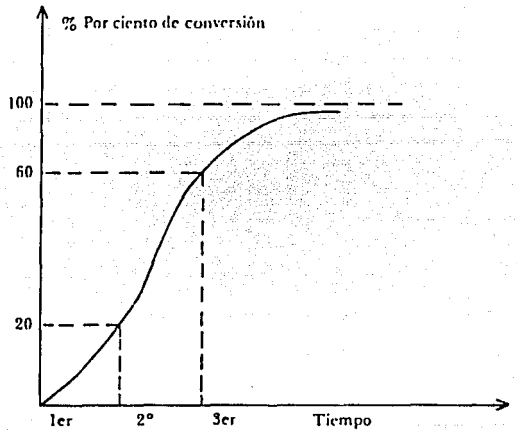
La nucleación micelar y heterogénea dependerá de la solubilidad en el agua del monómero y de la concentración de jabón. Así tenemos que mayor solubilidad del monómero en el agua y menor concentración de jabón favorecen la nucleación homogénea; y que menor solubilidad del monómero en el agua y mayor concentración de jabón favorecen la nucleación micelar.

EVOLUCION DE LA POLIMERIZACION

Se observan muchas tendencias en una gráfica de velocidad de polimerización contra conversión. De acuerdo con la siguiente figura pueden observarse tres intervalos, independientemente de la tendencia particular antes mencionadas.

En toda polimerización en emulsión pueden observarse estos tres intervalos, que

dependen del número de partícula N y la existencia de una fase separada de monómero. El número de partícula N puede definirse como la concentración de las partículas de polímero.



Curva de conversión

En (I) y (II) puede observarse la existencia de una fase independiente de monómero, pero no en (III). Esto porque al inicio de la reacción no hay muchas partículas de polímero ya que apenas se estan formando las micelas y las gotas de monómero. A medida que avanza la reacción, comienza a aumentar el número de parículas N, a medida van desapareciendo las gotas de monómero que son proveedoras de monómero para que entren a polimerizarse en las partículas de polímero.

N aumenta con el tiempo en (I) y permanece constante en (II) y (III). La entrada de monómero a micelas ocurre en (I) a medida que N aumenta. Las moléculas

de monómero se difunden a través de las micelas para reemplazar a aquellas que ya han polimerizado.

N se estabiliza en algún valor cerca del 0.1 concentración inicial de micelas. A medida que las partículas de polímero aumentan de tamaño y se vuelven contenedoras de polímero así como monómero, absorben más jabón que se encuentra en la solución, para mantener la estabilidad.

Debido a esto llega un momento en que la concentración de jabón es menor a la concentración micelar crítica (CMC). En este punto las micelas que han quedado inactivas se vuelven inestables y desaparecen al diluirse.

Al final de (I) o comenzando (II), todo o casi todo el jabón del sistema ha sido absorbido por las partículas de polímero. Como consecuencia, las gotas de monómero son inestables y se coalicionarían si se detuviera la agitación.

El intervalo (I) es el más corto de los tres, su duración esta en el rango de los 2-15 % de conversión. Sin embargo sera mayor si se cuenta con velocidades de iniciación bajas, ya que se requerirá mayor tiempo para alcanzar el estado de equilibrio del número de partículas.

Si se utilizan monómeros con alta solubilidad en el agua, (I) será un intervalo más corto que si se usan monómeros con bajas solubilidades en agua. Esto sería una explicación razonable para alcanzar el número de partículas estable más rápido, al provocar nucleaciones homogéneas y micelares simultaneamente.

La polimerización avanza en las partículas de polímero a medida que la concentración de monómero en las partículas se mantiene en equilibrio.

En el intervalo (II), la velocidad de polimerización puede ser constante o aumentar ligeramente con el tiempo (t). El comportamiento (t) es una consecuencia del efecto gel. Ya sabemos que las partículas de polímero aumentan de tamaño a medida disminuyen las gotas de monómero. El intervalo (II) finaliza cuando desaparecen todas las gotas de monómero.

La transición del intervalo (II) al (III) ocurre a bajas conversiones de reacción conforme aumenta la solubilidad en el agua del monómero y el hinchamiento de las partículas de polímero al introducirse los monómeros aumenta. Cuando se trata de monómero con bajas solubilidades en agua, esta transición de (II) a (III) ocurre en el 70-80 % de conversión. La transición se lleva a cabo bajo conversiones progresivamente bajas. Este decremento gradual en la conversión es proporcional a la disminución del monómero total del sistema. Para estireno y butadieno a 40-50 % de conversión. N permanece constante en (II) y (III), pero la concentración de monómero disminuye con el tiempo, ya que las gota de monómero se van acabando.

I.2.3 RESUMEN

Para concluir a continuación se presenta un resumen de lo que paso a paso es una polimerización en emulsión expuesto anteriormente :

1. Se agrega jabón. Este sobrepasa la Concentración micelar crítica y se aglomera formando micelas. En este punto decimos que el sistema esta emulsionado.

2. Se añade monómero y pasa lo siguiente :

a) Como los monómeros utilizados en polimerización en emulsión tienen baja solubilidad en agua, solo una pequeña fracción de monómero se solubilizará en en agua.

b) Otra pequeña fracción de monómero entra en las micelas de jabón.

c) La mayor parte del jabón formara gotas de monómero que se estabilizarán absorbiendo moléculas de jabón en su superficie.

3. El iniciador produce radicales en la fase acuosa, como los iniciadores utilizados en polimerización en emulsión no son solubles en compuestos orgánicos, los radicales formados a partir de ellos no pueden penetrar en las gotas de monómero. Sin embargo, los radicales, que están en la fase acuosa pueden actuar para iniciar la polimerización de la pequeña fracción de monómero que se encuentra también en el agua, pero ésta polimerización no es significativa debido a la baja concentración de M en el agua. No obstante se forman pequeñas cadenas propagantes llamadas oligómeros.

4. Los radicales si pueden entrar en las micelas, provocando la polimerización del monómero en ellas. A medida que avanza la polimerización, las micelas aumentan de tamaño, debido a que en el interior de ellas ya no sólo hay monómero, sino polímero que con el tiempo, va aumentando de tamaño.

5. Es lógico pensar que el monómero del agua se agote, en este punto, la concentración de monómero se reestablece al comenzar a diluirse monómero en las gotas de monómero.

6. En éste momento tendremos en el sistema tres tipos de partículas:

a) gotas de monómero

b) micelas inactivas (no hay polimerización todavía)

c) micelas activas (hay polimerización)

c') oligómeros

d) ya no se consideran micelas, sino partículas de polímero.

7. En la formación de las partículas de polímero intervienen los oligómeros de dos maneras:

a) entrando en las micelas (nucleación micelar)

b) precipitándose y absorbiendo moléculas de jabón en su superficie para estabilizarse (nucleación homogénea)

Por ambos mecanismos se obtendrán partículas de polímero equivalentes.

8. El aumento de partículas de polímero se expresará con la relación número de partículas de polímero/ ml que se conoce como N, número de partícula, y es una medida de la concentración de las partículas de polímero en la solución.

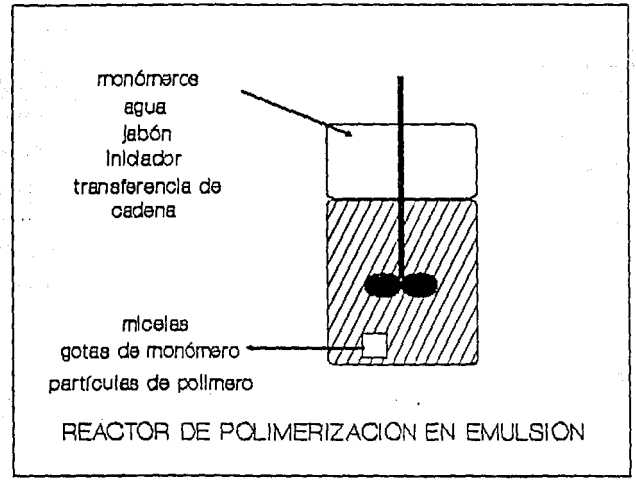
9. En toda polimerización en emulsión, si graficamos velocidad de polimerización contra tiempo, observamos tres etapas I, II y III:

a) I : Al inicio de la reacción comienzan a introducirse los monómeros a las micelas de jabón, y se forman las gotas de monómero, a medida que avanza la reacción comienzan a formarse las partículas de polímero. De aquí que en ésta etapa, N aumente drásticamente con el tiempo. Conforme ocurre esto, las gotas de monómero van desapareciendo. Las partículas de polímero comienzan a crecer, estan formadas por polímero, monómero y moléculas estabilizantes de jabón. Conforme aumentan de tamaño, requieren de más jabón para estabilizarse, mismo que comienzan a absorber de la solución. Pero llega un momento en que el jabón de la fase acuosa se agota hasta que su concentración es menor a la CMC. Entonces las micelas que han quedado inactivas, se vuelven inestables y se diluyen. Al final de la etapa I ó al comienzo de la II, todo o casi todo el jabón del sistema ha sido absorbido por las partículas de polímero. Como consecuencia, las gotas de monómero son inestables y se mantienen gracias a la continua agitación. El número de partículas comienza a alcanzar un estado de equilibrio.

b) II: La polimerización avanza en las partículas de polímero. La concentración de monómero en las partículas de polímero se mantiene en equilibrio por la difusión de monómero procedentes de la fase acuosa. La concentración de monómero en la fase acuosa se mantiene con la dilución de monómero de las gotas de monómero. Esta etapa finaliza cuando desaparecen las gotas de monómero. Dentro de esta etapa pueden observarse dos tendencias : i) la velocidad de polimerización constante y ii) ligero aumento en la velocidad de polimerización . Esto dependera del efecto gel.

c) III: Al crecer las partículas de polímero se hinchan de monómero. Conforme ocurre esto se experimenta un decremento gradual en la conversión. En éste momento se inicia la transición de la etapa II a la III. N permanece constante, pero la conversión se irá haciendo más lenta conforme el monómero final, ya en las partículas de polímero, comienza a agotarse.

La siguiente figura muestra un reactor típico para polimerización en emulsión y su receta típica :



REACTOR DE POLIMERIZACION EN EMULSION

I.3 MODELO MATEMATICO DE LA COPOLIMERIZACION EN EMULSION

Una vez que ya se ha explicado el mecanismo y demás aspectos desconocidos que habían sobre el proceso químico, en una forma general, y que ya se han establecido los conceptos generales, podemos ahora hacer la descripción del modelo matemático debidamente desarrollado por expertos en el área, que nos presenta la estructura matemática del proceso de polimerización en emulsión para así poder resolverlo.

La selección de dicho modelo se hizo en base a la comparación de los modelos existentes actualmente, es decir, que se hizo una busqueda de los principales modelos y se seleccionó el modelo que incluía todos los aspectos a considerarse en el mecanismo físico. Las principales diferencias que presenta este modelo en comparación con otros es que incluye todos los aspectos físicos que tiene el proceso, la capacidad que tiene de simular los tres tipos de reactores: batch, semi-batch y contínuo y que puede simular una operación isotérmica o no isotérmica.

Nos dimos cuenta que existen una gran variedad de modelos pero que la mayoria de los modelos con el fin de simplificar cálculos y asi tiempo, omiten partes que se dan por conocidas en base a la experimentación, o bien para hacerlos más simples dan por hecho algunos aspectos y los eliminan del modelo o bien no consideraban algunas partes que son de importancia por no requerirlo en su problema específico. Así tenemos que este modelo desarrollado por John R. Richards y John P. Congalidis del departamento de Productos de Polímero E.I. DuPont de Nemours Co. y por Robert G. Gilbert de la Escuela de Química de la Universidad de Sidney Australia, es el más completo que existe, ya que considera todas las fases que se dan en el mecanismo físico.

Este modelo combina la teoría de la nucleación gulativa de precursores homogéneos con los balances de materia y de energía para calcular el tiempo de evolución de la conversión, la composición de copolímero y el peso molecular en el sistema de emulsión.

La capacidad del modelo de polimerización en emulsión es demostrado por comparación de la predicción del modelo con datos experimentales de la literatura que de polimerización cubre estireno y estireno/metil-metacrilato.

Este modelo de polimerización en emulsión puede satisfactoriamente simular reactores continuos y batch sobre un amplio rango de adición de iniciador y de concentración de surfactante adicionado. Llamaremos al modelo EPM siglas en ingles de

Modelo de Polimerización en Emulsión. A continuación se presentan las ecuaciónes químicas que dan la cinética del proceso del cual se obtuvieron más de 100 ecuaciones, algúnas de las cuales se presentan más adelante. En el apéndice A se muestran todas las ecuaciones.

Descomposición de iniciador:

$$I \xrightarrow{k_i} 2I \tag{1}$$

Propagación en las fases :
acuosa w, y de polímero p

$$A_w \cdot + A_w \xrightarrow{k_{paa}} A_w \cdot \tag{2}$$

$$A_w \cdot + B_w \xrightarrow{k_{pab}} B_w \cdot \tag{3}$$

$$B_w \cdot + A_w \xrightarrow{k_{pba}} A_w \cdot \tag{4}$$

$$B_w \cdot + B_w \xrightarrow{k_{pbb}} B_w \cdot \tag{5}$$

Terminación en las fases :
acuosa w, y de polímero p

$$A_w \cdot + A_w \xrightarrow{k_{taa}} Q_w \cdot \tag{6}$$

$$A_w \cdot + B_w \xrightarrow{k_{tab}} Q_w \cdot \tag{7}$$

$$B_w \cdot + B_w \xrightarrow{k_{tbb}} Q_w \cdot \tag{8}$$

La estructura del EPM consiste de balance de materia, balance de concentración de número de partículas, balance de volumen, balance de energía, y el cálculo de algunas variables secundarias. Las siguientes son algunas de las ecuaciones usadas que se obtuvieron de las ecuaciones químicas anteriores.

BALANCE DE MATERIA

Se tienen 10 ecuaciones diferenciales que dan el tiempo de evolución de la concentración molar de cada especie en el reactor y asume que el reactor está perfectamente mezclado. Las especies consisten de : iniciador soluble en agua, electrolíto adicionado, monómeros A y B, agente de transferencia de cadena, surfactante adicionado, surfactante generado, monómeros A y B muertos, cadenas de polímero y copolímero muerto. Las 10 ecuaciones diferenciales son de la forma:

$$V_\omega \frac{dC_{i\omega}}{dt} = -C_{i\omega}\frac{dV_\omega}{dt} + F_i - C_{i\omega}Q_\omega - R_{i\omega}V_\omega \qquad (9)$$

donde los subindices cambian según la especie de que se trate, $C_{y\omega}$, C_{ae}, C_{be},C_{xe}, C_{se}, C_{ge}, C_{aqe}, C_{bqe}, C_{qe}.

VELOCIDADES DE REACCION

Las velocidades de reación que aparecen en el Balance de Materia anterior fueron calculadas usando ecuaciones de la forma siguiente respetando la especie de que se trate:

$$R_{pae} = \left[(k_{paa} + k_{xaa})C_{a\cdot p}C_{ap} + (k_{pba} + k_{xba})C_{b\cdot p}C_{ap}\right]V_p/V_e \qquad (10)$$

Las constantes cinéticas de velocidad fueron ajustadas para temperatura usando las expresión de Arrhenius; por ejemplo,

$$k_{paa} = A_{paa}e^{(-k_{paa}/RT_e)} \qquad (11)$$

CONCENTRACION EN AGUA Y EN PARTICULAS

El equilibrio de distribución de los dos monómeros y el agente de transferencia entre las partículas de látex, la fase acuosa y las gotas de monómero se describe usando coeficientes de partición. Para las especies $j = a, b, x$ el coeficiente de partición $K_{j\omega p}$ se define por la siguiente ecuación :

$$K_{j\omega p} = C_{j\omega}/C_{jp} \tag{12}$$

Si se presenta una fase de separación de monómeros, el coeficiente de partición puede ser definido entre las gotas de monómero y las partículas de polímrero. Lo cual deriva de la unión de los coeficientes de partición $K_{j\omega p}$ y K_{jmp}

$$K_{jmp} = C_{jm}/C_{jp} = \rho_j K_{j\omega p}/M_j C_{j\omega}^{sat} \tag{13}$$

La concentración de las especies j en las partículas de polímero se calculan del la siguiente ecuación :

$$C_{jp} = \frac{V_e C_{je}}{V_p + V_\omega K_{j\omega p} + V_m K_{jmp}} \tag{14}$$

La concentración de especies j en la fase acuosa esta dada por

$$C_{j\omega} = K_{j\omega p} C_{jp} \tag{15}$$

VOLUMEN EN LAS FASES ACUOSA Y DE PARTICULAS

Los volumenes de la fase acuosa y de partículas que aparece en la ecuación (19) junto con el volumen total de emulsión fueron calculados por las siguientes ecuaciones diferenciales :

$$\frac{dV_e}{dt} = Q_f - Q_e + \Delta Q_e = 0 \tag{16}$$

$$C_{\omega\omega}\frac{dV_\omega}{dt} = F_\omega - \frac{Q_e V_\omega C_{\omega\omega}}{V_e} \tag{17}$$

$$\frac{dV_p}{dt} = -V_p \rho_q - \frac{\phi_q \rho_q Q_e V_p}{V_e} + M_a R_{pac} V_e + M_b R_{pbe} V_e \tag{18}$$

CONCENTRACION RADICAL EN PARTICULAS

La concentración radical en las partículas es también necesaria para calcular las velocidades de reacción. EPM distingue entre dos tipos de especies de radicales libres en las partículas llamados, cadenas radicales finales con $A\cdot$ (concentración $C_{a\cdot p}$) y cadenas radicales finales con $B\cdot$ (concentración $C_{b\cdot p}$). La siguiente ecuación fue obtenida para $C_{a\cdot p}$, $C_{b\cdot p}$ en términos del número promedio de radicales por partícula $\bar{n}$:

$$C_{a\cdot p} = \frac{\bar{n} N_e V_e}{V_p N_A}\left(\frac{1}{1+\gamma_p}\right) \tag{19}$$

$$\gamma_p = \frac{C_{b\cdot p}}{C_{a\cdot p}} = \frac{(k_{pab} + k_{zab})C_{bp}}{(k_{pba} + k_{zba})C_{ap}} \tag{20}$$

BALANCE DE CONCENTRACION DE NUMERO DE PARTICULAS

En el desarrollo del EPM se asume que la dependencia de tamaño de los coeficientes de coagulación puede ser ignorado arriba de un cierto tamaño máximo ya que esto no afecta el producto final. La siguiente ecuación diferencial fue escrita para calcular el tiempo de evolución de la concentración de emulsión de las partículas de látex en partículas de cualquier tamaño :

$$V_e \frac{dN_e}{dt} = -N_e \frac{dV_e}{dt} - N_e Q_e + G_{ce} V_e \tag{21}$$

VELOCIDAD DE FORMACION DE PRECURSORES PRIMARIOS

Las siguientes ecuaciones fueron usadas para calcular la concentración de copolímero en la fase acuosa $C_{a\cdot\omega}$ y $C_{a\cdot\omega}$ que nos llevan a ecuación de formación de precursores primarios:

$$C_{a\cdot\omega} = \frac{-\lambda_2 + \sqrt{\lambda_2^2 - 4\lambda_1\lambda_3}}{2\lambda_1} \tag{22}$$

$$\gamma_\omega \equiv \frac{C_{b\cdot\omega}}{C_{a\cdot\omega}} = \frac{k_{pab} C_{b\omega}}{k_{pba} C_{a\omega}} \tag{23}$$

VELOCIDAD DE GENERACION DE PARTICULAS

Las partículas precursoras primarias crecen por propagación y coagulación hasta que alcanzan un tamaño coloidal estable. El modelo denota por v_k la concentración de particulas precursoras en donde el volumen es k veces el volumen v_1 . Cuando el número de partículas precursoras primaria en una partícula gana un valor crítico m este evento corresponde a la formación de partículas látex coloidalmente estables. La velocidad de generación de partículas látex G_{ce} :

$$G_{ce} = \frac{1}{2}\sum_{i=1}^{m-1} v_i \left(\sum_{j-m-i}^{m-1} B_{ij} v_j \right) + \frac{K_{vm-1} v_{m-1}}{v_1} - 2B_{m,m} N_r^2 \qquad (24)$$

NUMERO DE K FOLD PRECURSOR DE PARTICULAS

La siguiente ecuación diferencial ordinaria fué escrita para calcular la concentración de precursores $k - fold$ y contabilizar adición y muerte de monómero por coagulación, crecimiento por propagación y la formación de precursores primarios por nucleación homogénea. Hay $m - 1$ de estas ecuaciones y se calculan por :

$$\begin{aligned} & k = 1, \ldots, m-1 \\ & \frac{dv_k}{dt} = \frac{1}{2}\sum_{i=1}^{k-1} B_{i,k-1} v_i v_{k-i} - v_k \sum_{i=1}^{m-1} B_{i,k} v_i - B_{m,k} N_e v_k \\ & \quad - \frac{k_{vk} v_k - K_{vk-1} v_{k-1}}{v_1} + \delta_{k,1} G_{he} \end{aligned} \qquad (25)$$

COEFICIENTES DE COAGULACION

Los coeficientes de coagulación de Muller son calculados por un procedimiento que se basa en la teoría de estabilidad coloidal desarrollada por Derjaguin, Landau, Verwey y Oberbeek. Para llegar a obtener este coeficiente es necesario resolver antes una serie de ecuaciones para finalmente obtener el coeficente de coagulación. Unas de estas ecuaciones forman un sistema de ecuaciones no lineales :

$$\theta_g = \frac{b_g C_{g\omega}}{(1 + b_g C_{g\omega} + b_s C_{s\omega})} \qquad (26)$$

$$\theta_s = \frac{b_s C_{s\omega}}{(1 + b_g C_{g\omega} + b_s C_{s\omega})} \tag{27}$$

$$C_{g\omega} = \frac{C_{ge} V_e - (A_\nu \theta_g / a'_g N_A)}{V_\omega} \tag{28}$$

$$C_{s\omega} = \frac{C_{se} v_e - (A_e \theta_s / a'_s N_A}{V_\omega} \tag{29}$$

Teniendo esto, se calcula el máximo de la curva de energía potencial como sigue :

$$\Phi_{r\,max} = \max_s (\Phi_r(s)) = \max_s (\Phi_A(s) = \Phi_R(s)) \tag{30}$$

Y finalmente el coeficiente de coagulación, para cada partícula, usando la ecuación de Muller:

$$B_{ij} = B_{ji} = \left(\frac{2r_j k_B T_e}{3r_i \mu W_{ij}}\right)\left(1 + \frac{r_i}{r_j}\right)^2 \tag{31}$$

BALANCE DE ENERGIA

La ecuación diferencial para resolver el balance de energía, o sea la temperatura del reactor, fue calculada considerando el flujo de entrada, el de salida, temperaturas de reacción y la temperatura de la chaqueta de enfriamiento del reactor. Este balance de energía puede ser usado para calcular la temperatura del reactor en una operación no isotérmica.

$$V_e \rho_e c_e \frac{dT_e}{dt} = \rho_f c_f Q_f T_f - \rho_e c_e Q_e T_e + (-\Delta H_{pa}) R_{pae} V_e$$

$$M_a + (-\Delta H_{pb}) R_{pbe} V_e M_b - U_j A_j (T_e - T_j) \tag{32}$$

$$\frac{dT_e}{dt} = 0 \tag{33}$$

VARIABLES SECUNDARIAS

Mientras todas las variables primarias han sido calculadas otras variables secundarias también son resueltas, tales como la composición de polímero y la conversión de polímero. Las siguientes ecuaciones son usadas para su cálculo :

$$X_a = \frac{C_{aqe} M_a}{C_{aqe} M_a + C_{bqe} M_b} \tag{34}$$

$$X = \frac{C_{aqe} M_a + C_{bqe} M_b}{C_{aqe} M_a + C_{bqe} M_b + C_{ae} M_a + C_{be} M_b} \tag{35}$$

De esta forma se presenta el modelo de polimerización en emulsión que se resuelve más adelante.

Aclaramos que quizá muchas partes en la descripción del modelo no son muy claras, y por lo tanto, no es entendible, pero para el desarrollo de este trabajo no es necesario profundizar en ello basta con tener una idea general de lo que se va a simular. El propósito de presentar el modelo no es entenderlo, solamente es mostrar el tipo de ecuaciones y la gran cantidad de cálculos necesarios para resolverlo y así dar una idea de la complejidad de la solución. La nomenclatura del modelo esta en el apéndice B.

CAPITULO DOS

METODOS NUMERICOS

II.1. METODOS PARA LA SOLUCION DE ECUACIONES DIFERENCIALES ORDINARIAS.

II.1.1. INTRODUCCION

El comportamiento de muchos procesos físicos, particularmente de los sistemas que sufren cambios dependientes del tiempo, pueden describirse por ecuaciones diferenciales ordinarias. Así, métodos para la solución de ecuaciones diferenciales ordinarias son de gran importancia para ingenieros y científicos.

Ecuaciones diferenciales de orden n.

Consideremos la solución de ecuaciones diferenciales de orden enésimo de la forma

$$F(x, y, y', y'', y''', \ldots y^n) = 0$$

Una ecuación de este tipo se dice que es de enésimo orden porque la derivada de mayor indice de derivación es de indice n, y ordinaria porque sólo aparecen derivadas totales (no se presentan derivadas parciales, o alternativamente, sólo hay una variable independiente x). Una función $y(x)$ que satisfaga a esta ecuación, lo que implica que es al menos n veces diferenciable, se dice que es una solución de la ecuación. Para obtener una solución única (en general, hay muchas funciones $y(x)$ que satisfacen la ecuación) se necesita suministrar información adicional, es decir, valores de $y(x)$ y/o de sus derivadas en algunos valores especificados de x. Para una ecuación de orden n, normalmente n de tales condiciones son suficientes para determinar una solución única $y(x)$. Si todas las n condiciones se especifican para el mismo valor de x (por ejemplo x_0), entonces se dice que el problema es un problema con valores iniciales. Cuando intervienen más valores de x, se dice que se trata de un problema con valores a la frontera.

Una ecuación diferencial ordinaria de orden n puede escribirse como un sistema de n ecuaciones diferenciales ordinarias de primer orden mediante la definición de $n-1$

nuevas variables. Puesto que la mayor parte de las ecuaciones de orden superior (o un sistema equivalente de tales ecuaciones) pueden reescribirse en una forma semejante, y debido al tipo de ecuaciones que vamos a resolver, solo nos ocuparemos de la solución de numérica de ecuaciones de primer orden.

Aunque muchas ecuaciones diferenciales ordinarias pueden resolverse por técnicas analíticas bien conocidas, un número mucho mayor de ecuaciones diferenciales físicamente significativas no pueden resolverse de esa manera. Afortunadamente, pueden generarse numéricamente las soluciones de esas ecuaciones. En la siguiente parte se presentan algunos de los métodos numéricos que nos permiten resolver estas ecuaciones.

II.1.2. SOLUCION DE ECUACIONES DIFERENCIALES ORDINARIAS DE PRIMER ORDEN

Método de Taylor.

Un método aproximativo a la solución, numéricamente, se hace expresando la solución $y(x)$, respecto a algún punto de partida x usando un desarrollo de $y(x)$ en serie de Taylor:

$$y(x_0 + h) = y(x_0) + hf(x_0, y(x_0))$$

$$+(\frac{h^2}{2!})f\prime(x_0, y(x_0))$$

$$+(\frac{h^3}{3!})f\prime\prime(x_0, y(x_0))\ldots$$

Este método no es estrictamente un método numérico, pero en algunas ocasiones se utiliza en conjunto con esquemas numéricos, es de aplicabilidad general y sirve como introducción a otras técnicas que se estudiarán.

Cuando una serie de Taylor convergente se trunca, el error es sencillo de expresar. Simplemente se toma el siguiente término y se evalua la derivada en el punto $x = \xi, \quad 0 < \xi < h$, en lugar de hacerlo en el punto $x = x_2$. El término error de la serie de Taylor después del término h^4 es:

$$error = y^{(v)}(\xi) \qquad 0 < \xi < h$$

Sin embargo, esto no se puede calcular debido a que la evaluación de la derivada en $x = \xi$ es imposible con ξ desconocida, y aún limitándose en el intervalo $[0, h]$ es imposible debido a que las derivadas solo son conocidas en $x = 0$ y no en $x = h$.

Normalmente se trunca la serie de Taylor cuando la contribución del último término es despreciable con respecto a la cantidad de cifras decimales con las cuales se está trabajando. Sin embargo, esto es correcto solo cuando los términos sucesivos se hacen pequeños con suficiente rapidez, en algunos casos, la suma de muchos términos pequeños despreciables es significativa.

Método de Euler.

Se ha visto que el método de la serie de Taylor es un poco engorroso de aplicar si están complicadas varias derivadas, siendo difícil de determinar el error. Una crítica aún más significativa es que no resulta fácil codificar derivadas de funciones arbitrarias en un programa de computadora.

Es necesario saber que en la serie de Taylor el error será pequeño si el tamaño del paso h es pequeño. De hecho, si es lo suficientemente pequeño, solo pocos términos son necesarios para lograr una buena precisión. Se puede considerar el método de Euler como siguiendo esta idea para las ecuaciones diferenciales de primer orden. Supóngase que se escoge h lo suficientemente pequeño de manera que, se pueda truncar después del término de la primera derivada. Entonces

$$y(x_0 + h) = y(x_0) + hy'(x_0)$$

Se ha escrito la forma usual del término error para la serie de Taylor truncada.

Al usar esta ecuación, el valor de $y(x_0)$ está dado por la condición inicial y el valor $y(x_0)$ es evaluado de $f(x_0, y_0)$, dados por la ecuación diferencial $y' = f(x, y)$. Por supuesto será necesario utilizar este método iterativamente avanzando la solución a $x = x_0 + 2h$ después de que se ha encontrado $y(x_0 + h)$, después a $x = x_0 + 3h$, etc. Adoptando una notación de subíndices para los valores sucesivos y representando el error por la relación de orden, se escribe el algorítmo del método de Euler como:

$$y_{n-1} = y_n + hy'_n + O(h^2) error^*$$

II.1.3 METODO DE RUNGE-KUTTA.

Generalmente, no resulta práctica la solución de una ecuación diferencial por el desarrollo de Taylor directamente de la función objetivo si se tienen derivadas de orden superior al primero. Afortunadamente es posible desarrollar procedimientos de un paso que solo implican evaluaciones de primer orden de la derivada, pero que producen resultados de exactitud equivalente a las fórmulas de Taylor de orden superior.

Un avance más en eficiencia, se asegura con un grupo de métodos debidos a los matemáticos alemanes Runge y Kutta. Los métodos de cuarto orden de Runge-Kutta, son ampliamente usados en las soluciones de ecuaciones diferenciales por computadora. El desarrollo de esta técnica es complicado algebráicamente. Todos los métodos de Runge-Kutta tienen algorítmos de la forma:

$$y_{n+1} = y_n + h\phi(x_n, y_n, h)$$

en donde ϕ, que es la función incremento, es una aproximación adecuadamente escogida de $f(x,y)$ en el intervalo $x_i \leq x \leq x_{i+1}$.

Para dar una idea de como se desarrollan los métodos de Runge-Kutta, se muestran algunos pasos de la derivación de un método de segundo orden, ya que todos se desarrollan análogamente. Para la ecuación $y' = f(x,y)$,

$$y_{n+1} = y_n + ak_1 + bk_2,$$

$$k_1 = hf(x_n, y_n)$$

$$k_2 = hf(x_n + \alpha h, y_n + \beta k_1)$$

Se puede pensar que k_1 y k_2 son estimaciones del cambio en y cuando x avanza h unidades. El problema consiste en en diseñar un esquema para escoger los cuatro parámetros a, b, α, β. Se hace esto haciendo que la ecuación anterior concuerde con el desarrollo de la serie de Taylor:

$$y_{n+1} = y_n + hf(x_n, y_n) + \frac{h^2}{2} f'(x_n, y_n) + \ldots$$

Una forma equivalente es :

$$y_{n+1} = y_n + hf_n + h^2(\frac{1}{2}f_x + \frac{1}{2}f_y f)_n$$

Ahora se vuelve a escribir la ecuación substituyendo las definiciones de k_1 y k_2 :

$$y_{n+1} = y_n + ahf(x_n, y_n) + bhf[x_n + \alpha h, y + \beta hf(x_n, y_n)]$$

Entonces se tiene otra ecuación que sera igual a la primera que se planteo, si $a + b = 1$, $ab = \frac{1}{2}, \beta b = \frac{1}{2}$. Esta última ecuación que se obtiene es el método de Runge-Kutta de segundo orden.

Los métodos de Runge-Kutta de cuarto orden se usan más ampliamente y se derivan en una forma semejante. El siguiente es el algorítmo de cuarto orden :

$$y_{n+1} = y_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4),$$

$$k_1 = hf(x_n, y_n)$$

$$k_2 = hf(x_n + \frac{1}{2}h, y_n + \frac{1}{2}k_1),$$

$$k_3 = hf(x_n + \frac{1}{2}h, y_n + \frac{1}{2}k_2),$$

$$k_4 = hf(x_n + h, y_n + k_3).$$

II.1.4 APLICACION.

El problema planteado presenta un sistema de ecuaciones diferenciales de primer orden, son trece ecuaciones que resolver y se presentan en el capítulo uno.

Para esto, se utilizó un metodo de Runge-Kutta de orden cuarto que se explicó anteriormente. La subrutina utilizada forma parte de la biblioteca de subrutinas mátematicas IMSL programada en FORTRAN.

El método considera la solución de un sistema de ecuaciones ordinarias de primer orden simultaneas en las variables dependientes $y_1, y_2, \ldots, y_n$:

$$\frac{dy_1}{dx_1} = f_1(x, y_1, y_2, \ldots y_n),$$

$$\frac{dy_2}{dx_2} = f_2(x, y_1, y_2, \ldots y_n),$$

$$\vdots$$

$$\frac{dy_n}{dx_n} = f_n(x, y_1, y_2, \ldots y_n).$$

Con condiciones iniciales dadas en un punto común (x_0), esto es,

$$y_1(0) = y_{1,0},$$

$$y_2(0) = y_{2,0},$$

$$\vdots$$

$$y_n(0) = y_{n,0},$$

El algorítmo se aplica a cada una de la ecuaciones en paralelo en cada paso.

II.2 METODOS PARA LA SOLUCION DE ECUACIONES NO LINEALES

II.2.1 INTRODUCCION

Las raices de una ecuación no lineal $f(x) = 0$ generalmente no pueden ser expresadas en forma exacta. Por lo tanto para resolver las ecuaciones no lineales, debemos usar métodos de aproximación. Estos métodos son usualmente basados en la idea de aproximaciones sucesivas o en linearización. Tales métodos son llamados iterativos, esto es, comienzan con uno o más aproximaciones iniciales de la raíz, entonces producen una secuencia de puntos $x_1, x_2, x_3, \ldots$ los cuales presumiblemente convergen a la raíz descada. Con ciertos métodos, es suficiente con conocer el intervalo $[a, b]$ el cual contiene la raíz. Otros métodos requieren una aproximación inicial la cual se aproxima a la raíz deseada y estos métodos convergen más rápidamente. Por lo tanto, es adecuado comenzar con un método general y entonces cambiar a un método de rápida convergencia en el período final. Resolver un sistema de ecuaciones no lineales es un problema difícil, sin embargo muchas de los métodos para una sola ecuación son fácilmente generalizables.

Algunos de los métodos para resolver sistemas de ecuaciones no lineales, son generalizaciones del método de Newton, por lo cual, comenzaremos con este método para la comprensión del procedimiento. El objetivo de esta sección no es presentar los resultados teóricos de los métodos que se presentan, sino mostrar el procedimiento del método para solución al problema a resolver en este trabajo.

II.2.2. METODO DE NEWTON

El método de Newton puede ser generalizado a n dimensiones. Para $n = 1$, derivamos este método de la formula de Taylor

$$f(x) = f(x_k) + (x - x_k)f'(x_k) + O(|x - x_k|^2)$$

Abandonando el término cuadrático, obtenemos para $(f(x) = 0$ la iteración del

método:

$$f'(x_k)(x_{k+1} - x_k) + f(x_k) = 0 \qquad k = 0, 1, 2, \ldots$$

Análogamente, la formula de Taylor en n dimensiones es dada por

$$f(x) = f(x^{(k)}) + f'(x^{(k)})(x - x^{(k)} + O(\|x - x^{(k)}\|^2)$$

donde $f'(x)$ es una matriz de n x n, llamada matriz jacobiana, (denotada como J), con elementos

$$f'_{ij}(x) = \frac{df_i}{dx_i}(x), \qquad 1 \leq i, j \leq n$$

Esto da como resultado el método de Newton de n dimensiones

$$f'(x^{(k)})(x^{(k+1)} - x^{(k)}) + f(x^{(k)}) = 0$$

Estas son un sistema de ecuaciones lineales para $x^{(k+1)}$ y si $f'(x^{(k)})$ es no singular,es posible resolverlo por cualquier método de solución de sistemas lineales. En caso de que n sea muy grande y $f'(x)$ esparcida, se puede usar un método iterativo. El método de Newton en n dimensiones es de segundo orden.

II.2.3 METODO DE LEVENBERG-MAQUARDT

Dado un mapeo $F: D \epsilon \mathbf{R^n} \rightarrow \mathbf{R^n}$, si $x^k \epsilon D$ es una proximación de la solución x^* de la ecuación $f(x) = 0$. La idea básica de todos los métodos de linearización, consiste en construir alguna aproximación a

$$L: \mathbf{R_k^n} \rightarrow \mathbf{R^n},$$

$$L_k x = A_k(x - x^k) + f x^k,$$

$$A_k \epsilon L(\mathbf{R^n})$$

la cual coincide con F en x^k y usando solución de $L_k x = 0$ como una aproximación x^{k+1} de x^* . Hablamos de un método no singular de linealización si toda A_k, $k = 0, 1, \ldots$, es invertible, en este caso la iteración es especificada por

$$x^{k+1} = x^k - A_k^{-1} F x^k \qquad k = 0, 1, \ldots \qquad (a)$$

Es posible sustituir A_k por $A_k + \lambda_k I$ donde λ_k es un parámetro adecuado, este cambio sería de la siguiente forma:

$$x^{k+1} = x^k - (A_k + \lambda_k I)^{-1} F x^k \qquad k = 0, 1, \ldots$$

lo cual se llama regularización de la ecuación (a).

Ahora combinando el método de Newton y el de descenso más rápido, encontramos una aproximación inicial precisa para asegurar la convergencia, nos referimos al algorítmo de Levenberg-Marquardt.

El método de descenso más rápido determina un mínimo local para una función multivariada de la forma

$$G: \mathbf{R}^n \to \mathbf{R}$$

La conexión entre la minimización de una función de $\mathbf{R}^n$ a R y la solución de un sistema de ecuaciones no lineales se debe al hecho de que un sistema

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

$$\vdots$$

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

tiene una solución en $fx = (x_1, x_2, \ldots, x_n)^2$ precisamente cuando la función G definida como

$$G(x_1, x_2, \ldots, x_n) = \sum_{i=1}^{n} [f_i(x_1, x_2, \ldots, x_n)]^2$$

tiene un valor mínimo de cero.

II.2.4 APLICACION.

Para la solución de las ecuaciones no lineales de la (86) a la (89), en el modelo, utilizamos una subrutina de la biblioteca de subrutinas en FORTRAN para aplicaciones matemáticas, del paquete IMSL, la cual utiliza el algorítmo de Levenberg - Maquardt y la matriz Jacobiana.

Las bibliotecas de matemáticas tienen como finalidad facilitar las tareas de cálculo, y tienen las características de facilidad en su uso y de transportabilidad, el beneficio que proporcionan las bibliotecas es que reducen el trabajo de la programación del algorítmo y evita distraer la atención a la solución del problema, y además proporcionan la solución de manera óptima, porque estan especialmenente diseñadas para optimizar cálculos, facilitan la documetación porque tiene parámetros bien definidos.

II.3. METODOS PARA LA SOLUCION DE PROBLEMAS DE OPTIMIZACION SIN RESTRICCIONES

II.3.1. INTRODUCCION

La progración lineal es, sin duda, el mecanísmo más natural para formular una gran cantidad de problemas con el mínimo esfuerzo. Un problema de programación lineal se caracteriza, como su nombre lo indica, por funciones lineales de las incógnitas; el objetivo es lineal en las incognitas y las restricciones son igualdades o desigualdades lineales en las incógnitas. La popularidad de la programacion lineal estriba principalmente en la fase de formulación del ánalisis, más que en la fase de solución.

Dentro de la programación no lineal, pueden parecer que los problemas de optimización sin restricciones, están tan carentes de propiedades estructurales, que deberían excluirse de posibles aplicaciones como modelos útiles de problemas significativos. Sucede lo contrario por dos razones. Primera, puede argumentarse, de forma bastante convincente, que si se amplía el ámbito de un problema para tener en cuenta todas las variables de decisión relevantes, entonces no puede haber restricciones o, dicho de otra forma, las resticciones representan delimitaciones artificiales del ámbito de aplicación, y cuando se amplia el ámbito, las restricciones desaparecen. La segunda razón de que muchos problemas importantes puedan considerarse como si no tuvieran restricciones es que a veces es fácil convertir problemas con restricciones en problemas sin ellas. Por ejemplo el único efecto de las resticciones de igualdad no es otro que limitar las grados de libertad, sobre todo al hacer que ciertas variables sean funciones de otras. A veces se pueden caracterizar explícitamente estas dependencias, pudiendo determinarse un nuevo problema cuyo número de variables sea igual al verdadero grado de libertad. A pesar de los argumentos anteriores, muchos problemas prácticos se formulan como problemas con resticciones.

Una medida evidente de la complejidad de un problema de programación es su tamaño, medido en función del número de variables, incógnitas o del número de restricciones. El tamaño de los problemas que pueden resolverse con efectividad, ha ido en aumento con el avance de la tecnología de computación y con el progreso de la teoría. Sin embargo, con las posibilidades de computación actual, se pueden diferenciar en tres clases de problemas: problemas a pequeña escala, que tienen cinco o

menos de cinco incógnitas y restricciones, problemas a escala intermedia que poseen entre cinco y cien variables y problemas a gran escala con más de cien e incluso miles de variables y restricciones. Esta clasificación no es completamente rígida, sino que refleja, al menos aproximadamente, tanto el tamaño como las diferencias básicas de enfoque correspondientes a los problemas de distintos tamaños. Por regla general, los problemas a pequeña escala pueden resolverse a mano o con calculadora sencilla. Los problemas a escala intermedia se pueden resolver con un computador con lenguaje de programación matemática de uso general. Los problemas a gran escala necesitan lenguajes sofisticados que se sirven de una estructura especial y suelen requerir grandes computadores. Las características más importantes de una computadora de alta velocidad es su capacidad para efectuar eficientemente operaciones repetitivas, y para explotar ésta característica básica, la mayoria de los algorítmos diseñados para resolver grandes problemas de optimización son de naturaleza iterativa. Como norma, al buscar un vector que resuelva el problema de programación, se elige un vector inicial x_0 y el algorítmo genera un vector mejorado x_1 . El proceso se repite y se encuentra una sucesión de puntos más apropiados que tienden a un punto solución. Generalmente en los problemas de programación no lineal, la sucesión no alcanzará nunca el punto solución, sino que converge hacia él. El algorítmo termina cuando se obtiene un punto suficientemente próximo al punto de solución. A continuación presentamos algunos métodos para establecer las caracteristicas del algorítmo a utilizar para la solución de nuestro problema.

Como se mencionó en el planteamiento de nuestro modelo la función a optimizar es la máxima curva de energía potencial que se obtiene de la atracción de energía potencial ϕ_a y de la repulsión de energía potencial ϕ_r para cualquier par de partículas de latex. Por lo cual tenemos un problema de programación no lineal sin restricciones.

II.3.2. METODO DE NEWTON RAPHSON.

Consideramos conveniente presentar el método de Newton para mostrar en forma detallada el procedimiento a desarrollar ya que tienen una estructura fundamental los algorítmos descendentes que analizaremos. Se comienza en un punto inicial, se determina, de acuerdo a una regla fija una dirección de movimiento y despues se sigue esa dirección hacia un mínimo (relativo) de la función objetivo. En el punto nuevo se determina una nueva dirección, y se repite el proceso. Las diferencias fundamentales entre algorítmos radican en la regla mediante la cual se seleccionan las direcciones sucesivas del movimiento. Una vez que se ha hecho la selección, todos los algorítmos exigen movimientos hacia el punto mínimo de la recta correspondiente. Estas técnicas

de busqueda lineal, que no son otra cosa que procedimientos de resolución de problemas de minimización unidimensionales, constituyen la base fundamental de los algorítmos de programación no lineal, pues los problemas de dimensiones superiores se resuelven realizando una secuencia de busquedas lineales sucesivas.

Supongase que se va a minimízar la función f de una variable x, y supóngase que un punto x_k , donde se hace una medición, se pueden calcular los tres números $f(x_k)$, $f'(x_k)$ y $f''(x_k)$.

Entonces se puede construir una función q, que concuerde con f en x_k hasta la segunda derivada, es decir

$$q(x) = f(x_k) + f'(x_k)(x - x_k) + \frac{1}{2}f''(x_k)(x - x_k)^2$$

Entonces se puede calcular y estimar x_{k+1} del punto mínimo de f hallando el punto en el que se anula la derivada de q. Así, al hacer

$$0 = q'(x_{k+1}) = f'(x_k) + f''(x_k)(x_{k-1} - x_k),$$

resulta

$$x_{k+1} = x_k - \frac{f'(x_k)}{f''(x_k)}$$

que es la fórmula con la que se localiza el siguiente punto y se termina al alcanzar el error deseado.

II.3.3. METODO DE INTERPOLACION CUBICA

Este método fué desarrollado por Davidon y requiere que la función sea diferenciable. El método consiste en encontrar un valor óptimo de una variable α, denominada α^* , tal que la función

$$g(\alpha) = f(x + \alpha s)$$

obtenga un mínimo local, donde x y s son valores iniciales arbitrarios. El punto x es el punto de partida de la busqueda de los óptimos, mientras que s es la dirección de la búsqueda. Este mínimo local se obtiene en tres fases. Sea

$$g'(\alpha) = \frac{dg}{d\alpha}$$

Fase 1. Dados valores arbitrarios de x y s evalúe

$$g(\alpha) = f(x + \alpha s).$$

Fase 2. Evalué $g(\alpha)$ y $g'(\alpha)$ para valores de $\alpha = 0, 1, 2, 4, 8, 16 \ldots a, b$ donde b es el primer valor para el cual $g'(\alpha)$ es no negativo ó $g(\alpha)$ no ha decrecido. Resulta que el valor óptimo α^* se encuentra en el rango $a \leq \alpha^* \leq b$.

Fase 3. Se ajusta un polinómio cúbico tomando en consideración los valores $g(a), g(b) g'(a)$ y $g'(b)$. El valor mínimo α^* se representa en esta iteración por α_e donde

$$\alpha_e = b - \frac{g'(b) + u_2 - u_1}{g'(b) + g'(a) + 2u_2}(b - a)$$

$$u_1 = 3\left[\frac{g(a) - g(b)}{b - a}\right] + g'(a) + g'(b)$$

$$u_2 = (u_1^2 - g'(a)g'(b))^{\frac{1}{2}}$$

Si $g(a)$ y $g(b)$ no son menores a $g(\alpha_e)$, entonces se acepta que $\alpha^* = \alpha_e$ (valor aproximado). Si no, es decir, si $g(a) < g(\alpha_e)$ o $g(b) < g(\alpha_e)$, entonces

a) si $g'(\alpha_e) > 0$, se repite el mismo procedimiento en el intervalo $a \leq \alpha^* \leq b$ donde $a = a$ y $b = \alpha_e$. Regresar a la fase 3.

b) si $g'(\alpha_e) < 0$, se repite el mismo procedimiento en el intervalo $a \leq \alpha^* \leq b$ donde $a = \alpha_e$ y $b = b$. Regresar a la fase 3.

II.3.4 METODO DE LA SECANTE

Tomando como base el método de Newton-Raphson donde

$$x_{k+1} = x_k - \frac{f'(x_k)k}{f''(x_k)} \qquad (b)$$

Se sabe que la definición de la segunda derivada es :

$$f''(x_k) = \lim_{x \to x_k} \frac{f'(x) - f'(x_k)}{x - x_k}$$

Tomando $x = x_{k-1}$

$$f''(x_k) = \frac{f'(x_k) - f'(x_{k-1})}{x_k - x_{k-1}} \qquad (c)$$

Sustituyendo (c) en (b) tenemos

$$x_{k+1} = x_k - \frac{\frac{f'(x_k)}{f'(x_k) - f'(x_{k-1})}}{x_k - x_{k-1}}$$

$$x_{k+1} = x_k - f'(x_k) \frac{(x_k - x_{k-1})}{f'(x_k) - f'(x_{k-1})}$$

El algorítmo parte de las aproximaciones iniciales x_0 y x_1 y la secuencia - $x_2, x_3, \ldots$ se obtienen en forma recursiva, se termina el algorítmo cuando

$$|f(x_k) - f(x_{k-1})| < \epsilon$$

donde $\epsilon > 0$ es una tolerancia arbitraria.

II.3.5 APLICACION

Para la solución de la función de maximizar utilizamos una subrutina de la biblioteca de matemáticas anteriormente mencionada, pero para optimización de funciones sin restricciones, esta subrutina emplea el método de interpolación cúbica junto con el método de la secante.

Se inicia con un punto inicial y dos límites, inferior y superior. Si alguno de los tres puntos es el menor valor para la función, el algorítmo término con la solución. De otra forma, el punto que tenga el menor valor en la función se usará como punto inicial.

Para empezar se dice que x_c , el punto inicial, es evaluado en $f_c = f(x_c)$ la derivada $g_c = g(x_c)$ y el nuevo punto se define como $x_n = x_c - g_c$.

La función $f_n = f(x_n)$ y la derivada evaluada $g_n = g(x_n)$.

Si cualquier $f_n \geq f_c$ o g_n se toma el signo opuesto de g_c , entonces existe un punto mínimo entre x_c y x_n y un intervalo inicial es obtenido, de otra forma como x_c es guardado como el punto de menor valor de la función, se intercambian entre x_n y x_c

El método de la secante es entonces usado para obtener el nuevo punto:

$$x_s = x_c - g_c \left[\frac{g_n - g_c}{x_n - x_c}\right]$$

Sea $x_n \rightarrow x_s$ y se repite este proceso hasta que se encuentre un mínimo o el criterio de convergencia sea satisfecho.

Criterios:

Criterio 1. $|x_c - x_n| \leq \epsilon_c$

Criterio 2. $|g_c| \leq \epsilon_c$

donde $\epsilon_c = \{1.0, |x_c|\}, \epsilon$ error de tolerancia, ϵ_g gradiente de tolerancia.

Cuando la convergencia no es realizada, la interpolación cúbica obtiene un nuevo punto. La función y la derivada son evaluadas en el punto y por ende un intervalo pequeño el cual contiene un mínimo punto a escoger. Para asegurar que el método usado, es seguro, ese intervalo se reduce una mínima fracción del intervalo anterior. Otra interpolación cúbica se realiza y este procedimiento se repite hasta que se pare con algún criterio.

CAPITULO TRES

DISEÑO E IMPLANTACION

DEL PROGRAMA

III.1 INGENIERIA DE SOFTWARE.

III.1.1 INTRODUCCION.

La idea de construir una computadora nació de la necesidad humana de ser ayudado. Al principio, cuando surgieron las primeras computadoras los programas eran escritos en un código muy complicado y tardado, la programación era un proceso muy complicado y los programas eran muy difíciles de depurar y de mantener. Los problemas de software que entonces se tenían eran los que las limitaciones de hardware de esos tiempos marcaban. El hardware sufrió, con el tiempo, grandes avances tecnológicos que aunque lo hicieron más flexible el software no disminuyó en problemas, por el contrario crecieron. La complejidad de los sistemas de software se volvió inmanejable y las consecuencias naturales de ello fueron, principalmente, la falta de confiabilidad y la calidad pobre, retardos en los tiempos de entrega, costos excesivos en la producción y en el mantenimiento del producto de software. La situación realmente empezó a cambiar a mediados de los sesentas con el surgimiento de la Ingeniería de Software. Cabe señalar que en esta década se crearon tres generaciones de computadoras y al mismo tiempo se desarrollaron técnicas de multiprogramación y de tiempo compartido lo cual llevo a nuevas aplicaciones basadas en esta tecnología.

Cuando surgieron las primeras computadoras, éstas fueron usadas principalmente en el campo de la ciencia aplicada. Para el programador no era indispensable un conocimiento de informática especial, bastaba conocer un lenguaje de programación y la única dificultad era garantizar que el problema fuera correcto y eficiente esto además de que por lo general el programador era también el usuario. Los problemas que se presentaban antes de la década de los sesentas eran relativamente pequeños.

Sin embargo, conforme fué evolucionando este campo se volvió un problema más complejo el tratar de dar solución a problemas más grandes de aplicación comercial y problemas científicos complejos. Grupos de programadores tuvieron que trabajar en la producción de sistemas de programación con la necesidad de desarrollar productos de software de alta calidad que fueran usados por varios usuarios, ya no como en las primeras computadoras donde el programador era también el usuario. La especificación del problema y las demandas sobre el sistema cambiaban con frecuencia durante la fase de diseño, y posteriormente cuando el programa ya estaba en uso. Estos cambios en la

programación así como cada uno de los problemas de diseño a que se enfrentaban los programadores de sistemas, tales como la flexibilidad, la confiabilidad, la seguridad, la documentacion, el mantenimiento se convirtieron en los principales problemas de la producción de grandes sistemas de programación.

III.1.2. CONCEPTO DE LA INGENIERIA DE SOFTWARE.

Los objetivos primarios de la ingeniería de Software son: mejorar la calidad de los productos de software e incrementar la productividad y la satisfacción del trabajo de los que producen software. Uno de sus principios fundamentales es diseñar productos de software que minimícen la distancia intelectual entre el problema y la solución.

La Ingeniería de Software es interdisciplinaria. Se auxilia de diversas disciplinas tales como: las matemáticas para analizar y certificar algorítmos, ingeniería, administración, etc.

La Ingeniería de Software difiere de la programación tradicional en que la Ingeniería como técnica se usa para especificar, diseñar, implementar, validar y mantener los productos del software dentro del tiempo y de las restricciones de presupuesto establecidas por el proyecto. Además, está relacionada con los eventos administrativos que caen fuera del dominio de la programación tradicional.

Entre algunas definiciones podemos encontrar la de Boehm quien define la Ingeniería de Software como: "La aplicación práctica del conocimiento científico al diseño y a la elaboración de programas de cómputo y de la documentación asociada requerida para desarrollarlos, operarlos y mantenerlos."

Dennis la define como: "La aplicación de principios, habilidades y arte para el diseño y la elaboración de programas y sistemas de programación."

D.L. Parnas dice que: "Es programar al menos bajo una de las siguients condiciones: (1) Hay más de una persona involucrada en el desarrollo y/o en el uso del programa, y (2) se llegará a producir más de una versión del programa."

Para Pomberger es: "La aplicación práctica del entendimiento científico a la producción comercial y al uso de software confiable y eficiente."

Se puede observar que al analizar las diferentes opiniones o sugerencias sobre lo que es la Ingeniería de Software, existe poca diferencia entre ellas teniendo todas un común denominador: "producir programas de buena calidad". Por programas de buena calidad se pueden entender muchas cosas y regresar al problema anterior donde cada autor puede definirlos de diferente forma. En la siguiente parte de este capítulo trataremos de establecer cuáles deben ser las características de calidad del software.

III.1.3. CARACTERISTICAS DE LA CALIDAD DEL SOFTWARE.

Hasta el momento no existe una definición precisa de lo que significa calidad del software. A continuación se definirán algunas de las características primarias que debería tener un software de calidad:

Funcionamiento correcto.

Un sistema debe estar diseñado de tal forma que resulte correcto en todos los estados del desarrollo. En cada uno de los niveles de diseño, codificación o prueba, es necesario mostrar que el funcionamiento correcto es preservado por cualquiera de las nuevas adiciones al sistema. Un programa es correcto si concuerda con su especificación; de otro modo resulta incorrecto. Es decir, por funcionamiento correcto de un programa se entenderá que el programa satisface las especificaciones funcionales que son la base del desarrollo de programas.

Eficiencia.

Por eficiencia de un programa se entenderá la habilidad del programa para ejecutar la tarea con el uso óptimo de todos los recursos. Por recursos podemos entender espacio de memoria, tiempo de CPU, canales de entrada y salida, etc.

Robustez.

Se dice que un sistema de software es robusto cuando las consecuencias de errores en los datos de entrada o en el hardware, relacionados con una determinada aplicación, son inversamente proporcionales a la probabilidad de que ocurra un error en esa aplicación.

Efectividad.

Un sistema es efectivo cuando no solo realiza las tareas que tiene asignadas sino que también funciona durante un período largo.

Seguridad.

La seguridad es una medida de la probabilidad de que el sistema desarrollado por un usuario pueda destruir o hacer referencia, en forma accidental o intencional, a datos que son propiedad de otro usuario o interferir con la operación del sistema.

Adaptabilidad.

La adaptabilidad es la facilidad con la que se puede añadir al producto otras funciones.

Mantenibilidad.

Por mantenibilidad de un programa se entenderá la facilidad con la que los errores pueden localizarse y corregirse, y la facilidad con la que las funciones del programa pueden modificarse o expanderse. Esta definición indica también que la mantenibilidad de un programa depende de que sea legible, expandible y posible de probar.

La legibilidad de un programa depende de la forma en que ha sido representado, del estilo de la programación y su consistencia, de la legibilidad del lenguaje de programación, de la estructura del sistema y, más decisivamente, de la calidad de la documentación.

La expandibilidad de un programa depende de si es o no posible insertar los cambios deseados en los puntos lógicamente apropiados sin que se produzcan efectos no deseados.

Por la capacidad de prueba de un programa se entenderá la facilidad con que un programa permite la prueba en su ejecución y la localización de los errores que haya en él.

Generalidad.

La generalidad es una medida del número, potencia y alcance de las funciones desarrolladas por el usuario.

Portabilidad.

La portabilidad de un programa se entenderá la facilidad con la cual el programa puede ser implementado en diferentes sistemas de hardware. La portabilidad de un programa es, por lo tanto, una función de su independencia del hardware.

Confiabilidad.

El concepto de confiabilidad no debe confundirse con el hecho de que un programa funcione correctamente. Un programa correcto es aquel que ha sido probado para cumplir sus especificaciones. En contraste, un programa confiable no necesita ser correcto, pero si dar respuestas aceptables aún cuando los datos o el medio no cumplan con los supuestos acerca de ellos.

Todos los principios y prácticas para obtener confiabilidad pueden agruparse en cuatro módulos: en el primero se encuentran todos aquellos que se aplican para evitar las fallas; en el segundo se encuentran los que se encargan de detectar las fallas; en el tercero los que se ocupan de corregirlas; y finalmente en el cuarto, los que se encargan de su toleracia.

La Ingeniería de software pretende producir un sofware de buena calidad en el que estén presentes las características descritas anteriormente.

III.2 CICLO DE VIDA DE DESARROLLO DEL SOFTWARE.

III.2.1. INTRODUCCION.

Por lo general, cuando un sistema es muy grande sobrepasa la habilidad de cualquier individuo para enterderlo y construirlo. Para tener un mejor control de su desarrollo, es posible dividir el ciclo de vida del desarrollo de un sistema en las siguientes seis fases:

1.- Análisis de requerimientos.

2.- Definición de requerimientos (especificación del sistema).

3.- Diseño.

4.- Codificación.

5.- Instalación y Pruebas.

6.- Documentación y mantenimiento.

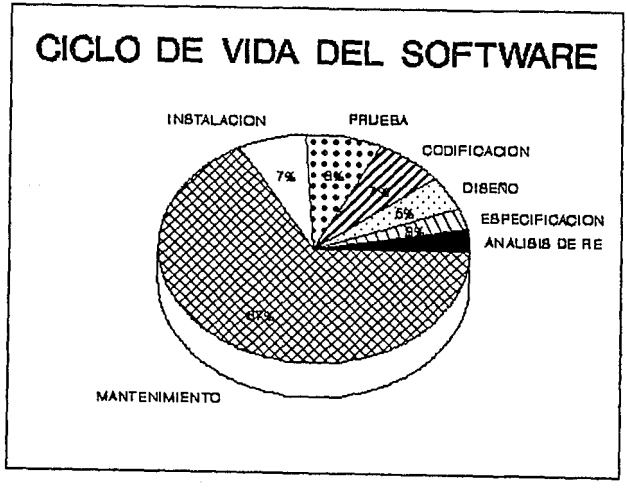
El propósito de la fase de análisis es determinar y documentar las funciones y pasos que se van a ejecutar y la naturaleza de las interacciones entre esas funciones. La fase de definición de requerimientos tiene como finalidad llegar a un contrato entre el usuario y el programador para determinar, con precisión, qué es lo que debe proporcionar el sistema. En la siguiente fase, es decir, la fase de diseño, se determina la forma en que serán implementados los requerimientos dados por la especificación del sistema. La fase de codificación tiene como tarea traducir el concepto, determinado en la fase de diseño, a un programa en un lenguaje de programación. La realización de pruebas del sistema intenta encontrar tantos errores en el producto de software como sea posible y garantizar que la implementación cumpla con la especificación. Finalmente, al terminar la fase de prueba, el sistema de software es instalado y puesto en uso. Posteriormente, durante la fase de mantenimiento se busca eliminar los errores que aparecieran durante la operación de implementar los cambios y expansiones al sistema. A estas seis fases en conjunto se les conoce como el "ciclo de vida del desarrollo del software".

Con el fin de direccionar varios aspectos del desarrollo del software y de su evolución, se han desarrollado diversos métodos, técnicas y herramientas. Es importante poder moverse de una fase del ciclo a otra en ambas direcciones y examinar el

progreso del trabajo en varios puntos intermedios. Esto permite identificar los problemas que surgen al inicio de los proyectos con el fin de tomar las acciones correctivas necesarias.

La experiencia muestra que frecuentemente cada una de las fases acarrea consecuencias en los resultados de las fases siguientes. En ocasiones, no queda claro sino hasta que se llega a la fase de diseño que la definición de requerimientos está incompleta; o es durante la fase de implementación o de prueba cuando se descubre donde se cometieron errores. La secuencia de las fases en el ciclo de vida del software se interrumpe con frecuencia y, en ocasiones el proceso de desarrollo debe ser reiniciado en una de las fases tempranas o, en el peor de los casos, desde la fase inicial.

La siguiente figura muestra las diferentes fases que conforman el ciclo de vida del desarrollo de software, así como la cantidad de tiempo promedio que se utiliza en cada una de ellas.



III.2.2 FASES DEL CICLO DE VIDA

1. ANALISIS DE REQUERIMIENTOS.

Durante esta primera fase, que curiosamente está ausente en la mayoría de los proyectos, se definen los requerimientos para llegar a una solución aceptable del problema. Su propósito es el de establecer cuáles son las necesidades del usuario con respecto de un producto en particular y, por supuesto, es deseable involucrar profundamente en la definición de los requerimientos tanto al ususario potencial del producto como al grupo que está encargado de desarrollar el software.

La computadora, únicamente, es una herramienta dentro de un complejo sistema hombre-máquina y el análisis de requerimientos se enfoca hacia la interfase entre ella y las personas que van a utilizarla. El análisis de requerimientos puede ayudar a comprender mejor el problema y los cambios posteriores que se le hagan al sistema, contribuyendo así a obtener una mejor solución, ya que, en general, la mayoría de los errores que ocurren en un sistema de software son ocasionados por la falta de entendimiento de las necesidades reales del ususario. Estos errores se originan cuando los requerimientos u objetivos se trasladan a las especificaciones externas. Desafortunadamente, poca gente pone suficiente énfasis en este proceso.

En términos de confiabilidad, el objeto es asegurar que los requerimientos del usuario se especifiquen de la manera más correcta y más precisa que sea posible y que el grupo de trabajo traslade esos requerimientos a un diseño del sistema con un mínimo de errores.

2. DEFINICION DE REQUERIMIENTOS.

Mientras que es en la fase de análisis de requerimientos donde se trata de determinar si se debe o no de usar una computadora, es en la fase de definición de requerimientos, también llamada especificación del sistema, donde se busca especificar que es lo que la computadora va a hacer.

Debido a que en esta fase se describe el ámbito de la solución, este documento debe dar una estimación inicial del tiempo de realización, del personal requerido y de los otros recursos que son necesarios para el proyecto.

La especificación describe el sistema en términos del problema que se va a resolver, indicando las funciones que lo conforman y las políticas que deben gobernar su comportamiento. La definición de requerimientos también debe imponer restricciones a la ejecución del sistema o restricciones de tipo económico y sugerir los atributos que

son deseables para los sistemas propuestos. Por lo tanto, una especificación puede ser caracterizada como una descripción orientada hacia el dominio del problema de un sistema de software.

La especificación define la función de un sistema desde el punto de vista del usuario y, por lo tanto, proporciona una liga entre éste y el diseñador del sistema. Los principales requerimientos del usuario sobre la especificación son la completez y la consistencia. Todas las especificaciones subsecuentes proporcionan una base para el diseño y la codificación.

La preparación de una definición de requerimientos que sea completa y que al mismo tiempo carezca de ambiguedad y que además, esté de acuerdo con las necesidades del usuario, es una de las tareas más difíciles de resolver en el desarrollo de un sistema. El análisis de los errores de software muestra que del 30 al 50 por ciento de todos los errores que surgen pueden atribuirse a una especificación incompleta, inconsistente o falsa.

Las partes del sistema que se especifican de una manera rápida y superficial, en general, son las que contienen más errores lógicos. Cualquier esfuerzo adicional que se realice durante la fase de especificación implicará un ahorro en el trabajo que se haga durante las fases de codificación, prueba e integración.

La descripción de las funciones del sistema es la parte esencial de la fase de especificación, comenzando con la descripción de las salidas deseadas, las entradas requeridas y el acoplamiento entre ellas.

Es de particular importancia que la definición de requerimientos contenga una descripción del ambiente en el que el nuevo sistema va a operar y una definición de la interfase entre este sistema y su ambiente. También debe contener una descripción de los límites superior e inferior para los valores de los datos de entrada y de salida. Sólo entonces será posible detectar los errores que aparezcan en los datos de entrada y verificar la verosimilitud de los resultados.

En resumen, el objetivo de la definición de requerimientos es establecer la especificación del sistema utilizando un metalenguaje. Esto impone restricciones al diseño y ayuda a establecer si las especificaciones inicales se cumplen. La definición de requerimientos define únicamente lo que es el sistema pero no como lo va a hacer.

3. DISEÑO.

En contraste con la fase de especificación, durante esta fase se describe el sistema en términos de la solución propuesta al problema y no en términos del problema

al cuál está dirigido. El diseño presenta esta solución como una colección de unidades de procesamiento conceptual o modulos, especificando los lineamientos para cada una de sus actividades individuales e indicando las iteraciones que se realizan entre ellos. Aunque tanto el diseño como las especificaciones pueden imponer restricciones y sugerir atributos deseables para el sistema propuesto, esas restricciones y atributos normalmente están relacionados directamente con las propiedades de la actividad e iteracción de los módulos.

Se usará la frase "proceso de diseño" para denotar la actividad de crear el diseño de un sistema basado en su especificación. Con el fin de minimizar el número de errores y de facilitar la producción de un diseño, el proceso deberá desarrollarse como una serie de pasos ordenados y verificables. La transición de la fase de especificación a la de diseño es aún muy abrupta y propensa a errores.

Cada metodología de diseño de software puede dividirse en dos partes: por un lado, un conjunto de las características deseadas de la solución y, por el otro, los lineamientos para el proceso de la solución. Al desarrollar una metodología de diseño normalmente se comienza definiendo las características deseadas de la solución y posteriormente se desarrolla el proceso de pensamiento requerido para llegar a la solución deseada.

Los programas diseñados de manera tradicional, en ocasiones, presentan exceso de complejidad, lo cual resulta en una falta de transparencia o propósito. Algunas de las causas más comunes de estos son las siguientes:

- Proliferación de instrucciones de transferencia de control;
- modificación del código, lo que puede hacer que el significado de un pedazo de texto cambie dinámicamente al tiempo de ejecución;
- existencia de banderas globales que se prenden y apagan en varios lugares sin explicación;
- ausencia u olvido de comentarios u otra documentación;
- nombres de identificadores que no concuerdan con el uso de la variable que están nombrando;
- inconsistencia al ejecutar subtareas
- uso de trucos inexplicables para optimizar la ejecución y,
- programas con una estructura monolítica o arbitraria.

Estas malas prácticas pueden eliminarse desarrollando programas con una estructura simple y consistente. La consistencia y la simplicidad de la estructura pueden obtenerse abordando la producción de programas de una manera sistemática y orde-

nada. El diseño sistemático tiene una ventaja adicional, proporciona una defensa contra el error humano. Al finalizar cada fase en el diseño de un programa, el programador puede revisar y eliminar cualquier error. Para que los programas sean simples de entender deben expresarse de tal forma que se encuentren relacionados con el problema que intentan resolver y no con la máquina en que se van a ejecutar.

Por lo tanto, el diseño de programas se debe llevar a cabo siguiendo tres fases:

1. Diseño de un algoritmo;
2. diseño de la forma en que se van a asociar los datos con los algoritmos que los van a representar y,
3. codificación de la representación en el lenguaje de programación que va a utilizarse.

Estas fases no son completamente independientes una de la otra y, por lo tanto, la separación no puede ser nunca completa en la práctica. El diseño del algoritmo y su representación constituyen lo que llamaremos "diseño del programa".

Idealmente, cualquier proceso de diseño debería ser capaz de garantizar :

a) Funcionamiento correcto. El programa debería de cumplir sus especificaciones con exactitud, es decir, para todos los posibles conjuntos de datos válidos el programa debería obtener la respuesta correcta.

b) Flexibilidad. En apariencia los cambios menores y razonables en el ambiente deberían realizarse sin mayor problema.

c) Completez. El programa debería de estar preparado para recibir todas las entradas inválidas o inesperadas y producir los mensajes de error apropiados antes de que se generen daños irreparables. No debe confiarse en salidas no válidas que se detecten después de la ejecución.

d) Eficiencia. Transparencia de propósito. El programa debería de ser sencillo de entenderse .

Los conceptos fundamentales del diseño de software incluyen: abstracción, estructura ocultamiento de información, modularidad, concurrencia, verificación y diseño estético.

Los esquemas de representación que se utilizan en el diseño de software son de importancia fundamental. Una buena notación puede clarificar las interrelaciones y las interacciones de interés, mientras que una notación pobre puede complicar e interferir con una buena práctica de diseño. Para representar los documentos de diseño se cuenta con: diagramas de flujo, diagramas de flujo de datos, mapas estructurales, diagramas

HIPO, especificaciones de procedimiento, seudocódigo, lenguaje natural estructurado y diagramas de flujo estructurado entre otros.

4. CODIFICACION.

En general, esta es la fase más fácil. Durante la fase de codificación de un sistema se lleva a cabo la traducción del diseño al programa. El programador debe asegurarse de que el diseño del sistema sea independiente del lenguaje en el que el sistema va a programarse posteriormente.

En la fase de codificación, la elección del lenguaje de programación resulta de gran importancia para el programador. No todos los lenguajes de programación son igualmente apropiados para la traducción del diseño a programa.

5. PRUEBAS E INSTALACION.

La calidad de un producto de software se distingue por la medida en la cual se satisface el funcionamiento correcto y la confiabilidad. Esto significa encontrar cuántos errores hay en el sistema total y cuántos errores graves ocurren durante el uso del producto de software.

La confiabilidad y el funcionamiento correcto del software, o por el contrario, la existencia de errores puede confirmarse solamente hasta que los resultados de las pruebas realizadas al programa a ser verificado, o el programa mismo, sean comparados con los criterios de aceptación que se han preparado a partir de los requerimientos del sistema.

El propósito principal de la fase de prueba es, primero asegurar que el sistema satisfaga las demandas, es decir, la definición de los requerimientos y segundo, descubrir tantos errores como sea posible. La experiencia muestra que la producción de software que esté libre totalmente de errores, en general no es posible. Por ello, las actividades de prueba que señalan la mayoría de los errores que pueden ocurrir son las más útiles.

Por error se entenderá, una desviación del comportamiento estipulado en la definición de los requerimientos. La causa de un error puede estar oculta en la especificación, en el diseño o en la codificación y no puede, si es robusto, encontrarse en la elección errónea de los valores de entrada al sistema. Está relacionada con la definición de requerimientos, con el diseño, con la codificación y con la fase de mantenimiento y

debe, también, forzar la verificación de la robustez. En general, la fase de prueba se realiza después de que han concluido las fases precedentes.

Los errores ocurren cuando alguno de los aspectos del desarrollo del producto de software está incompleto, inconsistente o incorrecto. Las tres categorías principales de error son: errores en los requerimientos, errores en el diseño y errores en la codificación, Los errores en los requerimientos son causados por la definición incorrecta de las necesidades del usuario, por fallas al especificar completamente los requerimientos funcionales y de ejecución, por inconsistencias entre los requerimientos y por requerimientos no factibles.

Los errores en el diseño se generan por fallas al traducir los requerimientos en estructuras de solución completas y correctas, por inconsistencia dentro de las especificaciones del diseño y por inconsistencia dentro de las especificaciones del diseño y los requerimientos.

Los errores en la codificación son los errores que resultan al traducir las especificaciones del diseño a código fuente. Estos errores pueden ocurrir en la declaración de los datos o en su referenciamiento, en la lógica del flujo de control, en las expresiones computacionales, en las interfases con los subprogramas y en las operaciones de entrada/salida.

Relacionadas muy cercanamente con la fase de prueba estan la verificación, la validación, la certificación y la depuración. Un sistema se valida probándolo para mostrar que actúa de acuerdo con sus especificaciones. Un sistema es verificado si puede probarse que el programa cumple con sus especificaciones. La certificación se refiere al proceso total de crear un programa correcto por medio de las técnicas de validación y de verificación. Mientras que la prueba es una actividad para descubrir errores, la depuración es una actividad para encontrar y anular fuentes de error.

Los objetivos de la verificación y de la validación son asegurar y mejorar la calidad de los productos del trabajo generados durante el desarrollo y la modificación del software. Los atributos de calidad de intéres incluyen: funcionamiento correcto, completez, consistencia, confiabilidad, utilidad usabilidad, eficiencia, conformancia a estándares y efectividad del costo total. La certificación no resolverá los problemas del software, aún cuando es una herramienta importante.

La fase de prueba debe abarcar la especificación del sistema, los modulos individuales, las interacciones entre los módulos, la integración de los modulos en el sistema total y la aceptabilidad del producto de software, es decir, su comportamiento durante un período de tiempo largo.

6. DOCUMENTACION Y MANTENIMIENTO.

La documentación de los productos de software puede dividirse de la siguiente forma:

1. Documentación para el usuario. Debe contener toda la información necesaria para aprender acerca del sistema de software y de su uso, sin necesidad de información adicional. Esta documentación incluye:

a) La descripción general del sistema,

b) el manual de instalación y el manual de usuario y,

c) el manual de operación.

2. Documentación del sistema. Debe contener todos los detalles necesarios para el entendimiento de la estructura y de la prueba del sistema. Debe servir de comunicación entre los programadores y de apoyo en el mantenimiento del sistema.

Debe describir todos los detalle de la elaboración del sistema de software, la estructura de los componentes individuales y las actividades de prueba. Debe contener toda la información necesaria para el entendimiento de la implementación total, la comunicación entre los programadores, la detección de errores y realizar las alteraciones y expansiones necesarias al sistema. Por lo tanto, dicho documento debe estar compuesto por todos los escritos elaborados en cada fase.

3. Documentación del proyecto. Debe contener todos los detalles del desarrollo del sistema desde un punto de vista organizacional y contable. Debe servir para supervisar el progreso del proyecto y para calcular los costos relacionados con éste.

El mantenimiento del software incluye todos los cambios que se hacen al software después de la terminación de los estados de desarrollo. Prácticamente, es imposible desarrollar un producto de software que no requiera mantenimiento. Ya se ha señalado con anterioridad que no es posible demostrar, sin ambiguedades, el funcionamiento correcto de un producto de software por medio de pruebas. Muchos errores se reconocen sólo hasta el momento en que se usa realmente el sistema y por ello estos errores sólo pueden eliminarse después de la fase de desarrollo. También durante la operación del sistema surgen nuevas demandas del usuario, lo cual implica nuevos cambios en el sistema.

Hay que considerar también que durante el tiempo de vida de un producto de software exitoso, con frecuencia se ha empleado más esfuerzo en la fase de mantenimiento que en cualquier otra de las fases del ciclo de desarrollo. Por ello los costos de mantenimiento pueden ser el factor más importante del ciclo de vida del software.

III.3 PROGRAMACION ESTRUCTURADA Y TECNICAS DE PROGRAMACION

III.3.1 INTRODUCCION.

La teoría de la programación estructurada tiene como idea principal el asegurar la existencia de una correspondencia única entre la notación estática del algoritmo y su comportamiento dinámico en la ejecución. De esta forma el control de flujo permanece claro, reduciéndose así la probabilidad de que ocurran errores durante el desarrollo del sistema lo que permite, al mismo tiempo, que la verificación del algoritmo sea más simple. Concretamente, en la programación estructurada se tratan de evitar las estructuras de datos no acotadas que resultan de un uso indisciplinado de ciertas estructuras de control como el GOTO.

La Programación estructurada puede verse como un conjunto de reglas, diseñadas para mejorar la legibilidad de un programa, reduciendo así las diferencias de estilo entre los programas escritos por varios individuos y mejorando la habilidad del programador para entender y modificar los programas existentes. La Programación Estructurada debe cumplir las siguientes reglas :

a) El código debe estar constituído por secuencias de las tres proposiciones básicas: secuencial, iterativa y condicional.

b) El uso de proposiciones GOTO debe evitarse siempre que sea posible. En particular, aquél que es el peor tipo de GOTO, el que regresa el control a una proposición anterior en el texto del programa.

c) El código debe escribirse siguiendo un estilo aceptable.

d) El código debe identarse correctamente en el texto, de tal forma que las interrupciones que aparezcan en la secuencia de ejecución puedan seguirse fácilmente.

e) Debe existir un único punto de entrada para cada módulo.

f) El código debe de estar segmentado físicamente en el texto del programa con el fin de mejorar la legibilidad. Las proposiciones ejecutables de un módulo deben aparecer en una página de listado únicamente.

g) El código debe representar una solución simple e íntegra del problema.

Aunque los siete puntos anteriores tratan de ilustrar los objetivos de la Programación Estructurada (complejidad mínima, claridad del pensamiento del programador y programas legibles), quedan algunos aspectos poco definidos. Por ejemplo el punto 3 habla de un estilo aceptable de programación pero no determina que significa "aceptable". En cuanto a la identación de un programa (punto 4), podríamos decir que depende mucho del sentido estético del programador, así como del tipo de estructuras y anidamientos que se esten utilizando. En el punto 6 se dice que las proposiciones de un módulo deben aparecer en una sola página del listado, pero no aclara de que tamaño debe ser la página (líneas por página). Estas reglas dejan mucho a juicio del programador.

Se pueden identificar como ventajas de la Programación Estructurada el incremento en la productividad del programador y claridad y legibilidad de los programas. Por otro lado la principal crítica que se hace a la teoría de la Programación Estructurada es que en esencia no es más que programación que evita cuidadosamente el uso de transferencias de control (proposición GOTO por ejemplo). Uno de los argumentos más comunes contra la programación estructurada es que conduce a escribir programas menos eficientes, es decir, el incremento en el énfasis que se pone en las llamadas a subrutinas, como una alternativa a la proposición GOTO, aumenta el tiempo de procesamiento del programa y puede, tal vez, añadir cantidades significaticas de requerimientos de memoria.

Basándose en los principios de la Programación Estructurada, la Ingeniería de Software puede lograr su objetivo de desarrollar un sistema de software confiable de buena calidad. Sin embargo es aconsejable seguir una metodología para el desarrollo del producto de software. A continuación se presentan algunas ideas sobre estas metodologías.

III.3.2 METODOLOGIAS DE DISEÑO.

El proceso de diseño implica desarrollar una visión conceptual del sistema, establecer la estructura del diseño, identificar los tipos de datos y donde almacenarlos, descomponer las funciones de alto nivel en subfunciones, establecer relaciones e interconexiones entre las componentes, desarrollar representaciones concretas de los datos y especificar los detalles algorítmicos.

Por ello es esencial que el grupo del diseño de software tenga un entendimiento conceptual claro de la naturaleza del sistema que va a construirse y que esté familiarizado con las herramientas y con las técnicas de las áreas de aplicación apropiadas. Es bastante común que un grupo de diseño este integrado por uno o más especialistas de

cada una de las áreas involucradas.

Desde el principio de esta década se han desarrollado muchas técnicas para el diseño de software. Algunas de estas técnicas se conocen como metodologías de diseño. A continuación se describiran brevemente dos de estas.

1. Técnicas TOP-DOWN y BOTTOM-UP.

En general las técnicas de diseño están basadas en las estrategias de diseño top-down y bottom-up. Al usar la aproximación top-down, la atención se enfoca, principalmente, en los aspectos globales del sistema total. Cuando el sistema progresa, el sistema se descompone en subsistemas y se da mayor consideración a los aspectos específicos. La retroalimentación es fundamental para el diseño top-down.

Como en esta técnica las decisiones de diseño se descomponen en niveles más elementales, podría parecer que una decisión de alto nivel resultara ineficiente o torpe, por lo que en ocaciones es necesario considerar de nuevo una decisión de alto nivel y reestructurar el sistema de acuerdo con ello. Con el fin de minimizar la retroalimentación, muchos diseñadores defienden una estrategia mixta que predominantemente es top-down, pero que involucra la especificación, al principio, de los módulos de más bajo nivel. Esto forma parte de la técnica de bottom-up.

En la práctica, el diseño de un sistema de software rara vez se lleva a cabo en una forma puramente top-down o bottom-up. Una estrategia predominantemente top-down tiene más éxito cuando existe un ambiente bien definido para el desarrollo del software. Por otro lado cuando el ambiente está bien definido la estrategia de diseño debe ser mixta o predominantemente bottom-up.

1.1. Técnica de arriba hacia abajo.

Uno de los principios más importantes para eliminar la complejidad es el principio de abstracción. El término de abstracción se utiliza cuando la solución de una tarea se considera sin conocer todos los detalles de su realización. La idea básica detrás de la abstracción es el diseño de arriba hacia abajo (top-down).

En el diseño de arriba hacia abajo el programador escribe primero una subrutina como una instrucción sencilla. A su vez esa instrucción se expande a mayor detalle. En cada nivel la función se va expandiendo con más y más detalle hasta que la descripción resultante es el programa fuente en algún lenguaje de programación.

En el diseño de arriba hacia abajo, el programa se estructura jerárquicamente y se escribe por refinamientos sucesivos. Cada refinamiento describe sus acciones refiriéndose a otros refinamientos en una manera de arriba hacia abajo.

En el desarrollo de arriba hacia abajo existen tres posibles "puntos críticos" para un sistema:

a) el inicio de la ejecución,

b) el foco de control,

c) la interfase con el usuario.

Al utilizar la técnica de arriba hacia abajo, el usuario conoce en una fase temprana las interfases de nivel alto en el sistema. Los cambios pueden realizarse durante el ciclo del desarrollo de una forma rápida y relativamente fácil. Sin embargo, la evaluación de la ejecución debe diferirse hasta que el sistema total se ensambla. Por el otro lado, el diseño de arriba hacia abajo y su implantación permiten la pronta demostración de las capacidades funcionales a nivel usuario.

La ventaja principal de esta estrategia es que la atención se dirige principalmente a las necesidades del usuario, a la interfase con él y a la naturaleza total del problema a ser resuelto.

1.2. Técnica de abajo hacia arriba.

Otra técnica de diseño es el llamado diseño de abajo hacia arriba (bottom-up). En esta técnica se procede de manera opuesta al método de diseño de arriba hacia abajo. Se comienza el diseño no al nivel de la especificación del sistema sino al nivel de ejecución. La idea básica es que el hardware y cada nivel de un producto de software pueden considerarse como máquinas abstractas. Por una máquina abstracta se entenderá un conjunto de operaciones fundamentales a través de las cuales se puede expresar la operación total de un sistema, o de un subsistema, en cualquiera de los niveles de abstracción. En el diseño bottom-up se comienza con las propiedades de una máquina concreta y se diseña una máquina abstracta después de otra, añadiendo sucesivamente las propiedades que se requieran hasta llegar a una máquina que sea capaz de cumplir las funciones que requiere el usuario .

En esta aproximación, el diseñador primero intenta identificar un conjunto de objetos primitivos, acciones y relaciones que darán una base para la solución del problema. Los conceptos de más alto nivel se formulan entonces en términos de estas primitivas. La estrategia de abajo hacia arriba requiere que el diseñador combine características proporcionadas por el lenguaje de implantación con entidades más sofisti-

cadas. Esas entidades se combinan a su vez hasta que un conjunto de funciones, estructuras de datos e interconexiones se han construido para resolver el problema utilizando los elementos disponibles en el ambiente actual de programación.

2. Diseño estructurado.

Programar aún sin utilizar proposiciones GOTO permite formas innumerables de escribir programas incomprensibles. Existen infinidad de programas estructurados cuyos módulos comparten un área de trabajo local o cuyos códigos pueden alterar el código de otros módulos; el resultado es que el hacer un cambio en un módulo puede causar problemas impredecibles en otro.

El diseño estructurado fué desarrollado por Larry Constantine, como una técnica de arriba hacia abajo para el diseño arquitectónico de sistemas de sofware. Principalmente, se concentra en las relaciones entre los módulos y su función principal consiste en convertir diagrámas de flujo de datos a gráficas estructuradas. Introduce los conceptos de acoplamiento y cohesión para guiar el proceso de diseño. Un sistema bien diseñado muestra un grado bajo de acoplamiento entre los módulos y un alto grado de cohesión entre los elementos de cada módulo.

El primer paso en el diseño estructurado, es la revisión y el refinamiento del diagrama o diagramas de flujo desarrollados durante la definición de los requerimientos y el diseño externo. El segundo paso lo determina el hecho de si el sistema está centrado en transformaciones o dirigido a transacciones. En un sistema centrado en transformaciones, el diagrama de flujo de datos contiene segmentos de entrada, procesamineto y salida que se convierten en subsistemas de entrada, procesamiento y salida respectivamente en el mapa de estructura. El tercer paso en el diseño estructurado es la descomposición de cada subsistema en módulos, lo que debe llevarse a cabo iterativamente hasta que cada módulo sea lo bastante pequeño, de tal forma que su implantación total pueda realizarse de una sola vez. En la fase inicial del diseño, el sistema debería poder subdividirse tanto como fuese posible, ya que los módulos pequeños pueden recombinarse después fácilmente.

Los resultados de aplicar ésta teoría se encuentran en el capítulo cuatro.

CAPITULO CUATRO

RESULTADOS

IV.1. RESULTADOS OBTENIDOS.

El primer paso del trabajo consistio en introducirse al campo en que se estaba incursionando. Se hizo un estudio de los principales conceptos y definiciones que nos ayudaron a entender físicamente el proceso químico. Recibimos platicas de investigadores en las cuales, sin profundizar, se nos explicó el modelo matemático.

Una vez conocido el problema con los principales conceptos de lo que físicamente es el proceso químico y teniendo los conocimientos indispensables de éste, se hizo un estudio de las ecuaciones que se involucraron en el modelo matemático.

Conociendo las ecuaciones que se tenian que resolver, se buscaron los posibles métodos números para darles solución, en el capítulo dos se explican los métodos números que se estudiaron y se explica cual se escogió para resolver las ecuaciones.

En el resultado de este estudio, se encontró que se tenían trece ecuaciones diferenciales ordinarias que resolver, y estas son de la ecuación (1) a la (10), la (27), la (28), y la (52), que estan en el apéndice A, las cuales fueron resueltas como un sistema de ecuaciones por el método de Runge-Kutta de orden cuarto. Ver ecuaciones en el apéndice A.

También hay un sistema de ecuaciones no lineales, ecuaciones (86) a (89) que se resolvieron por el método de Levenberg-Maquardt, y un problema de optimización sin restricciones la ecuación (103), la cual se resolvió por interpolación cúbica y por el método de la secante, el capítulo dos da más detalles.

Las ecuaciones restantes que presenta el modelo, son formulas sencillas de resolver, ya sea con la solución de alguna de las ecuaciones que se resulven por métodos números, con datos de entrada, con constantes ya definidas o por las tres.

Todas éstas ecuaciones estan formadas por variables, algunas de las cuales dependen del tiempo, pero que para poder resolverlas, antes hay que calcular una serie de fórmulas previas a estas. También hay algunas variables que son datos de entrada y que son constantes en toda simulación, esto permite que algunas de las ecuaciones solamante tengan que calcularse substituyendo los valores de sus variables. Otras de esas variables, que también son constantes, van a depender del tipo de simulación que en este caso, el investigador desee realizar y deben ser modificadas cada que se haga una simulación diferente, estas también son fáciles de resolver, solo substituyendo valores. Se hizo una clasificación de variables estableciendo cuáles eran constantes y cuales no, resultando datos variables y datos fijos. Para facilitar la solución se agruparon las ecua-

ciones dependiendo de la parte que resuelven, y resultaron once bloques que llamamos subrutinas. Estas subrutinas son las siguientes:

- Subrutina #1 Balance de materia
- Subrutina #2 Velocidades de reacción
- Subrutina #3 Concentración en agua y en partículas
- Subrutina #4 Volumen en fase acuosa y en partículas
- Subrutina #5 Concentración de radicales en partícula
- Subrutina #6 Balance de concentración de partículas
- Subrutina #7 Formación de precursores
- Subrutina #8 Velocidad de generación de partículas
- Subrutina #9 Precursor de partículas
- Subrutina #10 Coeficiente de coagulación
- Subrutina #11 Variables secundarias

Se trató de ir resolviendo aquellas ecuaciones que no dependen de ninguna otra, a fin de llegar a la solución y que todas las ecuaciones sean resueltas. El orden necesario para llegar a la solucion fué el siguiente :

- Subrutina #3 Concentración en agua y en partículas
- Subrutina #4 Volumen en fase acuosa y en partículas
- Subrutina #5 Concentración de radicales en partícula
- Subrutina #2 Velocidades de reacción
- Subrutina #1 Balance de materia
- Subrutina #7 Formación de precursores
- Subrutina #8 Velocidad de generación de partículas
- Subrutina #10 Coeficiente de coagulación
- Subrutina #9 Precursor de partículas
- Subrutina #6 Balance de concentración de partículas
- Subrutina #11 Variables secundarias

Este orden fué necesario ya que cada subrutina aporta resultados necesarios para resolver las ecuaciones precedentes. Sin embargo, cabe señalar, que llegar a la solución no fué tan sencillo como parece ya que en primera el modelo es complicado y en segunda fué necesario hacer pruebas con los métodos numéricos, investigaciones de las constantes, de las unidades en que se deben presentar las variables, relación que

deben tener entre sí, etc. En cuanto a la operación del modelo, es muy sensible por lo que no se pueden hacer pruebas con datos al azar puesto que se debe generar un ámbito coherente en cuanto a unidades y datos reales, esto nos llevó a tener que hacer consultas a expertos, tan sólo para hacer una corrida.

Una vez analizado el modelo para la solución numérica y resueltas las ecuaciones, el siguente paso fué aplicar al problema escogído, las fases del Ciclo de Vida del desarrollo del software, resultando el código objetivo de este trabajo. Los pasos que se siguieron son los que señala el capítulo tres.

Podemos decir que se creo un software con todas las características de calidad de software establecidas en el capítulo tres (III.1.3).

Siguiendo los pasos del Ciclo de vida de un sistema, durante en primer paso, de análisis de requerimientos, se determinó que se requiere de un programa de cómputo confiable y amigable, que simúle el proceso químico de copolimerización en emulsión. Este programa debe permitir el relizar diferentes modelaciones y aplicarle diferentes formulaciones a dichos modelos, para realizar experimentos previos a la relización de proceso en la planta.

Como ya se comentó, se tienen dos tipos de usuarios. Uno de ello es el modelador, esto es, un investigador que en base al programa que se diseñe podrá decidir que partes desea involucrar, si se complica o se hace muy simple,y bajo que formulaciones, con el fin de descubrir características, u observar comportamientos en el modelo y así poder obtener un producto con ciertas características. El otro usuario es el que hace experimentos con el modelo y la formulación fijada por el modelador, esto con el propósito de experimentar, ya sea tiempo de reacción o cantidades y así minímizar costos realizando los experimentos en el simulador y evitando hacer los experimentos en la planta piloto.

Debido a la complejidad de este proceso, para lograr lo anterior, se requiere también el que se realice un análisis que permita la correcta aplicación de los métodos numéricos que resuelven las ecuaciones involucradas en el sistema, ya que se requiere que el programa sea confiable en sus resultados. Para esto fué necesario el desarrollo del capítulo dos y los resultados son los que ya se mencionaron anteriormente.

En la siguiente parte, definición de requerimiento se encontró que específicamente se necesita que el programa permita cumplir con los siguientes puntos :

1. Hacer una descripción corta y otra detallada de lo que hace el modelo. Esto facilita que el usuario pueda hacer los cambios que desee al programa, y que al correrlo nuevamente se tenga la información elemental de las condiciones con que cuenta y no sea necesario estar en contacto con el modelador.

2. Presentar una lista con los datos de entrada, señalando cuáles son fijos (en todas

las modelaciones) y cuales varian en cada corrida. Con opción a modificar todo, algunos o ningún dato.

3. Presentar una lista de los datos de salida, con opción a decidir cuales datos interesan y cuales no.

4. Generar tablas estándar de los resultados.

5. Generar tablas que el usuario requiera dándole opciones de los posibles valores que puede obtener.

6. Presentar al usuario una corrida default, por si decide no alterar datos de entrada ni de salida.

Para esto se contó con un equipo HP-9000 serie 800, con el lenguaje FORTRAN y con algunas rutinas de métodos numéricos para la solución de las ecuaciones, el IMSL libreria de programas de aplicación matemática.

El siguiente paso fué diseñar un algorítmo, diseñar la forma en que se van a asociar los datos con los algorítmos que los van a representar y codificar esto en FORTRAN.

El código correspondiente a las fases anteriores, fue instalado en una HP-9000 serie 800 y se probó.

IV.2. CORRIDAS DE PRUEBA

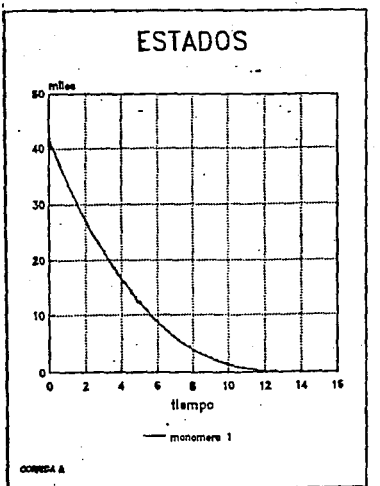
A continuación se presentan algunas corridas de prueba del simulador, para posteriormente hacer una comparación con los resultados que se presentan en la literatura, con respecto a resultados publicados de un modelo o experimentos deseados.

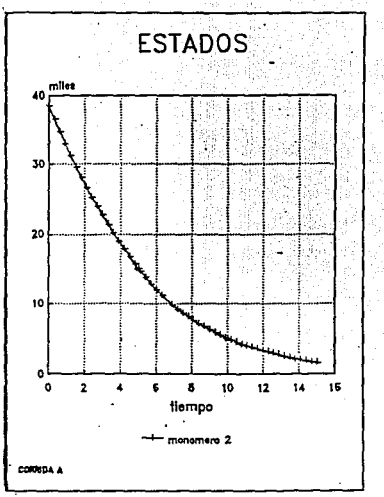
Solamente se presentan los resultados en forma de tabla y su respectiva gráfica. La corridas simulan un proceso de copolimerización en emulsion de estireno y acrilinitrilo en un reactor batch.

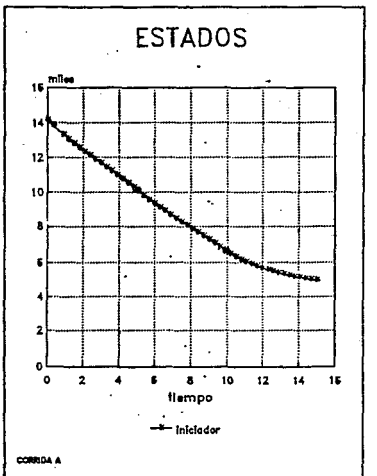
Se presentan curvas de conversión de polímero, de consumo de monómero A, estireno, del monómero B, y de iniciador y sus respectivos datos. En total son cinco corridas de prueba.

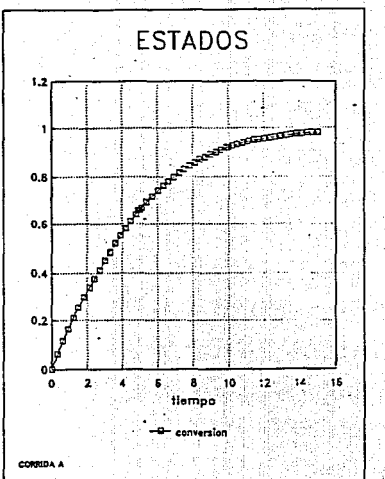
ESTADO DE LAS VARIABLES CORRIDA A

TIEMPO	MONOMERO 1	MONOMERO 2	INICIADOR	CONVERSION
0.0	41810.675883	38557.167074000	14.178082000000	1.474252D-04
0.3	39209.001175	36547.691736135	13.869467124135	5.751452D-02
0.6	36782.516859	34666.455009991	13.581855393487	0.1111066821
0.9	34508.753230	32896.785643951	13.311498078617	0.1614108424
1.2	32368.707771	31224.526776443	13.055196117268	0.2088395417
1.5	30346.255262	29637.608140867	12.810183561616	0.2537435588
1.8	28427.684574	28125.710383530	12.574016663797	0.2964218559
2.1	26601.345964	26680.010025485	12.344604128514	0.3371291392
2.4	24857.378272	25292.982954879	12.119953379088	0.3760816982
2.7	23187.497667	23958.253199144	11.898328799253	0.4134619148
3.0	21584.831914	22670.475378230	11.678120355097	0.4494217877
3.3	20043.787300	21425.241480330	11.457840228567	0.4840857480
3.6	18559.938128	20219.004578244	11.236106418078	0.5175529627
3.9	17129.930159	19049.013112062	11.011613238384	0.5498994537
4.2	15751.190776	17913.250320807	10.783221633001	0.5811797691
4.5	14422.840135	16810.374408736	10.549767632040	0.6114290328
4.8	13143.598607	15739.655711767	10.310256762393	0.6406647778
4.9	12318.149638	15043.596115274	10.146762351693	0.6595937920
5.0	12318.149638	15043.596115274	10.146762351693	0.6595937920
5.1	11916.780183	14703.533440157	10.064586769702	0.6688179168
5.4	10729.932059	13733.801977933	9.8223967838128	0.6950291219
5.7	9735.0653306	12833.771161331	9.5859645699180	0.7192221538
6.0	8774.1758044	11996.825191364	9.3545310937453	0.7415889409
6.3	7889.4994580	11217.344224985	9.1275275368795	0.7622926440
6.6	7074.5705319	10490.480359016	8.9045228790351	0.7814740158
6.9	6323.7376619	9811.9917106192	8.6851984721598	0.7992561377
7.2	5632.0042587	9178.1172550738	8.4693184026766	0.8157479630
7.5	4994.9112614	8585.4832583303	8.2567108767347	0.8310469188
7.8	4408.4486527	8031.0305730549	8.0472530924263	0.8452411025
8.1	3868.9899062	7511.9580122859	7.8408596257133	0.8584101508
8.4	3373.2427301	7025.6759862867	7.6374733182686	0.8706275301
8.7	2918.2140249	6569.7675929938	7.4370583265178	0.8819604558
9.0	2501.1851120	6141.9524302165	7.2395945986093	0.8924711211
9.3	2119.6976648	5740.0504824543	7.0450736413623	0.9022172320
9.6	1771.5508366	5361.9406879414	6.8534952434183	0.9112525566
9.9	1454.8123419	5005.5073795279	6.6648649293560	0.9196274554
9.9	1355.9207868	4891.1391587215	6.6026448183803	0.9222806103
10.0	1355.920786	4891.1391587215	6.6026448183803	0.9222806103
10.2	1171.437199	4672.8766572189	6.4822597165901	0.9272911555
10.5	929.1318126	4373.8371180892	6.3148445965752	0.9340260023
10.8	722.8822303	4103.5957868110	6.1617967337702	0.9399540034
11.1	547.9772910	3856.9110002956	6.0216879030212	0.9451989814
11.4	400.8508296	3628.9625488174	5.8932582662261	0.9498652708
11.7	278.9941796	3415.0476055578	5.7753929802075	0.9540425870
12.0	180.9307128	3210.2828944961	5.6671026662223	0.9578100606
12.3	106.1234175	3009.5041095250	5.5675068653585	0.9612386141
12.6	54.36767026	2808.1316295203	5.4758200145791	0.9643877706
12.9	23.89227278	2605.2723077163	5.3913394878545	0.9672906769
13.2	9.302130663	2406.9016699848	5.3134353629692	0.9699401119
13.5	3.444906756	2222.0163799427	5.2415416315520	0.9723131306
13.8	1.286546822	2055.6966618795	5.1751486225948	0.9744091585
14.1	0.498485501	1908.4184964127	5.1137964509835	0.9762512434
14.4	0.202335593	1778.5361749194	5.0570693362700	0.9778707877
14.7	8.61903D-02	1663.8970513482	5.004590663644	0.9792984526
15.0	3.84885D-02	1562.4413047546	4.9560186767825	0.9805612523


ESTADOS
miles
50
40
30
20
10
0
0 2 4 6 8 10 12 14 16
tiempo
monomero 1
CORRIDA A

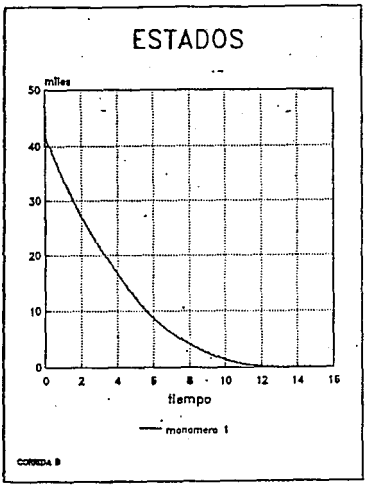

ESTADOS
miles
40
30
20
10
0
0 2 4 6 8 10 12 14 16
tiempo
monomero 2
CORRIDA A

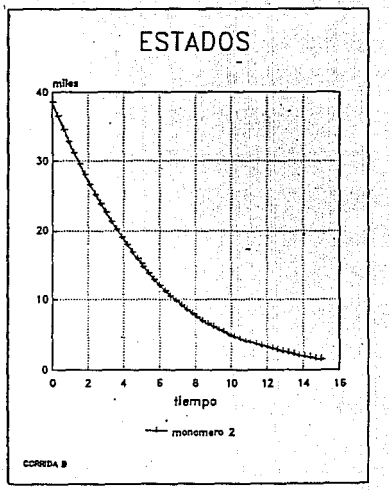

ESTADOS
miles
16
14
12
10
8
6
4
2
0
0 2 4 6 8 10 12 14 16
tiempo
Iniciador
CORRIDA A


ESTADOS
1.2
1
0.8
0.6
0.4
0.2
0
0 2 4 6 8 10 12 14 16
tiempo
conversion
CORRIDA A

ESTADO DE LAS VARIABLES CORRIDA B

TIEMPO	MONOMERO 1	MONOMERO 2	INICIADOR	CONVERSION
0.	41810.6758830	38557.167074	14.1780820000	1.474252D-04
0.3	39209.0011749	36547.691735	13.8694671240	5.751452D-02
0.6	36782.5168590	34666.455009	13.5818553934	0.1111066821
0.9	34508.7532300	32896.785643	13.3114980785	0.1614108424
1.2	32368.7077703	31224.526775	13.0551961171	0.2088395417
1.5	30346.2552619	29637.608140	12.8101835614	0.2537435588
1.8	28427.6845737	28125.710382	12.5740366372	0.2964218559
2.1	26601.3459632	26680.010024	12.3446041283	0.3371291392
2.4	24857.3782715	25293.982954	12.1199533789	0.3760816982
2.7	23187.4976668	23958.253198	11.8983287990	0.4134619148
3.0	21584.8319136	22670.475177	11.6781203549	0.4494217877
3.3	20043.7872991	21425.241479	11.4578402283	0.4840857480
3.6	18559.9381274	20219.004577	11.2361064178	0.5175529827
3.9	17129.9301584	19049.013111	11.0116323381	0.5498944537
4.2	15751.3907753	17913.250320	10.7832216328	0.5811797691
4.5	14422.8401344	16810.374407	10.5497676318	0.6114290329
4.8	13141.5986069	15739.655711	10.3102567621	0.6406647778
4.9	12318.1496590	15043.596332	10.1467623558	0.6595937915
5.0	12318.1496590	15043.596332	10.1467623558	0.6595937915
5.1	11916.7801829	14703.533439	10.0645867694	0.6688179168
5.4	10779.6622066	13731.801977	9.82239678361	0.6950291319
5.7	9735.06532995	12813.771160	9.58596456971	0.7192221538
6.0	8774.17580387	11996.825190	9.35453189154	0.7415889409
6.3	7889.49945746	11217.344224	9.12752753668	0.7622926440
6.6	7074.57051146	10490.480358	8.90452287884	0.7814740158
6.9	6323.73766352	9811.9917102	8.68519847198	0.7992561377
7.2	5632.00425836	9178.1172547	8.46931840250	0.8157479630
7.5	4994.91126312	8585.4832580	8.25671087657	0.8310469388
7.8	4408.44865240	8031.0305727	8.04725309227	0.8452410125
8.1	3868.98995592	7511.9580120	7.84085962557	0.8584101508
8.4	3373.24272992	7025.6759860	7.63747311814	0.8706275301
8.7	2918.21402477	6569.7675928	7.43705832640	0.8819604558
9.0	2501.18511190	6141.9524300	7.23959459850	0.8924711213
9.3	2119.69766470	5740.0504823	7.04507364127	0.9022172320
9.6	1771.55083640	5361.9406878	6.85349524334	0.9112525566
9.9	1454.81234191	5005.5073794	6.66486492929	0.9196274554
9.9	1354.92078625	4891.1391580	6.60264481797	0.9222806103
10.0	1355.92078625	4891.1391580	6.60264481797	0.9222806103
10.2	1171.43719917	4672.8766571	6.48225971654	0.9272911555
10.5	929.131812670	4373.8371180	6.31484459654	0.9340260023
10.8	722.882230321	4103.5957868	6.16179673375	0.9399540034
11.1	547.977291011	3856.9110002	6.02168790301	0.9451989814
11.4	400.850829684	3628.9625488	5.89325826622	0.9498652708
11.7	278.994379624	3415.0476055	5.77539298020	0.9540425870
12.0	180.930711824	3210.2828945	5.66710266622	0.9578100606
12.3	106.122417521	3009.5041095	5.56750686536	0.9612386141
12.6	54.3676702689	2808.1316295	5.47582001458	0.9643877706
12.9	23.8922727856	2605.2723077	5.39133948786	0.9672986769
13.2	9.30213066526	2406.9016700	5.31343536297	0.9699401119
13.5	3.44490675682	2222.0163799	5.24154163156	0.9723131306
13.8	1.28654682289	2055.6966619	5.17514862260	0.9744091585
14.1	0.49848551885	1908.4184964	5.11379645099	0.9762512434
14.4	0.20233553411	1778.5361749	5.05706933627	0.9778707877
14.7	8.61907361D-2	1663.8970513	5.00459066326	0.9792984526
15.0	3.84885149D-2	1562.4413047	4.95601867678	0.9805612523


ESTADOS
miles
50
40
30
20
10
0
0 2 4 6 8 10 12 14 16
tiempo
monomero 1
CORRIDA B

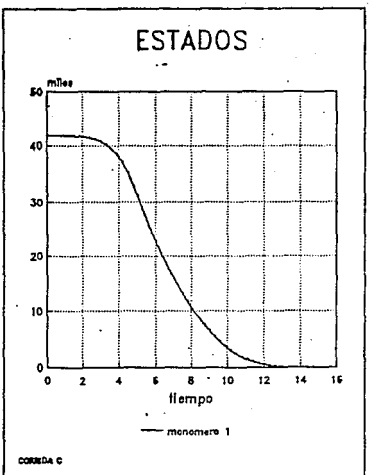

ESTADOS
miles
40
30
20
10
0
0 2 4 6 8 10 12 14 16
tiempo
monomero 2
CORRIDA B

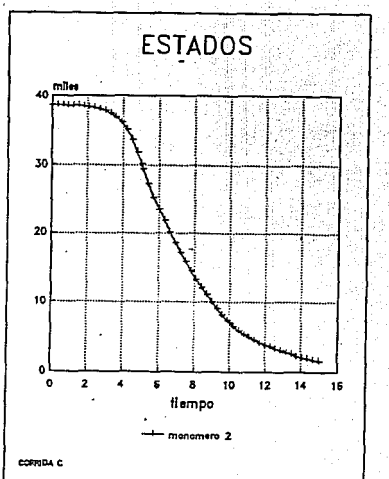
ESTADOS
miles
16
14
12
10
8
6
4
2
0
0 2 4 6 8 10 12 14 16
tiempo
iniciador
CORRIDA B

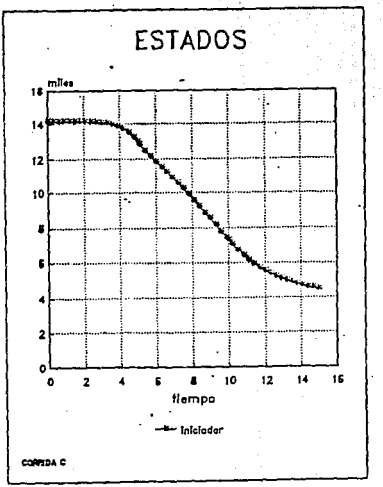
ESTADOS
1.2
1
0.8
0.6
0.4
0.2
0
0 2 4 6 8 10 12 14 16
tiempo
conversion
CORRIDA B

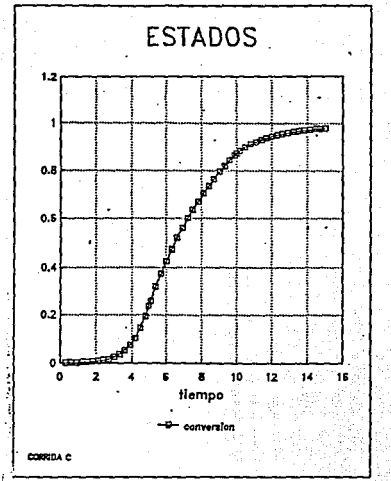
ESTADO DE LAS VARIABLES CORRIDA C

TIEMPO	MONOMERO 1	MONOMERO 2	INICIADOR	CONVERSION
0.	41810.6758830	38557.167074	14.17808200	3.474252D-04
0.3	41803.2568687	38551.447254	14.17775357	3.108849D-04
0.6	41791.2777995	38542.211658	14.17718236	5.748155D-04
0.9	41772.1768562	38527.484911	14.17620255	9.956647E-04
1.2	41742.0919192	38504.288852	14.17454452	1.658530D-03
1.5	41695.2717769	38468.187684	14.17177547	2.690150D-03
1.8	41621.2644935	38412.661183	14.1672104	4.276792D-03
2.1	41511.8264816	38328.260449	14.1597815	6.688331D-03
2.4	41349.5062964	38201.509576	14.1478504	1.030953D-02
2.7	41105.8985905	38013.545853	14.1289515	1.567869D-02
3.0	40749.6601093	37738.562540	14.0994558	2.353169D-02
3.3	40236.5647371	37342.255610	14.0541657	3.484552D-02
3.6	39510.1742334	36780.699150	13.9858847	0.086881D-02
3.9	38502.0910950	36000.365437	13.88505421	0.311842D-02
4.2	37135.0894770	34940.253494	13.73959666	0.033140D-01
4.5	35330.3243065	33537.019915	13.53508064	0.1432246492
4.8	33018.8269645	31733.260977	13.25512618	0.1944223010
4.9	31172.6558856	30286.854540	13.01769113	0.2353851034
5.0	31172.6558856	30286.854540	13.01769113	0.2353851034
5.1	30203.3039240	29525.184291	12.88934964	0.2569206696
5.4	27474.8498599	27172.292797	12.52241208	0.3176492607
5.7	24978.3540971	25389.419148	12.17613438	0.3733768855
6.0	22675.7331315	23547.795173	11.84455323	0.4249352424
6.3	20538.2841523	21825.609914	11.52185646	0.4729527709
6.6	18544.4616130	20206.953935	11.20401374	0.5179025381
6.9	16678.2661532	18677.792585	10.88764416	0.5601369216
7.2	14928.0323858	17230.781246	10.56987451	0.5999136980
7.5	13285.4774029	15858.870464	10.24842031	0.6374165252
7.8	11744.9256642	14557.587991	9.921442347	0.6727716513
8.1	10302.6579909	13323.988108	9.587516525	0.7060619941
8.4	8956.35455866	12156.234550	9.245610428	0.7373392592
8.7	7704.61711689	11053.224340	8.895066243	0.7666344374
9.0	6546.65598294	10014.315386	8.535587442	0.7939668417
9.3	5481.50848938	9038.9439518	8.167226128	0.8193517223
9.6	4508.69036332	8126.4770871	7.790369025	0.8428064644
9.9	3627.13063934	7275.9398481	7.405721023	0.8643554015
9.9	3353.34543766	7005.9461502	7.275938999	0.8711205293
10.0	3353.34543766	7005.9461502	7.275938999	0.8711205293
10.2	2854.67226624	6505.2540253	7.026978221	0.8835535861
10.5	2233.01634843	5860.6422402	6.692347529	0.8993071632
10.8	1732.34449628	5318.5385754	6.398121605	0.9122802845
11.1	1326.66487670	4856.97581132	6.138500124	0.9230696063
11.4	996.759521129	4458.9839050	5.908709453	0.9321253420
11.7	728.384564009	4111.0536728	5.704718908	0.9397927660
12.0	511.097560237	3801.9569191	5.52316723	0.9463414816
12.3	337.550833771	3521.7254240	5.36120418	0.9519869152
12.6	203.151298670	3260.6791689	5.216405529	0 .9569068862
12.9	105.839261184	3008.6624096	5.08669917	0.9612533673
13.2	44.7373709453	2756.5340703	4.970305803	0.9651496279
13.5	15.0542736151	2504.6788049	4.865690898	0.9686521090
13.8	4.44453549736	2267.5958794	4.771525648	0.9717336418
14.1	1.30688550793	2058.2330923	4.686654929	0.9743773498
14.4	0.40748679784	1878.5609107	4.610070934	0.9766238324
14.7	0.13702681683	1725.2185781	4.540891333	0.9785349219
15.0	4.97306369D-2	1593.1973265	4.478341116	0.9801685254


ESTADOS
miles
50
40
30
20
10
0
0 2 4 6 8 10 12 14 16
tiempo
monomero 1
CORRIDA C

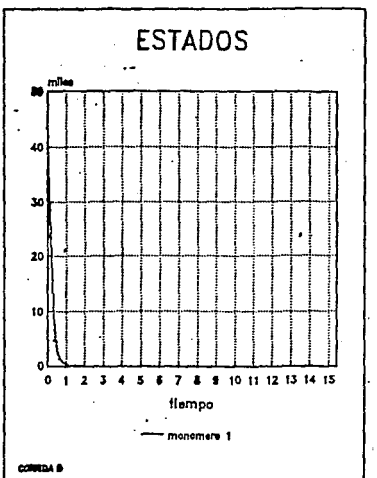

ESTADOS
miles
40
30
20
10
0
0 2 4 6 8 10 12 14 16
tiempo
monomero 2
CORRIDA C

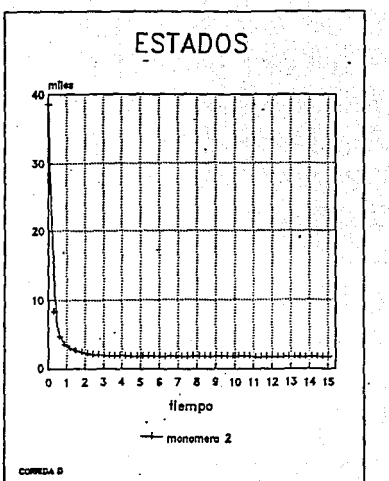

ESTADOS
miles
16
14
12
10
8
6
4
2
0
0 2 4 6 8 10 12 14 16
tiempo
Iniciador
CORRIDA C

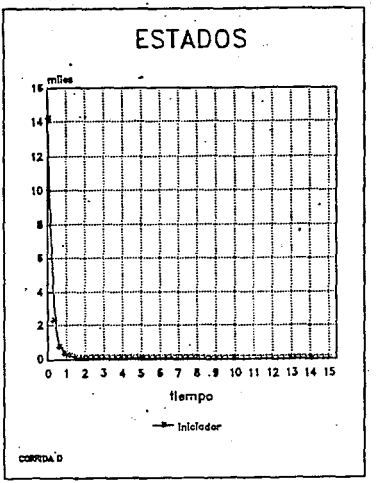

ESTADOS
1.2
1
0.8
0.6
0.4
0.2
0
0 2 4 6 8 10 12 14 16
tiempo
conversion
CORRIDA C

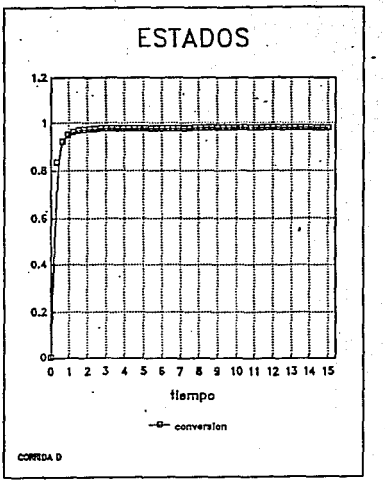
ESTADO DE LAS VARIABLES CORRIDA D

TIEMPO	MONOMERO 1	MONOMERO 2	INICIADOR	CONVERSION
0	41810.6758830	38557.16707	14.178082000	1.474252D-04
0.3	4771.66807173	8375.818003	2.2811430946	0.8364364644
0.6	1284.91709667	4807.943463	0.7015349961	0.9241989976
0.9	392.908280774	3615.879566	0.3031061964	0.9501268485
1.2	101.466083879	2994.521554	0.1672284223	0.9614829531
1.5	20.7514840820	2573.760093	0.1100024454	0.9677217932
1.8	5.01682826222	2289.676142	8.205834D-02	0.9714518225
2.1	1.77305473866	2108.249644	6.693676D-02	0.9737492966
2.4	0.85321827572	1990.469209	5.815839D-02	0.9752260412
2.7	0.50933212872	1911.611928	5.281711D-02	0.9762111179
3.0	0.35300714211	1857.592638	4.946684D-02	0.9768853754
3.3	0.27169372908	1819.964320	4.732597D-02	0.9773545192
3.6	0.22528184361	1793.520868	4.594388D-02	0.9776840784
3.9	0.19704212266	1774.854263	4.504782D-02	0.9779166600
4.2	0.17909354676	1761.663787	4.446682D-02	0.9780809854
4.5	0.16734862896	1752.350607	4.409120D-02	0.9781969217
4.8	0.15951809460	1745.611074	4.384957D-02	0.9782784518
4.9	0.15577535517	1742.577341	4.373910D-02	0.9783187291
5.0	0.15577535517	1742.577341	4.373910D-02	0.9783187291
5.1	0.15414599501	1741.147195	4.369178D-02	0.9783365418
5.4	0.14936293467	1736.667276	4.355008D-02	0.9783898475
5.7	0.14473374461	1732.603092	4.340878D-02	0.9784429556
6.0	0.14025351496	1728.354586	4.326788D-02	0.9784958669
6.3	0.13591729695	1724.121697	4.312737D-02	0.9785485820
6.6	0.13172005379	1719.904361	4.298726D-02	0.9786011019
6.9	0.12765723925	1715.702519	4.284755D-02	0.9786534273
7.2	0.12372441960	1711.516111	4.270822D-02	0.9787055592
7.5	0.11991730324	1707.345076	4.256930D-02	0.9787574982
7.8	0.11623174064	1703.189354	4.243076D-02	0.9788092452
8.1	0.11266371023	1699.048884	4.229261D-02	0.9788608010
8.4	0.10920934811	1694.923608	4.215486D-02	0.9789121663
8.7	0.10585649703	1690.813464	4.201749D-02	0.9789633420
9.0	1.02626883D-01	1686.718393	4.188051D-02	0.9790143289
9.3	9.9491592D-02	1682.638335	4.174392D-02	0.9790651277
9.6	9.6455809D-02	1678.573230	4.160771D-02	0.9791157392
9.9	9.3516226D-02	1674.523019	4.147190D-02	0.9791661643
9.9	9.25571955D-02	1673.176248	4.142671D-02	0.9791829314
10.0	9.25571955D-02	1673.176248	4.142671D-02	0.9791829314
10.2	9.0669888D-02	1670.488036	4.133648D-02	0.9792163988
10.5	8.7914547D-02	1666.469477	4.120151D-02	0.9792664277
10.8	8.5247439D-02	1662.467168	4.106701D-02	0.9793162510
11.1	8.2665608D-02	1658.481630	4.093295D-02	0.9793658695
11.4	8.0166103D-02	1654.512180	4.079936D-02	0.9794152843
11.7	7.7746196D-02	1650.558939	4.066621D-02	0.9794644965
12.0	7.5403219D-02	1646.621826	4.053351D-02	0.9795135071
12.3	7.3134605D-02	1642.700764	4.040126D-02	0.9795623171
12.6	7.0937883D-02	1638.795673	4.026946D-02	0.9796109274
12.9	6.8810665D-02	1634.906476	4.013811D-02	0.9796593392
13.2	6.6750662D-02	1631.033093	4.000720D-02	0.9797075534
13.5	6.4755642D-02	1627.175448	3.987673D-02	0.9797555710
13.8	6.2823469D-02	1623.333464	3.974670D-02	0.9798033930
14.1	6.0952063D-02	1619.507062	3.961711D-02	0.9798510204
14.4	5.9139427D-02	1615.696168	3.948796D-02	0.9798984541
14.7	5.7383623D-02	1611.900705	3.935925D-02	0.9799456951
15.0	5.5682784D-02	1608.120597	3.923097D-02	0.9799927444


ESTADOS
moles
40
30
20
10
0
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
tiempo
monomero 1
CORRIDA B

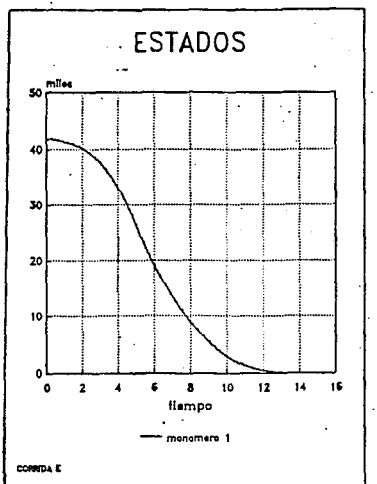

ESTADOS
moles
40
30
20
10
0
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
tiempo
monomero 2
CORRIDA B

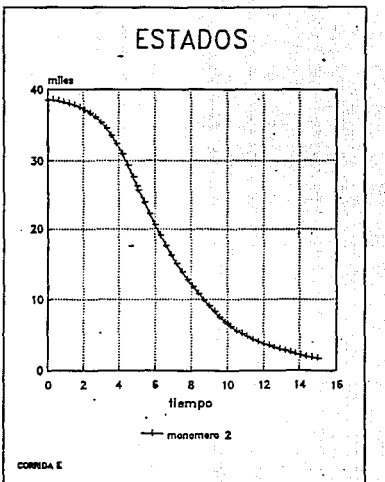

ESTADOS
miles
16
14
12
10
8
6
4
2
0
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
tiempo
iniciador
CORRIDA D

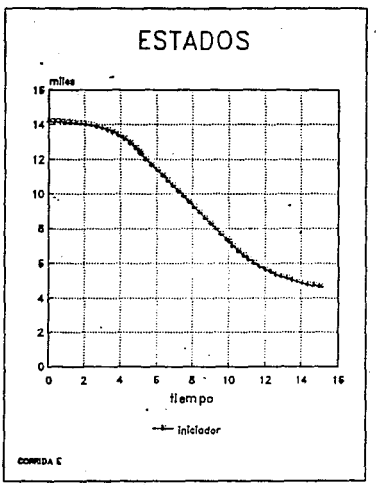

ESTADOS
1.2
1
0.8
0.6
0.4
0.2
0
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
tiempo
conversion
CORRIDA D

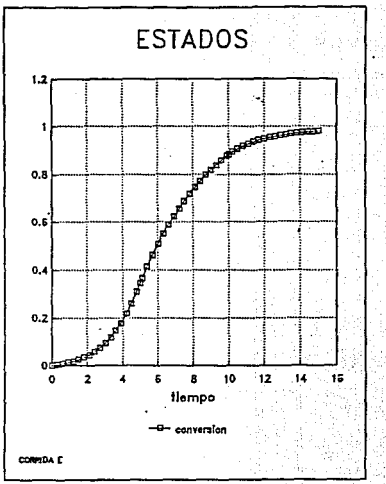
ESTADO DE LAS VARIABLES CORRIDA E

TIEMPO	MONOMERO 1	MONOMERO 2	INICIADOR	CONVERSION
0.	41810.675883	38557.1670	14.17808200	1.4742D-04
0.3	41697.236962	38469.7029	14.17031969	2.6468D-03
0.6	41550.280688	38396.3760	14.15987246	5.8850D-03
0.9	41360.967779	38210.3514	14.14580743	1.0056D-02
1.2	41118.594008	38023.3429	14.12703519	1.5398D-02
1.5	40810.145004	37785.4154	14.10216093	2.2193D-02
1.8	40421.141805	37484.8535	14.06945797	3.0775D-02
2.1	39933.606485	37108.1177	14.02682657	4.1527D-02
2.4	39328.216720	36639.9376	13.97176576	5.4883D-02
2.7	38583.725114	36061.6009	13.90136613	7.1316D-02
3.0	37677.911880	35361.4881	13.81233110	9.1320D-02
3.3	36588.708388	34515.8817	13.70102859	0.11539111
3.6	35295.665346	33510.0296	13.56356407	0.14399156
3.9	33781.650362	32329.1758	13.39584834	0.17751581
4.2	32034.565940	30962.7986	13.19361203	0.21625273
4.5	30048.832797	29403.6598	12.95229998	0.26035431
4.8	27826.450024	27650.5133	12.66677330	0.30981369
4.9	26218.097007	26375.7718	12.44883403	0.34568213
5.0	26218.097007	26375.7718	12.44883403	0.34568213
5.1	25402.179303	25727.1644	12.33506907	0.36389973
5.4	23097.714967	23886.2936	12.00505407	0.41547414
5.7	20979.962040	22182.5711	11.69038274	0.46301693
6.0	19021.898440	20595.3473	11.38409084	0.50712369
6.3	17203.029170	19109.0147	11.08339127	0.54824355
6.6	15507.922649	17711.7807	10.78544047	0.58671521
6.9	13924.983409	16394.7605	10.487980114	0.62279347
7.2	12445.522405	15151.2884	10.189228245	0.65666936
7.5	11063.033845	13976.3794	9.887802953	0.68840582
7.8	9772.9212526	12866.2999	9.582667446	0.71835020
8.1	8570.5498154	11818.2193	9.273089108	0.74634426
8.4	7453.8605399	10829.9257	8.958607551	0.77253226
8.7	6420.1009721	9899.59259	8.639008294	0.79696744
9.0	5467.0932033	9025.58768	8.314299944	0.81969722
9.3	4592.7744314	8206.31476	7.984693489	0.84076712
9.6	3795.0932991	7440.07960	7.650582961	0.86022373
9.9	3071.9638667	6724.96573	7.312527019	0.87811684
9.9	2847.1236551	6497.57493	7.199083004	0.88374303
10.0	2847.1236551	6497.57493	7.199083004	0.88374303
10.2	2435.3364228	6073.37099	6.981090346	0.89414356
10.5	1915.7913460	5520.10652	6.685959528	0.90749033
10.8	1491.9027771	5047.98685	6.424342877	0.91863752
11.1	1144.5842347	4640.49982	6.191747295	0.92802803
11.4	859.50085396	4284.54374	5.984399039	0.93600317
11.7	625.92399975	3969.37414	5.799107137	0.94283010
12.0	435.99154712	3685.73678	5.633157618	0.94872176
12.3	284.38913023	3425.06087	5.484230267	0.95385214
12.6	167.66948179	3178.67650	5.350332108	0.95836825
12.9	84.952188341	2937.56206	5.229743753	0.96239703
13.2	35.023319111	2695.09073	5.120975758	0.96603476
13.5	11.837887300	2455.06928	5.022732887	0.96930931
13.8	3.6410024802	2231.82356	4.933884672	0.97218868
14.1	1.1316720489	2025.08325	4.853441049	0.97466753
14.4	0.3726326912	1865.47145	4.780532072	0.97678711
14.7	0.1316091357	1719.79395	4.714390960	0.97860247
15.0	4.9864630-02	1594.33275	4.654339864	0.98016435


ESTADOS
miles
50
40
30
20
10
0
0 2 4 6 8 10 12 14 16
tiempo
monomero 1
CORRIDA E


ESTADOS
miles
40
30
20
10
0
0 2 4 6 8 10 12 14 16
tiempo
monomero 2
CORRIDA E

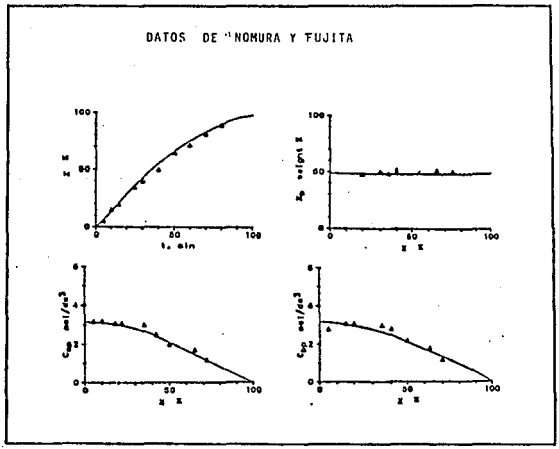

ESTADOS
miles
16
14
12
10
8
6
4
2
0
0 2 4 6 8 10 12 14 16
tiempo
iniciador
CORRIDA E


ESTADOS
1.2
1
0.8
0.6
0.4
0.2
0
0 2 4 6 8 10 12 14 16
tiempo
conversion
CORRIDA E

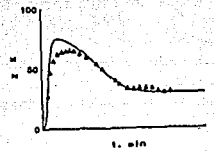
IV.3. COMPARACION DE LOS RESULTADOS CON LOS DE LA LITERATURA

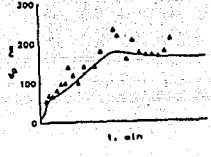
Existe una cantidad importante de datos en la literatura del efecto que causan algúnos factores (temperatura, monómeros, concentración de surfactante y tipos, fuerza iónica, configuración del reactor, etc.) en el tiempo de evolución de cantidades, tales como la conversión, número de partículas y tamaño de estas, peso molecular, composición, y muchos más. Aquí presentamos los resultados publicados de tres diferentes datos. Estos son de Goodwin, homopolimerización de estireno en un reactor batch sin adición de surfactante. Sttterlin, homopolimerización de estireno en un reactor batch con varias cantidades de surfactante adicionado. Nomura y Fujita, copolimerización de estireno y metil-metacrilato en un reactor batch.

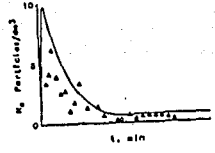
Las siguientes gráficas muestran una comparación con resultados obtenidos en otros simuladores, y con resultados obtenidos en realidad en este simulador.

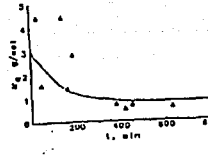


DATOS DE GOODWING

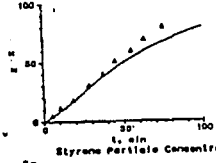


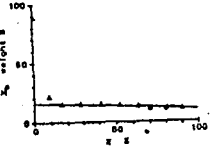




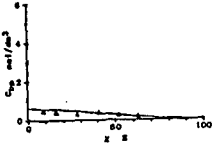


DATOS DE SUTTERLIN





Styrene Particle Concentration



CONCLUSIONES

La polimerización en emulsión es un complicado proceso físico-químico que ha tenido muchas contradicciones y en el cual se han invertido muchos años de investigación. El modelo de copolimerización en emulsión, que aquí se presenta (capítulo uno), es capaz de reproducir un amplio rango de datos sobre el mecanismo interno del proceso. Esto, aunado a otras carecterísticas que presenta el modelo, ha logrado un avance significativo en la teoría que se tenía antes de su publicación y por lo que fué considerado para crear un simulador. Por lo tanto, el contar con un programa que simule dicho modelo y que además sea amigable, contribuye a la investigación de propiedades de los polímeros lo cual es de gran utilidad en la industria.

El programa, permite que los usuarios puedan ralizar las experimentaciones que necesitan de una forma sencilla y obtener resultados en tablas para elaboración de gráficas de los datos que requieren, es decir, saber en cuanto tiempo se forman las partículas de polímero y otras características. Para el usuario resulta fácil hacer cambios en datos de entrada y salidas. Permite también a los usuarios, decidir que cálculos se quieren hacer, esto es, que si no les interesa saber el peso molecular de las partículas de polímero pueden eliminar esta parte del programa, lo cual implica que pueden jugar con el modelo y hacerlo lo mas simple posible en cuanto a cálculos, como también complicarlo.

Presentamos algunos resultados en tablas y gráficas que el usuario puede generar a partir de los resultados del experimento para facilitar el manejo de los datos. Presentamos también, un ejemplo de una corrida del programa. Al hacer las comparaciones con resultados publicados, podemos darnos cuenta de que el simulador es confiable.

Consideramos que se ha desarrollado una herramienta computacional con futuro en el mercado.

A partir de este punto, una continuación al trabajo pude ser la optimización

de un reactor de polimerización real. Esta optimización pretendería realizar el proceso en el menor tiempo posible, respetando las condiciones de operación reales, esto es, sin cambiar el equipo con se cuenta actualmente. El producir en el menor tiempo posible un producto conservando la misma calidad, significa poder competir. Otra posible optimización seria minimizar la concentración final de algun monómero o bien buscar minimizar algún residuo. Podrían minimizarse residuos contaminantes, por ejemplo. Esto resulta de gran interés ya que como el simulador permite jugar con el proceso, de tal forma, es factible una segunda etapa que implicaría considerables beneficios.

BIBLIOGRAFIA

1. BURDEN R.L., DOUGLAS FAIRES J.
   ANALISIS NUMERICO.
   GRUPO EDITORIAL IBEROAMERICANO.
2. CARNAHAN B., LUTHER H. A., WILKES J. O.
   APPLIED NUMERICAL METHODS.
   JOHN WILEY & SOONS, INC. 1969.
3. CLOSE CHARLES M. & FREDERICK DEAN K.
   MODELING AND ANALYSIS OF DINAMIC SYSTEMS.
   HOUGHTON MIFFLIN COMPANY, BOSTON 1978.
4. CONTE S. D., DE BOOR C.
   ELEMENTARY NUMERICAL ANALYSIS.
   INTERNATIONAL BOOK COMPANY, USA. 1983.
5. FAIRLEY R. E.
   SOFTWARE ENGINEERING CONCEPTS.
   Mc GRAW HILL INC. 1985.
6. FLETCHER R.
   PRACTICAL METHODS OF OPTIMIZATION VOL. 1.
   UNCONSTRAINED OPTIMIZATION.
   JOHN WILEY & SONS, INC. 1981.
7. HALL G. & MATT J. M.
   MODERN NUMERICAL METHODS FOR ORDINARY

DEFFERENTIAL EQUATIONS.

CLARENDON PRESS, OXFORD 1976.

8. GERMUND DAHLQUIST, AKE BJORCK

NUMERICAL METHODS.

PRENTICE HALL, 1974.

9. HIMMELBLAU DAVID M. & BISCHOFF KENNETH B.

ANALISIS Y SIMULACION DE PROCESOS.

REVERTE S.A. 1976.

10. JOURNAL OF APPLIED POLYMER SCIENCE.

VOL 37. pags. 2727 - 2756.

JHON WILEY & SONS INC.

11. LUENBERGER, DAVID G.

INTRODUCTION TO LINEAR AN NO LINEAR PROGRAMING.

ADISSON WESLEY PUBLISHING CO. ING. 1973.

12. MARCK H. F., GAYLOR N.G.

ENCYCLOPEDIA OF POLYMER SCIENCE AND TECHNOLOGY.

VOL. 2, 4 Y 5.

JOHN WILEY & SONS INC., 1972

13. MATH/LIBRARY.

FORTRAN SOUBRUTINES FOR MATHEMATICAL APPLICATIONS.

IMSL INC., 1987.

14. Mc. CRACKEN Y DORN.

METODOS NUMERICOS Y PROGRAMACION FORTRAN.

LIMUSA.

15. ODIAN G.

PRINCIPLES OF POLYMERIZATION.

Mc GRAW HILL, INC. 1970.

16. PRAWDA J.

METODOS Y MODELOS DE INVESTIGACION DE OPERACIONES.

VOLUMEN 1. MODELOS DETERMINISTICOS.

LIMUSA.

17. SEYMOUR LIPSCHUTZ, ARTHUR POE.

PROGRAMACION CON FORTRAN.

Mc. GRAW HILL.

18. WERNER C. RHEINBOLDT.

METHODS FOR SOLVING SYSTEMS OF NONLINEAR EQUATIONS.

SOCIETY FOR INDUSTRIAL AND APPLIED MATHEMATICS, 1984.

19. WIRTH NIKLAUS.

ALGORITMOS Y ESTRUCTURAS DE DATOS.

PRENTICE HALL.

20. ZELKOWITZ, M.

PRINCIPLES OF SOFTWARE ENGINEERING AND DESIGN.

PRENTICE HALL INTERNATIONAL, 1979.

APENDICE A

ECUACIONES DEL MODELO

$$V_\omega \frac{dC_{i\omega}}{dt} = -C_{i\omega}\frac{dV_\omega}{dt} + F_i - C_{i\omega}Q_\omega - R_{i\omega}V_\omega \tag{1}$$

$$V_\omega \frac{dC_{y\omega}}{dt} = -C_{y\omega}\frac{dV_\omega}{dt} + F_y - C_{y\omega}Q_\omega \tag{2}$$

$$V_e \frac{dC_{ae}}{dt} = -C_{ae}\frac{dV_e}{dt} + F_a - C_{ae}Q_e - R_{pae}V_e \tag{3}$$

$$V_e \frac{dC_{be}}{dt} = -C_{be}\frac{dV_e}{dt} + F_b - C_{be}Q_e - R_{pbe}V_e \tag{4}$$

$$V_e \frac{dC_{xe}}{dt} = -C_{xe}\frac{dV_e}{dt} + F_x - C_{xe}Q_e - R_{pxe}V_e \tag{5}$$

$$V_e \frac{dC_{se}}{dt} = -C_{se}\frac{dV_e}{dt} + F_s - C_{se}Q_e \tag{6}$$

$$V_e \frac{dC_{ge}}{dt} = -C_{ge}\frac{dV_e}{dt} - C_{ge}Q_e + R_{ge}V_e \tag{7}$$

$$V_e \frac{dC_{aqe}}{dt} = -C_{aqe}\frac{dV_e}{dt} - C_{aqe}Q_e + R_{pae}V_e \tag{8}$$

$$V_e \frac{dC_{bqe}}{dt} = -C_{bqe}\frac{dV_e}{dt} - C_{bqe}Q_e + R_{pbe}V_e \tag{9}$$

$$V_e \frac{dC_{qe}}{dt} = -C_{qe}\frac{dV_e}{dt} - C_{qe}Q_e + R_{qe}V_e \tag{10}$$

$$R_{i\omega} = k_i C_{i\omega} \tag{11}$$

$$R_{pae} = \left[(k_{paa} + k_{xaa})C_{a\cdot p}C_{ap} + (k_{pba} + k_{xba})C_{b\cdot p}C_{ap}\right]V_p/V_e \tag{12}$$

$$R_{pbe} = \left[(k_{pbb} + k_{xbb})C_{b\cdot p}C_{bp} + (k_{pab} + k_{xab})C_{b\cdot p}C_{bp}\right]V_p/V_e \tag{13}$$

$$R_{pxe} = \left[k_{xax}C_{a\cdot p}C_{xp} + k_{xbx}C_{b\cdot p}C_{xp}\right]V_p/V_e \tag{14}$$

$$R_{ge} = 2(1-f)R_{i\omega}V_\omega/V_e \tag{15}$$

$$R_{ie} = 2fR_{i\omega}V_\omega/V_e \tag{16}$$

$$R_{qe} = \frac{R_{ie}}{2} + R_{pxe} + \left[k_{xaa}C_{a\cdot p}C_{ap} + k_{xba}C_{b\cdot p}C_{ap} + k_{xbb}C_{b\cdot p}C_{bp} + k_{xab}C_{a\cdot p}C_{bp}\right]\frac{V_p}{V_e} \tag{17}$$

$$k_{paa} = A_{paa}e^{(-k_{paa}/RT_e)} \tag{18}$$

$$V_m = V_e - V_p - V_\omega \tag{19}$$

$$K_{j\omega p} = C_{j\omega}/C_{jp} \tag{20}$$

$$K_{jmp} \equiv C_{jm}/C_{jp} = \rho_j K_{j\omega p}/M_j C_{j\omega}^{sat} \tag{21}$$

$$C_{jp} = \frac{V_e C_{je}}{V_p + V_\omega K_{j\omega p} + V_m K_{jmp}} \tag{22}$$

$$C_{j\omega} = K_{j\omega p}C_{jp} \tag{23}$$

$$\phi_j = C_{jp}M_j/\rho_j \tag{24}$$

$$\phi_q = 1 - \phi_a - \phi_b - \phi_x \tag{25}$$

$$\frac{dV_e}{dt} = Q_f - Q_e + \Delta Q_e = 0 \tag{26}$$

$$C_{\omega\omega}\frac{dV_\omega}{dt} = F_\omega - \frac{Q_e V_\omega C_{\omega\omega}}{V_e} \tag{27}$$

$$\frac{dV_p}{dt} = -V_p\rho_q - \frac{\phi_q\rho_q Q_e V_p}{V_e} + M_a R_{pae} V_e + M_b R_{pbe} V_e \tag{28}$$

$$\Delta Q_e = \left(\frac{1}{\rho_q} - \frac{1}{\rho_a}\right) M_a R_{pae} V_e + \left(\frac{1}{\rho_q} - \frac{1}{\rho_b}\right) M_b R_{pbe} V_e \tag{29}$$

$$Q_e = Q_f + \Delta Q_e \tag{30}$$

$$C_{\omega\omega} = \rho_\omega / M_\omega \tag{31}$$

$$C_{a\cdot p} = \frac{\bar{n} N_e V_e}{V_p N_A}\left(\frac{1}{1+\gamma_p}\right) \tag{32}$$

$$\gamma_p \equiv \frac{C_{b\cdot p}}{C_{a\cdot p}} = \frac{(k_{pab} + k_{xab}) C_{bp}}{(k_{pba} + k_{xba}) C_{ap}} \tag{33}$$

$$\rho = \frac{R_{ic} N_A}{Ne} + \rho_{\tau_e} + \alpha k_d \bar{n} \tag{34}$$

$$k_d = (k_{da} + \gamma_p k_{db})\left(\frac{1}{1+\gamma_p}\right) \tag{35}$$

$$c_t = \frac{1}{2 v_p N_A}(k_{taa} + 2k_{tab}\gamma_p + k_{tbb}\gamma_\rho^2)\left(\frac{1}{1+\gamma_p}\right)^2 \tag{36}$$

$$\alpha = \frac{\alpha_a k_{da} + \alpha_b k_{db}\gamma_p}{k_{da} + k_{db}\gamma_p} \tag{37}$$

$$k_{taa} = k_{taa0} exp[2(k_{taa1}\chi_a + k_{taa2}\chi_a^2 + k_{taa3}\chi_a^3)] \tag{38}$$

$$k_{tbb} = k_{tbb0} exp[2(k_{tbb1}\chi_b + k_{tbb2}\chi_b^2 + k_{tbb3}\chi_b^3)] \tag{39}$$

$$k_{tab} = \sqrt{k_{taa} k_{tbb}} \tag{40}$$

$$\chi_a = \frac{C_{aqe}}{C_{aqe} + C_{ae}} \tag{41}$$

$$\chi_b = \frac{C_{bqe}}{C_{bqe} + C_{be}} \tag{42}$$

$$\frac{1}{k_{da}} = \frac{1}{[(k_{zaa} + k_{zba})C_{ap} + k_{zax}C_{xp}]} + \left[k_{0a}\left(\frac{k_{zaa}C_{ap} + k_{zba}k_{pab}C_{bp}/k_{pba} + k_{zax}C_{xp}}{k_{paa}C_{ap} + k_{0a}\bar{n} + k_{pab}C_{bp}}\right)\right]^{-1} \tag{43}$$

$$k_{Qa} = \frac{3D_{\omega a}K_{a\omega p}}{[r_p^2(1 + D_{\omega a}K_{a\omega p}/2D_{pa})]} \tag{44}$$

$$\frac{1}{k_{db}} = \frac{1}{[(k_{xbb} + k_{zab})C_{bp} + k_{xbz}C_{xp}]} + \left[k_{0b}\left(\frac{k_{xbb}C_{bp} + k_{zab}k_{pba}C_{ap}/k_{pab} + k_{xbz}C_{xp}}{k_{pbb}C_{bp} + k_{ob}\bar{n} + k_{pba}C_{ap}}\right)\right]^{1} \tag{45}$$

$$k_{0b} = \frac{3D_{\omega b}K_{b\omega p}}{\left[r_p^2\left(1 + \frac{D_{\omega b}k_{b\omega p}}{2D_{pb}}\right)\right]} \tag{46}$$

$$\xi = (8\rho/c_t)^{\frac{1}{2}} \tag{47}$$

$$\nu = k_d/c_t \tag{48}$$

$$V_e\frac{dN_e}{dt} = -N_e\frac{dV_e}{dt} - N_eQ_e + G_{ce}V_e \tag{49}$$

$$v_p = V_p/N_eV_e \tag{50}$$

$$r_p = (3v_p/4\pi)^{\frac{1}{3}} \tag{51}$$

$$r_q = r_p\phi_q^{\frac{1}{3}} \tag{52}$$

$$2R_{i\omega} = R_{e\omega} + R_{i\omega} \tag{53}$$

$$R_{e\omega} = k_{ea}\left(\frac{N_e}{N_A}\right)C_{a\cdot\omega} + k_{eb}\left(\frac{N_e}{N_A}\right)C_{b\cdot\omega} \tag{54}$$

$$R_{t\omega} = \frac{1}{2}k_{taa0}C_{a\cdot\omega}^2 + k_{tab0}C_{a\cdot\omega}C_{b\cdot\omega} + \frac{1}{2}k_{tbb0}C_{b\cdot\omega}^2 \tag{55}$$

$$k_{tab0} = \sqrt{k_{taa0}k_{tbb0}} \tag{56}$$

$$C_{a\cdot\omega} = \frac{-\lambda_2 + \sqrt{\lambda_2^2 - 4\lambda_1\lambda_3}}{2\lambda_1} \tag{57}$$

$$\lambda_1 = \frac{1}{2}k_{taa0} + k_{tab0}\gamma_\omega + \frac{1}{2}k_{tbb0}\gamma_\omega^2 \tag{58}$$

$$\lambda_2 = (k_{ea} + k_{eb}\gamma_\omega(N_e/N_A) \tag{59}$$

$$\lambda_3 = -2R_{i\omega} \tag{60}$$

$$\gamma_\omega \equiv \frac{C_{b\cdot\omega}}{C_{a\cdot\omega}} = \frac{k_{pab}C_{b\omega}}{k_{pba}C_{a\omega}} \tag{61}$$

$$R_{p\omega} = k_{paa}C_{a\cdot\omega}C_{a\omega} + k_{pba}C_{b\cdot\omega}C_{a\omega} + k_{pab}C_{a\cdot\omega}C_{b\omega} + k_{pbb}C_{b\cdot\omega}C_{b\omega} \tag{62}$$

$$\frac{1}{k_{ea}} = \frac{1}{k_{ca}^{\max}} + \frac{1}{4\pi r_p^2 k_{ma}N_A + \beta_e B_{1m}N_A} \tag{63}$$

$$\frac{1}{k_{eb}} = \frac{1}{k_{cb}^{\max}} + \frac{1}{4\pi r_p^2 k_{mb}N_A + \beta_e B_{1m}N_A} \tag{64}$$

$$f = R_{e\omega}/R_{i\omega} \tag{65}$$

$$G_{hc} = \frac{2R_{i\omega}N_A(V_\omega/V_c)}{(1 + R_{e\omega}/R_{p\omega})^{j_{cr}-1}} \tag{66}$$

$$j_{cr} = \frac{v_1\rho_q N_A}{M_v} \tag{67}$$

$$v_1 = \frac{4}{3}\pi r_1^3 \tag{68}$$

$$M_v = \frac{M_a + (j_b/j_a)M_b}{(j_b/j_a) + 1} \tag{69}$$

$$\left(\frac{j_b}{j_a}\right) = \frac{(K_{pab} + k_{pbb}\gamma_\omega)C_{b\omega}}{(k_{paa} + k_{pba}\gamma_\omega)C_{a\omega}} \tag{70}$$

$$G_{ee} = \frac{1}{2}\sum_{i=1}^{m-1} v_i \left(\sum_{j-m-i}^{m-1} B_{ij} v_j\right) + \frac{K_{vm-1}v_{m-1}}{v_1} - 2B_{m,m}N_r^2 \tag{71}$$

$$k = 1, \ldots, m-1$$

$$K_{vk} = \bar{n}_v \left[\frac{M_a C_{ap}(k_{paa} + k_{pba}\gamma_v + M_b C_{bp}(k_{pab} + k_{pbb}\gamma_v)}{N_A \Phi_q \rho_q (1+\gamma_v)}\right] \times \tanh\left(\frac{k^{\frac{1}{3}} r_1}{r_{FH}}\right) \tag{72}$$

$$\gamma_v \equiv \frac{C_{b\cdot v}}{C_{a\cdot v}} = \frac{k_{pab}C_{bp}}{k_{pba}C_{ap}} \tag{73}$$

$$k = 1, \ldots, m-1$$

$$\frac{dv_k}{dt} = \frac{1}{2}\sum_{i=1}^{k-1} B_{i,k-1} v_i v_{k-i} - v_k \sum_{i=1}^{m-1} B_{i,k} v_i - B_{m,k} N_e v_k - \frac{k_{vk}v_k - K_{vk-1}v_{k-1}}{v_1} + \delta_{k,1} G_{he} \tag{74}$$

$$\sigma_{vi} = \sigma_s + \sigma_g + \sigma_R, \qquad i = 1, \ldots, m \tag{75}$$

$$r_i = i^{\frac{1}{3}} r_1, \qquad i = 1, \ldots, m-1 \tag{76}$$

$$r_m = r_p \tag{77}$$

$$\sigma_R = \frac{\rho_q e_L N_A r_i}{3M_v j_{cr}} \tag{78}$$

$$\sigma_{Im} = \frac{2\rho_q e_L N_A r_p \omega}{3M_q} \tag{79}$$

$$M_q = \frac{C_{aqe}M_a + C_{bqe}M_b}{C_{qe}} \tag{80}$$

$$\sigma_s = \frac{z_+ e_L \theta_s}{a_v'} \tag{81}$$

$$\sigma_g = \frac{z_+ e_L \theta_g}{a'_g} \tag{82}$$

$$\theta_g = \frac{b_g C_{g\omega}}{(1 + b_g C_{g\omega} + b_s C_{s\omega})} \tag{83}$$

$$\theta_s = \frac{b_s C_{s\omega}}{(1 + b_g C_{g\omega} + b_s C_{s\omega})} \tag{84}$$

$$C_{g\omega} = \frac{C_{ge} V_e - (A_\nu \theta_g / a'_g N_A)}{V_\omega} \tag{85}$$

$$C_{s\omega} = \frac{C_{se} v_e - (A_e \theta_s / a'_s N_A}{V_\omega} \tag{86}$$

$$A_e = 4\pi r_p^2 N_e V_e + \sum_{k-1}^{m-1} 4\pi r_k^2 v_k V_e \tag{87}$$

$$\psi_{0i} = \frac{4\pi r_i \sigma_{vi}}{\epsilon(1 + k r_i)}, \quad i = 1, \ldots, m-1 \tag{88}$$

$$\psi_{0m} = \left(\frac{2k_B T_e}{z_+ e_L}\right) \sin h^{-1} \left(\frac{2\pi e_L z_+ \sigma_{vm}}{\epsilon k_B T_v k}\right) \tag{89}$$

$$I_e \equiv \frac{1}{2} \sum_j (C_{+j\omega} z_{+j\omega}^2 + C_{-j\omega} z_{-j\omega}^2) = 3C_{i\omega} + C_{y\omega} \tag{90}$$

$$\epsilon = 4\pi \epsilon_0 \epsilon_r \tag{91}$$

$$k \equiv (8\pi e_L^2 N_A I_e / \epsilon k_B T_e)^{\frac{1}{2}} \tag{92}$$

$$\varsigma_i = \left(\frac{2k_B T_e}{z_+ e_L}\right) \ln \left(\frac{e^{\lambda_4} + 1}{e^{\lambda_4} - 1}\right) \tag{93}$$

$$\lambda_4 = k8 + \ln \left(\frac{2^{\lambda_5} + 1}{e^{\lambda_5 + 1}}\right) \tag{94}$$

$$\lambda_5 = \frac{z_+ e_L \psi_{0i}}{2k_B T_e} \tag{95}$$

$$i = 1, \ldots, m$$
$$j = 1, \ldots, i$$

$$R_{ij} = \frac{(r_i + r_j)s}{2} \tag{96}$$

$$L_{ij} = R_{ij} - (r_i + r_j) \tag{97}$$

$$\Phi_A = \frac{-A}{6}\left[\frac{2r_i r_j}{R_{ij}^2 - (r_i + r_j)^2} + \frac{2r_i r_j}{R_{ij}^2 - (r_i - r_j)^2} + \ln\left(\frac{R_{ij}^2 - (r_i + r_j)^2}{R_{ij}^2 - (r_i - r_j)^2}\right)\right] \tag{98}$$

$$\Phi_R = \frac{\epsilon r_i r_j (\zeta_i^2 + \zeta_j^2)}{4(r_i + r_j)}\left[\frac{2\zeta_i\zeta_j}{(\zeta_i^2 + \zeta_j^2)} \ln\left(\frac{1 + e^{-\alpha L_{ij}}}{1 - e^{-\alpha L_{ij}}}\right) + \ln\left(1 - e^{-2\alpha L_{ij}}\right)\right] \tag{99}$$

$$\Phi_{r\,\max} = \max_s\left(\Phi_r(s)\right) = \max_s\left(\Phi_A(s) = \Phi_R(s)\right) \tag{100}$$

$$W_{ij} = W_{ji} = 2\int_2^\infty \left\{\frac{e^{(\Phi_r(s)/k_s r_e)}}{s^2}\right\} ds \tag{101}$$

$$W_{ij} \approx \left(\frac{r_i + r_j}{4kr_i r_j}\right) e^{(\Phi_{r\,\max}/k_s r_e)} \tag{102}$$

$$B_{ij} = B_{ji} = \left(\frac{2r_j k_B T_e}{3r_i \mu W_{ij}}\right)\left(1 + \frac{r_i}{r_j}\right)^2 \tag{103}$$

$$\begin{aligned} V_e \rho_e c_e \frac{dT_e}{dt} = {} & \rho_f c_f Q_f T_f - \rho_e c_e Q_e T_e + (-\Delta H_{pa}) R_{pac} V_e M_a \\ & + (-\Delta H_{pb}) R_{pbc} V_e M_b - U_j A_j (T_e - T_j) \end{aligned} \tag{104}$$

$$\frac{dT_e}{dt} = 0 \tag{105}$$

$$X_a = \frac{C_{aqe} M_a}{C_{aqe} M_a + C_{bqe} M_b} \tag{106}$$

$$\chi = \frac{C_{aqe} M_a + C_{bqe} M_b}{C_{aqe} M_a + C_{bqe} M_b + C_{ae} M_a + C_{be} M_b} \tag{107}$$

APENDICE B

SIMBOLOGIA EMPLEADA

A	área, constante de Hamaker, factor de Arrhenius
A	monómero A
B	coeficiente de coagulación de Muller
B	monómero B
C	concentración
D	difusitividad
E	energía
F	velocidad de flujo molar
G	velocidad de formación de partículas
H	entalpía
I	fuerza iónica, función modificada de Bessel
I	iniciador
K	coeficiente de partición
L	distancia de superficie a superficie
M	peso molecular
N	número de concentración de partículas
N_A	número de Avogadro
Q	velocidad de flujo volumétrico
Q	polímero muerto
R	velocidad de reacción, constante de los gases
R	distancia de centro a centro
T	temperatura
U	coeficiente de transferencia de calor

V	volumen
W	estabilidad de radio
X	fracción peso
X	agente de transferencia de cadena
Y	electrolíto adicionado
a	moléculas por unidad de área
b	constante isotérmica
c	coeficiente de terminación, capacidad calorífica
e	base natural de logarítmos
e_L	carga del electrón
f	eficiencia del iniciador
j	longitud de cadena
k	constande de velocidad cinética
k_B	constante de Botzmann
m	número de precursores en partículas latex
n	número de radicales por partícula
$\bar{n}$	número promedio de radicales por partícula
r	radio
r_{FH}	radio de Fory - Huggins
t	tiempo
v	número de concentración de precursores
z	valencia iónica
α	parámetro destino
β	parámetro coagulativo
γ	radio de concentración de radicales Ca a Cb
δ	grosor de la capa
Δ	operador diferencial
ϵ_0	permitividad del vacio
ϵ_r	constante dieléctrica
ς	potencial zeta
θ	fracción de superficie

λ variable intermedia

μ viscosidad

ν orden en la función modificada de Bessel

ξ argumento en la función modificada de Bessel

ρ densidad, velocidad de radicales por partícula

σ densidad de la carga superficial

τ factor filtro

υ volumen de partículas o precursores

ϕ fracción volumen

Φ energía potencial

χ conversión

ψ potencial eléctrico de la superficie

ω fracción de carga

SUBINDICES Y SUPERINDICES

a monómero A

b monómero B

cr crítico

d desorción

e entrada, emulsión

f alimentación

g surfactante generado

h homogéneo

i iniciador

i índices

I grupos con extremo iónico

j indices, chaqueta

k indices

m monómero, transferencia de masa

max máximo

n	número de radicales por partícula
o	valor estándar
p	propagación
q	polímero muerto
r	radicales
s	surfactante adicionado
sat	saturado
t	terminación
T	total, térmico
v	precursor de partículas
w	fase acuosa
x	transferencia de cadena
y	electrolíto adicionado
z	valencia
.	especies radicales
0	condición inicial

VARIABLES

Con la combinación de ésta simbología formamos el nombre de las variables utilizadas en el programa, las siguentes son algunos ejemplos.

Riw	velocidad de reacción de iniciador en la fase acuosa
Rge	vel. de reacción de surfactante generado en emulsión
Rie	velocidad de reacción del iniciador en emulsión
Rqe	velocidad de reacción de polímero muerto en emulsión
Rpxe	velocidad de reacción de transferencia de cadena en emulsión
Rpbe	velocidad de reacción en la fase de propagación del
kpaa	constantes cinéticas de velocidad de propagación del polímero aa
kxaa	constantes cinéticas de velocidad de transferencia de cadena
Cxp	concentración de propagación de transferencia de cadena

Cap	concentración de propagación del polímero a
Carp	concentración radical de propagación de polímero a
Ciw	concentración de iniciador en la fase acuosa
Cww	concentración de agua en fase acuosa
Caqe	concentración total de monómero a en emulsión
Cae	concentración de monómero a en emulsión
Caw	concentración de monómero a en la fase acuosa
Carw	concentración radical de monómero a en la fase acuosa
Vm	volumen de monómero
Ve	volumen de emulsión
Vp	volumen de propagación
Vw	volumen fase acuosa
fiq	fracción de volumen de polímero muerto
eQe	flujo volumétrico en emulsión
roq	densidad de polímero muerto
roa	densidad de monómero a
Ma	peso molecular de el monómero a
row	densidad del agua
gamap	concentración radical de radio de ca a cb
ktaa	constante cinética de velocidad de terminación de monómero a
up	número de concentración de precursores de propagación
gama	radio de concentración radical de Ca a Cb
k0a	constante cinética de velocidad inicial de monómero a
Dwa	difusitividad en la fase acuosa del monómero a
kawp	constante cinética de velocidad de propagación de en la fase acuosa
rp	radio de polímero
Dpa	difusividad de propagación del monómero a
k0b	constante cinética de velocidad inicial de monómero b
kda	constante cinética de vel. de desorcion del monómero a
k0a	constante cinética de velocidad inicial de monómero a
kea	constante cinética de velocidad de monómero a en emulsión

kma	constante cinética de transferencia de masa de monómero a
$B(1,m)$	coeficiente de coagulación de Muller b en emulsión
Rtw	velocidad de reacción de terminación en la fase acuosa
$jbja$	longitud de cadena del monómero b entre la longitud de cadena
Ghe	velocidad de formación de partículas homogéneas en emulsión
$Kv(k)$	coeficiente de partición k de precursores de partículas
$Gce(i)$	velocidad de formación de partículas látex
roe	densidad de entrada
ce	capacidad calorífica de entrada
rof	densidad de alimentación
cf	capacidad calorífica de alimentación
Hpa	entalpía de la fase de propagación del monómero a
Uj	coeficiente de transferencia de calor de la chaqueta
Aj	área de la chaqueta
Tj	temperatura de la chaqueta
Xa	composición de polímero
ji	conversión promedio de copolímero