



49
205

UNIVERSIDAD NACIONAL AUTONOMA
DE MEXICO

FACULTAD DE QUIMICA

CODIFICACION DE METODOS
NUMERICOS CON LOTUS 1-2-3

T E S I S

QUE PARA OBTENER EL TITULO DE:

INGENIERO QUIMICO

P R E S E N T A :

JUAN CARLOS JIMENEZ BEDOLLA

MEXICO, D. F.

1991

FALLA DE ORIGEN



UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso

DERECHOS RESERVADOS © PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

CONTENIDO.

	Página
Introducción	i
Objetivos	iii
Capítulo I: Generalidades	
Generalidades sobre métodos numéricos	i
Capítulo II: Breve descripción de los métodos numéricos y su representación en diagramas de flujo	
Raíces.	
Métodos que usan intervalos	4
Método de Bisección	6
Método de la regla falsa	10
Método de sustitución directa	14
Método de Newton-Raphson	16
Método de la Secante	19
Método de Wegstein	21
Método de Muller	23
Método de Hooke-Jeves	25
Ecuación cuadrática	27
Ecuación cúbica	29
<i>Solución de sistemas de ecuaciones</i>	
Ecuaciones algebraicas lineales y su aplicación a la ingeniería	31
Inversión de matrices	33

	Página
Metodo de Gauss-Seidel	35
Metodo de Jacobi	38
<i>Ajuste de curvas</i>	
Regresión lineal	40
Regresión polinomial	43
Regresión múltiple	44
Polinomio de interpolación	45
Polinomio fundamental de Newton	47
<i>Integración</i>	
Regla del Trapecio	49
Regla de Simpson	52
Regla de Simpson 3/8	55
Integración de Romberg	57
Método de Gauss generalizado	59
<i>Ecuaciones diferenciales</i>	
Metodo de Euler	63
Metodo de Euler modificado	65
Metodo del Punto medio	67
Metodo de Heun	70
Metodo de Runge-Kutta 3er. orden 5 constantes	72
Metodo de Runge-Kutta 3er. orden 6 constantes	75
Metodo de Runge-Kutta 4o. orden 7 constantes	78
Metodo de Runge-Kutta 4o. orden 10 constantes	80
Metodos de multipasos de integración abierta	82
Metodo de Adams-Bashforth de dos pasos	82
Metodo de Adams-Bashforth de tres pasos	83
Metodo de Adams-Bashforth de cuatro pasos	84

	Página
Metodo de Adams-Bashfoth de cinco pasos	85
Métodos de multipasos de integración cerrada	88
Metodos predictores correctores	89
 Capitulo III: Descripción general de Lotus 1-2-3	
Generalidades sobre Lotus 1-2-3	93
Manejo de fórmulas	95
Funciones incorporadas en la hoja de cálculo de Lotus 1-2-3	95
Manejo de comandos o mandatos	96
Funciones incorporadas a Lotus 1-2-3	97
Creación de Gráficas con Lotus 1-2-3	97
Macroinstrucciones del teclado y su lenguaje de programación	98
 Capitulo IV: Descripción de la hoja de cálculo	
Panel de control	100
La hoja de cálculo	101
Introducción de datos en la hoja de cálculo	102
Utilización de las funciones @	102
Uso de rangos dentro de la hoja de trabajo	103
 Capitulo V: Comandos de Lotus 1-2-3	
Menús de comandos	104
Comandos aplicables a rangos	105
Nombrar un rango	105

	Pagina
Análisis de regresión	106
Sistemas de ecuaciones lineales simultáneas usando Lotus 1-2-3	108

Capítulo VI: Macroinstrucciones de Lotus 1-2-3

Macroinstrucciones o macros de Lotus 1-2-3	110
Configuración de las macroinstrucciones	111
Lenguaje de comandos para macros	112
Elementos de configuración de las macros	113
Los comandos invisibles de las macros	116
Creación y actividades de macros	117
Lenguaje de macrocomandos de Lotus 1-2-3	120
Reglas gramaticales que rigen a los comandos para macros	120
Pruebas condicionales en una macroinstrucción	124
Término de la ejecución de una macroinstrucción	126
Comandos para verificación	126
Macrocomandos para almacenar información	126
El comando para almacenar Put	128
Macrocomandos para desactivar la pantalla	128

Capítulo VII: Codificación de los métodos numéricos

Raíces.

Método de Bisección	134
Método de la regla falsa	135
Método de sustitución directa	136
Método de Newton-Raphson	137

	Página
Metodo de la Secante	138
Metodo de Wengstein	139
Metodo de Muller	140
Metodo de Hooke-Jeves	141
Ecuación cuadrática	142
Ecuación cubica	143
<i>Solucion de sistemas de ecuaciones</i>	
Inversión de matrices	144
Metodo de Gauss-Seidel	145
Metodo de Jacobi	146
<i>Ajuste de curvas</i>	
Regresión lineal	147
Regresión polinomial	148
Regresión múltiple	149
Polinomio de interpolación	150
Polinomio fundamental de Newton	151
<i>Integración</i>	
Regla del Trapecio	152
Regla de Simpson	153
Regla de Simpson 3/8	154
Integración de Romberg	155
Metodo de Gauss generalizado	156
<i>Ecuaciones diferenciales</i>	
Metodo de Euler	157
Metodo de Euler modificado	158
Método del Punto medio	159
Metodo de Heun	160

	Página
Método de Runge-Kutta 3er. orden 5 constantes	161
Método de Runge-Kutta 3er. orden 6 constantes	162
Método de Runge-Kutta 4o. orden 7 constantes	163
Método de Runge-Kutta 4o. orden 10 constantes	164
Método de Adams-Bashfoth de dos pasos	165
Método de Adams-Bashfoth de tres pasos	166
Método de Adams-Bashfoth de cuatro pasos	167
Método de Adams-Bashfoth de cinco pasos	168
Método de Milne de 4o. orden	169
Método de Milne de 6o. orden	170
Método de Adams-Moulton	171
Capítulo VIII: Aplicaciones	172
Capítulo IX: Conclusiones	206
Bibliografía	210

INTRODUCCION

Para el ingeniero químico moderno el hecho de "ir a la par con su profesión" implica inevitablemente el uso de las computadoras. Hay pocas actividades cotidianas que de alguna manera no tienen contacto con estas máquinas tan poderosas y rápidas. Ciertamente las computadoras han sido por años un gran aliado de la Ingeniería al desempeñar millares de tareas, tanto analíticas como prácticas, en el desarrollo de proyectos y la solución de problemas en forma más eficiente.

Muchos estudiantes de Ingeniería no explotan bien la capacidad de solución de problemas que tienen las computadoras hasta que están adentrados en su educación.

La revolución de la microelectrónica nos da la oportunidad de integrar la Computación de una manera más efectiva. Las computadoras personales pueden aumentar la capacidad del estudiante de Ingeniería para resolver problemas.

Se eligió el tema de los métodos numéricos como punto principal por sus muchas aplicaciones a la Ingeniería. Ya sea que los ingenieros utilicen software comercial o propio, es necesaria una base sólida en los métodos numéricos para la aplicación efectiva de las computadoras en la solución de problemas de Ingeniería.

Por consiguiente se elaboró esta tesis de tal forma que se puedan programar no solo los métodos numéricos, sino también se dan las bases para poder crear sus propios programas para hacer diferentes tareas en las que se necesite programar.

Los temas están dedicados al área de los métodos numéricos, que tiene importancia directa para el candidato a ingeniero: raíces de

ecuaciones no lineales, ecuaciones algebraicas lineales, ajuste de curvas, integracion y ecuaciones diferenciales ordinarias.

Se dispone de un paquete de software denominado Lotus 123 en donde se codifican los diferentes metodos numericos. Lotus 123 cuenta con una gran potencialidad para crear tablas de resultados y graficar tales resultados; esta potencialidad se puede explotar al maximo para presentar un reporte de resultados mucho mas legible y presentable.

Estos programas proporcionan los criterios de programacion necesarios para crear nuevas aplicaciones. Puede lograrse un progreso mas rapido cuando se emplean con el software conjuntamente.

¿Por que se deben dominar los metodos numericos y la programacion de computadoras para resolver los problemas? Ademas del hecho de que a diario se observa que las computadoras intervienen en las actividades mas comunes en la vida diaria, los metodos numericos y las multiples tareas que se pueden desarrollar mediante la computadora hace de esta combinacion una gran herramienta para solucionar problemas comunes en los procesos ingenieriles.

OBJETIVOS

Los métodos numéricos combinan dos de las herramientas más importantes en el repertorio de la Ingeniería: matemáticas y computadoras. Los métodos numéricos se pueden definir (sin que esto sea muy exacto) como las matemáticas por computadora. Las buenas técnicas de programación aumentan la habilidad para aplicar los conocimientos de los métodos numéricos. En particular, las potencialidades y limitaciones de las técnicas numéricas se aprecian mejor cuando se usan estos métodos para resolver los problemas de Ingeniería utilizando como herramienta una computadora.

Debido a la gran disponibilidad de computadoras personales y dispositivos de memoria magnética, los programas se pueden conservar y usar para toda la carrera. Por lo tanto, uno de los principales objetivos de la presente tesis es que el estudiante obtenga programas útiles para la solución de los problemas que se presentan en el diseño.

Todas las técnicas numéricas van acompañadas de su codificación en macroinstrucciones. Estos programas, desarrollados para computadoras personales (IBM-PC), pueden servir como base para una biblioteca de programas propios.

Están escritos bajo la suposición de que ya se ha tenido una experiencia previa en la programación de computadoras.



Capítulo: I

Generalidades

GENERALIDADES SOBRE METODOS NUMERICOS

Los métodos numéricos son técnicas mediante las cuales es posible formular problemas de tal manera que se puedan resolver utilizando operaciones aritméticas. Aunque hay muchos tipos de métodos numéricos, todos comparten una característica común: Invariablemente los métodos numéricos llevan a cabo un buen número de tediosos cálculos. No es raro que con el desarrollo de las computadoras personales y los paquetes de software, el papel de los métodos numéricos en la solución de problemas de ingeniería haya aumentado considerablemente.

Desde finales de la década de 1940, la multiplicación y disponibilidad de las computadoras y paquetes de software ha llevado a un verdadero avance en cuanto al uso y desarrollo de los métodos numéricos. Al principio, este crecimiento estaba algo limitado por el costo de acceso a computadoras grandes, por lo que se hacían planteamientos analíticos en buena parte de la solución de problemas. Ahora las computadoras personales son de bajo costo y ha dado a mucha gente la oportunidad de tener acceso a las poderosas capacidades de cómputo.

Los métodos numéricos se deben estudiar por lo siguiente:

1. Los métodos numéricos son herramientas extremadamente poderosas para la solución de problemas. Son capaces de manejar sistemas de ecuaciones grandes, no lineales y geométricamente complicadas que son comunes en la práctica de la ingeniería y que,

frecuentemente son imposibles de resolver por medios analíticos.

2. En el transcurso de la carrera es posible que se presente la ocasión de usar software disponible comercialmente que contenga métodos numéricos.

Las técnicas que se presentan son :

1. Raíces de ecuaciones. Estos problemas están relacionados con el valor de una variable o de un parámetro que satisface una ecuación. Son especialmente valiosos en proyectos de ingeniería donde con frecuencia resulta imposible despejar analíticamente parámetros de ecuaciones de diseño.
2. Sistemas de ecuaciones algebraicas lineales. Estos problemas son similares a los de raíces de ecuaciones en el sentido de que están relacionados con valores que satisfacen ecuaciones. Sin embargo, a diferencia de satisfacer una sola ecuación, se busca un conjunto de valores que satisfaga simultáneamente a un conjunto de ecuaciones algebraicas. Las ecuaciones lineales simultáneas resultan de una variedad de problemas. Casi siempre se originan a partir de modelos matemáticos de sistemas grandes de elementos interconectados, como: estructuras, circuitos eléctricos y redes de flujo de fluidos.
3. Ajuste de curvas. Con frecuencia se presenta la necesidad de ajustar curvas a un conjunto dado de datos que representan puntos. Las técnicas que se han desarrollado para este fin pueden dividirse en dos categorías generales: Cuando hay un grado significativo de error asociado a los datos. Para estas

situaciones, la solución es encontrar una curva que represente la tendencia general de los datos sin necesidad de tocar los puntos individuales. En contraste, la interpolación se maneja cuando lo que se quiere es determinar valores intermedios entre datos que estén relativamente libres de error. Para estas situaciones, la solución es ajustar una curva directamente a través de los puntos y usar esta curva para predecir valores intermedios.

4. *Integración.* Una interpretación gráfica de la integración numérica es la determinación del área bajo una curva. La integración tiene muchas aplicaciones para el ingeniero, empezando por la determinación de los centroides de objetos con formas irregulares hasta el cálculo de cantidades totales basadas en conjuntos de medidas discretas.
5. *Ecuaciones diferenciales ordinarias.* Las ecuaciones diferenciales ordinarias tienen un enorme significado en la práctica de la ingeniería. Esto se debe a que muchas leyes físicas están expresadas en términos de la razón de cambio de una cantidad más que en términos de su magnitud. Los ejemplos típicos son desde la predicción demográfica (razón de cambio de la población) hasta la aceleración de un cuerpo en descenso (razón de cambio de la velocidad).

A decorative frame with a central white box containing text. The frame consists of a thick black border with a textured, stippled appearance. The central white box is also bordered by a thick black line. The text is centered within this box.

Capítulo: II

**Breve descripción de los métodos
numéricos y su representación
en diagramas de flujo**

MÉTODOS QUE USAN INTERVALOS

A los métodos que aprovechan el hecho de que una función, típicamente, cambia de signo en la vecindad de una raíz se les llama métodos que usan intervalos porque se necesita de dos valores iniciales para la raíz. Como su nombre lo indica, estos valores deben "encerrar" o estar a cada lado de la raíz. Estos métodos emplean diferentes estrategias para reducir progresivamente el tamaño del intervalo y de esta manera converger a la respuesta correcta.

MÉTODOS GRÁFICOS.

Un método simple para obtener una aproximación a la raíz de la ecuación $f(x)=0$ consiste en graficar la función y observar en donde cruza el eje x . Este punto, que representa el valor de x para el cual $f(x)=0$, proporciona una aproximación inicial de la raíz.

En este punto Lotus 123 es una herramienta muy útil, ya que el paquete cuenta con la opción en el menú principal de graficación. Esta opción es de gran utilidad porque se puede ampliar o reducir el rango de puntos a graficar con gran facilidad y, de esta manera tener una idea más clara del comportamiento de la función para poder detectar la posición aproximada de la raíz, y de esta manera proporcionar los intervalos de búsqueda. La posibilidad de graficar aumenta considerablemente la utilidad de los programas.

Las interpretaciones geométricas, además de proporcionar aproximaciones iniciales de la raíz, son herramientas importantes en el aislamiento de las propiedades de las funciones previendo las fallas de los métodos numéricos. Por ejemplo, las raíces múltiples, esto es, funciones tangenciales al eje x y las funciones discontinuas. La existencia de estos casos dificulta el desarrollo de algoritmos generales que garanticen la localización de todas las raíces en el intervalo. Sin embargo, cuando se usan los métodos expuestos en conjunción con esquemas gráficos, son de gran utilidad en la solución de problemas de muchas raíces.

USO DE GRAFICAS POR COMPUTADORA PARA LA LOCALIZACION DE RAICES.

Las gráficas por computadora pueden informar y acelerar los esfuerzos para localizar raíces de una función. Esto se puede lograr utilizando las opciones que contiene Lotus 123 para este fin. De esta manera es posible entender cómo la graficación por computadora ayuda a localizar las raíces.

METODO DE BISECCION

Los métodos de búsqueda incremental se aprovechan donde la función cambie de signo. Por lo tanto, la localización de cambio de signo (y por ende de la raíz), se logra dividiendo el intervalo en una cantidad definida de subintervalos. Se rastrea cada uno de estos subintervalos para encontrar el cambio de signo. El proceso se repite y la aproximación a la raíz mejora cada vez más a medida que los subintervalos se dividen en subintervalos cada vez más pequeños.

El método de bisección, conocido también como de corte binario, de partición en dos intervalos iguales o método de Bolzano, es un método de búsqueda incremental donde el intervalo se divide siempre en dos. Si la función cambia de signo sobre un intervalo, se evalúa el valor de la función en el punto medio. La posición de la raíz se determina situándola en el punto medio del subintervalo dentro del cual ocurre un cambio de signo. El proceso se repite hasta tener una mejor aproximación. A continuación se muestra un algoritmo para la bisección.

Paso 1. Escójanse los valores iniciales de x_0 y x_1 de forma tal que la función cambie de signo sobre el intervalo. Esto se puede verificar asegurándose de que $f(x_1) \cdot f(x_0) < 0$.

Paso 2. La primera aproximación a la raíz x se determina como:

$$x = \frac{x_0 + x_1}{2}$$

Paso 3. Realicéense las siguientes evaluaciones y determinese en qué subintervalo cae la raíz:

a) Si $f(x_1) \cdot f(x_0) < 0$, entonces la raíz se encuentra en dentro del primer subintervalo. Por lo tanto, hágase $x_1 = x_0$, y continúese en el paso 4.

b) Si $f(x_1) \cdot f(x_0) > 0$, entonces la raíz se encuentra en el segundo subintervalo. Por lo tanto, hágase $x_0 = x_1$, y continúese en el paso 4.

c) Si $f(x_1) \cdot f(x_0) = 0$, entonces la raíz es igual a x_0 y se terminan los cálculos.

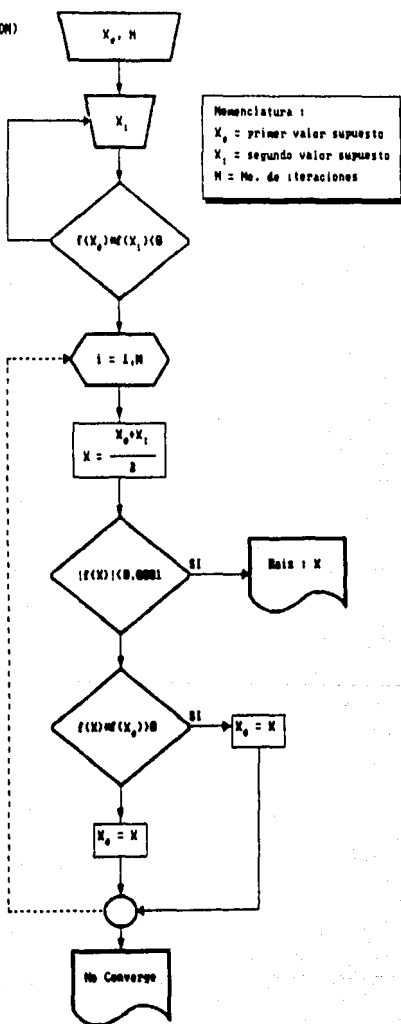
Paso 4. Calcúlese una nueva aproximación a la raíz mediante :

$$x = \frac{x_0 + x_1}{2}$$

Paso 5. Decidase si la nueva aproximación es tan exacta como se desea. Si es así entonces los cálculos se terminan, de otra manera, regrese al paso 3.

El diagrama de flujo se presenta en la siguiente página.

MEDIO INTERVALO (BISECCION)



METODO DE LA REGLA FALSA

Aunque el método de bisección es una técnica perfectamente aceptable para determinar raíces, su enfoque es relativamente ineficiente. Una alternativa mejorada es la del método de la regla falsa, está basada en una idea para aproximarse en forma más eficiente a la raíz.

Una deficiencia del método de bisección es que al dividir el intervalo x_0 a x_1 en mitades iguales, no se toma en cuenta la consideración de la magnitud de $f(x_1)$ y $f(x_0)$. Por ejemplo, si $f(x_1)$ está más cerca de cero que $f(x_0)$ es lógico que la raíz se encuentre más cerca de x_0 que de x_1 . Este método alternativo aprovecha la idea de unir los puntos con una línea recta. La intersección de esta línea con el eje x proporciona una mejor aproximación de la raíz. El reemplazamiento de la curva por una línea recta da una "posición falsa" de la raíz, de aquí el método de regla falsa o en latín, regla falsi. También se le conoce como método de interpolación lineal.

Con el uso de triángulos semejantes, la intersección de la línea recta y el eje x se puede calcular de la siguiente manera:

$$\frac{f(x_1)}{x - x_1} = \frac{f(x_0)}{x - x_0}$$

que se puede resolver para:

$$x = \frac{x_0 f(x_1) - x_1 f(x_0)}{f(x_1) - f(x_0)}$$

Esta es la fórmula de la regla falsa. El valor de x calculado para la ecuación, reemplaza a uno de los dos valores, x_0 o x_1 que produzca un valor de la función que tenga el mismo signo de $f(x)$. De esta manera los valores de x_0 y x_1 siempre encierran a la raíz. El proceso se repite hasta que la aproximación de la raíz sea adecuada.

A continuación se muestra un algoritmo para la regla falsa

Paso 1. Escójanse los valores iniciales de x_0 y x_1 de forma tal que la función cambie de signo sobre el intervalo. Esto se puede verificar asegurándose de que $f(x_1)f(x_0) < 0$.

Paso 2. La primera aproximación a la raíz x se determina como:

$$x = \frac{x_0 f(x_1) - x_1 f(x_0)}{f(x_1) - f(x_0)}$$

Paso 3. Realícense las siguientes evaluaciones y determínese en qué subintervalo cae la raíz:

- a) Si $f(x_1)f(x_0) < 0$, entonces la raíz se encuentra dentro del primer subintervalo. Por lo tanto, hágase $x_1 = x$, y continúese en el paso 4.
- b) Si $f(x_1)f(x_0) > 0$, entonces la raíz se encuentra en el segundo subintervalo. Por lo tanto, hágase $x_0 = x$, y continúese en el paso 4.
- c) Si $f(x_1)f(x_0) = 0$, entonces la raíz es igual a x y se terminan los cálculos.

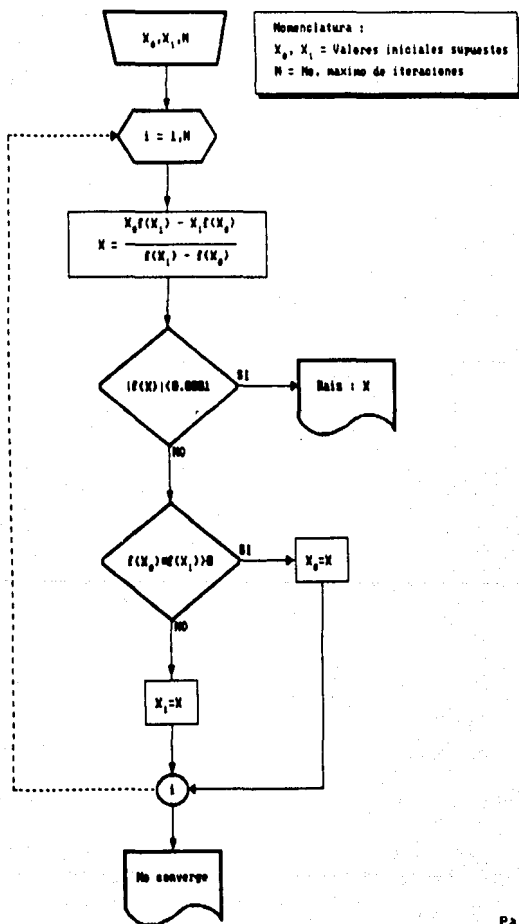
Paso 4. Calcúlese una nueva aproximación a la raíz mediante:

$$x = \frac{x_0 f(x_1) - x_1 f(x_0)}{f(x_1) - f(x_0)}$$

Paso 5. Decidase si la nueva aproximación es tan exacta como se desea. Si es así entonces los cálculos se terminan, de otra manera, regrese al paso 3.

El diagrama de flujo se presenta en la siguiente página.

METODO DE REGULA FALSI



MÉTODOS ABIERTOS

Los métodos abiertos se basan en fórmulas que requieren de un solo valor de x o de un par de ellos pero que no necesariamente encierran a la raíz. Como tales algunas veces divergen o se alejan de la raíz a medida que crece el número de iteraciones. Sin embargo, cuando los métodos abiertos convergen, en general lo hacen mucho más rápido que los métodos que usan intervalos.

MÉTODO DE SUSTITUCIÓN DIRECTA

Los métodos abiertos emplean una fórmula que predice una aproximación a la raíz. Tal fórmula se puede desarrollar para la iteración de punto fijo, reorganizando la ecuación $f(x)=0$ de tal forma que x quede del lado izquierdo de la ecuación:

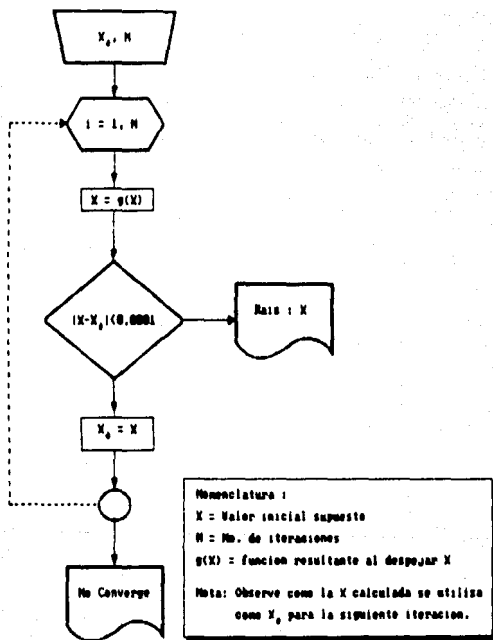
$$x = g(x)$$

La utilidad de esta ecuación es que proporciona una fórmula para predecir un valor de x en función de x . De esta manera, dada una aproximación inicial a la raíz, x_0 , la ecuación se puede usar para obtener una nueva aproximación.

La convergencia ocurre únicamente cuando el valor de la pendiente $g'(x)$ es menor al valor de la pendiente x , esto es cuando $|g'(x)| < 1$.

Ver diagrama de flujo en la página siguiente.

SUSTITUCION DIRECTA



METODO DE NEWTON - RAPHSON

Tal vez, dentro de las fórmulas para localizar raíces, la fórmula de Newton - Raphson, sea la más ampliamente usada.

El método de Newton - Raphson se puede derivar de la serie de Taylor. Esta derivación es muy útil en el sentido de que muestra la penetración de la velocidad de convergencia del método.

La serie de Taylor se puede representar como:

$$f(x_{i+1}) = f(x_i) + f'(x_i)(x_{i+1} - x_i) + \dots + \frac{f''(x_i)}{2}(x_{i+1} - x_i)^2$$

en donde ξ se encuentra en alguna parte del intervalo entre x_i y x_{i+1} . Truncando la serie de Taylor después de la primera derivada, se obtiene una versión aproximada:

$$f(x_{i+1}) \cong f(x_i) + f'(x_i)(x_{i+1} - x_i)$$

En la intersección con el eje x , $f(x_{i+1})$ debe ser igual a cero:

$$0 \cong f(x_i) + f'(x_i)(x_{i+1} - x_i)$$

que se puede resolver para

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$$

Obviamente, una pendiente cero ($f'(x)=0$) es un real desastre que causa una división por cero en la fórmula de Newton - Raphson. Gráficamente, esto significa que la solución se dispersa horizontalmente y jamás toca el eje x .

El diagrama de flujo se presenta en la siguiente pagina.

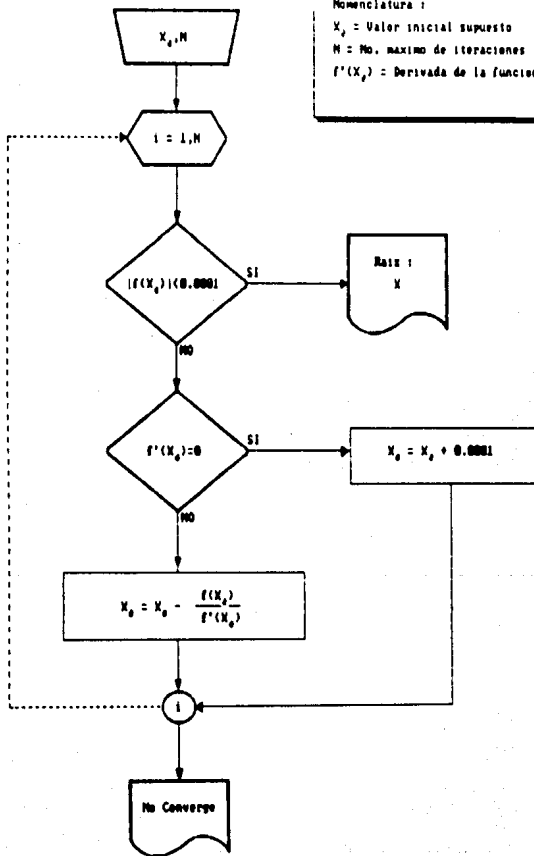
METODO DE NEWTON-RAPHSON.

Nomenclatura :

X_0 = Valor inicial supuesto

M = No. maximo de iteraciones

$f'(X_0)$ = Derivada de la funcion.



METODO DE LA SECANTE

Un problema fuerte en la implementación del método de Newton - Raphson es el de la evaluación de la derivada. Aunque esto no es un inconveniente para los polinomios y para muchas otras funciones, existen algunas de estas cuyas derivadas pueden ser extremadamente difíciles de evaluar. En estos casos la derivada se puede evaluar mediante una diferencia dividida.

$$f'(x) \approx \frac{f(x_{i-1}) - f(x_i)}{x_{i-1} - x_i}$$

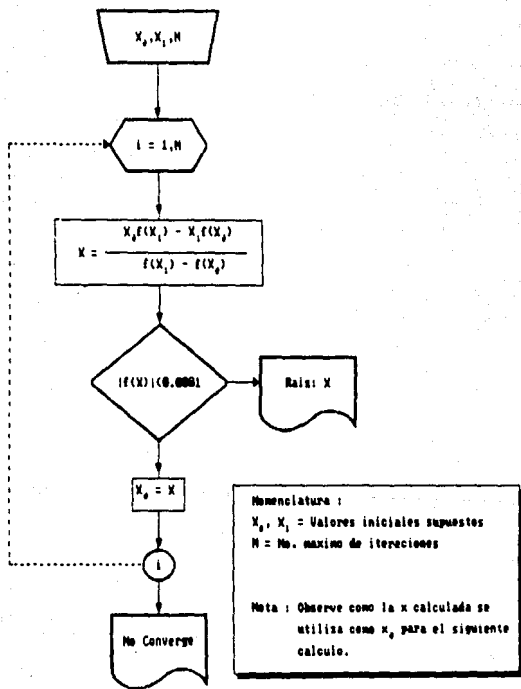
Esta aproximación se puede sustituir en la ecuación obteniendo la ecuación iterativa :

$$x = \frac{x_0 f(x_1) - x_1 f(x_0)}{f(x_1) - f(x_0)}$$

Esta ecuación es la fórmula para el método de la secante. Nótese que el planteamiento requiere de dos puntos iniciales de x . Sin embargo, debido que no se requiere que $f(x)$ cambie de signo entre los valores, a este método no se le clasifica como aquellos que usan intervalos.

El diagrama de flujo se presenta en la siguiente página.

METODO DE LA SECANTE



METODO DE WENGSTEIN

Se desarrolla de acuerdo a los siguientes pasos:

Paso 1. Se transforma la función $f(x)$ a $g(x)$, donde $g(x)=f(x)+x$

Paso 2. Se expande linealmente mediante el polinomio de Newton a $g(x)$

Paso 3. Se aplica la condición de la existencia de una raíz x , para la cual $f(x)=0$.

Desarrollo del algoritmo :

$$g(x) = f(x) + x$$

$$g(x) = g(x_0) + \frac{g(x_1) - g(x_0)}{x_1 - x_0} (x - x_0)$$

Si x es raíz, entonces $g(x) = x$

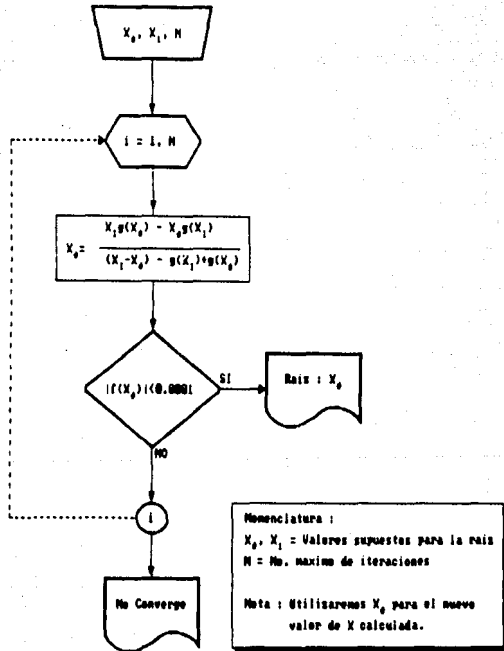
$$x = g(x_0) + \frac{g(x_1) - g(x_0)}{x_1 - x_0} (x - x_0)$$

Desarrollando y despejando x

$$x_0 = \frac{x_1 g(x_0) - x_0 g(x_1)}{(x_1 - x_0) - g(x_1) + g(x_0)}$$

El diagrama de flujo se presenta en la siguiente página.

METODO DE WEGSTEIN



METODO DE MULLER

Se desarrolla mediante los siguientes pasos:

Paso 1. Expansión cuadrática según el polinomio fundamental de Newton para $f(x)$.

Paso 2. Aplicación de la existencia de una raíz x

Desarrollo del algoritmo:

$$f(x) = f(x_0) + f(x_0, x_1)(x - x_0) + f(x_0, x_1, x_2)(x - x_0)(x - x_1)$$

Aplicando la condición de existencia de raíz y agrupando:

$$0 = f(x_0) - x_0 f(x_0, x_1) + x_0 x_1 f(x_0, x_1, x_2) + \\ (f(x_0, x_1) - (x_0 + x_1)f(x_0, x_1, x_2))x + f(x_0, x_1, x_2)x^2$$

Se observa que tiene la forma de una ecuación cuadrática

$ax^2 + bx + c = 0$, donde:

$$a = f(x_0, x_1, x_2)$$

$$b = (f(x_0, x_1) - (x_0 + x_1)f(x_0, x_1, x_2))$$

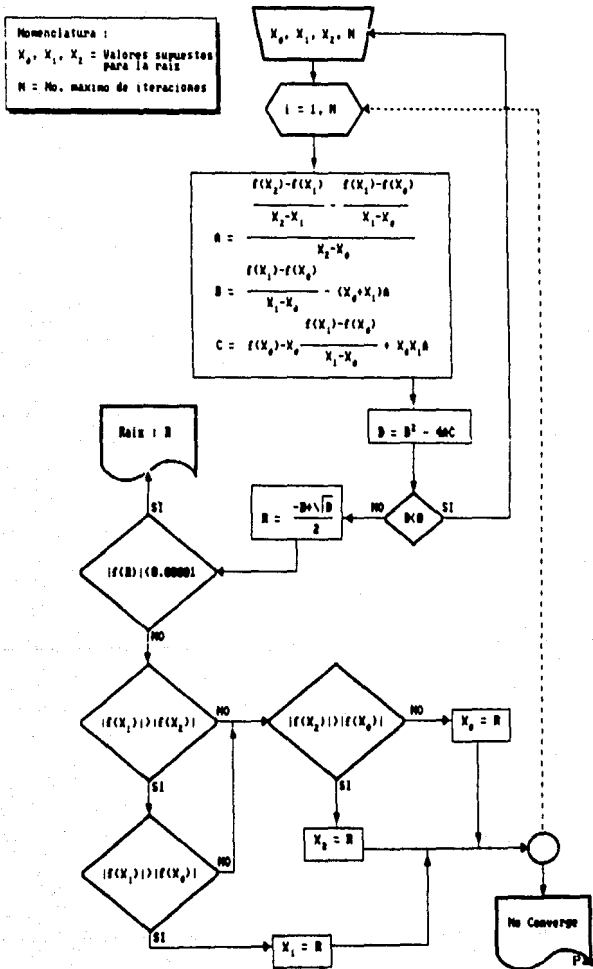
$$c = f(x_0) - x_0 f(x_0, x_1) + x_0 x_1 f(x_0, x_1, x_2)$$

que cuyas raíces son:

$$r_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a} \quad \text{y} \quad r_2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a}$$

El diagrama de flujo se presenta en la siguiente página.

METODO DE MULLER



METODO DE HOOKE - JEEVES

Se desarrolla mediante los siguientes pasos:

Paso 1. Suponga x_0

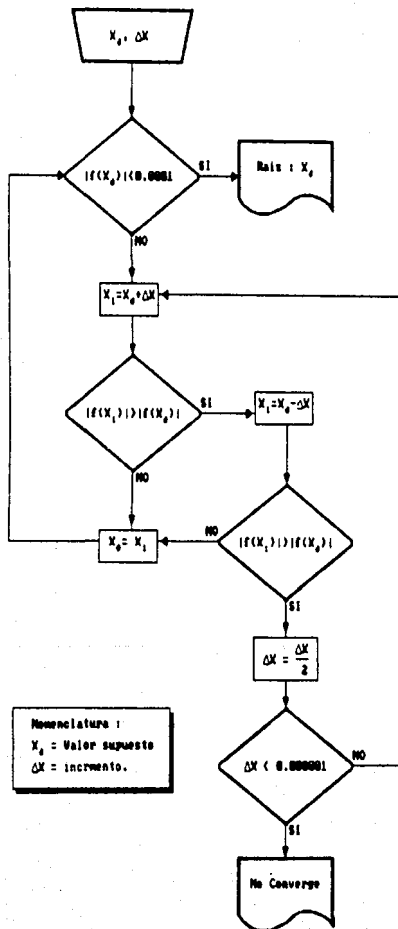
Paso 2. Se da un incremento a x_0 y se verifica si el valor absoluto de la función, en este punto incrementado, es menor que el valor absoluto de la función en x_0 , si es así se sustituye x_0 por x_0 incrementado y se repite el proceso. En caso contrario, se prueba con el incremento de signo contrario.

Paso 3. Si no funciona la técnica para dicho incremento, este se reduce a la mitad y se repite el paso 2.

Si el incremento es demasiado pequeño el método no converge.

El diagrama de flujo se presenta en la siguiente página.

METODO DE HOOKE-JEVES



ECUACION CUADRATICA

Se aplica a las funciones que tienen la forma $ax^2 + bx + c = 0$ y tendrá dos raíces (reales o complejas).

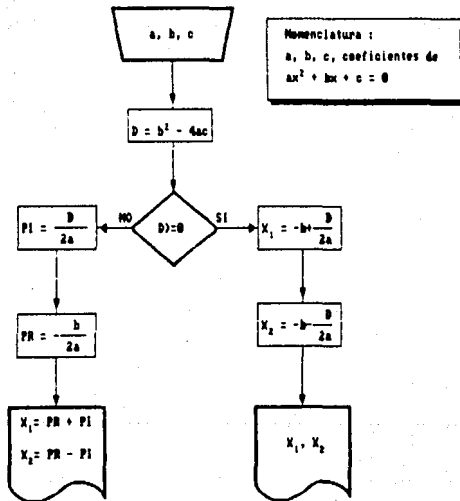
La podemos tratar de la siguiente manera :

$$x^2 + \frac{b}{a}x = -\frac{c}{a}$$
$$x^2 + \frac{b}{a}x + \frac{1}{4}\frac{b^2}{a^2} = \frac{1}{4}\frac{b^2}{a^2} - \frac{c}{a}$$
$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Esto genera dos raíces, según sea \pm y serán complejas si el discriminante ($b^2 - 4ac$) es negativo.

El diagrama de flujo se presenta en la siguiente página.

ECUACION CUADRATICA



ECUACION CUBICA

Se aplica a una ecuación cubica de la siguiente forma:

$$x^3 + 3b_1x^2 + 3b_2x + b_3 = 0$$

La solución de esta ecuación se encuentra realizando los siguientes cambios:

$$b_1 = -\frac{a_1}{3}, \quad b_2 = \frac{a_2}{3} \quad \text{y} \quad b_3 = a_3$$

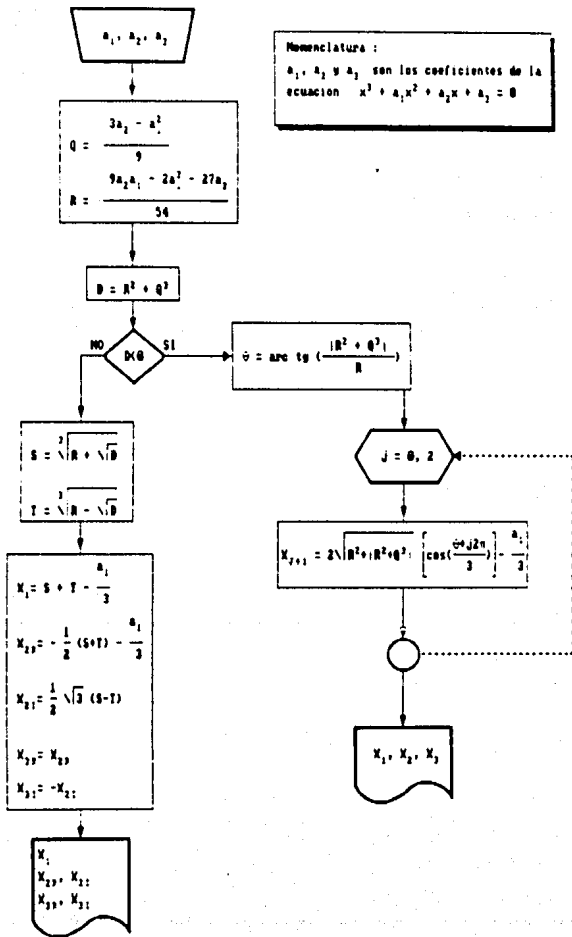
Obteniendo :

$$R = \frac{9a_2a_1 - 2a_1^3 - 27a_3}{54}$$
$$Q = \frac{3a_2 - a_1^2}{9}$$

Con estas modificaciones, podrían utilizarse las ecuaciones ya obtenidas para calcular x_1 , x_2 y x_3 de acuerdo al caso determinado por el valor de $R^2 + Q^3$.

El diagrama de flujo se presenta en la siguiente página.

ECUACION CUBICA



ECUACIONES ALGEBRAICAS LINEALES Y SU APLICACION EN LA INGENIERIA

Muchas de las ecuaciones fundamentales de la ingeniería se basan en las leyes de conservación de la masa, la fuerza, la energía y el momento. En términos matemáticos esto lleva a ecuaciones de equilibrio que representan el comportamiento del sistema; estas ecuaciones consideran las incógnitas a obtener del modelo matemático, las respuestas de la cantidad que se está modelando, las propiedades o características del sistema y los estímulos externos que actúan sobre el sistema.

Cuando estas dependencias se expresan en forma matemática, las ecuaciones que resultan contienen las variables "x", "a" y "c". Las x's, miden las magnitudes y respuestas de los componentes individuales. Las a's, representan las propiedades y características que se refieren a las interacciones entre las componentes. Finalmente, las c's representan los estímulos externos que actúan sobre el sistema.

Los problemas de variables discretas implican componentes finitos acoplados como reactores y circuitos eléctricos. Estos tipos de problemas usan modelos que proporcionan el comportamiento de un sistema en función de ciertas variables.

Los problemas microescalados describen los sistemas con una base continua o semicontinua. La distribución de sustancia sobre un reactor rectangular alargado es un ejemplo. Las ecuaciones diferenciales que se derivan de las leyes de conservación marcan la distribución de la variable dependiente para estos sistemas. Estas ecuaciones diferenciales se pueden resolver numéricamente para convertirlas a un sistema de ecuaciones algebraicas simultáneas. La

solucion de estas ecuaciones en conjunto representa una importante aplicacion en ingenieria. Estas ecuaciones están unidas porque las variables se relacionan entre si y cada una depende de sus vecinas.

METODOS GRAFICOS

En este punto Lotus 123 es una herramienta muy util, ya que el paquete cuenta con la opción de graficacion en el menu principal. Esta opción es de gran utilidad porque se puede ampliar o reducir el rango de puntos a graficar con gran facilidad y, de esta manera tener una idea mas clara del comportamiento de la funcion para poder detectar la intersección de las curvas, y de esta manera proporcionar los intervalos de busqueda. La posibilidad de graficar aumenta considerablemente la utilidad de los programas.

Se obtiene una solución grafica de dos ecuaciones representandolas en coordenadas cartesianas en un eje que corresponda a x_1 y el otro a x_2 . Ya que el problema es para ecuaciones lineales, cada ecuación representa una linea recta. Esto puede ilustrarse facilmente por las ecuaciones generales:

$$a_{11}x_1 + a_{12}x_2 = C_1$$

$$a_{21}x_1 + a_{22}x_2 = C_2$$

Ambas ecuaciones se pueden resolver para x_2 :

$$x_2 = -\left(\frac{a_{11}}{a_{12}}\right)x_1 + \frac{C_1}{a_{12}}$$

$$x_2 = -\left(\frac{a_{21}}{a_{22}}\right)x_1 + \frac{C_2}{a_{22}}$$

De esta manera, las ecuaciones se encuentran ahora en la forma de líneas rectas; esto es, $x_2 = x_1(\text{pendiente}) + (\text{ordenada al origen})$. Estas líneas se pueden graficar en coordenadas cartesianas con x_2 como ordenada y x_1 como abscisa. Los valores de x_1 y x_2 en la intersección de las líneas satisfacen las dos ecuaciones simultáneamente.

INVERSION DE MATRICES

Si una matriz es cuadrada, entonces existe otra matriz $[A]^{-1}$, llamada la matriz inversa de $[A]$, para la cual :

$$[A][A]^{-1} = [A]^{-1}[A] = [I]$$

La aplicación de esta inversa ocurre cuando se necesita resolver varios sistemas de ecuaciones de la forma :

$$[A][X] = [C]$$

En lugar de resolver cada sistema por separado, una alternativa diferente consiste en determinar la inversa de la matriz de coeficientes. Entoces se puede usar la ecuación

$$[X] = [A]^{-1}[C]$$

para obtener las soluciones simplemente multiplicando la matriz $[A]^{-1}$ por el vector de términos independientes correspondiente $[C]$.

Los elementos de $[X]$ representan los valores de las variables que se están equilibrando para cada una de las partes del sistema.

El vector $[C]$ de términos independientes contiene aquellos elementos del balance que son independientes del comportamiento del sistema, esto es, son constantes. Como tales, representan las fuerzas externas o los estímulos que manejan al sistema.

Finalmente, la matriz $[A]$ de coeficientes contiene, en general, los parámetros que expresan cómo interactúa el sistema. Por consiguiente la ecuación se puede escribir como :

$$[\text{iteraciones}] [\text{respuestas}] = [\text{estímulos}]$$

El uso de la matriz inversa lleva a un resultado particularmente interesante. La solución formal se puede expresar como :

$$[X] = [A]^{-1}[C]$$

De esta manera el coeficiente a_{11}^{-1} es una constante de proporcionalidad que proporciona el valor de x_1 debido al nivel unitario c_1 . Este resultado es independiente de los efectos de c_2 y c_3 sobre x_1 , los cuales se reflejan sobre a_{12}^{-1} y a_{13}^{-1} respectivamente. Por lo tanto, se puede decir que el elemento a_{11}^{-1} de la matriz invertida representa el valor de x_2 debido a la cantidad unitaria de c_1 .

Debido a que la inversión y la multiplicación de matrices es una función integrada de 1-2-3, no se presentará diagrama de flujo.

METODO DE GAUSS - SEIDEL

El método de Gauss - Seidel es el método iterativo más usado. Supongase que se tiene un sistema de n ecuaciones del tipo

$$[A][X] = [C]$$

si los elementos de la diagonal son diferentes de cero, la primera ecuación se puede resolver para x_1 , la segunda para x_2 , etcetera, lo que lleva a :

$$x_1 = \frac{c_1 - a_{12}x_2 - a_{13}x_3 - \dots - a_{1n}x_n}{a_{11}}$$

$$x_2 = \frac{c_2 - a_{21}x_1 - a_{23}x_3 - \dots - a_{2n}x_n}{a_{22}}$$

$$x_3 = \frac{c_3 - a_{31}x_1 - a_{32}x_2 - \dots - a_{3n}x_n}{a_{33}}$$

⋮
⋮
⋮
⋮

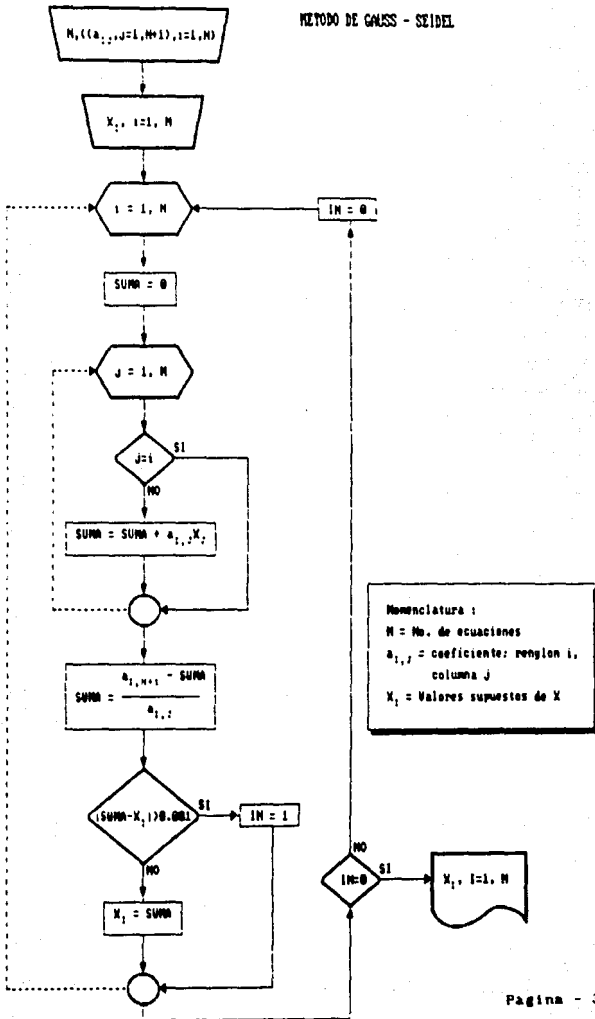
$$x_n = \frac{c_n - a_{n1}x_1 - a_{n2}x_2 - \dots - a_{n,n-1}x_{n-1}}{a_{nn}}$$

Ahora se puede empezar el proceso de solución usando un valor inicial para las x .

Notese que en este método, a medida que se calcula un nuevo valor de x , este mismo se usa inmediatamente en la siguiente ecuación que a su vez determina una nueva x . De esta manera, si la solución es convergente, se usa la mayor aproximación posible.

En la siguiente pagina se muestra el diagrama de flujo.

METODO DE GAUSS - SEIDEL



METODO DE JACOBI

En vez de usar el último valor calculado de las x , usa la ecuación:

$$x_1 = \frac{C_1 - a_{12}x_2 - a_{13}x_3 - \dots - a_{1n}x_n}{a_{11}}$$

$$x_2 = \frac{C_2 - a_{21}x_1 - a_{23}x_3 - \dots - a_{2n}x_n}{a_{22}}$$

$$x_3 = \frac{C_3 - a_{31}x_1 - a_{32}x_2 - \dots - a_{3n}x_n}{a_{33}}$$

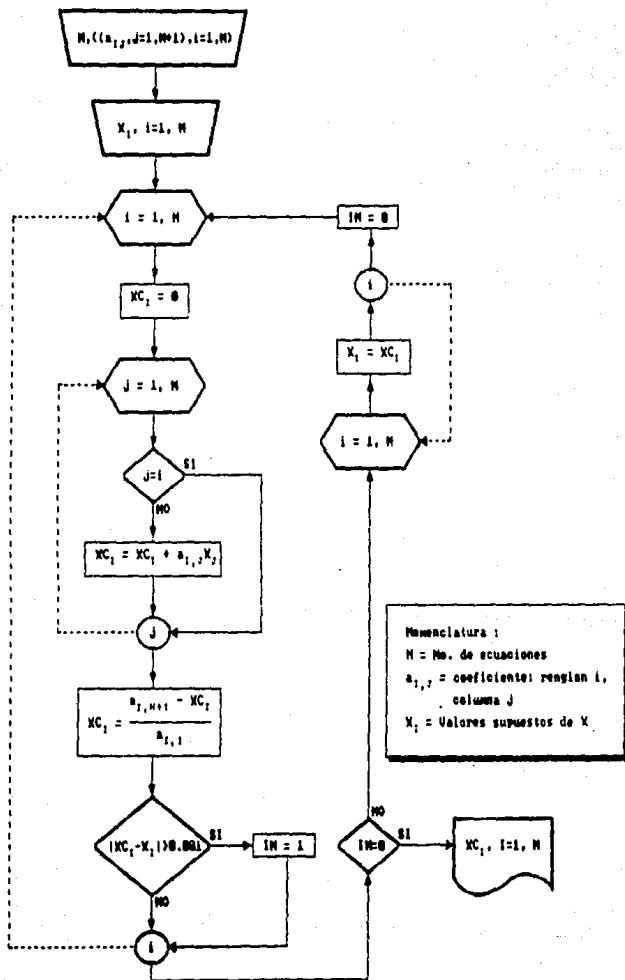
⋮
⋮
⋮

$$x_n = \frac{C_n - a_{n1}x_1 - a_{n2}x_2 - \dots - a_{nn-1}x_{n-1}}{a_{nn}}$$

para calcular un nuevo valor de x en base a la aproximación anterior de las x . De esta forma, al generar un nuevo valor no se usa de inmediato sino que se almacena para la siguiente iteración.

En la siguiente página se muestra el diagrama de flujo.

METODO DE JACOBI



REGRESION LINEAL

Los datos experimentales generalmente están acompañados de errores y por lo tanto la interpolación polinomial resulta inadecuada. Una estrategia apropiada es obtener una función aproximada que ajuste el comportamiento o la tendencia general de los datos, sin pasar necesariamente por todos los puntos.

La manera de encontrar este ajuste es obtener una curva que minimice la diferencia de los datos y la curva (regresión lineal con mínimos cuadrados).

El ejemplo más simple de una aproximación con mínimos cuadrados es el ajuste de una línea recta a un conjunto de parejas de datos. La expresión matemática de una línea recta es :

$$y = a_0 + a_1 x + E$$

en donde a_0 y a_1 son coeficientes que representan la intersección con el eje de las abscisas y la pendiente respectivamente y E es el residuo entre el modelo y las observaciones.

$$E = y - a_0 - a_1 x$$

Por lo tanto el error o residuo es lo que difiere entre el valor real de y y el valor aproximado, $a_0 + a_1 x$, predicho por la ecuación lineal.

Una estrategia para obtener la mejor línea que pase más cerca de los puntos es minimizar la suma de los cuadrados de los residuos,

$$\sum_{i=1}^n E_i^2 = \sum_{i=1}^n (y_i - a_0 - a_1 x_i)^2$$

En donde las constantes a_0 y a_1 están dadas por

$$a_1 = \frac{n \sum x_i y_i - \sum x_i \sum y_i}{n \sum x_i^2 - (\sum x_i)^2}$$

$$b_0 = \bar{y} - a_1 \bar{x}$$

en donde \bar{y} y \bar{x} son la media de y y x respectivamente.

La regresión lineal proporciona una técnica muy poderosa para poder ajustar datos a una mejor línea. Sin embargo, se considera que la relación de las variables es lineal. Esto no se cumple en todos los casos, y el primer paso en cualquier análisis de regresión es graficar los puntos para analizar visualmente la tendencia, en este paso Lotus 123 es muy útil graficando rápidamente los puntos a analizar y de esta manera decidir si es correcto aplicar un modelo lineal. De no ser así, se pueden hacer transformaciones que expresen los datos de manera que se pueda aplicar la regresión lineal.

Un ejemplo es el modelo exponencial :

$$y = a_1 \text{Exp}(b_1 x)$$

en donde a_1 y b_1 son constantes.

Otro ejemplo de un caso no lineal es la ecuación elevada a una potencia :

$$y = a_2 x^{b_2}$$

en donde a_2 y b_2 son coeficientes.

Un tercer ejemplo de un modelo no lineal es la ecuación de promedio de crecimiento de saturación :

$$y = a_3 \frac{x}{b_3 + x}$$

en donde a_3 y b_3 son coeficientes constantes.

En seguida se puede aplicar la regresión lineal para ajustar las ecuaciones a los datos.

Por ejemplo, el modelo exponencial se puede linearizar mediante logaritmos naturales para obtener :

$$\text{Ln } y = \text{Ln } a_1 + b_1 x$$

Por lo tanto una gráfica de Ln y contra x forma una línea recta con pendiente b_1 y ordenada al origen $\text{Ln } a_1$.

La segunda ecuación se puede linearizar Tomando logaritmos de base 10 :

$$\text{Log } y = b_2 \text{Log } x + \text{Log } a_2$$

Por lo tanto una gráfica de Log y contra Log x forma una línea recta con pendiente b_2 y ordenada al origen $\text{Log } a_2$.

Y por último la tercera ecuación se linealiza mediante :

$$\frac{1}{y} = \frac{b_3}{a_3} \frac{1}{x} + \frac{1}{a_3}$$

Por lo tanto, una gráfica de $1/y$ contra $1/x$ forma una línea recta con pendiente b_3/a_3 y ordenada al origen $1/a_3$.

En este caso no se presenta el diagrama de flujo ya que la regresión lineal es una función integrada dentro de Lotus 123.

REGRESION POLINOMIAL

Otra alternativa es ajustar polinomios a los datos usando regresión polinomial.

El mismo procedimiento para los mínimos cuadrados se extiende para ajustar los datos a un polinomio de n -ésimo grado :

$$y = a_0 + a_1x + a_2x^2 + \dots + a_nx^n$$

En este caso no se presenta el diagrama de flujo ya que la regresión polinomial es una función integrada dentro de Lotus 123.

REGRESION MULTIPLE

Una extension util en la regresion lineal es el caso en que y es una funcion lineal de dos o mas variables. Por ejemplo, y pudiera ser una funcion lineal de x_1 y x_2 , de manera que :

$$y = a_0 + a_1 x_1 + a_2 x_2$$

Esta ecuacion es util cuando se ajustan datos experimentales en donde la variable que se esta analizando es funcion de varias variables. En este caso bidimensional, la "linea" de regresion viene a ser un plano.

En este caso no se presenta el diagrama de flujo ya que la regresion lineal multiple es una funcion integrada dentro de Lotus

123.

POLINOMIO DE INTERPOLACION

Cuando se tiene que estimar valores intermedios entre los valores conocidos. El metodo mas comunente empleado es la interpolacion polinomial.

Esta tecnica consiste en determinar los coeficientes del polinomio de grado $n-1$ que representa en forma mas exacta a los n datos, o bien para determinar los coeficientes del modelo lineal de pendiente de $x_{11}, x_{21}, \dots, x_{n-11}$.

Notese que esta tecnica sólo se aplica a los dos siguientes modelos :

$$y_1 = a_0 + a_1 x_{11} + a_2 x_{11}^2 + \dots + a_{n-1} x_{11}^{n-1}$$

$$y_1 = a_0 + a_1 x_{11} + a_2 x_{21} + \dots + a_{n-1} x_{n-11}$$

afortunadamente, la gran mayoria de los modelos formulados para los resultados experimentales pueden transformarse a las dos formas anteriores. La tecnica de estimación de parametros consiste en substituir cada uno de los datos en el modelo lineal propuesto. Lo que conduce a un sistema de n ecuaciones con n incognitas.

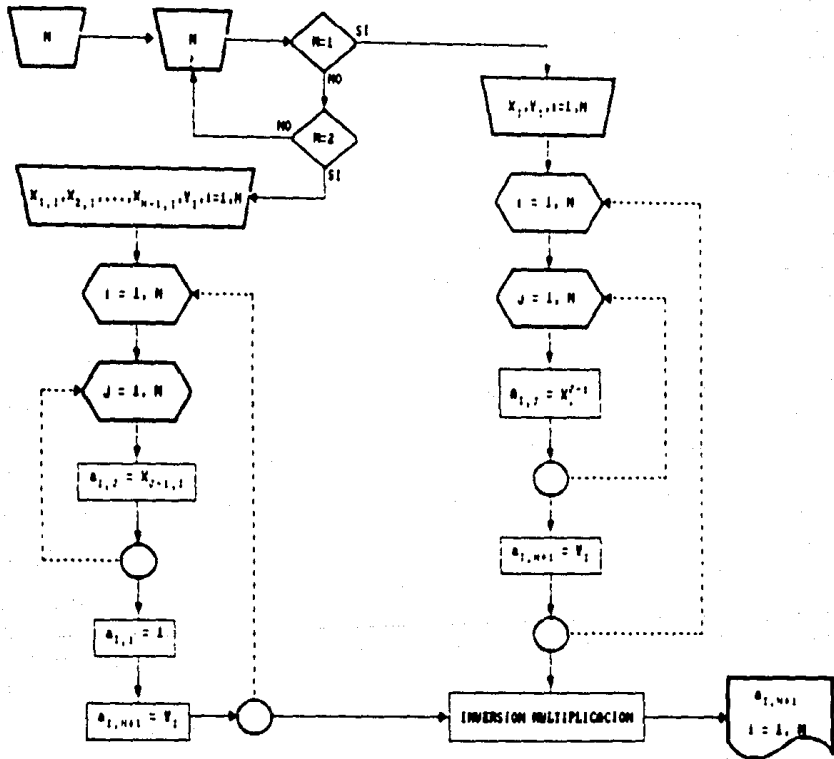
De manera similar para el otro modelo se obtendrán n ecuaciones con n incognitas.

La desventaja de este procedimiento es que para muchos datos, el grado del polinomio será muy grande, e inmanejable.

Por otro lado, este procedimiento unicamente garantiza el que todos los datos están garantizados pero no así en puntos intermedios, y puede haber problemas de oscilacion del polinomio.

En la siguiente pagina se muestra el diagrama de flujo.

POLINOMIO DE INTERPOLACION



Notación:
 N = No. de datos
 N = Tipo de Modelo
 para N = 1 : $V_i = a_0 + a_1 X_i + a_2 X_i^2 + \dots$
 para N = 2 : $V_i = a_0 + a_1 X_{i,1} + a_2 X_{i,1}^2 + \dots$
 Después de Inversión Multiplicación : $a_i = a_{i+1, N+1}$ para $i=0, N$

POLINOMIO FUNDAMENTAL DE NEWTON

Consiste en expandir la función (variable dependiente), en un polinomio de diferencias finitas divididas. Esta técnica difiere de la anterior en la estructura del polinomio que representa a los n datos. Debido a dicha diferencia se evita el resolver un sistema de n ecuaciones, con lo cual el proceso de estimación de parámetros será rápido, pero el polinomio tendrá la siguiente estructura :

$$y_i = a_0 + a_1(x - x_1) + a_2(x - x_1)(x - x_2) + \dots + a_{n-1}(x - x_1)(x - x_2)\dots(x - x_{n-1})$$

en donde $a_0, a_1, a_2, \dots, a_{n-1}$ son diferencias finitas divididas evaluadas en x_1 .

$a_0 = f[x_1]$ diferencia de orden cero

$a_1 = f[x_1, x_2]$ diferencia de orden uno

$a_2 = f[x_1, x_2, x_3]$ diferencia de orden dos

.

.

.

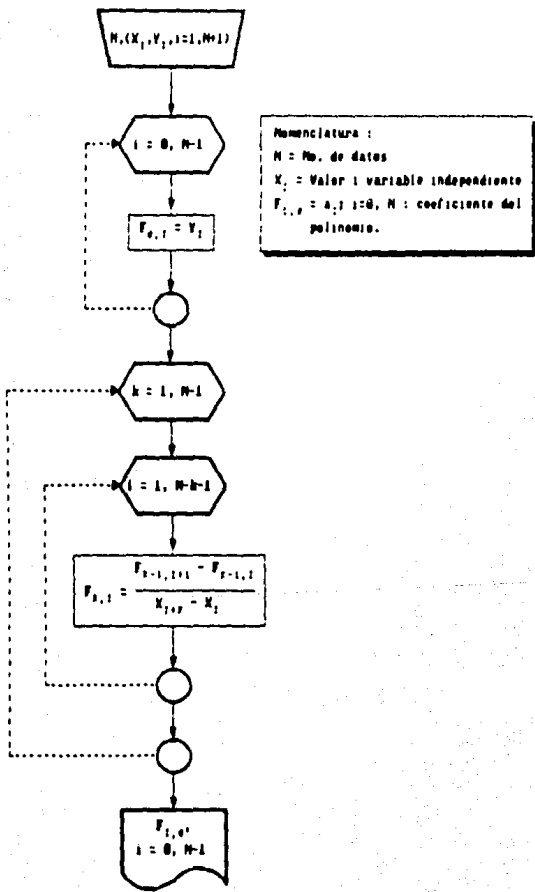
etc.

de manera que pueden ser evaluadas a partir de los datos.

Presenta las mismas desventajas de la técnica anterior y ambas conducen al mismo polinomio. Un polinomio que represente los n datos requerirá expandirse hasta la diferencia de $n-1$.

En la siguiente página se muestra el diagrama de flujo.

POLINOMIO FUNDAMENTAL DE NEWTON



Nomenclatura :
 N = No. de datos
 Xj = Valor i variable independiente
 Fj,i = aij i=0, N : coeficiente del polinomio.

REGLA DEL TRAPECIO

La regla del trapecio o regla trapezoidal se puede representar como :

$$f(x) = f(a) + \frac{f(b) - f(a)}{b - a} (x - a)$$

El Area bajo la linea recta es la aproximación de la integral de $f(x)$ entre los límites a y b :

$$I \approx \int_a^b \left[f(a) + \frac{f(b) - f(a)}{b - a} (x - a) \right] dx$$

El resultado de la integración es :

$$I \approx (b - a) \frac{f(a) + f(b)}{2}$$

al que se llama regla trapezoidal.

Una manera de mejorar la exactitud de la regla trapezoidal es la de dividir el intervalo de integración de a a b en un conjunto de segmentos y aplicar el método a cada uno de los segmentos, en seguida se suman las áreas de los segmentos individuales y se obtiene la integral sobre el intervalo completo.

Hay $n+1$ puntos base igualmente espaciados; por consiguiente, hay n segmentos de igual anchura.

$$h = \frac{b - a}{n}$$

Si a y b se igualan a x_0 y a x_n , respectivamente, la integral se representa como :

$$I = \int_{x_0}^{x_1} f(x)dx + \int_{x_1}^{x_2} f(x)dx + \dots + \int_{x_{n-1}}^{x_n} f(x)dx$$

sustituyendo la regla trapezoidal para cada uno de las integrales, se obtiene :

$$I \approx h \left[\frac{f(x_0) + f(x_1)}{2} + h \frac{f(x_1) + f(x_2)}{2} + \dots + h \frac{f(x_{n-1}) + f(x_n)}{2} \right]$$

agrupando terminos :

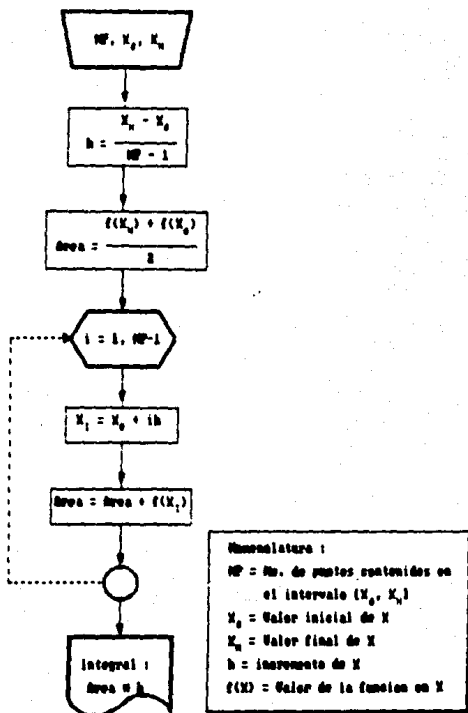
$$I \approx \frac{h}{2} \left[f(x_0) + \sum_{k=1}^{n-1} f(x_k) + f(x_n) \right]$$

En la forma mas general de la ecuacion se obtiene :

$$I \approx (b - a) \frac{f(x_0) + \sum_{k=1}^{n-1} f(x_k) + f(x_n)}{2n}$$

El diagrama de flujo se presenta en la siguiente pagina.

REGLA DEL TRAPEZIO



Abreviatura :

NP = No. de puntos contenidos en el intervalo (X₀, X_n)

X₀ = Valor inicial de X

X_n = Valor final de X

h = incremento de X

f(X) = Valor de la función en X

REGLA DE SIMPSON

La regla de Simpson de 1/3 resulta cuando se sustituye un polinomio de segundo orden en la ecuación

$$I = \int_a^b f(x) dx \approx \int_a^b f_2(x) dx$$

si a y b se denominan como x_0 y x_2 , y $f_2(x)$ se representa mediante un polinomio de Lagrange de segundo orden, entonces la integral es :

$$I \approx \int_{x_0}^{x_2} \left[\frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)} f(x_0) + \frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)} f(x_1) + \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)} f(x_2) \right] dx$$

Después de integrar y de reordenar términos

$$I \approx \frac{h}{3} [f(x_0) + 4f(x_1) + f(x_2)]$$

donde en este caso $h = (b-a)/2$, $a = x_0$, $b = x_2$ y x_1 es el punto medio entre a y b dado por $(b+a)/2$.

Así como la regla trapezoidal, la regla de Simpson se puede mejorar dividiendo el intervalo de integración en segmentos de igual anchura.

$$h = \frac{b - a}{n}$$

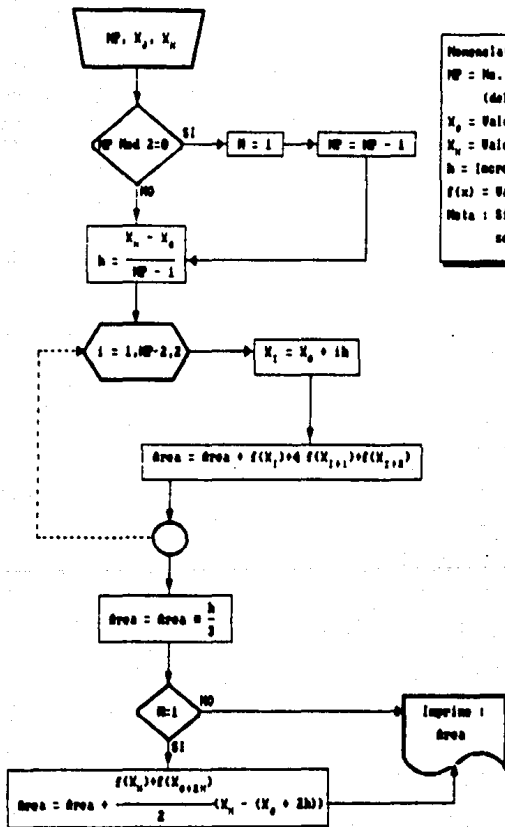
quedando :

$$S \approx (b - a) \frac{f(x_0) + \frac{2}{3} f(x_1) + \frac{2}{3} f(x_2) + f(x_3)}{3n}$$

Notese que se debe usar un numero par de segmentos para utilizar esta tecnica.

En la siguiente pagina se muestra el diagrama de flujo.

REGLA DE SIMPSON



Notación:

NP = No. de puntos contenidos en el intervalo
(debe ser par) (X_0, X_N)

X_0 = Valor inicial de X

X_N = Valor final de X

h = incremento de X

$f(x)$ = Valor de la función en X

Nota: Si NP es impar, el último subintervalo se integra con la regla del trapecio.

REGLA DE SIMPSON DE 3/8

Se puede ajustar polinomios de Lagrange de tercer orden a cuatro puntos e integrar :

$$I = \int_a^b f(x)dx \approx \int_a^b f_3(x)dx$$

para obtener

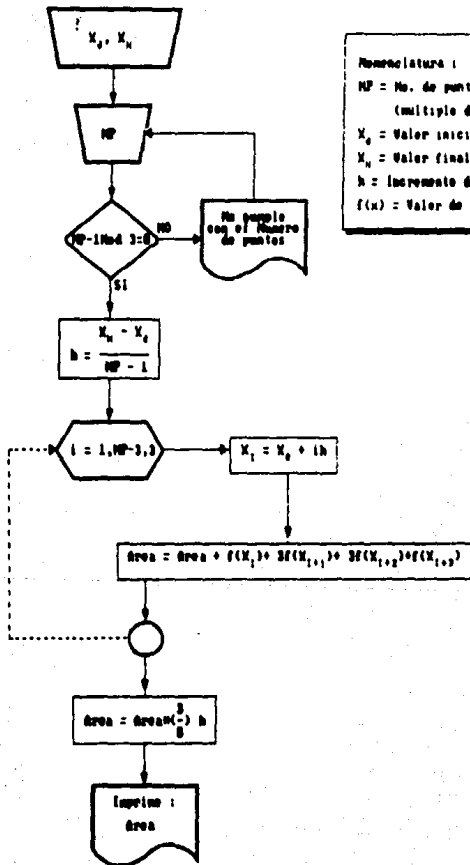
$$I \approx \frac{3h}{8} (f(x_0) + 3f(x_1) + 3f(x_2) + f(x_3))$$

en donde $h=(b-a)/3$. A esta ecuación se le llama regla de Simpson de 3/8 porque h es un múltiplo de 3/8. La regla de Simpson se puede expresar en la forma de la ecuación :

$$I \approx (b-a) \frac{f(x_0) + 3f(x_1) + 3f(x_2) + f(x_3)}{8}$$

El diagrama de flujo se presenta en la siguiente pagina.

REGLA DE SIMPSON 3/8



Nomenclatura :
 NP = No. de puntos contenidos en el intervalo
 (multiplo de tres) (X₀, X_N)
 X₀ = Valor inicial de X
 X_N = Valor final de X
 h = incremento de X
 f(x) = Valor de la funcion en X

INTEGRACION DE ROMBERG

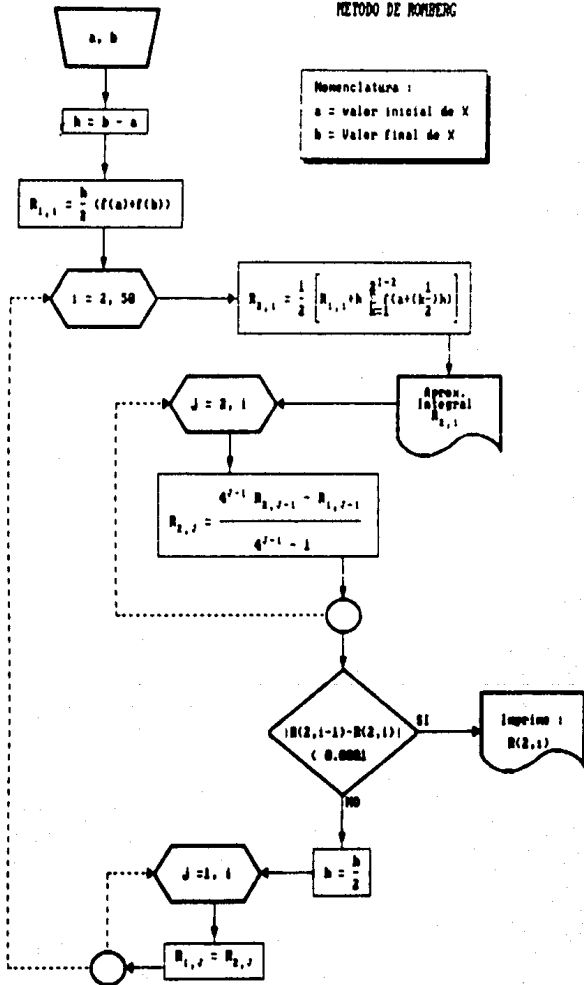
La formula general representada mediante la ecuación

$$I \approx \frac{4^{j-1} I_{j+1,k+1} - I_{j,k-1}}{4^{j-1} - 1}$$

se le atribuye a Romberg, y a la aplicacion sistemática en la evaluacion de integrales se le conoce como integraci3n de Romberg. En donde $I_{j+1,k+1}$ y $I_{j,k-1}$ son las integrales mäs y menos exactas respectivamente, $I_{k,1}$ es la integral mejorada.

El diagrama de flujo se presenta en la siguiente página.

METODO DE ROMBERG



Nomenclatura :
 a = valor inicial de X
 b = Valor final de X

METODO DE GAUSS GENERALIZADO

Supongase que la restriccion de fijar los puntos base se elimina y se va a evaluar libremente el area bajo la linea recta que une dos puntos cualesquiera de la curva, colocando estos puntos de manera que se pueda definir una linea recta que balancee los errores negativos y positivos.

La cuadratura Gaussiana es el nombre de uno de estos métodos que implementa esta estrategia. Las fórmulas particulares de cuadratura Gaussiana se llaman fórmulas de Gauss - Legendre.

La cuadratura Gaussiana determina los coeficientes de una ecuación de la forma

$$I \approx c_1 f(x_1) + c_2 f(x_2)$$

en donde las c son los coeficientes incógnitas, los argumentos de la función x_1 y x_2 no están fijados a los puntos extremos, sino que son incógnitas. Por lo tanto se tiene un total de cuatro incógnitas y por consiguiente se requieren de cuatro ecuaciones para conocer las incógnitas.

Se pueden obtener dos de estas condiciones suponiendo que la ecuación $I \approx c_1 f(x_1) + c_2 f(x_2)$ ajusta exactamente la integral de una constante y de una función lineal. Entonces, para obtener las otras dos condiciones se supone que la integral también se ajusta a una función parabólica ($y = x^2$) y a una función cúbica ($y = x^3$). Así se determinan las cuatro incógnitas derivando una fórmula de integración de doble punto que sea exacta para cúbicas. Las cuatro ecuaciones por resolver son :

$$c_1 f(x_1) + c_2 f(x_2) = \int_{-1}^1 f(x) dx = 2$$

$$c_1 f(x_1) + c_2 f(x_2) = \int_{-1}^1 x dx = 0$$

$$c_1 f(x_1) + c_2 f(x_2) = \int_{-1}^1 x^2 dx = 2/3$$

$$c_1 f(x_1) + c_2 f(x_2) = \int_{-1}^1 x^3 dx = 0$$

Estas se resuelven simultaneamente,

$$c_1 = c_2 = 1$$

$$x_1 = -1/\sqrt{3}$$

$$x_2 = 1/\sqrt{3}$$

Las cuatro se pueden sustituir en la ecuación $1 \approx c_1 f(x_1) + c_2 f(x_2)$ para obtener:

$$1 \approx f\left[-\frac{1}{\sqrt{3}}\right] + f\left[\frac{1}{\sqrt{3}}\right]$$

Ademas de la formula de dos puntos, se pueden desarrollar tambien versiones de mas de dos puntos las cuales se representan en la formula general :

$$1 \approx c_1 f(x_1) + c_2 f(x_2) + \dots + c_n f(x_n)$$

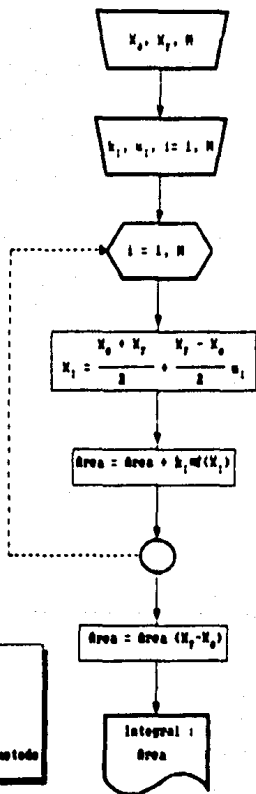
En el siguiente cuadro se resumen los valores de las c y de las x de las formulas de hasta cinco puntos ($c_n = k_n$ y $x_n = u_n$):

Puntos	Factores de peso	Argumentos de la funcion
1	$k_1 = 1$	$u_1 = 0$
2	$k_1 = 1/2$	$u_1 = -1/\sqrt{3}$
	$k_2 = 1/2$	$u_2 = 1/\sqrt{3}$
3	$k_1 = 5/18$	$u_1 = -\sqrt{3/5}$
	$k_2 = 4/9$	$u_2 = 0$
	$k_3 = 5/18$	$u_3 = \sqrt{3/5}$

Puntos	Factores de peso	Argumentos de la función
4	$k_1 = 0.1739$	$u_1 = -0.8611$
	$k_2 = 0.3261$	$u_2 = -0.34$
	$k_3 = 0.3261$	$u_3 = 0.34$
	$k_4 = 0.1739$	$u_4 = 0.8611$
5	$k_1 = 0.118463$	$u_1 = -0.90618$
	$k_2 = 0.239314$	$u_2 = 0.538469$
	$k_3 = 0.284444$	$u_3 = 0$
	$k_4 = 0.239314$	$u_4 = 0.538469$
	$k_5 = 0.118463$	$u_5 = 0.90618$

El diagrama de flujo se presenta en la siguiente página.

MÉTODO DE GAUSS GENERALIZADO



Notación:
 X_0 = Valor inicial de X
 X_f = Valor final de X
 N = Orden del Método
 h_i, u_i = parámetros i del método

METODO DE EULER

$$y_{i+1} = y_i + \phi h$$

De acuerdo a esta ecuación, se usa la aproximación a la pendiente ϕ para extrapolar a partir de un valor anterior y_i a un valor actual y_{i+1} en una distancia h . Esta fórmula se aplica paso a paso para calcular una solución futura y , de aquí, trazar la trayectoria de la solución.

La primera derivada proporciona la aproximación directa de la pendiente en x_i

$$\phi = f(x_i, y_i)$$

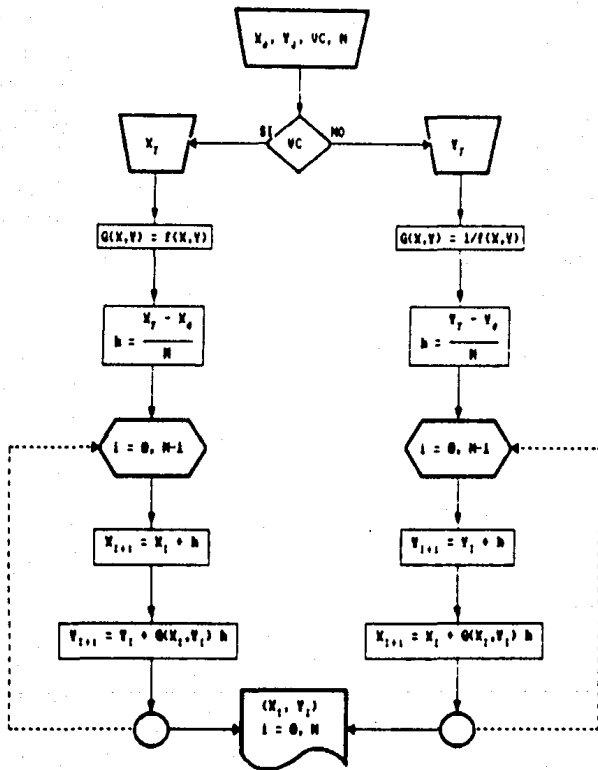
donde $f(x_i, y_i)$ es la ecuación diferencial evaluada en x_i y y_i . Esta aproximación se sustituye en la ecuación anterior para dar:

$$y_{i+1} = y_i + f(x_i, y_i)h$$

A esta ecuación se le conoce como método de Euler (o método de Euler-Cauchy). Se predice un nuevo valor de y usando la pendiente (igual a la primera derivada en el valor original x) para extrapolar linealmente sobre el tamaño de paso h .

En la siguiente página se muestra el diagrama de flujo.

METODO DE EULER



Nomenclatura :

X_0 = Valor inicial de X

V_0 = Valor inicial de V

VC = Variable de control

N = No. de subintervalos

X_f = Valor final de X

V_f = Valor final de V

METODO DE EULER MODIFICADO

El método de Euler simple considera que y se comporta como una línea recta en el intervalo $[x_i, x_{i+1}]$. Si se observa la ecuación

$$y_{i+1} = y_i + f(x_i, y_i)h$$

representativa de este algoritmo, la pendiente de esta línea recta es $f(x_i, y_i)$, es decir:

si $y_{i+1} = y_i + f(x_i, y_i)(x_{i+1} - x_i)$ entonces

$$f(x_i, y_i) = \frac{y_{i+1} - y_i}{x_{i+1} - x_i}$$

el Euler modificado considera que el comportamiento lineal es buena aproximación, si la pendiente de la línea se calcula como el promedio de dos pendientes, de tal manera que el algoritmo queda

$$y_{i+1} = y_i + f^*h,$$

donde f^* es la pendiente promedio; esta se calcula con la ecuación:

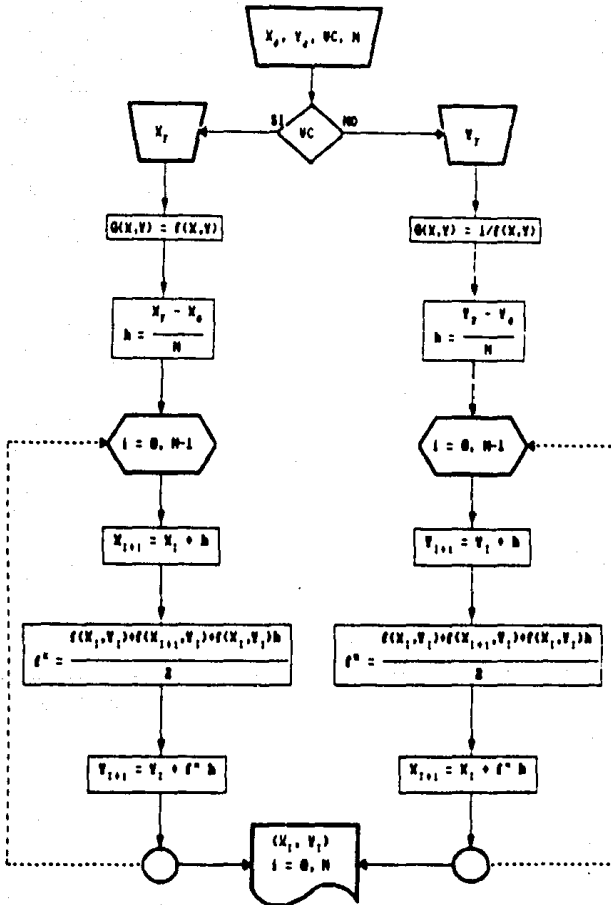
$$f^* = (f(x_i, y_i) + f(x_{i+1}, y_{i+1}))/2$$

dado que no se conoce la pendiente en (x_{i+1}, y_{i+1}) esta se aproxima mediante el cálculo de y_{i+1} con el método de Euler simple de modo que

$$f^* = (f(x_i, y_i) + f(x_{i+1}, y_i + f(x_i, y_i)h))/2$$

En la siguiente página se muestra el diagrama de flujo.

METODO DE EULER MODIFICADO



Notación :

X_0 = Valor inicial de X
 Y_0 = Valor inicial de Y
 VC = Variable de control

N = No. de subintervalos
 X_f = Valor final de X
 Y_f = Valor final de Y

METODO DE PUNTO MEDIO

Observando la ecuación de Euler $y_{i+1} = y_i + f(x_i, y_i)h$ se concluye que la expansión es lineal con pendiente igual al valor de la función f en el punto (x_i, y_i) . Runge y Kutta trabajaron en forma independiente para mejorar esta ecuación proponiendo la siguiente :

$$y_{i+1} = y_i + f^* h$$

donde f^* debe ser una pendiente representativa del intervalo $[x_i, x_{i+1}]$; dicha pendiente es evaluada por comparación de modelos propuestos con expansión de Taylor equivalente.

Dependiendo del número de puntos que se utilicen en la evaluación de la pendiente, así será el orden del método.

Sean k_1, k_2, \dots, k_n valores de la pendiente en diferentes puntos dentro del intervalo cerrado $[x_i, x_{i+1}]$; con estos puede determinarse el valor de f^* con un promedio pesado de la forma siguiente :

$$f^* = \frac{\sum_{i=1}^n a_i k_i}{\sum_{i=1}^n a_i}$$

siendo n el valor del método, a_i el peso de cada pendiente y k_i la pendiente respectiva.

El método de punto medio resulta del sistema de ecuaciones resuelto para $a_1 = 0$ por lo que $a_2 = 1$ y $p = 1/2$ y el algoritmo será :

paso 1) $k_1 = f(x_i, y_i)$

paso 2) $k_2 = f(x_i + h/2, y_i + k_1 h/2)$

paso 3) $f^* = k_2$

paso 4) $y_{i+1}^f = y_i + f^* h$

para iniciar este algoritmo deberán conocerse x_0, y_0 para la ecuación $dy/dx = f(x,y)$.

En la siguiente página se muestra el diagrama de flujo.

METODO DEL PUNTO MEDIO

X_0, Y_0, X_T, N

$$h = \frac{X_T - X_0}{N}$$

X_0, Y_0

$i = 0, N-1$

$$h_1 = f(X_1, Y_1)$$

$$h_2 = f\left(X_1 + \frac{h}{2}, Y_1 + \frac{h}{2} h_1\right)$$

$$Y_{101} = Y_1 + h_2 h$$

$$X_{101} = X_1 + h$$

X_{101}, Y_{101}



Monoclitura :

X_0 = Valor inicial de X

Y_0 = Valor inicial de Y

X_T = Valor final de X

N = No. de incrementos

METODO DE HEUN

Se obtiene asignándole el valor a $a_1 = 1/4$; al resolver el sistema obtenemos $a_2 = 3/4$ y $p = 2/3$. El algoritmo se resuelve con los siguientes pasos :

paso 1) $k_1 = f(x_i, y_i)$

paso 2) $k_2 = f(x_i + 2h/3, y_i + 2k_1h/3)$

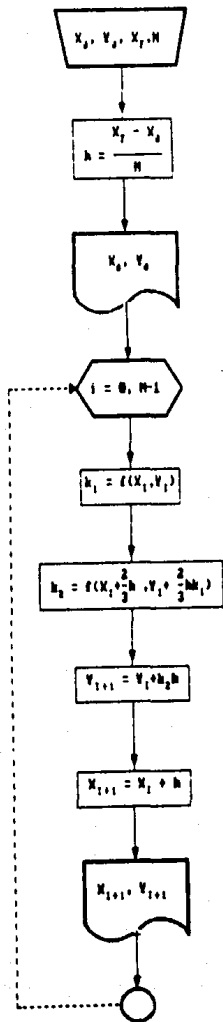
paso 3) $f^* = k_1/4 + 3k_2/4$

paso 4) $y_{i+1} = y_i + f^*h$

para iniciar este algoritmo deberán conocerse x_0, y_0 para la ecuación $dy/dx = f(x,y)$.

En la siguiente pagina se muestra el diagrama de flujo.

METODO DE HEUN



Nomeclatura :

X_0 = Valor inicial de X

Y_0 = Valor inicial de Y

X_f = Valor final de X

N = No. de incrementos

METODO DE RUNGE - KUTTA DE 3ER ORDEN UTILIZANDO 5 CONSTANTES:

El método tendrá la siguiente forma:

$$y_{i+1} = y_i + f^{\circ} h$$

donde $f^{\circ} = a_1 k_1 + a_2 k_2 + a_3 k_3$

$$k_1 = f(x_i, y_i) \quad \text{punto de referencia}$$

$$k_2 = f(x_i + mh, y_i + mhk_1)$$

$$k_3 = f(x_i + ph, y_i + phk_2)$$

al ser tres puntos, el método será equivalente a un polinomio de Taylor de tercer grado.

Para poder comparar las formulas de Taylor y Runge - Kutta, k_2 y k_3 deberán expandirse hasta un polinomio de segundo grado, para que al hacer el producto $f^{\circ} h$ se obtenga la h^3 correspondiente al último término de la expansión de Taylor. Al realizar estas expansiones los polinomios obtenidos son los siguientes:

$$k_1 = f$$

$$k_2 = f + mh f_x + m^2 h^2 / 2 (f_{xx} + 2f_{xy} + f^2 f_{yy})$$

$$k_3 = f + ph f_x + phk_2 f_y + p^2 h^2 / 2 (f_{xx} + 2k_2 f_{xy} + k_2^2 f_{yy})$$

Uno de los algoritmos de tercer orden de Runge - Kutta es el obtenido al resolver el sistema de ecuaciones para $m = 1/2$; con lo cual se obtiene $a_1 = 2/9$, $a_2 = 3/9$, $a_3 = 4/9$, $p = 3/4$. Con estos valores establecemos el siguiente algoritmo.

paso 1) $k_1 = f(x_i, y_i)$

paso 2) $k_2 = f(x_i + h/2, y_i + k_1 h/2)$

paso 3) $k_3 = f(x_i + 3h/4, y_i + 3k_2 h/4)$

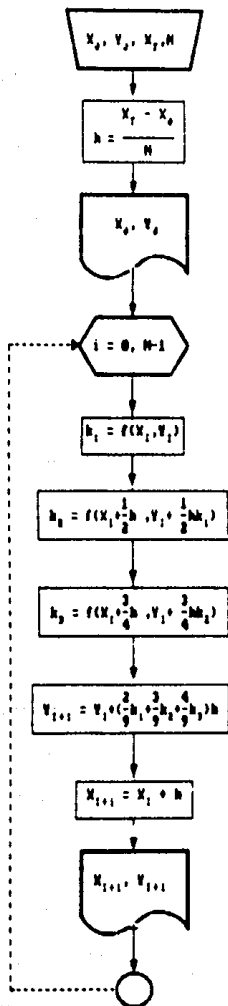
paso 4) $f^{\circ} = 2k_1/9 + 3k_2/9 + 4k_3/9$

paso 5) $y_{i+1} = y_i + f \cdot h$

para iniciar este algoritmo deberán conocerse x_0, y_0 para la ecuación $dy/dx = f(x,y)$.

En la siguiente página se muestra el diagrama de flujo.

METODO DE RUNGE-KUTTA 3er. ORDEN, 5 CONSTANTES



Nomenclatura :
 X_0 = Valor inicial de X
 Y_0 = Valor inicial de Y
 X_r = Valor final de X
 N = No. de incrementos

METODO DE RUNGE - KUTTA 3ER ORDEN, UTILIZANDO 6 CONSTANTES

Con el objeto de tener mayor aproximación en el calculo de y_{i+1} para definir k_3 se aprovecha el conocimiento que se tiene de k_2 y k_1 de tal manera que k_3 se obtiene como :

$$k_3 = f(x_i + ph, y_i + qhk_2 + (p-q)hk_1)$$

Notese que con respecto al metodo anterior, se ha introducido una nueva variable q , de tal manera que $k_3 = f(x_i + ph, y_i + qhk_2 + (p-q)hk_1)$. Al hacer las expansiones correspondientes para k_1 , k_2 , k_3 mediante un polinomio de Taylor de segundo grado y al comparar $y_{i+1} = y_i + f' h$, con el polinomio de expansion de series de Taylor de tercer grado, se obtiene el siguiente sistema de ecuaciones:

$$a_1 + a_2 + a_3 = 1$$

$$ma_2 + pa_3 = 1/2$$

$$m^2 a_2 + p^2 a_3 = 1/3$$

$$mqa_3 = 1/6$$

Notese que existen cuatro ecuaciones con seis incognitas, lo cual implica fijar dos de ellas para obtener una solución. Fijando $m = 1/2$ y $p = 1$ se obtiene lo siguiente :

$$a_1 = 1/6, a_2 = 4/6, a_3 = 1/6 \text{ y } q = 2$$

con esto se obtiene el siguiente algoritmo :

paso 1) $k_1 = f(x_i, y_i)$

paso 2) $k_2 = f(x_i + h/2, y_i + k_1 h/2)$

paso 3) $k_3 = f(x_i + h, y_i + 2k_2 h - hk_1)$

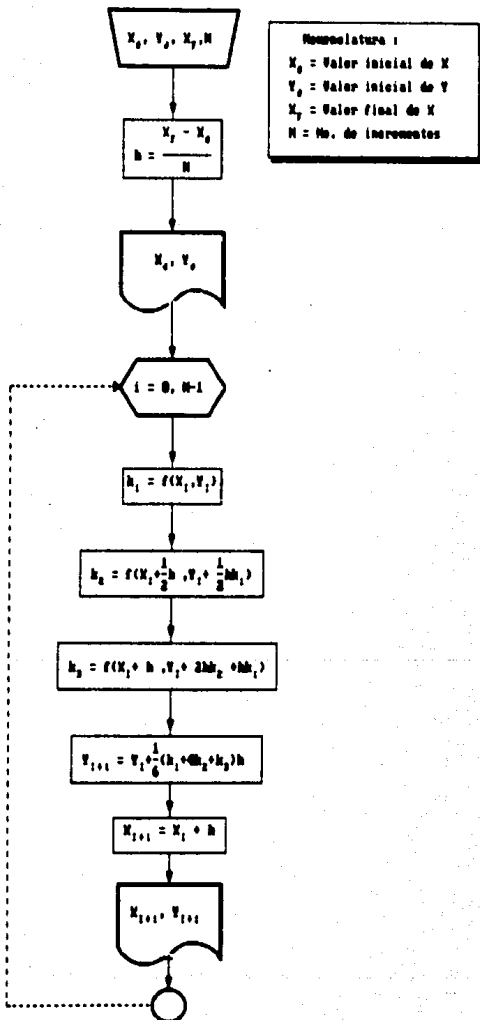
paso 4) $f^* = (k_1 + 4k_2 + k_3)/6$

paso 5) $y_{i+1}^j = y_i^j + f^* h$

para iniciar este algoritmo deberán conocerse x_0, y_0 para la ecuación $dy/dx = f(x, y)$.

En la siguiente página se muestra el diagrama de flujo.

METODO DE RUNGE-KUTTA, 3 ORDEN, 6 CONSTANTES



METODO DE RUNGE - KUTTA DE 4O ORDEN, UTILIZANDO 7 CONSTANTES

Uno de los métodos más utilizados es el que se obtiene mediante la expansión y comparación con el polinomio de Taylor de cuarto grado respectivo de k_2 , k_3 , k_4 , representadas en el sistema de ecuaciones siguiente :

$$f' = k_1 a_1 + k_2 a_2 + k_3 a_3 + k_4 a_4$$

$$k_1 = f(x_1, y_1)$$

$$k_2 = f(x_1 + mh, y_1 + mhk_1)$$

$$k_3 = f(x_1 + ph, y_1 + phk_2)$$

$$k_4 = f(x_1 + qh, y_1 + qhk_3)$$

k_2 , k_3 y k_4 deben expandirse a un polinomio de grado 3, para que al multiplicarlo por la h del término $f'h$, se genere el término h^4 , el cual corresponderá al término de la expansión en serie de Taylor de grado 4 de y_{i+1} .

La solución del sistema de ecuaciones obtenidas, generará el siguiente algoritmo :

paso 1) $k_1 = f(x_1, y_1)$

paso 2) $k_2 = f(x_1 + h/2, y_1 + k_1 h/2)$

paso 3) $k_3 = f(x_1 + h/2, y_1 + hk_1/2)$

paso 4) $k_4 = f(x_1 + h, y_1 + hk_3)$

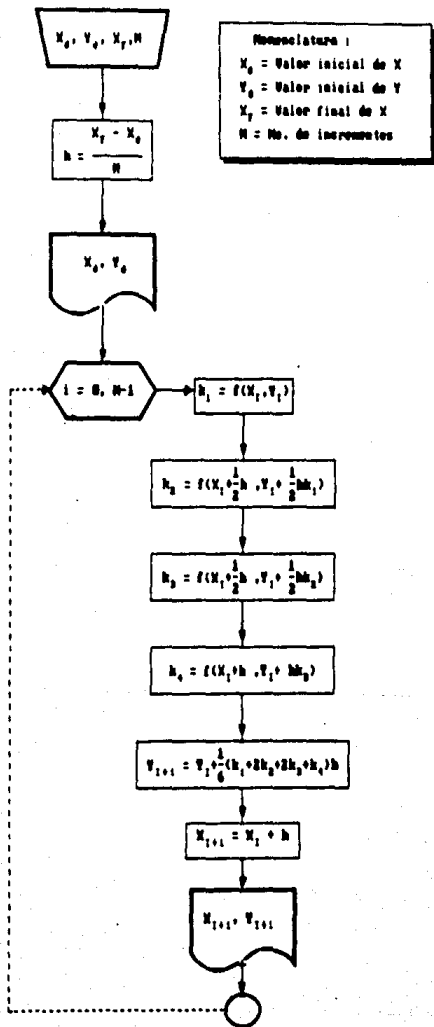
paso 5) $f' = (k_1 + 2k_2 + 2k_3 + k_4)/6$

paso 6) $y_{i+1} = y_1 + f'h$

para iniciar este algoritmo deberán conocerse x_0 , y_0 para la ecuación $dy/dx = f(x, y)$.

En la siguiente página se muestra el diagrama de flujo.

METODO DE RUNGE-KUTTA 4^o ORDEN, 7 CONSTANTES



VARIANTE DEL METODO DE RUNGE - KUTTA 4º ORDEN

Otro algoritmo de cuarto orden de diez constantes, usado con menos frecuencia que el anterior, es el que se obtiene al considerar que $k = f(k_2, k_1)$ y que $a_4 = f(k_3, k_2, k_1)$.

Las ecuaciones que genera el algoritmo son las siguientes :

$$\begin{aligned}
 f^{\circ} &= k_1 a_1 + k_2 a_2 + k_3 a_3 + k_4 a_4 \\
 k_1 &= f(x_1, y_1) \\
 k_2 &= f(x_1 + ph, y_1 + phk_1) \\
 k_3 &= f(x_1 + ph, y_1 + qhk_2 + (p-q)hk_1) \\
 k_4 &= f(x_1 + rh, y_1 + thk_3 + (r-t)hk_2 + shk_1)
 \end{aligned}$$

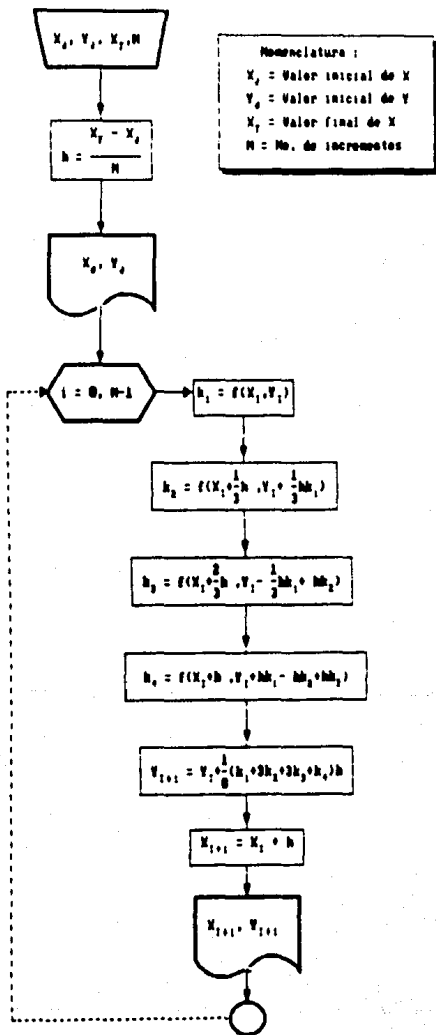
De la comparación de los polinomios se obtiene el siguiente algoritmo :

- paso 1) $k_1 = f(x_1, y_1)$
- paso 2) $k_2 = f(x_1 + h/3, y_1 + k_1 h/3)$
- paso 3) $k_3 = f(x_1 + 2h/3, y_1 - hk_1/3 + hk_2)$
- paso 4) $k_4 = f(x_1 + h, y_1 + hk_1 + hk_2 + hk_3)$
- paso 5) $f^{\circ} = (k_1 + 3k_2 + 3k_3 + k_4)/8$
- paso 6) $y_{1,1} = y_1 + f^{\circ} h$

para iniciar este algoritmo deberán conocerse x_0, y_0 para la ecuación $dy/dx = f(x, y)$.

En la siguiente página se muestra el diagrama de flujo.

VARIANTE DEL METODO DE RUNGE-KUTTA 4^o ORDEN



MÉTODOS DE MULTIPASOS DE INTEGRACION ABIERTA

Todos los métodos anteriores se apoyan en un punto anterior para estimar el nuevo punto. Existe otra familia de métodos, los cuales para evaluar el punto (x_{i+1}, y_{i+1}) se apoyan en varios puntos anteriores, conocidos como métodos de multipasos. Para desarrollar estos métodos la expansión de la función $f(x,y)$ se lleva a cabo mediante el polinomio fundamental de Newton, con diferencia finitas sin dividir hacia atrás. Lo anterior implica que en lugar de utilizar valores de y_i , se utilizan valores de $f(x_i, y_i)$ los cuales pueden ser representados como f_i .

MÉTODO DE ADAMS - BASHFORTH DE DOS PASOS

El nombre del método indica que se apoya en dos puntos: (x_i, y_i) y (x_{i-1}, y_{i-1}) . Para que estos puntos sean los únicos que participen en el proceso recursivo, la integral deberá ser resuelta de x_i a y_i .

Si la ecuación se trunca a partir de la segunda diferencia obtenemos:

$$y_{i+1} = y_i + h \left[\alpha f_i + (\alpha^2/2) \nabla f_i \right]_0^1$$

$$y_{i+1} = y_i + hf_i + (1/2)h(f_i - f_{i-1})$$

$$y_{i+1} = y_i - (1/2)hf_{i-1} + (3/2)hf_i = y_i - (1/2)hf(x_{i-1}, y_{i-1}) + (3/2)hf(x_i, y_i).$$

Para poder utilizar este algoritmo se deben conocer los puntos (x_0, y_0) , (x_1, y_1) para calcular el punto (x_2, y_2) y así sucesivamente. Dado que (x_0, y_0) es un dato se recomienda calcular el punto (x_1, y_1) con el método de Euler.

METODO DE ADAMS - BASHFORTH DE TRES PASOS

Es un método que se apoya en tres puntos anteriores (al punto en que queremos evaluar), es decir, participan los puntos (x_i, y_i) , (x_{i-1}, y_{i-1}) y (x_{i-2}, y_{i-2}) . Si se trunca la ecuación a partir de la tercera diferencia se obtiene :

$$y_{i+1} = y_i + h \left[f_i + \left(\frac{5}{24}\right) \nabla f_i + \left(\frac{5}{24}\right) \nabla^2 f_i + \left(\frac{1}{24}\right) \nabla^3 f_i \right] \Big|_0^1$$

$$y_{i+1} = y_i + h \left[f_i + \left(\frac{1}{24}\right) (f_i - f_{i-1}) + \left(\frac{5}{6}\right) (f_i - 2f_{i-1} + f_{i-2}) \right]$$

$$y_{i+1} = y_i + (h/12) \left[23f(x_i, y_i) - 16f(x_{i-1}, y_{i-1}) + 5f(x_{i-2}, y_{i-2}) \right]$$

Para iniciar este algoritmo, deben conocerse los puntos (x_0, y_0) , (x_1, y_1) y (x_2, y_2) para calcular el punto (x_3, y_3) y así sucesivamente. Dado que (x_2, y_2) es un dato, se recomienda calcular los puntos (x_1, y_1) y (x_2, y_2) con el método de Euler por simplicidad.

METODO DE ADAMS - BASHFORTH DE CUATRO PASOS

Se obtiene evaluando la ecuación desde x a $x+1$, truncando dicha ecuación a partir de la cuarta diferencia. La ecuación resultante de y_{i+1} dependerá de cuatro puntos anteriores: (x_i, y_i) , (x_{i-1}, y_{i-1}) , (x_{i-2}, y_{i-2}) , (x_{i-3}, y_{i-3}) . De manera análoga al desarrollo algebraico de las secciones anteriores, se obtiene la siguiente ecuación:

$$y_{i+1} = y_i + (h/24) [55f(x_i, y_i) - 59f(x_{i-1}, y_{i-1}) + 37f(x_{i-2}, y_{i-2}) - 9f(x_{i-3}, y_{i-3})]$$

Para iniciar este algoritmo, se deben conocer los puntos (x_0, y_0) , (x_1, y_1) , (x_2, y_2) y (x_3, y_3) para calcular el punto (x_4, y_4) y así sucesivamente. Dado que el punto (x_0, y_0) es un dato, se recomienda calcular los puntos (x_1, y_1) , (x_2, y_2) y (x_3, y_3) con el método de Euler por simplicidad.

METODO DE ADAMS - BASHFORTH DE CINCO PASOS

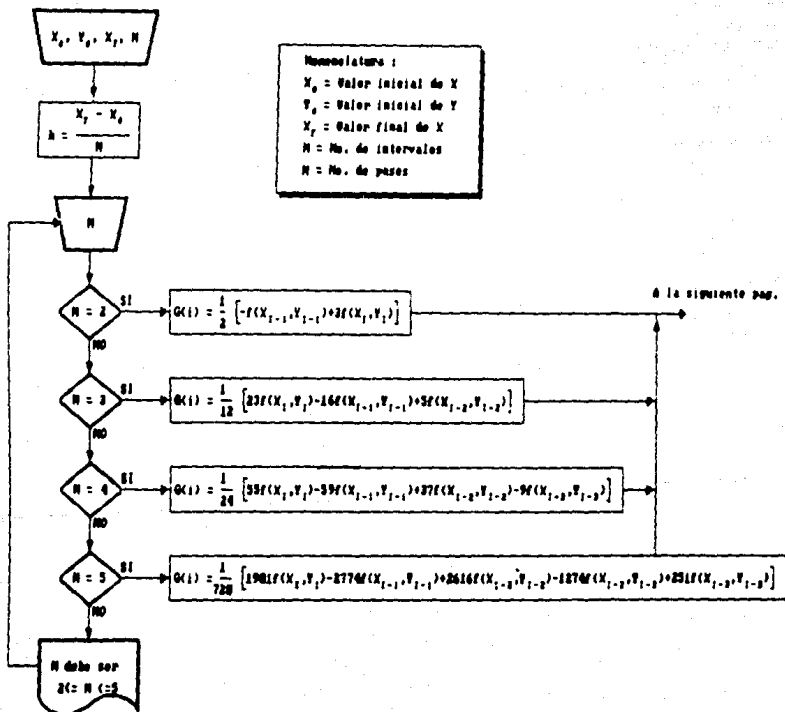
Se obtiene evaluando la ecuación desde x_i a x_{i+1} , truncando dicha ecuación a partir de la quinta diferencia. La ecuación resultante de y_{i+1} dependerá de cinco puntos anteriores: (x_i, y_i) , (x_{i-1}, y_{i-1}) , (x_{i-2}, y_{i-2}) , (x_{i-3}, y_{i-3}) y (x_{i-4}, y_{i-4}) . De manera análoga al desarrollo algebraico de las secciones anteriores, se obtiene la siguiente ecuación :

$$y_{i+1} = y_i + (h/720) [1901f(x_i, y_i) - 2774f(x_{i-1}, y_{i-1}) + 2616f(x_{i-2}, y_{i-2}) - 1274f(x_{i-3}, y_{i-3}) + 251f(x_{i-4}, y_{i-4})]$$

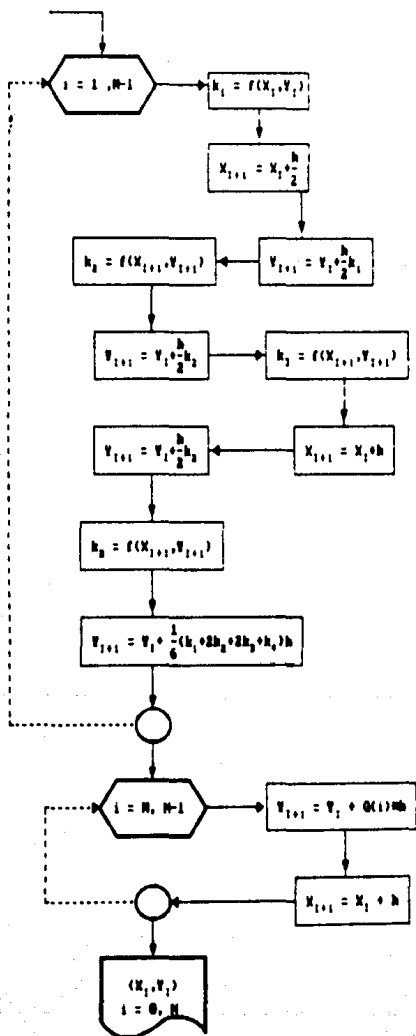
Para iniciar este algoritmo, se deben conocer los puntos (x_0, y_0) , (x_1, y_1) , (x_2, y_2) , (x_3, y_3) y (x_4, y_4) para calcular el punto (x_5, y_5) y así sucesivamente. Dado que el punto (x_0, y_0) es un dato, se recomienda calcular los puntos (x_1, y_1) , (x_2, y_2) , (x_3, y_3) y (x_4, y_4) con el método de Euler por simplicidad.

El diagrama de flujo se presenta en la siguiente página.

MULTIPASOS DE INTEGRACION ABierta



De la pagina anterior



MÉTODOS DE MULTIPASOS DE INTEGRACION CERRADA

La diferencia de estos métodos con los de la sección anterior es que las diferencias finitas hacia atrás se evalúan a partir del punto (x_{i-1}, y_{i-1}) . Esto indica que la ecuación para calcular y_{i+1} dependerá a su vez del punto (x_{i-1}, y_{i-1}) que no se conoce. Este criterio genera una familia de métodos, conocidos como métodos implícitos. Truncando la ecuación e integrando tenemos:

Ecuación de Adams - Moulton de dos pasos

$$y_{i+1} = y_i + (h/12) [5f(x_{i-1}, y_{i-1}) + 8f(x_i, y_i) - f(x_{i+1}, y_{i+1})]$$

Ecuación de Adams - Moulton de tres pasos

$$y_{i+1} = y_i + (h/12) [9f(x_{i-1}, y_{i-1}) + 19f(x_i, y_i) - 5f(x_{i+1}, y_{i+1}) + f(x_{i-2}, y_{i-2})]$$

Ecuación de Adams - Moulton de cuatro pasos

$$y_{i+1} = y_i + (h/120) [251f(x_{i-1}, y_{i-1}) + 646f(x_i, y_i) - 264f(x_{i+1}, y_{i+1}) + 106f(x_{i-2}, y_{i-2}) - 19f(x_{i-3}, y_{i-3})]$$

Dado que se desconoce y_{i+1} , estos métodos no se utilizan directamente, sino que deberán combinarse con otro, el cual vaya prediciendo valores de y_{i+1} , para que nuestra última familia de métodos vaya prediciendo valores de y_{i+1} . A la combinación de dos métodos donde uno predice y otro corrige se le conoce como método predictor-corrector. Dado que esta familia de métodos no se usan independientemente no se presentará diagrama de flujo.

MÉTODOS PREDICTORES - CORRECTORES

Como se menciona en la sección anterior, estos métodos se generan al combinar un método de integración abierta, con un método de integración cerrada. Por lo tanto, el desarrollo de las fórmulas seguirá un procedimiento análogo al descrito anteriormente, por lo que en esta sección solo se presentarán las fórmulas correspondientes a los métodos de este tipo más utilizados :

Método de Milne de cuarto orden.

Predictor :

$$y_{i+1} = y_{i-3} + (4h/3) [2f(x_i, y_i) - f(x_{i-1}, y_{i-1}) + 2f(x_{i-2}, y_{i-2})]$$

Corrector :

$$y_{i+1} = y_{i-1} + (h/3) [f(x_{i+1}, y_{i+1}) + 4f(x_i, y_i) + f(x_{i-1}, y_{i-1})]$$

Método de Milne de sexto orden.

Predictor :

$$y_{i+1} = y_{i-5} + (3h/10) [11f(x_i, y_i) - 14f(x_{i-1}, y_{i-1}) + 26f(x_{i-2}, y_{i-2}) - 14f(x_{i-3}, y_{i-3}) + 11f(x_{i-4}, y_{i-4})]$$

Corrector :

$$y_{i+1} = y_{i-3} + (2h/45) [7f(x_{i+1}, y_{i+1}) + 32f(x_i, y_i) + 12f(x_{i-1}, y_{i-1}) + 32f(x_{i-2}, y_{i-2}) + 7f(x_{i-3}, y_{i-3})]$$

Método de Adams - Moulton o Adams modificado.

Predictor :

$$y_{i+1} = y_i + (h/24) [55f(x_i, y_i) - 59f(x_{i-1}, y_{i-1}) + 37f(x_{i-2}, y_{i-2}) - 9f(x_{i-3}, y_{i-3})]$$

Corrector :

$$y_{i+1} = y_i + (h/24) [9f(x_{i+1}, y_{i+1}) + 19f(x_i, y_i) - 5f(x_{i-1}, y_{i-1}) + f(x_{i-2}, y_{i-2})]$$

Se utilizan estos métodos cuando se desea mucha precisión con pocos subintervalos de integración ya que, como puede observarse en las fórmulas, se requieren muchas evaluaciones de la función. Por ende, en pos de la exactitud, se recomienda que los puntos que inician el cálculo se estimen con un Runge - Kutta de cuarto orden.

El diagrama de flujo se presenta en la siguiente página.

MULTIPASOS DE INTEGRACION CERRADA

Para N=1 usa el metodo de Runge de 4^o orden

Para N=2 usa el metodo de Runge de 6^o orden

Para N=3 usa el metodo de Adams - Runout

Notaciatura :

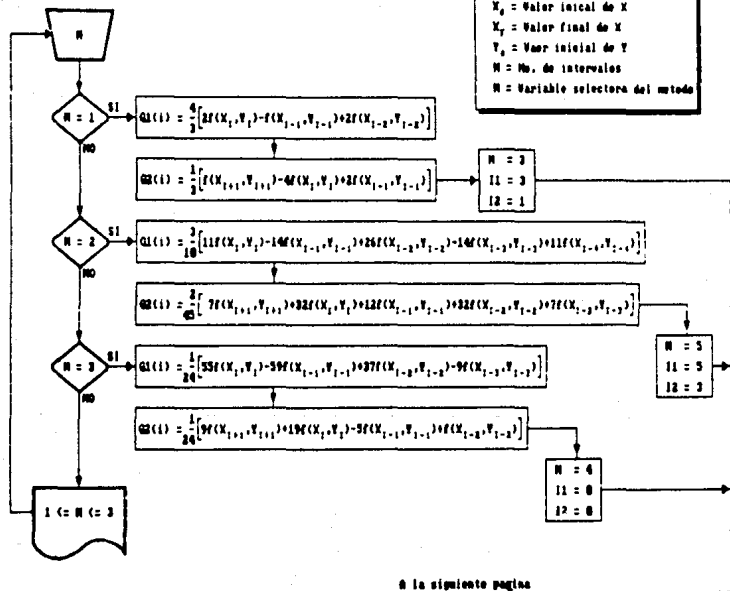
X_0 = Valor inicial de X

X_f = Valor final de X

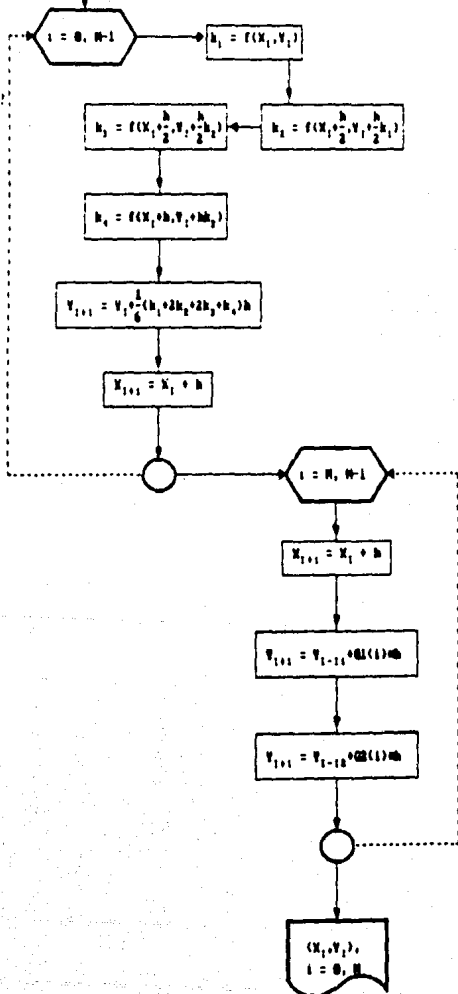
Y_0 = Valor inicial de Y

N = No. de intervalos

N = Variable selectora del metodo



De la pagina anterior





Capitulo: III

Descripción general de

Lotus 1-2-3

GENERALIDADES SOBRE LOTUS 123

Antes de involucrarnos en la discusión sobre el desarrollo de la hoja electrónica de trabajo, es necesario aclarar y entender que significa este término.

La "hoja electrónica de trabajo" o *Worksheet* se define como una matriz compuesta por celdas para introducir los datos o ecuaciones en la intersección de las filas (horizontales) y columnas (verticales).

Desde la introducción al mundo de la computación el paquete de software *Lotus 123* ha logrado mantener su lugar por encima de cualquier programa existente en el mercado; reemplazando así a *Visi Calc*, que ocupaba dicho lugar desde 1978.

Desde su aparición en 1983, *1-2-3* logró convertirse en el programa de mayor venta para las computadoras IBM PC y sus compatibles.

El nombre *1-2-3* viene de la combinación de tres de los principales programas de aplicación gerencial en un sofisticado paquete de software. Las aplicaciones integradas son: la hoja electrónica de cálculo, la generación de gráficos, y la creación y manejo de una base de datos.

Una de las más recientes revisiones de software, denominada Versión 2.0, trae consigo lo siguiente: (1) una hoja de trabajo más grande, (2) nuevas y más funciones incorporadas a la hoja de cálculo y, lo más importante para nosotros, (3) un lenguaje de comandos para macroinstrucciones que ha sido notablemente ampliado por sobre el de la Versión 1A.

Básicamente *1-2-3* es una hoja electrónica de cálculo. Su estructura

integra a la hoja de trabajo los elementos necesarios para la representación de gráficos y para la creación y manejo de una base de datos.

Los gráficos se generan en la pantalla o en la impresora, mediante los comandos contenidos en la hoja de trabajo. Las operaciones para el manejo de datos se usan en la configuración de columnas y renglones de la hoja electrónica de cálculo.

Requerimientos mínimos del Hardware:

Computadora Personal IBM (IBM PC)

Monitor de video: Monocromático o a color

Capacidad de los disquetes: Dos unidades escritoras/lectoras de disco de doble cara de 360K de capacidad

Capacidad de memoria RAM: 256K

Máxima capacidad de memoria RAM utilizable: 640 K

Sistema Operativo: PC-DOS Version 2.0 o mayor

Otro hardware: Adaptador de color/gráfico, impresora, graficador, capacidad de memoria RAM expandida (hasta 4 megabytes), coprocesador 8087.

El programa puede ser utilizado con las siguientes computadoras:

IBM PC AT

IBM 3270 PC

COMPAQ PC

COMPAQ PLUS

COMPAQ DESKPRO

COMPAQ 286

AT&T PC 6300

MANEJO DE FORMULAS

Las hojas electrónicas de cálculo nos permiten crear interrelaciones entre distintas celdas. Por ejemplo si la celda C1 contiene la fórmula $C1=A1+B1$, la celda C1 desplegará la suma del valor que contiene la celda A1 más el valor de la celda B1.

Como se puede observar, las referencias o nombres de las celdas (tales como A1, B1 y C1) sirven como variables en la ecuación. No importa qué valores contengan las celdas A1 y B1. La celda C1 siempre nos dará la suma de los dos valores contenidos en ellas.

El valor contenido en una celda puede ser sumado, restado multiplicado o dividido por el valor contenido en cualquier otra celda. En resumen, las funciones evaluadas en las hojas electrónicas de cálculo pueden ser utilizadas para realizar cualquier función en todas las celdas.

FUNCIONES ESPECIALES INCORPORADAS EN LA HOJA DE CALCULO DE LOTUS

Las funciones incorporadas a 1-2-3 son simplificaciones que permiten al usuario realizar operaciones matemáticas con un mínimo de tecleo.

Estas funciones incorporadas al programa equivalen a abreviaciones de largas y tediosas formulas que se incluyen en el paquete para hacer la vida más fácil al usuario.

1-2-3 contiene las funciones básicas que se encontraban en los programas de la primera generación, tales como VisiCalc. Ellas son Sum (sumar), Count (contar), Avg (promediar), Max (máximo), Min

(mínimo); las funciones básicas del álgebra booleana tales como: IF (si lógico), AND (y lógico) y OR (o lógico); y las funciones trigonométricas, incluyendo SIN (seno), COS (coseno), TAN (tangente) y PI (3.1416). Las hojas de cálculo más recientes y la versión 2.0 de 1-2-3 incluyen muchas otras y más avanzadas funciones.

MANEJO DE COMANDOS O MANDATOS

Al igual que las demás hojas de trabajo, 1-2-3 incluye varios comandos útiles que permiten manejar la hoja electrónica de cálculo de diferentes maneras. Por ejemplo, todas las hojas de trabajo incluyen un comando para formatear o para dar forma al contenido de una celda en la hoja. Tales comandos pueden modificar la manera de cómo se despliegan los resultados para hacer que se puedan visualizar de diferentes maneras.

En casi todos estos programas, los comandos del menú se activan primero presionando la barra diagonal "/". Oprimir esta tecla hace que se despliegue el menú principal de selección de comandos sobre la pantalla. De este menú el usuario puede seleccionar el comando que le permita al programa hacer lo que la persona desee.

La mayoría de los programas para hojas electrónicas de cálculo utilizan esta estructura básica de comando implementada por primera vez en VisiCalc.

El programa 1-2-3 está escrito en lenguaje ensamblador, que es el que más se aproxima a los números hexadecimales que utiliza la computadora. Esto proporciona a 1-2-3 más velocidad sobre otras hojas de cálculo y sobre los paquetes integrados que compiten con él.

FUNCIONES INCORPORADAS A LOTUS.

Las hojas de trabajo de segunda generación fueron desarrolladas para satisfacer la necesidad de los usuarios con funciones matemáticas, trigonométricas y financieras que fuesen parte del programa y permitieran así aprovechar al máximo todo su potencial.

Las funciones incorporadas a 1-2-3 no solo se pueden usar en aplicaciones de negocios, sino que también se pueden utilizar en aplicaciones tanto de ingeniería como científicas.

1-2-3 incluye funciones matemáticas tan comunes como: @Log, @Exp, @Sqrt, @Int, @Abs; un juego completo de funciones trigonométricas; una función específica para generar números aleatorios @Rand, así como funciones que sirven para redondear cantidades y otras para determinar el residuo de la división de dos números @Mod.

La versión 2.0 de 1-2-3 también incluye un comando para la regresión múltiple; este comando (que no es una función incorporada sino una instrucción) puede realizar una regresión hasta con 16 variables independientes.

CREACION DE GRAFICAS CON LOTUS.

1-2-3 integra la capacidad de generar gráficas directamente en la hoja de cálculo de manera que no se necesita utilizar un programa de comunicaciones para el intercambio de datos, tal como sucede con VisiCalc y Multiplan.

Esta capacidad gráfica de 1-2-3 es notablemente versátil y fácil de usar. 1-2-3 permite desplegar cinco tipos diferentes de gráficas:

lineal, de barras, de barras apiladas, de dispersión y de sectores.

Cada gráfica permite representar hasta 6 diferentes rangos de datos a la vez, excepto con las gráficas de sectores.

Las gráficas de 1-2-3 se generan utilizando el comando /Graph. Aunque el programa ofrece una gran variedad de opciones, todo lo que se necesita especificar es el tipo de gráfica y el rango de datos. Una vez que se le proporciona esta información, solo se oprime la letra V, de View, con el efecto de que la gráfica se despliega en la pantalla.

Esta es una de las grandes ventajas que ofrece trabajar, en el paquete de software 1-2-3, los métodos numéricos porque se puede analizar el comportamiento de una función cualquiera para estimar, por ejemplo, la localización aproximada de una raíz o de varias raíces. Se puede analizar también el comportamiento de una dispersión de puntos a los cuales se requiere representar por medio de una función. O simplemente para una excelente presentación de un reporte gráfico, que en cualquier otro lenguaje convencional como Basic o Pascal tomaría muchas horas de esfuerzo y programación para poder lograr lo que 1-2-3 realiza en cuestión de segundos.

MACROINSTRUCCIONES DEL TECLADO Y SU LENGUAJE DE PROGRAMACION.

Una de las más relevantes características de 1-2-3 es la de poder acomodar el programa a nuestras necesidades específicas y automatizar su ejecución mediante macroinstrucciones. Esta característica permite crear, dentro de la hoja de trabajo, programas que se pueden utilizar para una variedad de propósitos de solución de ecuaciones o problemas

de Ingeniería.

En su nivel más elemental estos programas o macros son "alternativas de teclado" porque reducen a dos, el número de teclas que hay que oprimir para que realice una secuencia completa de instrucciones.

En una situación más compleja, la versión 2.0 de 1-2-3 proporciona un lenguaje completo de programación para comandos o macroinstrucciones. Esta habilidad de poder configurar las macros estaba también en las versiones 1.0 y 1A, pero era mucho más limitada.

Una vez creada la macro se le da un nombre, se podrá activar la secuencia de comandos simplemente pulsando dos teclas: la tecla [Alt] y la letra que identifica la macro.

La macro puede ser estructurada de tal manera que pueda tomar decisiones al ejecutarse. Estas decisiones pueden estar basadas en los datos que se encuentran sobre la hoja de trabajo o el cálculo que realice con dichos datos, por ejemplo, la condición de paro para la localización de una raíz.

El lenguaje para macroinstrucciones de 1-2-3 es muy parecido a un lenguaje de programación (como Basic).



Capitulo: IV

**Descripción de la hoja
de cálculo**

PANEL DE CONTROL DE LOTUS.

El panel de control o de información es el área que se encuentra encima de la línea horizontal de video inverso. Esta zona consta de tres líneas, cada una de las cuales cumple con un fin específico.

La primera línea contiene toda la información sobre la celda activa. La celda activa es aquella donde se encuentra actualmente el cursor o indicador de celda.

La primera información que aparece en esta línea, corresponde a las coordenadas de la celda; seguida de la indicación (en paréntesis) del formato de dicha celda. La última información de esta línea del panel de control representa el contenido actual de la celda activa.

La segunda línea del panel de control contiene los caracteres que se están introduciendo en ese momento a la hoja de trabajo o que se está editando; mientras que la tercera línea despliega explicaciones relacionadas con el menú de comandos actual.

A medida que se desplaza el indicador de celda de una opción a otra en el menú de comandos, cambiará la explicación que aparece en la tercera línea del panel de control. Si no se encuentra activo el menú de comandos esta línea permanecerá en blanco.

LA HOJA DE CALCULO DE LOTUS.

La hoja de calculo de la versión 2.0 de 1-2-3 está constituida por 8,192 rengiones y 256 columnas para un total de 2'097,152 celdas.

La nomenclatura de las columnas, de izquierda a derecha, comienza con la letra A, continuando con la B, C, D, hasta la Z. Luego se nombran mediante la duplicación de letras, es decir, AA, AB, AC, hasta la AZ. Después se numeran BA, BB, BC y así sucesivamente hasta la columna 256 la cual corresponde al nombre IV.

Mientras que los rengiones se nombran mediante números consecutivos comenzando con el número 1 (para la primera de arriba) hasta llegar a la 8,192 (2,048 para las versiones anteriores).

De tal manera que cada celda se identifica con la(s) letra(s) de la columna seguida del número del rengion al cual corresponde; a esta combinación se le conoce como la "dirección de celda". Por ejemplo, la celda formada por la intersección de la columna R con el rengion 1 se conoce como la celda R1.

Todas las celdas podrán llenarse con solo tres tipos de información: números, fórmulas matemáticas y cadenas de caracteres.

Las dimensiones de la hoja electrónica de 1-2-3 en medidas reales sería una enorme hoja de papel que mide aproximadamente 7.5 m de ancho por 52 m de longitud.

INTRODUCCION DE DATOS EN LA HOJA DE CALCULO.

Existen tres tipos de entradas diferentes para las celdas: números, fórmulas y rótulos. Los datos son introducidos en una celda simplemente posicionando el indicador de celda sobre la misma y tecleando la información.

1-2-3 determina el tipo de entrada que se hace sobre la celda, basándose en el primer carácter que se introduce. Si la entrada de datos empieza con uno de los siguientes caracteres: 0 1 2 3 4 5 6 7 8 9 + - . (@ # \$ 1-2-3 considerará la información introducida en la celda como un número o una fórmula. Pero si la información empieza con un carácter, 1-2-3 identificará la entrada como un rótulo.

UTILIZACION DE LAS FUNCIONES ●

1-2-3 ofrece el uso de funciones incorporadas al programa. Estas funciones son simplemente abreviaciones de largas o complejas fórmulas matemáticas que el paquete realiza automáticamente.

Todas las funciones incorporadas a 1-2-3 constan de tres partes: el signo "=", el nombre de la función, y un argumento o rango. El signo "=" indica a 1-2-3 que la entrada que se está haciendo es una función incorporada. El nombre de la función que sigue sirve para indicarle al programa que tipo de operación se debe realizar. El argumento o rango es un dato que necesita conocer 1-2-3 para llevar a cabo la función.

USO DE RANGOS DENTRO DE LA HOJA DE TRABAJO

Algunos de los comandos del programa requieren del uso de rangos. Un rango se define como la asociación de una o más celdas en forma rectangular localizada en la hoja electrónica.

De acuerdo a esta definición, una celda individual es el rango más pequeño posible, mientras que el rango más grande será la totalidad de las celdas de la hoja.

Entre las ventajas que ofrece el uso de rangos y los nombres de rangos se encuentra la de permitir utilizar los comandos y fórmulas sobre conjuntos o bloques de celdas. Se puede nombrar un rango de celdas que represente una matriz y poder utilizar todos o parte de los valores contenidos en este rango para realizar operaciones específicas.



Capitulo: V

Comandos de Lotus 1-2-3

MENUS DE COMANDOS

Al pulsar la tecla con la barra inclinada "/", 1-2-3 despliega sobre la pantalla una lista de comandos que presentan al usuario un menu de diferentes alternativas de donde elegir. Estos menus son de gran utilidad para los programas presentados en esta tesis.

1-2-3 presenta a los comandos como una lista de palabras completas. El menu aparece situado en la segunda linea del panel de control. Para visualizar este menu, pulse la tecla con la barra inclinada "/". Recuerdese que se debe pulsar esta tecla siempre que se desee desplegar el menu o utilizar uno de los comandos de 1-2-3.

La segunda caracteristica del menu de comandos aparece en la tercera linea del panel de control. Esta linea contiene una explicacion de la opcion donde esta posicionado el indicador en ese momento.

A medida que se desplaza el indicador de celda por la lista de opciones del menu, aparecerá una nueva explicacion en esta tercera linea.

El mensaje desplegado allí, indica lo que se puede hacer si se elige ese comando.

El tercer aspecto a considerar sobre los menus de comandos es la forma de indicar un comando. Para esto se puede elegir entre señalar la opcion con el indicador de celda o teclear la primera letra del nombre del comando deseado. En esta caracteristica se basa la utilizacion de las macros.

Si se selecciona el comando con el indicador de celda, utilice las flechas de direccion para mover el indicador. Una vez situado el indicador en la posicion correspondiente, pulse [Enter].

COMANDOS APLICABLES A RANGOS.

1-2-3 contiene una serie de comandos que le permiten realizar una serie de operaciones sobre zonas específicas o rangos sobre la hoja de trabajo. Luego de seleccionar este comando es posible nombrar, borrar, formatear o proteger los rangos.

El comando que se usa para tener acceso a este submenú es /Range. Cuando se tecléa /Range (/r) desde la modalidad activo, aparece el siguiente menú:

Format Label Erase Name Justify Protect Unprotect Input Value
Transpose

NOMBRAR UN RANGO.

Usando /Range Name Create (/rnc), se puede asignar cualquier nombre a una celda o a un rango de celdas, así como cambiar, suprimir o desplegar un listado de los nombres previamente asignados a celdas o rangos.

El nombre que se asigne a un rango puede contener hasta 15 caracteres y, de preferencia, deberá ser descriptivo de la información contenida al rango que se especifica. Por ejemplo, si en la celda R1 se almacena el valor de la variable X, se puede nombrar a esta celda como X y subsecuentemente se utilizará con este nombre.

La ventaja que tiene poder identificar a un rango por su nombre es que los nombres son mucho más fáciles de manejar que las direcciones de celda.

Los nombres de rangos se crean usando la secuencia de comandos /Range Name Create (/rnc). Una vez definidos los nombres de rango, estos pueden ser usados en situaciones de las direcciones de celda tanto en los comandos como en las fórmulas. Esto quiere decir que, una vez nombrado el rango, se puede borrar el rango denominado como $R1$, o se puede realizar la operación $R1^2$ en vez de $R1^2$.

1-2-3 no diferencia las letras mayúsculas de las minúsculas. Sin embargo, se recomienda no utilizar las vocales acentuadas. Puede usar las letras A a la Z, 0 al 9 y el signo de subrayar.

Una vez definido el nombre de un rango, 1-2-3 automáticamente utilizará dicho nombre en toda la hoja de trabajo y al desplegar una fórmula en el panel de control, dará preferencia al nombre sobre las direcciones de la celda correspondiente.

Sin embargo, si se suprime el nombre de un rango de la hoja de trabajo, 1-2-3 ya no lo usará más y en su lugar sustituirá el nombre del rango borrado con las respectivas direcciones de celda.

ANÁLISIS DE REGRESIÓN

El comando /Data Regression (/dr) de 1-2-3, ofrece el beneficio de poder disponer de un mecanismo que permite realizar análisis de regresión múltiple.

El comando /Data Regression (/dr) se usa al determinar la relación existente entre un conjunto de valores (llamados la variable dependiente) y uno o más conjuntos de otros valores (llamados la variable independiente).

Cuando pensemos en una regresión lineal, pensemos en que este es el mejor método que tenemos a nuestro alcance para encontrar una línea "óptima" que relacione (en una gráfica) la mayor cantidad de puntos de la variable dependiente con la mayor cantidad de valores de la variable independiente.

Una regresión múltiple es, por tanto, una regresión lineal aplicada a varias variables simultáneamente de tal forma que se determina la mejor relación lineal existente entre la variable dependiente y el conjunto de variables independientes.

Una vez utilizado el comando /Data Regression (/dr) se puede dibujar una línea para determinar que tan bien se ajusta la línea a los puntos representativos de los datos.

Al invocar este comando aparece el siguiente menú desplegado en el panel de control.

X-Range Y-Range Output-Range Intercept Reset Go Quit

La opción X-Range se usa para seleccionar una o más variables independientes para el análisis de regresión. El comando /Data Regression (/dr) puede utilizar hasta 16 variables independientes.

En un análisis de regresión, las variables siempre serán los valores que estén contenidos en una columna.

La opción Y-Range especifica la variable dependiente. Este debe consistir de una sola columna.

La opción Output-Range especifica la esquina superior izquierda donde se desplegarán los resultados. Para evitar escribir sobre celdas que contienen información, este espacio se debe escoger de tal manera que no interfiera con celdas ya llenas.

La opción Intercept permite especificar si se desea calcular la

constante de regresión. El programa también calcula esta constante por definición.

Los resultados también incluyen un mínimo de regresiones estadísticas que describen qué tan bien se ajusta la línea de regresión a los datos.

SISTEMAS DE ECUACIONES LINEALES SIMULTANEAS USANDO LOTUS

El comando /Data Matrix (/dm) es un comando matemático especializado que permite al usuario resolver sistemas de ecuaciones lineales simultáneas y manipular las soluciones resultantes.

Este comando es muy poderoso, se encuentran aplicaciones muy amplias en la solución de sistemas de ecuaciones que son muy comunes en problemas de Ingeniería y como ayuda en la solución de otros métodos numéricos.

El comando /Data Matrix (/dm) despliega un menú con las siguientes opciones:

Invert Multiply

Como lo indica su nombre, la opción Invert permite invertir una matriz cuadrada de hasta 90 columnas por 90 renglones.

Para tal propósito, simplemente se digita la opción Invert y señale (en modo inverso) el rango que se desea invertir. Luego indique un rango de salida donde se pueda colocar la matriz ya invertida.

El rango de salida puede quedar colocado en cualquier sitio de la hoja de trabajo, hasta encima de la matriz que se está invirtiendo.

El tiempo que se requiere para poder invertir una matriz es directamente proporcional al número de renglones y columnas de la matriz elevadas al cubo.

De tal manera que, para poder invertir una matriz de 10 columnas X 10 renglones se tarda cerca de 10 segundos mientras que para una de 90 columnas X 90 renglones se tardará casi una hora en una computadora IBM PC estándar con procesador 8088.

La opción Multiply permite multiplicar dos matrices rectangulares de acuerdo a las reglas de álgebra matricial. Esto es, el número de columnas de la primera matriz debe ser igual al número de renglones de la segunda.

Además la matriz resultante tendrá el mismo número de renglones de la primera matriz y el mismo número de columnas de la segunda matriz.

Al usar el comando /Data Matrix Multiply (/dmm), 1-2-2 permite que se introduzcan los tres rangos: el rango que comprende la primera matriz, el rango de la segunda matriz y el rango de salida.

La multiplicación de matrices es mucho más rápida que la inversión; sin embargo, el multiplicar matrices de gran orden se puede tardar algún tiempo.



Capítulo: VI

Macroinstrucciones de

Lotus 1-2-3

MACROINSTRUCCIONES O MACROS DE LOTUS.

Las macroinstrucciones son simplemente colecciones de multiples digitaciones de tecla que reducen, de muchas a dos, el numero de pulsaciones que hay que hacer para impartir una secuencia completa de instrucciones.

Cualquier secuencia de golpes de tecla que incluya comandos especiales para macro, asi como los comandos comunes de 1-2-3 que deben ser introducidos por el teclado, se convierte en un sencillo programa escrito en el lenguaje de comandos para macro que posee 1-2-3.

Una macroinstrucción o macro es un conjunto de instrucciones que 1-2-3 ejecutara automaticamente.

Por ejemplo: /rncx~R1~

Esta macroinstrucción esta compuesta de una serie de pulsaciones; cada elemento de dicha macro representa una pulsacion.

Elemento	Pulsaciones que representa en la macro
/	Pulse la tecla / para visualizar el menu principal
r	Teclee r para seleccionar Range del menu
n	Teclee n para seleccionar Name
c	Teclee c para seleccionar Create
x	Teclee x para dar el nombre al rango que se va a seleccionar
-	Pulse [Enter] para aceptar el nombre del rango
R1	Teclee R1 para dar el rango de celda que llevara ese nombre
-	Pulse [Enter] para aceptar el rango

Esta macro al activarse, automáticamente llamará a al rango comprendido por la celda F:

CONFIGURACION DE LAS MACROINSTRUCCIONES.

Para configurar una macroinstrucción, se tiene que hacer de la siguiente manera:

1. Teclrear la macro en una celda al igual que se hace con un letrero alineado a la izquierda. Se debe tener en cuenta que la celda que se encuentra inmediatamente por debajo de la que contiene la macro este vacia.

2. Posicione el cursor en la celda inmediatamente a la izquierda de la celda donde tecléo la macro y active el comando /Range Name Create (/rnc). Con esto asignara a dicha celda el nombre que identificará a esta macro.

3. Para activar la macro, pulse la tecla [Alt], y manteniéndola pulsada, oprima la letra que identifica a la macro (una unica letra). La macro está activada.

Una macro puede extenderse a cualquier numero de celdas adyacentes, siempre y cuando la celda que se encuentre inmediatamente por debajo de la ultima celda de la macro esté vacia.

Al grabar la hoja de trabajo al disco, tambien se graba con ella las macros que contenga.

LENGUAJE DE COMANDOS PARA MACROS

Una macro puede ser más que una simple colección de golpes de tecla. Para ello existen comandos especiales que sólo pueden ser utilizados en las macroinstrucciones y, sólo como parte de ellas.

Durante la ejecución de una macro, estos comandos realizan tareas tales como la de detener temporalmente la ejecución de una instrucción para aceptar la introducción de información por vía del teclado.

También se pueden efectuar pruebas condicionales con el comando *IF-THEN* (similares a las que se incluyen en los lenguajes de programación como Basic o Cobol) para proporcionar al programa trayectorias lógicas referente a donde continuar la ejecución si se cumple o no la condición.

Si se tiene alguna experiencia o conocimiento de programación, ya sea en Basic o en cualquier otro lenguaje, las macroinstrucciones serán más fáciles de entender y de utilizar.

Las macroinstrucciones pueden resultar difíciles de someter a depuración (buscar y eliminar pequeños errores de programación). En tales casos, la opción que permite hacer que se ejecute una macro paso a paso ayuda a resolver este problema.

ELEMENTOS DE CONFIGURACION DE LAS MACROS.

Existen varios elementos de configuración en las macros como, por ejemplo, (?), la cual se asemeja al comando Input del lenguaje de programación Basic. Cuando 1-2-3 encuentra el macrocomando (?) en una macroinstrucción, se hace una pausa en la ejecución y espera a que el usuario introduzca alguna información por vía del teclado. Una vez que se digita tal información se almacena y se mantiene en la celda actual.

La siguiente tabla muestra una colección completa de los elementos de las macros usados para representar a las teclas especiales:

Elemento	Función
(Edit)	Permite la edición de la celda en la que se encuentra el cursor.
(Name)	Muestra el menú de con los nombres de rango actuales en la modalidad (Point) y en conjunción con el macro comando (Goto).
(Abs)	Hace que una dirección de celda pase de relativa a absoluta a mixta y de nuevo a relativa en forma ciclica en las modalidades (Edit) y (Point).
(Goto)	Desplaza el cursor a la celda especificada.

- (Window) Cambia el cursor entre las dos ventanas cuando hay una pantalla dividida.
- (Query) Repite el procedimiento /Data Query más reciente.
- (Table) Repite el procedimiento /Data Table más reciente.
- (Calc) Recalcula fórmulas de la hoja de trabajo en la modalidad (Ready); convierte una fórmula en su valor actual en modalidades (Value) y (Edit).
- (Graph) Muestra la gráfica que fue especificada más recientemente.
- (Up) Desplaza el indicador una celda hacia arriba.
- (Down) Desplaza el indicador una celda hacia abajo.
- (Right) Desplaza el indicador una celda hacia la derecha.
- (Left) Desplaza el indicador una celda hacia la izquierda.
- (Bigleft) Desplaza el indicador una pantalla hacia la izquierda.
- (Bigrigh) Desplaza el indicador una pantalla hacia la derecha.
- (Pgup) Desplaza el indicador a la pantalla anterior.

- (Pgdn) Desplaza el indicador a la pantalla siguiente.
- (Home) Desplaza el indicador a la celda A1.
- (End) Desplaza el indicador a la última celda del área de trabajo activa.
- (Delete) Borra el carácter a la derecha del cursor en la modalidad [Edit].
- (Insert) Alterna entre las modalidades inserción y sobrescritura en la modalidad [Edición].
- (Esc) Cancela la entrada actual o rango, o vuelve al paso anterior del comando.
- (Backspace) Borra el carácter a la izquierda del cursor; si un rango es seleccionado, borra el rango en cuestión.
- (?) Detiene la ejecución de la macro temporalmente, permitiendo al usuario introducir información o realizar otra tarea; la macro continuará ejecutándose cuando el usuario pulse [Enter].
- Corresponde a pulsar la tecla [Enter].

(~) Genera el tilde ~

(()) Genera la llave ()

(|) Genera la llave |

Para poder incluir cualquier tecla de función especial del teclado en una macro, encierre el nombre de dicha tecla entre las llaves (()). La única excepción es la tecla [Enter] que se indica con un tilde (~).

Para indicar que se debe pulsar la tecla varias veces, solo se tiene que poner el número de repeticiones entre las llaves a continuación del nombre de la tecla. Por ejemplo, (Right 10) significa que el cursor se desplazará 10 veces a la derecha.

LOS COMANDOS INVISIBLES DE LAS MACROS.

1-2-3 incluye un juego de comandos especiales de macro a los que se les llama macrocomandos invisibles, ya que no pueden digitarse desde el teclado, sino que únicamente pueden ser activados desde la macroinstrucción de la cual forman parte.

Muchos de estos comandos son comparables a los comandos de los lenguajes de programación, tales como Cobol o Basic, que permiten realizar comparaciones lógicas, bifurcar, asignar valores a celdas, ejecutar ciclos, etc.

1-2-3 incluye un número de comandos para macro o macrocomandos que permite a las macroinstrucciones hacer algo más que simplemente

reproducir las secuencias de tecla.

En la versión 2.0, Lotus ha actualizado los comandos para macro, cambiando el lenguaje de comandos de Symphony con los comandos originales para macro que tenía 1-2-3.

El resultado ha sido un nuevo y poderoso paquete de comandos para macros que ofrece las ventajas que posee un verdadero lenguaje de programación. Esta serie de comandos incluye tanto los comandos especiales originales de la versión 1A, como una colección del lenguaje de comandos de Symphony.

La mayoría de los macrocomandos necesitan uno o más datos adicionales para poder operar. Por ejemplo, el macrocomando (Branch) necesita la inclusión de la dirección de la celda donde debe bifurcar.

Esta información adicional es conocida como el "argumento" del macrocomando. Si fuera necesario incluir más de un argumento, estos se separan mediante un "separador de argumentos" el cual por definición es la coma "," para la versión en inglés.

CREACION Y ACTIVACION DE MACROS.

Toda macroinstrucción que sea incluida en una hoja de trabajo como una serie de rótulos debe llevar un nombre. De igual forma a los rangos que contienen a estas macros se les debe asignar un nombre que los identifique.

La diferencia consiste en que el nombre de una macroinstrucción que va a ser activada directamente desde el teclado, tiene que reunir los siguientes requisitos:

1. El nombre de la macro no puede tener más de un carácter.
2. El carácter elegido para nombrar la macro debe ser alfabético o el número 0, y
3. La macroinstrucción debe ir antecedida de una barra diagonal invertida "\".

Todas las macroinstrucciones, con excepción de aquellas que llevan el nombre de \0 (las cuales se ejecutan automáticamente cuando se carga el archivo a la memoria de la computadora) se activan oprimiendo la tecla (Alt) seguida de la letra correspondiente al nombre de la macro.

Por ejemplo, si a la macroinstrucción que queremos ejecutar se le da el nombre \a, para ejecutarla no hay más que oprimir las teclas (Alt)-(A). Tan pronto como se emita dicho comando de activación, la macro comienza a realizar su tarea.

El símbolo "\", que va incluido en todo nombre de macro, representa que se debe pulsar la tecla (Alt) y a continuación la letra que le sigue.

Si no existen ni errores ni instrucciones especiales dentro de la macroinstrucción, la ejecución continuara hasta el final.

Una sola celda puede contener varios golpes de tecla o macrocomandos. Los que sean muy largos, o que incluyan comandos especiales, se tienen que dividir en dos o más celdas.

Cuando 1-2-3 comienza a ejecutar una macroinstrucción, la ejecución del programa permanece en la primera celda hasta finalizar con todos los golpes de tecla o macrocomandos que hayan sido almacenados en la misma. Terminando con esta celda, 1-2-3 comenzara a recorrer la

columna hacia abajo, celda por celda, para seguir con la ejecución de la macro.

Si 1-2-3 se encuentra con una celda que está en blanco, la ejecución se detiene.

Las macroinstrucciones son autómatas que carecen de la capacidad de detectar la existencia de un error al momento de su codificación. Por ejemplo, si se se teclea (Got) no ocurre ningún error hasta el momento de ejecutar la macro. Si se encuentra algo como esto, tratará de interpretarlo y, al no lograrlo, emitirá un mensaje de error.

Se debe tener mucho cuidado al construir las macros para evitar estos errores; ya que un espacio en blanco o un tilde mal colocado pueden ocasionar muchos problemas en la solución de sus métodos numéricos.

Al igual que sucede con programas codificados o escritos en otros lenguajes, las macros de 1-2-3 generalmente necesitan ser sometidas a una depuración antes de que puedan ser utilizadas.

Las dos versiones de 1-2-3 disponen de un instrumento sumamente útil que facilitan la labor de depuración, dicho instrumento es la función [Step]. Cuando 1-2-3 pasa a esta modalidad de operación, todas las macros se ejecutan paso a paso.

Cuando se encuentra bajo esta modalidad de operación, 1-2-3 realiza una pausa entre cada pulsación de tecla que contiene la macroinstrucción.

Para activar la modalidad de operación [Step] digite la secuencia de teclas [Alt][F2] en la versión 2.0.

Al hacer esto, el indicador de modalidad de operación cambiara su mensaje al de [Step], el cual será reemplazado por el de [SST] tan

pronto como se ejecute la macro. Al iniciar la ejecución después de activada dicha modalidad, la macro avanzará paso a paso.

Luego de ejecutar cada paso, la macroinstrucción hará una pausa esperando que se oprima cualquier tecla antes de proseguir con la tarea.

LENGUAJE DE MACROCOMANDOS DE LOTUS.

Este lenguaje de macrocomandos contiene todos los elementos necesarios para desarrollar los métodos numéricos que se presentan en esta tesis que permiten automatizar el control de las hojas de trabajo, eliminando así el requisito de que para configurar dichas macroinstrucciones el alumno deba tener el conocimiento completo de cómo usar 1-2-3.

Con el lenguaje de macrocomandos, es posible configurar los diferentes métodos numéricos en macroinstrucciones que permitan que una persona que apenas sepa de qué trata Lotus 1-2-3, realice tales tareas.

REGLAS GRAMATICALES QUE RIGEN A LOS COMANDOS PARA MACROINSTRUCCIONES.

El formato de los comandos para macro del tipo /x se basa en la forma de cómo se activan todos los comandos de 1-2-3 desde los menús. El comando debe comenzar con una barra invertida "/" seguida de la letra que active el comando o subcomando deseado. Los argumentos, si se requieren, deben escribirse a continuación de dicho comando. Cada

argumento debe estar seguido de un tilde '~', el cual representa la tecla [Enter].

Los comandos para macro derivados del lenguaje de comandos de Symphony obedecen a unas reglas gramaticales o sintaxis especial, similar a las de las funciones @. La sintaxis general de los comandos para macro es:

(Macrocomando)

o

(Macrocomando argumento1,argumento2,...,argumentoN)

donde los argumentos 1 a N estan separados entre si por una coma para la versión en inglés.

Cada macrocomando que se incluye en una macroinstrucción debe tener la estructura gramatical adecuada.

(Macrocomando arg1,arg2,...,argN)

↑	↑	↑	↑
↑	↑	↑	Liave derecha
↑	↑	Separador de argumento	
↑	Espacio en blanco		
Liave izquierda			

El programador deberá suministrar los argumentos que necesite el macrocomando; por ejemplo si se dese que /-2-5 coloque el numero 10 en el rango de celda nombrada como x, se usa el macrocomando (Let) cuya sintaxis es la siguiente:

(Let posición,entrada:sufijo)

asi que para nuestro proposito tendremos que escribir:

{Let x,10}

Los macrocomandos incorrectos en su sintaxis generaran un mensaje de error al activar la macroinstruccion, no cuando se configura. A continuacion se dan las reglas gramaticales de 1-2-3 para los macrocomandos avanzados.

1. Los macrocomandos avanzados deben ser introducidos en la hoja de trabajo como letreros o como formulas cuyo valor genere un letrero al ser evaluado.
2. Cada macrocomando debe comenzar con un llavecilla "{", seguida de la palabra que identifica al macrocomando.
3. Inmediatamente despues del macrocomando debe aparecer un espacio en blanco y a continuacion uno o mas argumentos, cada uno separado por una coma; no debe haber espacios en los argumentos. Cada argumento debe ser del tipo correcto: numerico o alfanumerico o de situacion (indicado por la direccion de celda o por un rango). Algunos macrocomandos requieren de mas de un argumento otros no necesitan ninguno.

Los argumentos deben ser de un solo tipo, o contienen un valor numerico o una cadena alfanumerica, aunque a veces se puede usar cualquier tipo, siempre y cuando se especifique. Esta especificacion se hace en el ultimo argumento. Por ejemplo:

{Let x,10:string} introduce el letrero "10" en la celda llamada x

4. Cada comando debe terminar con una llave "}". Todo el macrocomando debera estar contenido en una sola celda. Nunca podra tener la llave de comienzo "{" en una celda y la llave de cierre "}" en otra.
5. Se pueden incluir varios macrocomandos en una misma celda, siempre

y cuando no se exceda de 240 caracteres. También se pueden mezclar en una misma celda macrocomandos avanzados con pulsaciones de tecla que correspondan al menú de 1-2-3.

6. Se pueden utilizar mayúsculas, minúsculas o una combinación de ambas en una macroinstrucción.

Los siguientes macrocomandos avanzados demuestran la sintaxis apropiada, utilizando el macrocomando (Blank posición) como ejemplo. Este macrocomando borra el contenido de una celda o rango determinado.

(Blank 1) Borra el contenido del rango llamado 1

Los macrocomandos de 1-2-3 pueden ser agrupados en seis categorías, ellos son:

1. Los comandos para operaciones lógicas.
2. Los comandos para bifurcaciones y subrutinas.
3. Los comandos para verificar la introducción de datos hecha por vía del teclado.
4. Los comandos para guardar y borrar información.
5. Los comandos para el procesamiento de archivos.
6. Los comandos para la edición y actualización de los procesamientos de archivos.

PRUEBAS CONDICIONALES EN UNA MACROINSTRUCCION

El macrocomando *IF* se usa para hacer una prueba condicional que hace que la macroinstrucción ejecute los comandos o las instrucciones que sigan a dicho macrocomando en la misma celda si la condición resulto ser verdadera.

Cuando se utiliza este comando, se puede implementar el tipo de logica condicional *IF-THEN-ELSE* que se encuentran en los lenguajes de programación tales como Basic, Cobol o Pascal. El formato general para dicho comando es:

```
(if condicion)(comando(s) a ejecutar si la condición se cumple)
```

Si la condición se cumple se ejecutarán los comandos restantes que se encuentren en la misma celda después de la prueba condicional.

La aplicación más común del macrocomando (if) consiste en hacer bifurcaciones a otra parte de la macro o para llamar a otra subrutina. Si es necesario bifurcar, si se cumple alguna condición, simplemente se coloca el macrocomando (Branch) en la misma celda a continuación del macrocomando (if).

Supongase, por ejemplo, que se desea comprobar el valor de la función $f(x)$ que sea cercano a cero. Si el absoluto de dicho valor es menor o igual a 0.00001 entonces que termine la ejecución de la macro. La macro incluirá las siguientes instrucciones:

```
(if @Abs(fx)<1e-5)(Quit)
```

```
(Branch j30)
```

La primera línea está contenida en una sola celda. A la parte de la celda que sigue después de la porción *if* se le llama cláusula *Then*. Esta cláusula se ejecutará únicamente si el resultado de la prueba

lógica es verdadera.

En este caso, la cláusula Then contiene el macrocomando (Quit) que significa suspender.

La segunda línea contiene implícita la cláusula Else, la cual se ejecutará únicamente si el resultado de la prueba lógica de la instrucción (if) resultara ser falsa.

En este caso, la cláusula Else contiene la instrucción (Branch), la cual bifurca la continuación de la macro hacia la celda j30. Obsérvese que en este caso, la cláusula Else Transfiere el control a otra parte de la macro para realizar otra iteración.

Si las instrucciones de la macro de la cláusula Then no transfieren el control, la celda que se halla por debajo de la instrucción (if) (que es la cláusula Else) se ejecutará después de la instrucción o las instrucciones de la cláusula Then.

El macrocomando (if) le imparte mucha fuerza al lenguaje para macros de 1-2-3. De hecho todas las codificaciones de los métodos numéricos utilizan este macrocomando para hacer pruebas condicionales en determinado momento. Sin embargo, su desventaja consiste en que, si se requiere ejecutar más de un comando para macro después de la prueba lógica, la cláusula Then deberá contener una instrucción que obligue a una bifurcación o una llamada de subrutina.

Si el código de la cláusula Then no se bifurca ni termina con el comando (Quit), la macroinstrucción continuará con la ejecución sin detenerse, pasando inclusive a través de la cláusula Else.

El macro comando (Branch) corresponde a uno de los comandos más sencillos disponibles para hacer que 1-2-3 transfiera el control de la macro a otro sitio.

TERMINO DE LA EJECUCION DE UNA MACROINSTRUCCION.

El comando (Quit) hace que la macroinstrucción se detenga tan pronto como este comando se ejecute. Resulta muy útil detener la ejecución de una macro tan pronto se cumpla alguna prueba condicional.

Aun cuando no se incluya la instrucción (Quit) al final de una macro, 1-2-3 se detiene automáticamente cuando el programa encuentra una celda en blanco o una celda que contenga un valor numérico.

Sin embargo, no causa ningún problema incluir el comando (Quit) al final de cada macro para indicar que la ejecución se detenga en ese punto.

COMANDOS PARA VERIFICACION.

Una variante de la instrucción (Get) es el macrocomando (Getnumber).

La instrucción (Getnumber) acepta la entrada como una fórmula e introduce su valor numérico en la celda indicada y se usa únicamente para números. La sintaxis para este comando es:

`(Getnumber cadena-solicitud,posición)~`

MACROCOMANDOS PARA ALMACENAR INFORMACION.

1-2-3 contiene varios comandos para almacenar información de las celdas. Estos comandos no funcionan con entradas hechas directamente por el teclado, sino con valores ya incluidos en la hoja de trabajo.

El más sencillo de los comandos para almacenar es (Let). Guarda un valor numérico o de cadena alfanumérica en una celda dada. La sintaxis general para este comando es:

(Let posición, entrada:sufijo)~

Este comando es especialmente útil para llenar aquellas celdas que se utilizan como variables en la macro.

También es posible utilizar cadenas alfanuméricas con el macrocomando (Let). Incluso, se puede usar una fórmula para cadena alfanumérica.

Una aplicación muy usada en la codificación de los métodos numéricos es el cálculo de una variable por medio de valores ya contenidos en otras celdas. Por ejemplo:

(Let x, x-fx/df)~

Que significa: Colocar en el rango llamado x, el valor contenido en el rango llamado x menos el valor contenido en el rango llamado fx entre el valor contenido en el rango llamado df. De este modo, se puede cambiar el valor anterior del rango x por otro.

EL COMANDO PARA ALMACENAR PUT.

La instrucción (Put) se utiliza cuando la celda de destino se especifica en referencia a las coordenadas de intersección de un renglón y una columna comprendida dentro de un rango especificado. En este caso, el renglón superior del rango especificado en el macrocomando se toma como el renglón cero y la columna más a la izquierda (en ese mismo rango) como la columna cero. La sintaxis general de dicho comando es:

(Put posición, columna, renglón, entrada:sufijo)~

donde posición se refiere a un rango de celda, columna y renglón pueden ser cero o un número entero con signo positivo, y entrada puede ser un número, una fórmula o un letrero.

Es importante hacer notar que la posición, columna renglón es relativa a la coordenada (renglón 0, columna 0) que corresponde a la esquina superior izquierda del rango especificado.

MACROCOMANDOS PARA DESACTIVAR LA PANTALLA.

Un problema que se presenta con las macroinstrucciones de 1-2-3 consiste en que la pantalla se desplaza para reflejar los distintos pasos por los que atraviesa una macro durante su ejecución; y este problema, en el caso de las macroinstrucciones más complejas puede tornarse en algo grave.

El continuo centelleo de la pantalla constituye un problema por dos razones: la primera, por el tiempo que 1-2-3 pierde cada vez que

cambia lo que despliega la pantalla, lo cual se refleja en la disminución de la velocidad de ejecución; y segundo, porque los cambios que tienen lugar en la pantalla pueden resultar incómodos para el usuario.

Mediante el macrocomando (Windowsoff) se puede "congelar" la parte inferior de la pantalla y dejar a la vista solo el panel de control. Mientras la macro realice la aplicación de que se trate, es posible congelar la pantalla con el macrocomando (Windowsoff).

Cuando la macroinstrucción ya haya concluido su labor, se puede restaurar la pantalla a la actividad normal con la instrucción (Windowson). Si la ejecución de la macro no funciona bien cuando el macrocomando (Windowsoff) está activado se pueden presentar problemas.

Se recomienda tener preparada una macro sencilla, solo para emitir el macrocomando (Windowson) para poder recuperar el uso de la pantalla. Porque, de otra manera, se tendrá que reinicializar la computadora, volver a cargar 1-2-3 y comenzar la aplicación de la macro desde el principio.

Por esta causa, se deben desarrollar y probar las macros sin incluir los macrocomandos (Windowsoff) y (Windowson). Una vez depuradas se anexas estos macrocomandos.

Se puede también activar y desactivar el panel de control de la pantalla con los macrocomandos (Paneloff) y (Panelon). Estos dos macrocomandos funcionan igual que (Windowsoff) y (Windowson).

Al utilizar los macrocomandos (Paneloff) y (Panelon) se puede tener un importante efecto sobre el tiempo de ejecución de la macroinstrucción. Desactivando el control de la pantalla se puede simplificar la

aparición de una macroinstrucción impidiendo que el texto aparezca desplegado.

En una aplicación compleja, el uso de estos macrocomandos (para congelar la pantalla por completo) redujo el tiempo de ejecución de las macros en un cincuenta por ciento: de 5 minutos a 2 1/2 minutos. Claro que, el mayor o menor aumento de velocidad dependerá de la aplicación de que se trate.

A decorative frame consisting of a central white rectangle with a thick black border. This central rectangle is enclosed within a larger, more complex frame made of thick, textured horizontal bars at the top and bottom, and vertical bars on the left and right sides. The vertical bars have small rectangular protrusions at their ends, resembling a wooden frame or a display case.

Capítulo: VII

**Codificación de los
Métodos Numéricos**

CODIFICACION DE LOS METODOS NUMERICOS

En este capítulo se presenta la codificación de los métodos numéricos.

El listado consta de dos partes principales como, por ejemplo,

```
H21: '(Paneloff)/rncfx~f3~/rncx0~f4~/rncxb~f5~'
```

1. La referencia de la izquierda indica en qué celda debe ser colocada la macroinstrucción, en este caso H21.

Si se desea se puede cambiar la posición de la macro siempre y cuando se respeten las direcciones de bifurcación. Por ejemplo, si se cambia la posición de la macro de:

```
H21: '(Paneloff)/rncfx~f3~/rncx0~f4~/rncxb~f5~'
```

.

.

.

```
H32: '(if @Abs(fxx)<0.00001){Branch H35}'
```

.

.

.

```
H35: '(Down 2){Right}~Raiz = ~(Right)/rvx~~(Down 2){Up 2}'
```

a la celda L38, se debe escribir:

```
L38: '(Paneloff)/rncfx`f3`/rncx0`f4`/rncxb`f5`
```

```
L50: '(if @Abs(fxx)<0.00001)(Branch L53)
```

```
L53: '(Down 2)(Right)^Raiz =^(Right)/rvx^(Down 2)(Up 2)
```

2. La macroinstrucción: que comienza desde el apostrofe (incluyéndolo), en este caso:

```
'(Paneloff)/rncfx`f3`/rncx0`f4`/rncxb`f5`
```

COMENTARIOS.

1. Se debe tener cuidado de incluir todos y cada uno de los comandos de la macroinstrucción.
2. No se deben incluir espacios entre los argumentos de los macrocomandos porque 1-2-3 tomaría el argumento como una cuerda alfanumérica.
3. No es indispensable que la codificación se haga con mayúsculas o minúsculas ya que 1-2-3 no diferencia entre las dos formas.
4. El nombre de un rango no debe ser igual que la dirección de una celda. De esta forma, no se puede nombrar al rango F3 como X1 porque

1-2-3 tomará la dirección de la celda X1 y no el rango nombrado como X1.

5. Para comenzar la ejecución de la macro, la celda activa en ese momento debe ser la celda que tomará el nombre de fx (.../rncfx~f3~...). En este caso la celda activa en el momento de iniciar la macro debe ser f3.

6. No debe estar activa la tecla Scroll Lock. Esta tecla permite desplazar la pantalla entera un renglón o una columna en cualquier dirección cada vez que el indicador de celda se desplace. Esto no es conveniente porque el indicador de celda no cambia de posición, sino la pantalla.

7. Para introducir la ecuación a resolver se debe hacer desde el teclado. Si la ecuación comienza con una letra, o nombre de variable es necesario antecederla del signo + o - según sea el caso. Si la ecuación contiene alguna función integrada de 1-2-3 tales como, @sin, @cos, @tan, @Exp, @Ln, @Log, @Abs, deben comenzar con el símbolo especial "@". No deben incluirse espacios. Presione [Enter].

8. Los valores de las variables a utilizar, se puede introducir mediante una fórmula matemática o función integrada de 1-2-3. No deben incluirse espacios. Presione [Enter].

9. Si la macroinstrucción no funciona correctamente durante su ejecución, se puede interrumpir presionando simultáneamente las teclas [Ctrl]-[Break], esta interrupción genera un error, acepte el error presionando [Enter]. Regrese a la parte de la hoja de cálculo donde se encuentra colocada la macro e inicie la búsqueda del error.

10. Una vez depurada la macroinstrucción no olvide salvar la hoja de cálculo, en ella también se salvará la macro.

METODO DE BISECCION

```
H21: '(paneloff)/rncrx`r3`/rncx0`r4`/rncxb`r5`
H22: '/rncfxx`h4`/rncf0`h5`/rncfb`h6`/rncx`h2` (panelon)
H23: ' (?)` (down) (?)` (down)
H24: ' (?)` (paneloff)
H25: '(let x,x0)` (let f0,fx)` (let x,xb)` (let fb,fx)`
H26: '(if f0*fb>0) (panelon) (branch h24)
H27: '/rea3..a16` (goto)a4` iteraciones a la convergencia` (down) x`
H28: '(right 2) f(x)` (goto)a6` /uth
H29: '(let x,(x0+xb)/2)`
H30: '(let fxx,fx)`
H31: '/rvx` (right 2) /rvfxx` (down) (left 2)
H32: '(if @abs(fxx)<0.00001) (branch h35)
H33: '(if fxx=f0>0) (let x0,x)` (branch h29)
H34: '(let xb,x)` (branch h29)
H35: '(down 2) (right) Raiz: x=` (right) /rvx` (down 2) (up 2)
```


METODO DE LA REGLA FALSA

```
H21: '(paneloff)/rncix`f3~/rncx0`f4~/rncxb`f5`
H22: '/rncfxx`ha~/rncf0`h5~/rncfb`hb~/rncx`h2'(panelon)
H23: '{?}`(down){?}`(down){?}`(paneloff)
H24: '/rea3..a14`(goto)a4`iteraciones a la convergencia`(down)`x`
H25: '(right 2)`f(x)`(goto)a6~/with
H26: '((let x,x0)`((let f0,fx)`((let x,xb)`((let fb,fx)`
H27: '((let x,(x0+fb-xb*f0)/(fb-f0))`
H28: '((let fxx,fx)`
H29: '/rvx`^(right 2)~/rvfxx`^(down)`((left 2)`
H30: '(if @abs(fxx)<0.00001)(branch h33)
H31: '(if f0=fb>0)(let x0,x)`(branch h26)
H32: '((let xb,x)`(branch h26)
H33: '(down 2)(right)`Raiz: x=`(right)/rvx`^(down 2)(up 2)
```

METODO DE SUSTITUCION DIRECTA

```
H21: '(Paneloff){?}~/rea3..gl1~/wcr/rncgx`f4~/rncx`f5~/rncxx`h2~(Gotoigx~/w  
ca27`  
H22: '(Let e4,g(x) = )^(Let e5,x = )^(Panelon)  
H23: '{?}^(Down){?}^(Paneloff)  
H24: '(Goto)a5`x^(Right)`g(x)^(Down)(Left)/wth  
H25: '/rvx^(Right)/rvgx^(Down)(Left)  
H26: '(Let xx,gx)^(lt @Abs(xx-x)<0.0001)(Branch h28)  
H27: '(Let x,xx)^(Branch h25)  
H28: '(Down)(Right)Raiz =^(Right)/rvx`
```

METODO DE NEWTON - RAPHSON

```
G21: '(paneloff)//rncfx`r3`/rncdf`f4`/rncn`a102`
G22: '/rncx`f5`(panelon){?}`(down){?}`(down){?}`(paneloff)/res3..a9`
G23: '{goto}a4`iteraciones a la convergencia;`
G24: '(down)`x`(right 2)`f(x)`(goto)a6`/wth
G25: '{if @abs(fx)<0.00001}(branch g31)
G26: '/rvX``(right 2)/rvfX``(down)(left 2)
G27: '{if df=0}(let x,x+0.0001)`(branch g25)
G28: '{let x,x-fx/df}`(let n,n+1)`
G29: '{if n>65}(down)(right 2)`No converge`(down 2)(up 2)(quit)
G30: '(branch g25)
G31: '/rvf5``(right 2)/rvf3``
G32: '(down 2)(left 2)`Raiz: x=`(right)/rvf5``(down 2)(up 2)
G33: '/rva6`x`
```

METODO DE LA SECANTE

```
H21: [W27] '(paneloff)/rncfx`f3~/rncx0`f4~/rncxb`f5~
H22: [W27] '/rncfxx`h4~/rncf0`h5~/rncfb`h6~/rncx`h2~(panelon)
H23: [W27] '(?)^(down){?}^(down){?}^(paneloff)
H24: [W27] '/rea3..a12'(goto)a4`iteraciones a la convergencia^(down)`x`
H25: [W27] '(right 2)`f(x)^(goto)a6~/wth
H26: [W27] '((let x,x0)^(let f0,fx)^(let x,xb)^(let fb,fx)`
H27: [W27] '((let x,(x0+fb-xb*f0)/(fb-f0))`
H28: [W27] '((let fxx,fx)`
H29: [W27] '/rvx^(right 2)/rvfxx^(down){left 2}
H30: [W27] '(if @abs(fxx)>0,00001)(let x0,x)(branch h26)
H31: [W27] '(down 2)(right)`Raiz: x^(right)/rvx^(down 2)(up 2)
```

METODO DE WENGSTEIN

```
H21: '(paneloff)/rncfx`f3~/rncx0`f4~/rncxb`f5`
H22: '/rncg0`h5~/rncgb`h6~/rncx`h2~(panelon)
H23: ' (?)~(down) (?)~(down) (?)~(paneloff)
H24: '/rea3..a12^(goto)a4^iteraciones a la convergencia^(down)`x`
H25: '(right 2)^(f(x)^(goto)a6~/wth
H26: '(let x,x0)^(let g0,fx*x0)^(let x,xb)^(let gb,fx*xb)~
H27: '(let x,(xb*g0-x0*gb)/(xb-x0-gb*g0)^(let xb,x0)^(let x0,x)~
H28: '/rvx0^(right 2)/rvfx^(down)(left 2)
H29: '(if @abs(fx)>0.00001)(branch h26)
H30: '(down 2)(right)~Raiz: x=(right)/rvx^(down 2)(up 2)
```

METODO DE MULLER

```

H21: '(paneloff)/rncrx`f3`/rncxs0`fa`/rncxsl`f5`/rncxs2`f6`/rncx`n2`/rncp`
h1`
H22: '/rncfs0`h5`/rncfs1`h6`/rncfs2`h7`/rncfs1`h8`/rncfs2`h5`/rncdel`n10`
H23: '/rncde2`h11`/rncdm`h12`/rnce`h13`/rncd`h14`/rncb`h15`/rncn`n16` (pane
on)
H24: '(?)` (down) (?)` (down) (?)` (down) (?)` (paneloff)
H25: '/rea3...a12` (goto) a5` iteraciones a la convergencia` (down) `x`
H26: '(right 2)` f(x)` (goto) a7` /wth
H27: '(let x, xs0)` /rvfx` fs0` (let x, xs1)` /rvfx` fs1` (let x, xs2)` /rvfx` fs2`
H28: '(let hs1, xs1-xS0)` (let hs2, xs2-xS1)` (let del, (fs1-fs0)/hs1)`
H29: '(let de2, (fs2-fs1)/hs2)` (let dm, (de2-del)/(hs2-hs1))`
H30: '(let b, de2+hs2*dm)` (let d, @sqrt(b`2-4*fs2*dm))`
H31: '(if @abs(b-d)<@abs(b+d)) (let e, b+dm) (branch h35)
H32: '(let e, b-d)`
H33: '(let h, -2*fs2/e)` (let p, xs2+h)`
H34: '(if @abs(h)<1e-5) (branch h40)
H35: '(let xs0, xs1)` (let xs1, xs2)` (let xs2, p)` (let hs1, xs1-xS0)`
H36: '(let hs2, xs2-xS1)` (let x, xs1)` (let fs1, fx)` (let x, xs0)` (let fs0, fx)`
H37: '(let x, xs2)` (let fs2, fx)` (let del, (fs1-fs0)/hs1)` (let de2, (fs2-fs1)
hs2)`
H38: '(let dm, (de2-del)/(hs2-hs1))`
H39: '(branch h30)
H40: '(down) (right) `Raiz: x=` (right) /rvx` (down 2) (up 2)

```

METODO DE HOOKE - JEEVES

```

H21: '(paneloff)/rncdx`f3`/rncx0`f4`/rncxb`h3`/rncdx`f5`
H22: '/rncf0`h5`/rncfb`h6`/rncx`h2`(panelon)
H23: '(?)`((down)(?))`((down)(?))`(paneloff)
H24: '(let x,x0)`(let f0,fx)`
H25: '/rea3.,a14`((goto)a4`Iteraciones a la convergencia`((down)`x`
H26: `(right 2)`fx)`((goto)a6`/wth
H27: '(let x,x0)`(let f0,fx)`(if @abs(f0)<0.00001)(branch h38)
H28: '(let xb,x0-dx)`(let x,xb)`
H29: '/rvx0`((right 2)/rvf0`((down)(left 2)
H30: '(let fb,fx)`
H31: '(if @abs(fb)>@abs(f0))(branch h33)
H32: '(let x0,xb)`(branch h27)
H33: '(let xb,x0-dx)`(let x,xb)`(let fb,fx)`(if @abs(fb)>@abs(f0))(let dx,d
x/2)`(branch h35)
H34: '(branch h32)
H35: '(if dx<0.00000001)(branch h37)
H36: '(branch h28)
H37: `(down 2)(right 2)`No converge`quit)
H38: '/rvx0`((right 2)/rvf0`((down)(left 2)
H39: `(down 2)(right)`Raiz: x=`(right)/rvx`((down 2)(up 2)(quit)

```

ECUACION CUADRATICA

```

H21: '(paneloff)/rnc`f3~/rncb`f4~/rnc`f5~
H22: '/rncd`h1~/rncpr`h2~/rncpl`h3~/rncxb`h4~/rncxc`h5~(panelon)~
H23: '(?)^(down){?}^(down){?}^(paneloff)
H24: '{let d,b^2-4*a*c}~
H25: '{if d<0}{branch h32}~
H26: '{let xb,(-b+sqrt(d))/(2*a)}~
H27: '{let xc,(-b-sqrt(d))/(2*a)}~
H28: '{goto}b12` RAICES REALES:~(down 2){left}`x1 =~
H29: '{right}/rvxb~
H30: '{down}{left}`x2 =~
H31: '{right}/rvxc~{goto}f1`{quit}
H32: '{let pr,-b/(2*a)}~
H33: '{let pl,sqrt(@abs(d))}~
H34: '{goto}b12` RAICES COMPLEJAS:~(down 2){left}`x1 =~
H35: '{right}/rvpr~(right)`+~(right)/rvpl~(right)`i~(down){left 4}`x2 =~
H36: '{right}/rvpr~(right)`-~(right)/rvpl~(right)`i~{goto}f1`

```


ECUACION CUBICA

```

H21: '(paneloff)/rncas1a`f3`/rncas2a`f4`/rncas3a`f5`
H22: '/rncq`h1`/rncr`h2`/rncd`h3`/rncs`h4`/rncf`h5`/rncx1`h6`
H23: '/rncx2`h7`/rncx21`h8`/rncx3`h9`/rncx31`h10`/rncfet`h11`(panelon)
H24: '(?)`t`down(?)`t`down(?)`t`paneloff)
H25: '(let q,(3*as2a-as1a`2)/9)`(let r,(9*as1a*as2a-27*as3a-2*as1a`3)/54)`
H26: '(let d,q`3+r`2)`
H27: '(if d<=0)(branch h35)
H28: '(let s,@sqrt(r+@sqrt(d)))`t`let t,@sqrt(r-@sqrt(d))`t`
H29: '(let xs1,s+t-as1a/3)`t`let xs2,-0.5*(s+t)-as1a/3)`t`
H30: '(let x21,0.5*sqrt(3)*s-t)`t`let xs3,xs2)`t`let x31,-x21)`t`
H31: '(goto)b12` RAICES COMPLEJAS:`t`down 2)`t`left)`t`x1 =`t`
H32: '(right)/rvxs1`t`down)`t`left 2)`t`x2 =`t`
H33: '(right)/rvxs2`t`right)`t`rvx21`t`right)`t`
H34: '(down)`t`left 4)`t`x3 =`t`right)`t`rvxs3`t`right)`t`rvx31`t`right)`t`left)`t`quit)
H35: '(let tet,57.3*@acos(-r/@sqrt(-q`3))`t`let xs1,2*@sqrt(-q)*@cos(tet*0.017453/3)`t`
H36: '(let xs2,2*@sqrt(-q)*@cos((tet/3+120)*0.017453)`t`let xs3,2*@sqrt(-q)*@cos((tet/3+240)*0.017453)`t`
H37: '(goto)b12` RAICES REALES:`t`down 2)`t`left)`t`x1 =`t`
H38: '(right)/rvxs1`t`down)`t`left)`t`x2 =`t`right)`t`rvxs2`t`down)`t`left)`t`x3 =`t`
H39: '(right)/rvxs3`t`goto)b13`

```

INVERSION DE MATRICES

```
121: "(paneloff)(?)\wcr/rea4.g10(goto)a4/rnci~11~/rncj~12~
122: /rnccl~13~/rnccl~14~(panelon)
123: "(getnumber Número de renglones de la matriz = ,1)(paneloff)
124: "(let c3,"Elementos de la Matriz :")~
125: "(let j,1+1)~
126: /rnc~(right j-1)(down i-1)~(panelon)
127: "(let ci,ci+1)~
128: "(let cj,cj+1)~(?)~(right)(if cj<j)(branch 128)
129: "(let cj,0)~(beep 3)(down)(left j)(if ci<i)(branch 127)
130: "(paneloff)
131: "(down 2)(right 2)Solución de la Matriz :~
132: "(down)(left 2)/rncb~(right j-1)(down i-1)~(goto)a~
133: /dmi.(right j-2)(down i-1)~b~/dm(down i+3).(right j-2)(down i-1)~a~b

134: "(let ci,1)~(goto)b~
135: /re(right i-1)~"x"(right)/rvcl~(right)~(down)(left 2)
136: "(let ci,ci+1)~(if ci<j)(branch 135)
```

METODO DE GAUSS - SEIDEL

```

121: '(paneloff){?}~/wcr/rea4..gl9~/rnci~11~(goto)a4~
122: '/rncsuma~15~/rncal~j~16~/rncxi~17~/rncu~18~/rncal~1~19~/rncal~1~110~(pa
nelon)
123: '(getnumber Número de renglones de la Matriz = .1)~(paneloff)
124: '{let c3,"Elementos de la Matriz"}~/rnci~12~/rncj~13~/rncj~14~(up)
125: '/rncup~111~/rncin~114~(let j,1+1)~(right j)
126: '/rncxsd~112~/rncxcd~113~(panelon)
127: '{let ci,ci+1}~(down){left j}
128: '{let cj,cj+1}~(?)~(right){if cj<j}{branch 128}
129: '{let cj,0}~(beep 3){if ci<1}{branch 127}
130: '(paneloff){down 2}{left j-2}X's supuestas~(down){left 2}/rncxs~(pane
lon)
131: '{let cj,cj+1}~(?)~(right){if cj<1}{branch 131}
132: '(paneloff){down 2}{left 1-2}X's calculadas~(down){left 2}
133: '/rncxc~.~(right 1-1)~
134: '{let ci,0}~(let up,1+3)~
135: '{let ci,ci+1}~(let cj,0)~(let suma,0)~(goto)xs~(let up,up-1)~
136: '{let cj,cj+1}~/c~*i~(up up)/c~*i~j~
137: '{if ci<1}{let xi,0}~/c~*i~
138: '{let suma,suma+ai~*xi}~(down up){right}
139: '{if cj<1}{branch 136}
140: '{up up}/c~*i~1~
141: '{put xc,ci-1,0,(a1i-suma)/ai}~
142: '{let xcd,(a1i-suma)/ai}~(goto)xs~(right ci-1)/rv~xsd~
143: '{if @abs(xcd-xsd)>0.001}{let in,1}~
144: '/rvxcd~
145: '{if ci<1}{branch 135}
146: '{if in=0}{quit}
147: '(goto)xc~/re.(right 1-1)~(let in,0)~(branch 134)

```

METODO DE JACOBI

```

121: '(paneloff)(?)~/wcr/res4..g19~/rnci~11^(goto)A4~
R21: '/frjacobi.wk1~
122: '/rncsuma~15~/rncalj~16~/rncxi~17~/rncu~18~/rncali~19~/rncaini~110^(pa
nelon)
123: '(getnumber Nbeero de renglones de la Matriz = ,i)^(paneloff)
124: '(let c3,"Elementos de la Matriz")~/rnccl~12~/rncclj~13~/rncj~14^(up)
125: '/rncup~11~/rncin~114^(let j,i+1)^(right j)
126: '/rncxsd~112~/rncxcd~113^(panelon)
127: '(let ci,ci+1)^(down)(left j)
128: '(let cj,cj+1)^(?)^(right)(if cj<j)(branch 128)
129: '(let cj,0)^(beep 3)(if ci<i)(branch 127)
130: '(paneloff)(down 2)(left j-2)X's supuestas^(down)(left 2)/rncxs^(pane
lon)
131: '(let cj,cj+1)^(?)^(right)(if cj<i)(branch 131)
132: '(paneloff)(down 2)(left i-2)X's calculadas^(down)(left 2)
133: '/rncxc^(right i-1)~
134: '(let ci,0)^(let up,i+3)~
135: '(let ci,ci+1)^(let cj,0)^(goto)xs^(let up,up-1)~
136: '(let cj,cj+1)/c~xi^(up up)/c~aij~
137: '(if ci=cj)(let xi,0)/c~aii~
138: '(let suma,suma+aij*xi)^(down up)(right)
139: '(if cj<i)(branch 136)
140: '(up up)/c~aini~
141: '(put xc,ci-1,0,(aini-suma)/aii)~
142: '(let suma,0)^(if ci<i)(branch 135)
143: '(goto)xc^(let ci,0)
144: '(let ci,ci+1)~
145: '/c~xcd^(up 3)/c~xsd^(down 3)(right)
146: '(if @abs(xsd-xcd)>1e-4)(let in,1)~
147: '(if ci<i)(branch 144)
148: '(if in=0)(quit)
149: '(goto)xc~/c.(right i-1)~xs~
150: '/re.(right i-1)^(let in,0)^(branch 134)

```

REGRESION LINEAL

```

Z21: *(Paneloff)(Goto/a3~/rea3..h20~/rncn'z1~/rncm'z2~/rnci'z3'X"
Z22: *(Right)Y'(Getnumber Número de datos = ,n)"(Down)
Z23: *(Left)/rncx"(Down n-1)"(Right)/rncy"(Down n-1)"(Panelon)
Z24: *(Let 1,i+1)"(Left)(?)"(Right)(?)"(Down)
Z25: *(if i<n)(Branch z24)
Z26: *(Getnumber "Método que se utilizará [1,2,3,4] = ",m)"(Paneloff)
Z27: *(if m>4)(Quit)
Z28: *(Goto)y"(Right)~/re(Right 3)(Down n-1)"(if m=1)(Branch z32)
Z29: *(if m=2)(Branch z35)
Z30: *(if m=3)(Branch z38)
Z31: *(if m=4)(Branch z42)
Z32: */drrxx"yy"oi(Down n+2)(Left 2)"g
Z33: */(Left 2)*(Down n+9)(Abs)+*(Down n+3)(Right)(Abs)"
Z34: */grgotfRegresión Lineal"qq(Branch z46)
Z35: */@Ln((Left)~/c".(Down n-1)~/drrxx"y.(Down n-1)"oi(Down n+2)(Left
2)"g(Right)
Z36: */@Expi(Down n+3)(Abs)*/@Expi(Left 3)*/(Down n+9)(Left)(Abs))"
Z37: */grgotfRegresión Exponencial"qq(Branch z46)
Z38: */@Log((Left 2)~/c".(Down n-1) (Right)*/@Log((Left 2)~/c".(Down n-1)
Z39: */drrx(Left).(Down n-1)"y.(Down n-1)"oi(Down n+2)(Left 3)"g(Right)
Z40: */10"((Down n+3)(Left)(Abs)*/(Left 4)"/(Down n+9)(Left 2)"/(Abs)"
Z41: */grgotfRegresión Potencial"qq(Branch z46)
Z42: */1/(Left 2)~/c".(Down n-1)"(Right)*/1/(Left 2)~/c".(Down n-1)"(Right)
Z43: */drrx(Left 2).(Down n-1)"y(Left).(Down n-1)"oi(Down n+2)(Left 4)"
g(Left 4)"/(Down n+3)
Z44: */(Left)(Abs)*/(Down n+9)(Left 2)(Abs)*/(Down n+3)(Left)(Abs)*/(Left 4))"
Z45: */grgotfRegresión de Crecimiento"qq
Z46: */c".(Down n-1)~/gtxxx"ay"b.(Down n-1)"o
Z47: */laExperimentales"lbRegresión"fasqtXX"tyY"qvqi(Branch z26)

```

REGRESION POLINOMIAL

```

Z21: *(Paneloff)(Goto)a3/rea3..h20/rncn`z1`/rnca`z2`/rncl`z3`X
Z22: *(Right)Y(Getnumber Número de datos = n)(Down)
Z23: *(Left)/rncx(Down n-1)(Right)/rncy(Down n-1)(Panelon)
Z24: *(Let 1,1+1)(Left)(?)(Right)(?)(Down)
Z25: *(If 1<n)(Branch z24)
Z26: *(Getnumber "Método que se utilizará [1,2,3,4] = ",m)(Paneloff)
Z27: *(If m=4)(Quit)
Z28: *(Goto)y(Right)/re(Right 3)(Down n-1)(If m=1)(Branch z32)
Z29: *(If m=2)(Branch z35)
Z30: *(If m=3)(Branch z38)
Z31: *(If m=4)(Branch z42)
Z32: */drrxx`yy`o(Down n-2)(Left 2)`g
Z33: *+(Left 2)*(Down n+9)(Abs)+(Down n+3)(Right)(Abs)
Z34: */grgotfRegresión Lineal`qq(Branch z46)
Z35: *@Ln((Left))/`c`.(Down n-1)/drrxx`y.(Down n-1)o(Down n+2)(Left 2)`g(Right)
Z36: *@Exp((Down n+3)(Abs))/@Exp((Left 3)*(Down n+9)(Left)(Abs))
Z37: */grgotfRegresión Exponencial`qq(Branch z46)
Z38: *@Log((Left 2))/`c`.(Down n-1)(Right)@Log((Left 2))/`c`.(Down n-1)
Z39: */drrx(Left).(Down n-1)`y.(Down n-1)o(Down n+2)(Left 3)`g(Right)
Z40: *10`((Down n+3)(Left)(Abs))*((Left 4)`(Down n+9)(Left 2)(Abs)
Z41: */grgotfRegresión Potencial`qq(Branch z46)
Z42: *+1/(Left 2)/`c`.(Down n-1)(Right)+1/(Left 2)/`c`.(Down n-1)(Right)
Z43: */drrx(Left 2).(Down n-1)`y(Left).(Down n-1)o(Down n+2)(Left 4)`g`
(Left 4)/(Down n+3)
Z44: *(Left)(Abs)/((Down n+9)(Left 2)(Abs)/(Down n+3)(Left)(Abs)+(Left 4))
Z45: */grgotfRegresión de Crecimiento`qq
Z46: */`c`.(Down n-1)/gtxxx`ay`b.(Down n-1)`o
Z47: *|aExperimentales`|bRegresión`fasqtx`ty`qvq(Branch z26)

```

REGRESION MULTIPLE

```

Z21: '(Paneloff)(Goto)a3~/rea3..h20~/rncn~z1~/rncm~z2~/rnci~z3~ X
Z22: '(Right) Y'(Getnumber Número de datos = ,n)(Down)
Z23: '(Left)/rncx'(Down n-1)'(Right)/rncy'(Down n-1)'(Panelon)
Z24: '(Let 1,1+1)'(Left)(?)'(Right)(?)'(Down)
Z25: '(if {n})(Branch z24)
Z26: '(Getnumber "Método que se utilizará [1,2,3,4] = ",m)'(Paneloff)
Z27: '(if m=4)(Quit)
Z28: '(Goto)y'(Right)~/re(Right 3)(Down n-1)'(if m=1)(Branch z32)
Z29: '(if m=2)(Branch z35)
Z30: '(if m=3)(Branch z38)
Z31: '(if m=4)(Branch z42)
Z32: '/drrxx^yy^o(Down n+2)(Left 2)'g
Z33: '*((Left 2)*(Down n+9)(Abs))*((Down n+3)(Right)(Abs)~'
Z34: '/grgotfRegresión Lineal~qq(Branch z46)
Z35: '@Ln((Left)~/c^,(Down n-1)~/drrxx^y,(Down n-1)~o(Down n+2)(Left
2)'g(Right)
Z36: '@Exp((Down n+3)(Abs))*@Exp((Left 3)*(Down n+9)(Left)(Abs)~'
Z37: '/grgotfRegresión Exponencial~qq(Branch z46)
Z38: '@Log((Left 2)~/c^,(Down n-1)~/Right)@Log((Left 2)~/c^,(Down n-1)~'
Z39: '/drrx(Left),(Down n-1)~y,(Down n-1)~o(Down n+2)(Left 3)'g(Right)
Z40: '*10^((Down n+3)(Left)(Abs))*((Left 4)~/Down n+9)(Left 2)(Abs)~'
Z41: '/grgotfRegresión Potencial~qq(Branch z46)
Z42: '*1/(Left 2)~/c^,(Down n-1)~/Right)*1/(Left 2)~/c^,(Down n-1)~/Right)
Z43: '/drrx(Left 2),(Down n-1)~y(Left),(Down n-1)~o(Down n+2)(Left 4)'g*
(Left 4)/(Down n+3)
Z44: '(Left)(Abs)/(Down n+9)(Left 2)(Abs)/(Down n+3)(Left)(Abs)+((Left 4)~'
Z45: '/grgotfRegresión de Crecimiento~qq
Z46: '/c^,(Down n-1)~/gtxxx^ay^b,(Down n-1)~o
Z47: 'laExperimentales~lbRegresión~fasqtxX^ty~qvq(Branch z26)

```

POLINOMIO DE INTERPOLACION

```

121: '(paneloff)(?)~/wcr/res4..g20\goto)a4~/rnci~11~/rncj~12~/rnca~15~
122: ~/rnci~13~/rncj~14~(panelon)
123: '(getnumber Método que se empleará (1 ó 2) ,a)~
124: '(if a=1)(branch 142)
125: '(if a=2)(branch 127)
126: '(branch 123)
127: '(getnumber Número de Ecuaciones = ,i)~(paneloff)
128: '(let c3,"Elementos de las Ecuaciones :")~
129: '(let j,1+1)~
130: ~/rnca~(right j-1)(down i-1)~(panelon)
131: '(let ci,ci+1)~
132: '(let cj,cj+1)~(?)~(right)(if cj<j-1)(branch 132)
133: '(let cj,0)~(beep 3)(down)(left j-1)(if ci<1)(branch 131)
134: '(paneloff)
135: '(down 2)(right 3)Solución i~(down)(left 3)
136: ~/rncb~(right j-1)(down i-1)~(right)/ca~~(left)1~
137: '/c~(down i-1)~(down i+5)/rnc~(right j-2)(down i-1)~(up i+5)
138: '/dmi.(right j-2)(down i-1)~c(home)/dmc~b~b~/rec~(down i+6)(let ci,0
j~
139: '/re(right i-1)~A~(right)/rvci~~(right)=~(let ci,ci+1)~
140: '(if ci<1)(down)(left 2)(branch 139)
141: '(home)(quit)
142: '(getnumber Número de Datos = ,i)~
143: '~xi~(right)~yi~(down)(left)
144: '(?)~(right)(?)~(down)(left)(let ci,ci+1)~(if ci<1)(branch 144)
145: '(paneloff)(goto)d5~/rnca~(right i)(down i-1)~/rnci~16~/rncj~17~
146: '(goto)a5~(let ci,0)~(let f4,Resolviendo)~
147: '/rv~xi~
148: '(put a,cj,ci,xi~(cj))~(let cj,cj+1)~(if cj<i)(branch 148)
149: '(right)/rv~yi~(put a,i,ci,yi)~(down)(left)
150: '(let ci,ci+1)~(let cj,0)~(if ci<1)(branch 147)
151: '(goto)d5~/dmi.(right i-1)(down i-1)~(down i+1)~
152: '/dem(down i+1).(right i-1)(down i-1)~a~
153: '(down i+1)/re(right i-1)(down i-1)~(up i+1)
154: '/re(right i-1)(down i-1)~(let cj,0)~(let f4,"Solución :")~
155: 'A~(right)/rv~j~~(right)=~(down)(left 2)
156: '(let cj,cj+1)~(if cj<i)(branch 155)
157: '(goto)f4~

```


POLINOMIO FUNDAMENTAL DE NEWTON

```

121: '(paneloff){?}~vcr/rea3..g20~/rncn~z1~/rncxa~z2~/rncxs~z3~/rncyl~z4~
122: '/rncfs~z5~/rncfa~z6~/rnck~z7~/rnci~z8~
123: '(goto)a4'(panelon)(getnumber Número de Datos = ,n)~
124: '~x1'(right)~y1~
125: '(down){left}{?}~(right){?}~(let i,i+1)~(if i<n){branch 125}
126: '~(let d4,Resolviendo)~(paneloff)
127: '~(blank 1)~(goto)d5~/rncf~(right n-1)(down n-1)~(left 2)
128: '/rv~y1~(put f,i,0,y1)~(let i,i+1)~
129: '(down)(if i<n){branch 128}
130: '~(let i,0)~(let k,1)~
131: '(goto)f~(down k)(right i)
132: '/rv(escape)(up)~fa~/rv(escape)(up)(right)~fa~(goto)a5~
133: '/rv(escape)(down i)~xa~/rv(escape)(down i+k)~xa~
134: '(put f,i,k,(fs-fa)/(xs-xa))~(let i,i+1)~
135: '(if i<n-k){branch 131}
136: '~(let k,k+1)~(let i,0)~(if k<n){branch 131}
137: '(goto)f~(right)/re(right n-1)(down n)~(goto)d4~Solución~

```

REGLA DEL TRAPECIO

```
G21: '(paneloff)/rncfx`f3`/rncx0`f4`/rncxn`f5`/rncnp`f6`/rncx`h6`
G22: '/rncx`h1`/rncfn`h2`/rncf0`h3`/rnc1`h4`/rncx1`h5`/rncarea`h12` (panelon
)
G23: '(?)^(down){?}^(down){?}^(down){?}^(paneloff)
G24: '(let h,(xn-x0)/(np-1))^(goto)s12`"Area ="
G25: '(let x,xn)^(let fn,fx)^(let x,x0)^(let f0,fx)`
G26: '(let area,(fn+f0)/2)^(let i,1)`
G27: '(let i,i+1)`
G28: '(let xi,x0+(i-1)*h)`
G29: '(let x,xi)^(let f0,fx)^(let area,area+f0)`
G30: '(let f12,area=h)`
G31: '(if i<np-1)(branch g27)
```

REGLA DE SIMPSON

```

G21: '(paneloff)/rncfx^f3~/rncx0^f4~/rncxn^f5~/rncnp^f6~
G22: '/rncx^h6~/rnci1^h7~/rnci11^h8~/rncm^h9~
G23: '/rncx^h1~/rncfn^h2~/rncf0^h3~/rnci^h4~/rncarea^h12~(panelon)
G24: '(?)^(down)(?)^(down)(?)^(down)(?)^(paneloff)
G25: '(if (np/2-@int(np/2))>0){let m,1}^(let np,np-1)^
G26: '(let h,(xn-x0)/(np-1))^(goto)e12^"Area ="
G27: '{let x,x0+i*h}^(let i,i+1)^(let i1,i+1)^
G28: '{let f0,fx}^(let x,x0+i*h}^(let area,area+f0)^
G29: '{let f0,4*fx}^(let area,area+f0)^(let x,x0+i1*h}^(let f0,fx)^
G30: '{let area,area+f0}^
G31: '{let i,i+2}^
G32: '{let f12,area*h/3}^
G33: '{if i<*(np-2){branch g27}
G34: '{if m=0}{quit}
G35: '{let x,xn}^(let fn,fx)^(let area,area+(fn*f0)/2*(x-xn)}
G36: '{let f12,area*h/3}^

```

REGLA DE SIMPSON 3/8

```

G21: '(paneloff)/rncfx`f3~/rncx0`f4~/rncxn`f5~
G22: '/rncnp`f6~/rncx`h6~/rnci1`h7~/rnci11`h8~/rnci111`h9~
G23: '/rncx`h1~/rncfn`h2~/rncf0`h3~/rnci`h4~/rncarea`h12~(panelon)
G24: '(?)^(down)(?)^(down)(?)^(down)
G25: '(panelon)(?)^(paneloff)
G26: '(if (np/3-@int(np/3))>0)(goto)f6^(beep 4)~/re^(branch g25)
G27: '(let h,(xn-x0)/(np-1))^(goto)e12~"Area ="
G28: '(let x,x0+i*h)^(let i1,i+1)^(let i11,i1+1)^(let i111,i11+1)~
G29: '(let f0,fx)^(let x,x0+i1*h)^(let area,area+f0)~
G30: '(let f0,3*fx)^(let area,area+f0)^(let x,x0+i11*h)^(let f0,fx)~
G31: '(let area,area+3*f0)^(let x,x0+i111*h)^(let f0,fx)^(let area,area+f0)

G32: '(let i,i+3)~
G33: '(let f12,area*3*h/8)~
G34: '(if i<=np-3)(branch g28)

```

INTEGRACION DE ROMBERG

```

G21: '(paneloff)/rncfx`f3~/rnca`f4~/rncb`f5~/rnch`h1~/rnca`h2~/rncfb`h3`
G22: '/rnci`h5~/rncsuma`h6~/rnck`h7~/rncj`h10~/rncx`j7~/rncr2_1`a33~/rncr2_
  1a`a34`
G23: '/rncr`11..bg3~/rncr2_j`a30~/rncr1_j`a31~/rncr1_1`a32`(panelon)`(?)`{d
  own}`
G24: '(?)`{down}`(?)`{paneloff}`
G25: '(if a>b){beep 4}/ref4..f5`{goto}f4`(panelon)`{branch g26}`
G26: '(let h,b-a)`{let x,a}`{let fa,fx}`{let x,b}`{let fb,fb}`{let r1,1,0.5
  #h=(fa+fb)}`
G27: '(put r,1,1..5#h=(fa+fb)`{let i,1}`{let j,1}`{let @i1,"Integral ="}`
G28: '{let i,i+1}`
G29: '{let k,0}`
G30: '{let k,k+1}`
G31: '{let x,a+(k-.5)*h}`
G32: '{let suma,suma+fx}`
G33: '{if k<2^(i-2)}{branch g30}`
G34: '{put r,1,2..5*(r1_1+h*suma)}`
G35: '{let j,j+1}`{goto}r`{down 2}(right j-1)/rv`r2_j`{up}/rv`r1_j`
G36: '{put r,j,2,(4^(j-1)*r2_j-r1_j)/(4^(j-1)-1)}`
G37: '{if j<1}{branch g35}`
G38: '{goto}r`{down 2}(right i)/rv`r2_1`{left}/rv`r2_1a`
G39: '{if @abs(r2_1a-r2_1)<=0.000001}{home}{let f11,r2_1}`{quit}`
G40: '{let h,h/2}`{goto}r`{down 2}`
G41: '/ci3.bg3`i2`
G42: '{branch g28}`

```

METODO DE GAUSS GENERALIZADO

```

G21: '(paneloff)(?)~/mip3..fo20^a3^(?)~/rea3..g20^
G22: '(goto)e3~"f(x) =~(down)"x0 =~(down)"xf =~(down)"H =~(goto)f3^
G23: '/rncf(x)^f3~/rncx0^f4~/rncxf^f5~/rncm^f6~/rncxi^18^
G24: '/rncx^11~/rnci^12~/rncki^13~/rncui^14~/rncarea^15~/rncxp^16~/rncxs^17^
      (panelon)
G25: '(?)^(down)(?)^(down)(?)^(down)(?)^(paneloff){goto}e8^
G26: '{let xp,(x0+xf)/2}^{let xs,(xf-x0)/2}^
G27: '{let i,i+1}^
G28: '{down}{left 3}
G29: 'k^{(right)/rv}i^{(right)}=^(right){panelon}(?)^(paneloff)
G30: '/rv^ki^(down){left 3}'u^{(right)/rv}i^{(right)}
G31: '(right)=^(right){panelon}(?)^(paneloff)/rv^ui^
G32: '{let xi,xp+xs*ui}^
G33: '{let x,xi}^
G34: '{let area,area+k{f(x)}^
G35: '{if i<m}{branch g27}
G36: '{goto}e13^"Integral =~(let f13,area+(xf-x0))^

```

METODO DE EULER

```

G21: '(paneloff)/rncf(xy)`e3~/rncx0`e4~/rncy0`e5~/rncvc`e6~/rncn`e7~
G22: '/rncxf`e8~/rncx`h1~/rncy`h2~/rncyf`h3~/rnc`h4~/rncxi`h5~
G23: '/rncxii`h6~/rnci`h7~/rncyi`h8~/rncyii`h9~(panelon)
G24: '(?)^(down)(?)^(down)(?)^(down)(?)^(down)(?)^(paneloff)(down)(left)
G25: '(if vc="y")"yf ="^(right)(panelon)(?)^(paneloff)(branch g36)
G26: ""xf ="^(right)(panelon)(?)^(paneloff)/rea3...a15~
G27: '(goto)a8`xi^(right)`yi^(down)(left)/wth
G28: '(let h,(xf-x0)/n)^(let xi,x0)^(let yi,y0)~
G29: '(let x,xi)^(let y,yi)~
G30: '(let xii,xi+h)~
G31: '(let yii,yi+f(xy)*h)~
G32: '/rvxi^(right)/rvyi^(down)(left)
G33: '(let yi,yii)^(let xi,xiii)~
G34: '(if i<=(n-1))(let i,i+1)^(branch g20)
G35: '(quit)
G36: '/rea3...a15^(goto)a8`xi^(right)`yi^(down)(left)/wth
G37: '(let yf,xf)~
G38: '(let h,(yf-y0)/n)^(let xi,x0)^(let yi,y0)~
G39: '(let x,xi)^(let y,yi)~
G40: '(let yii,yi+h)~
G41: '(let xii,xi+h/f(xy))~
G42: '/rvxi^(right)/rvyi^(down)(left)
G43: '(let yi,yii)^(let xi,xiii)~
G44: '(if i<=n-1)(let i,i+1)^(branch g39)

```

METODO DE EULER MODIFICADO

```

G21: '(paneloff)/rncf(xy)~e3~/rncx0~e4~/rncy0~e5~/rncvc~e6~/rncn~e7~
G22: '/rncxf~e8~/rncx~h1~/rncy~h2~/rncyf~h3~/rnc~h4~/rncxi~h5~/rncfxi~h1;
/rncfxi~h12~
G23: '/rncxi~h6~/rnci~h7~/rncyi~h8~/rncyi~h9~/rncff~h10~(panelon)
G24: '(?)~(down)~(?)~(down)~(?)~(down)~(?)~(down)~(?)~(paneloff)~(down)~(left)
G25: '(if vc="y")~yf =~(right)~(panelon)~(?)~(paneloff)~(branch g36)
G26: "xf =~(right)~(panelon)~(?)~(paneloff)~rea3..a17~
G27: '(goto)a8~"xi"~(right)~"yi"~(down)~(left)~/wth
G28: '(let h,(xf-x0)/n)~(let xi,x0)~(let yi,y0)~
G29: '(let x,xi)~(let y,yi)~(let fx,f(xy))~
G30: '(let xii,xi+h)~(let x,xii)~(let y,yi+fxi*h)~(let fxii,f(xy))~(let ff,
(fxi+fxii)/2)~
G31: '(let yii,yi+ff*h)~
G32: '/rvxi~(right)~/rvyi~(down)~(left)
G33: '(let yi,yii)~(let xi,xii)~
G34: '(if i<n-1)~(let i,i+1)~(branch g29)
G35: '(quit)
G36: '/rea3..a17~(goto)a8~"xi"~(right)~"yi"~(down)~(left)~/wth
G37: '(let yf,yf)~
G38: '(let h,(yf-y0)/n)~(let xi,x0)~(let yi,y0)~
G39: '(let x,xi)~(let y,yi)~(let fx,f(xy))~
G40: '(let yii,yi+h)~(let x,xi)~(let y,yii+fxi*h)~(let fxii,f(xy))~(let ff,
(fxi+fxii)/2)~
G41: '(let xii,xi+ff*h)~
G42: '/rvxi~(right)~/rvyi~(down)~(left)
G43: '(let yi,yii)~(let xi,xii)~
G44: '(if i<n-1)~(let i,i+1)~(branch g39)

```


METODO DEL PUNTO MEDIO

```

F21: '(paneloff)/rncf(xy)^e3~/rncx0^e4~/rncy0^e5~/rncxf^e6~/rncn^e7~
F22: '/rncn^g1~/rnci^g2~/rncxi^g3~/rncyi^g4~/rncck(1)^g5~/rncck(2)^g6~/rncfxi
^g7~
F23: '/rncxii^g8~/rncyii^g9~/rncx^g10~/rncy^g11~(panelon)
F24: '(?)^(down)(?)^(down)(?)^(down)(?)^(down)(?)^(paneloff)/rea3..a14^(got
o)a7~
F25: '~xi^(right)^yi^(down)(left)/wth/rvx0^(right)/rvy0^(down)(left)
F26: '(let h,(xf-x0)/n)~
F27: '(let xi,x0)^(let yi,y0)~
F28: '(let i,i+1)~
F29: '(let x,xi)^(let y,yi)^(let k(1),f(xy))~
F30: '(let x,xi+h/2)^(let y,yi+h*k(1)/2)^(let k(2),f(xy))~
F31: '(let y1i,yi+k(2)*h)^(let x1i,xi+h)~
F32: '/rvx1i^(right)/rvy1i^(down)(left)(let xi,x1i)^(let yi,y1i)~
F33: '(if i<n){branch f28}

```

METODO DE HEUN

```
F21: '(paneloff)/rncf(xy)`e3~/rncx0`e4~/rncy0`e5~/rncxf`e6~/rncn`e7`
F22: '/rnrch`g1~/rnci`g2~/rncxi`g3~/rncyi`g4~/rnc(1)`g5~/rnc(2)`g6~/rncfxi
`g7`
F23: '/rncxii`g8~/rncyii`g9~/rncx`g10~/rncy`g11` (panelon)
F24: `?(?)` (down)(?)` (down)(?)` (down)(?)` (down)(?)` (paneloff)/rea3...a14` (got
o)a7`
F25: `x1` (right) y1` (down) (left) /vth/rvx0` (right) /rvy0` (down) (left)
F26: `(let h,(xf-x0)/n)`
F27: `(let xi,x0)` (let yi,y0)`
F28: `(let i,i+1)`
F29: `(let x,xi)` (let y,yi)` (let k(1),f(xy))`
F30: `(let x,xi+2*h/3)` (let y,yi+2*h*k(1)/3)` (let k(2),f(xy))`
F31: `(let yii,yi+(k(1)/4+3*k(2)/4)*h)` (let xii,xi+h)`
F32: '/rvxii` (right) /rvyii` (down) (left) (let xi,xii)` (let yi,yii)`
F33: `(if i<n)(branch f28)
```

METODO DE RUNGE - KUTTA DE 3ER. ORDEN 5 CONSTANTES

```

F21: '(paneloff)/rncf(xy)~e3~/rncx0~e4~/rncy0~e5~/rncxf~e6~/rncn~e7~
F22: '/rncg1~/rnci~g2~/rncxi~g3~/rncyi~g4~/rnck1i~g5~/rnck2i~g6~/rncfxi
~g7~
F23: '/rncxii~g8~/rncyii~g9~/rncx~g10~/rncy~g11~/rncck(3)~g12~(panelon)
F24: '(?)~(down)(?)~(down)(?)~(down)(?)~(down)(?)~(paneloff)/rea3..a16~(got
o)~a7~
F25: '~xi~(right)~yi~(down)(left)/wth/rvx0~(right)/rvy0~(down)(left)
F26: '(let h,(xf-x0)/n)~
F27: '(let xi,x0)~(let yi,y0)~
F28: '(let i,i+1)~
F29: '(let x,xi)~(let y,yi)~(let k(1),f(xy))~
F30: '(let x,xi+h/2)~(let y,yi+h*k(1)/2)~(let k(2),f(xy))~
F31: '(let x,xi+3*h/4)~(let y,yi+3*h*k(2)/4)~(let k(3),f(xy))~
F32: '(let y1,yi+(2*k(1)/9+3*k(2)/9+4*k(3)/9)*h)~(let x1,xi+h)~
F33: '/rvxii~(right)/rvyii~(down)(left)(let xi,x1)~(let yi,y1)~
F34: '(if i<n)(branch f28)

```

METODO DE RUNGE - KUTTA DE 3ER ORDEN 6 CONSTANTES

```

F21: '(paneloff)/rncf(xy)'e3'/rncx0 e4'/rncv0'e5'/rncxt e6'/rncn e7
F22: '/rnc'h`g1'/rnc1`g2'/rncx1`g3'/rncv1`g4'/rncck(1)`g5'/rncck(2)`g6'/rncfx1
`g7'
F23: '/rncx11`g8'/rncv11`g9'/rncx`g10'/rncv`g11'/rncck(3)`g12'(panelon)
F24: '(?)^(down)(?)^(down)(?)^(down)(?)^(down)(?)^(paneloff)/rea3..a13`igt
o1a7'
F25: '`xi^(right)`yi^(down)(left)/rvc0^(right)/rvy0^(down)(left)
F26: '(let h,(xf-x0)/n)'
F27: '(let xi,x0)(let yi,y0)'
F28: '(let i,1)'
F29: '(let x,xi)(let y,yi)(let k(1),f(xy))'
F30: '(let x,xi+h/2)(let y,yi+h*k(1)/2)(let k(2),f(xy))'
F31: '(let x,xi+h)(let y,yi+2*h*k(2)-h*k(1))(let k(3),f(xy))'
F32: '(let yi,yi+(k(1)+4*k(2)+k(3))*h/6)(let xi,xi+h)'
F33: '/rvx1^(right)/rvy1^(down)(left)(let xi,xi1)(let yi,yi1)'
F34: '(if i<n)(branch f28)

```

METODO DE RUNGE - KUTTA DE 4o. ORDEN 7 CONSTANTES

```

F21: '(paneloff)/rncf(xy)'e3~/rncx0'e4~/rncy0'e5~/rncx1'e6~/rncn'e7~
F22: '/rncn'g1~/rnci'g2~/rncx1'g3~/rncy1'g4~/rnck(1)'g5~/rnck(2)'g6~/rncfxi
'g7~
F23: '/rncx11'g8~/rncy11'g9~/rncx'g10~/rncy'g11~/rnck(3)'g12~/rnck(4)'g13~
(panelon)
F24: '(?)'(down)(?)'(down)(?)'(down)(?)'(down)(?)'(paneloff)/rea3...a9'(goto
)a7~
F25: '~x1'(right)'y1'(down)((left)/wth/rvx0''(right)/rvy0''(down)((left)
F26: '~(let h,(xf-x0)/n)~
F27: '~(let xi,x0)~(let yi,y0)~
F28: '~(let i,i+1)~
F29: '~(let x,xi)~(let y,yi)~(let k(1),f(xy))~
F30: '~(let x,xi+h/2)~(let y,yi+h*k(1)/2)~(let k(2),f(xy))~
F31: '~(let x,xi+h/2)~(let y,yi+h*k(2)/2)~(let k(3),f(xy))~
F32: '~(let x,xi+h)~(let y,yi+h*k(3))~(let k(4),f(xy))~
F33: '~(let y1,yi+k(1)+2*k(2)+2*k(3)+k(4))*h/6)~(let x11,xi+h)~
F34: '/rvx11''(right)/rvy11''(down)((left))(let xi,x11)~(let yi,y11)~
F35: '~(if i<n)(branch f28)

```

METODO DE RUNGE - KUTTA DE 4o. ORDEN 10 CONSTANTES

```

F21: '(paneloff)/rncf(xy)'e3'/rncx0'e4'/rncy0'e5'/rncxf'e6'/rncn'e7'
F22: '/rncf'g1'/rncf'g2'/rncf'g3'/rncyf'g4'/rncf(1)'g5'/rncf(2)'g6'/rncf'x1'
'g7'
F23: '/rncf'g8'/rncyf'g9'/rncf'g10'/rncy'g11'/rncf(3)'g12'/rncf(4)'g13'
(panelon)
F24: '(?)'(down)(?)'(down)(?)'(down)(?)'(down)(?)'(paneloff)/rea3..a9'(goto
ja7'
F25: 'x'(right)'y'(down)(left)/wth/rvx0''(right)/rvy0''(down)(left)
F26: '(let h,(xf-x0)/n)'
F27: '(let x1,x0)'(let y1,y0)'
F28: '(let i,i+1)'
F29: '(let x,x1)'(let y,y1)'(let k(1),f(xy))'
F30: '(let x,x1+h/3)'(let y,y1+h*k(1)/3)'(let k(2),f(xy))'
F31: '(let x,x1+2*h/3)'(let y,y1+h*k(1)/3+h*k(2))'(let k(3),f(xy))'
F32: '(let x,x1+h)'(let y,y1+h*k(1)+h*k(2)+h*k(3))'(let k(4),f(xy))'
F33: '(let y1,y1+k(1)+3*k(2)+3*k(3)+k(4))*h/6)'(let x11,x1+h)'
F34: '/rvx11''(right)/rvy11''(down)(left)'(let x1,x11)'(let y1,y11)'
F35: '(if i<n){branch f28)

```

METODO DE ADAMS - BASHFORTH DE DOS PASOS

```

121: '(panel off) (?) ~ /rea3...e20~/wcr(goto)a3~/rncrx y b3~/rncx0~b4~/rncy0~b5~/
/rncxf~b6~/rncn~b7~
122: '/rnc h~j1~/rnc i~j2~/rnc x1~j3~/rnc x11~j4~/rnc y1~j5~/rnc y11~j6~/rnc f i u~k
11~/rnc g i~k12~
123: '/rnc x~k1~/rnc y~k2~/rnc k u~k3~/rnc k d~k4~/rnc k t~k5~/rnc k c~k6~/rnc x i u~k7~/
/rnc x i d~k8~/rnc y i u~k9~/rnc y i d~k10~
124: '"f(x,y) = "(down)"x0 = "(down)"y0 = "(down)"xf = "(down)"M = "(goto)b3"(pa
nel on)
125: '(?)~(down) (?)~(down) (?)~(down) (?)~(down) (?)~(panel off) (goto)d7~"x1"(r
ight)~"y1"(down)~(left)~/with
126: '(let h,(xf-x0)/n)~/rvx0~(right)~/rvy0~(down)~(left)~(let xi,x0)~(let y
i,y0)~
127: '(let x,x1)~(let y,y1)~(let ku,fx)~
128: '(let x11,x1+h/2)~(let y11,y1+h*ku/2)~(let x,x11)~(let y,y11)~(let kd,
fx)~
129: '(let y11,y1+h*kd/2)~(let y,y11)~(let kt,fx)~(let x11,x1+h)~/rvx11~
(right)
130: '(let y11,y1+kt*h)~(let x,x11)~(let y,y11)~(let kc,fx)~
131: '(let y11,y1+(ku+2*kd+2*kt+kc)*h/6)~/rvy11~(down)~(left)~(let xi,x11)~(
let yi,y11)~
132: '(if >=1)~(let i,i+1)~(branch 134)
133: '(let i,i+1)~(branch 127)
134: '(up 2)~/c~xiu~(right)~/c~yiu~(down)~/c~yid~(left)~/c~xid~(down)~(right)
135: '(let x,xiu)~(let y,yiu)~(let fiu,fx)~(let x,xid)~(let y,yid)~(let gi
,-fiu+3*fx)~
136: '(let y11,y1+gi*h)~/rvy11~(left)~(let x11,x1+h)~/rvx11~(down)~(let xi,
x11)~(let yi,y11)~
137: '(if >=n-1)~(quit)
138: '(let i,i+1)~(branch 134)

```

METODO DE ADAMS - BASHFORTH DE TRES PASOS

```

121: '(paneloff)(?)~/rea3..e20~/vcr(goto)a3~/rncfxy`b3~/rncx0`ba~/rncy0`b5~/
rncxf`b6~/rncn`b7`
122: '/rncx`j1~/rnci`j2~/rncx1`j3~/rncxii`ja~/rncyl`j5~/rncyii`j6~/rncfliu`k
13~/rncfid`k14~/rncgi`k15`
123: '/rncx`k1~/rncy`k2~/rncku`k3~/rnckd`k4~/rnckt`k5~/rncck`k6~/rncxiu`k7~/
rncxid`k8~/rncxlt`k9~/rncyiu`k10~/rncyld`k11~/rncyit`k12`
124: '"f(x,y) ="(down)"x0 ="(down)"y0 ="(down)"xf ="(down)"N ="(goto)b3"(pa
nelon)
125: '?(?)~(down)(?)~(down)(?)~(down)(?)~(down)(?)~(paneloff)(goto)d7`xi`r
ight`yi`~(down)(left)/wth
126: '(let h.(xf-x0)/n)~/rvx0`~(right)/rvy0`~(down)(left)(let xi,x0)~(let y
i,y0)`
127: '(let x,x1)~(let y,y1)~(let ku,fx)~
128: '(let xii,x1+h/2)~(let yii,y1+h*ku/2)~(let x,x11)~(let y,y11)~(let kd,
fx)~
129: '(let yii,y1+h*kd/2)~(let y,y11)~(let kt,fx)~(let xii,x1+h)~/rvxii`~(
right)
130: '(let yii,y1+kt*h)~(let x,x11)~(let y,y11)~(let kc,fx)~
131: '(let yii,y1+(ku*2*kd+2*kt*kc)/h/6)~/rvyii`~(down)(left)(let xi,x11)~(
let yi,y11)`
132: '(if i>=2)(let i,i+1)~(branch 134)
133: '(let i,i+1)~(branch 127)
134: '(up 3)/c`xiu`~(right)/c`yiu`~(down)/c`yid`~(left)/c`xid`~(down)/c`xit`~(ri
ght)/c`yit`~(down)
135: `~(let x,xiu)~(let y,yiu)~(let fiu,fx)~(let x,xid)~(let y,yid)~(let fi
d,fx)~
136: '(let x,xit)~(let y,yit)~(let gi,(23*fx-16*fid+5*fiu)/12)`
137: '(let yii,yi+gi*h)~/rvyii`~(left)(let xii,x1+h)~/rvxii`~(down)(let xi,
x11)~(let yi,y11)~
138: '(if i=n-1)(quit)
139: '(let i,i+1)(branch 134)

```


METODO DE ADAMS - BASHFORTH DE CUATRO PASOS

```

121: '(paneloff)(?)~/rea3..e20~/vcr(goto)a3~/rncfxy`b3~/rncx0`b4~/rncy0`b5~/
rncxf`b6~/rncn`b7`
122: '/rnc`j1~/rnci`j2~/rncxi`j3~/rncxii`j4~/rncyi`j5~/rncyii`j6~/rncfiu`k
15~/rncfid`k16~/rncfit`k17~/rncgi`k18`
123: '/rncx`k1~/rncy`k2~/rncku`k3~/rnckd`k4~/rnckt`k5~/rnckc`k6~/rncxiu`k7~/
rncxid`k8~/rncxlt`k9~/rncxic`k10~/rncyiu`k11~/rncyid`k12~/rncyit`k13~/
rncyic`k14`
124: ``f(x,y) = `(down)"x0 = `(down)"y0 = `(down)"xf = `(down)"N = `(goto)b3`(pa
nelon)
125: '(?)^(down)(?)^(down)(?)^(down)(?)^(down)(?)^(paneloff)(goto)d7`xi`lr
ight)`yi^(down)(left)/wth
126: '(let h,(xf-x0)/n)/rvx0^(right)/rvy0^(down)(left){let xi,x0}{let y
i,y0}`
127: '{let x,xi}{let y,yi}{let ku,fx}`
128: '{let x1,xi+h/2}{let y1,yi+h*ku/2}{let x,x1}{let y,y1}{let kd,
fx}`
129: '{let y1,yi+h*kd/2}{let y,y1}{let kt,fx}{let x1,xi+h}/rvxii^(ri
ght)
130: '{let y1,yi+kt*h}{let x,x1}{let y,y1}{let kc,fx}`
131: '{let y1,yi+(ku+2*kd+2*kt*kc)*h/6}/rvyii^(down)(left){let x1,x1}{
let y1,y1}`
132: '{if i>=3}{let i,i+1}{branch 13a}
133: '{let i,i+1}{branch 127}
134: '(up 4)/c`xic^(right)/c`yic^(down)/c`yit^(left)/c`xit^(down)/c`xid^(ri
ght)/c`yid^(down)/c`yiu^(left)/c`xiu^(down)(right)
135: '{let x,xiu}{let y,yiu}{let fiu,fx}{let x,kid}{let y,yid}{let fi
d,fx}{let x,xit}{let y,yit}{let fit,fx}`
136: '{let x,xic}{let y,yic}{let gi,(55*fiu-50*fid+37*fit-9*fy)/24}`
137: '{let y1,yi+gi*h}/rvyii^(left){let x1,xi+h}/rvxii^(down){let x1,
x1}{let y1,y1}`
138: '{if i>=n-1}{quit}
139: '{let i,i+1}{branch 13a}

```

METODO DE ADAMS - BASHFORTH DE CINCO PASOS

```

121: '(paneloff)(?)~/rea3..e20~/vcr(goto)a3~/rncfxy~b3~/rncx0~b4~/rncy0~b5~/rncxf~b6~/rncn~b7'
122: '/rncx~j1~/rncf~j2~/rncx1~j3~/rncx11~j4~/rncy1~j5~/rncy11~j6~/rncfliu~k17~/rncfid~k18~/rncf11~k19~/rncfic~k20~/rncg1~l1'
123: '/rncx~k1~/rncy~k2~/rncxk~k3~/rncxk~k4~/rncxk~k5~/rncxk~k6~/rncxk~k7~/rncxk~k8~/rncxk~k9~/rncxk~k10~/rncxk~k11~/rncyiu~k12~/rncyid~k13~/rncyit~k14~/rncyic~k15~/rncyici~k16'
124: '"f(x,y) = (down)*x0 + (down)*y0 + (down)*xf + (down)*N = (gotob3)tpanelon)'
125: '(?)~(down)(?)~(down)(?)~(down)(?)~(paneloff)(gotod7)~x1~(right)~y1~(down)(left)/wth'
126: '(let h,(xf-x0)/n)~/rvx0~(right)/rvy0~(down)(left)(let xi,x0)(let yi,y0)'
127: '(let x,xi)~(let y,y1)~(let ku,fx)'
128: '(let x11,x1+h/2)~(let y11,y1+h*ku/2)~(let x,x11)~(let y,y11)~(let kd,fx)'
129: '(let y11,y1+h*kd/2)~(let y,y11)~(let kt,fx)~(let x11,x1+h)~/rvx11~(right)'
130: '(let y11,y1+kteh)~(let x,x11)~(let y,y11)~(let kc,fx)'
131: '(let y11,y1+(ku+2*kd+2*kt+kc)*h/6)~/rvy11~(down)(left)(let xi,x11)~(let yi,y11)'
132: '(if i>=4)(let i,i+1)~(branch 134)'
133: '(let i,i+1)~(branch 127)'
134: '(up 5)/c~xic~(right)/c~yic~(left)/c~xic~(down)/c~xit~(right)/c~yit~(down)/c~yid~(left)/c~xid~(down)/c~xiu~(right)/c~yiu~(down)'
135: '(let x,x1u)~(let y,y1u)~(let fiu,fx)~(let x,xid)~(let y,yid)~(let fid,fx)~(let x,xit)~(let y,yit)~(let fit,fx)'
136: '(let x,xic)~(let y,yic)~(let fic,fx)~(let x,xici)~(let y,yici)~(let gi,(190*fiu-272*fid+2616*fit-1274*fic+251*fx)/720)'
137: '(let y11,y1+g1*h)~/rvy11~(left)(let x11,x1+h)~/rvx11~(down)(let xi,x11)~(let yi,y11)'
138: '(if i>=n-1)(quit)'
139: '(let i,i+1)(branch 134)'

```

METODO DE MILNE DE 4o. ORDEN

```

121: '(paneloff){?}/rea3..e20~/wcr(goto)a3~/rncfxy~b3~/rncx0~b4~/rncy0~b5~/rncx1~b6~/rncn~b7~
122: '/rncn~j1~/rnci~j2~/rncx1~j3~/rncx11~j4~/rncyl~j5~/rncy11~j6~/rncfliu~k13~/rncfid~k14~/rncy11~k16~/rncgl~k17~
123: '/rncx~k1~/rncy~k2~/rncu~k3~/rncd~k4~/rnc~k5~/rnc~k6~/rncxiu~k7~/rncxid~k8~/rncxit~k9~/rncyiu~k10~/rncyid~k11~/rncyit~k12~
124: '"f(x,y)="(down)"x0="(down)"y0="(down)"xf="(down)"N="(goto)b3"(panelon)
125: '({?})~(down){?}~(down){?}~(down){?}~(paneloff)(goto)d7~"xi"(right)"yi"(down)(left)/wth
126: '(let h,(xf-x0)/n)/rvx0~(right)/rvy0~(down)(let xi,x0)~(let yi,y0)~
127: '(let x,x1)~(let y,y1)~(let ku,fx)~(let x,x1+h/2)~(let y,y1+hku/2)~
128: '(let kd,fx)~(let y,y1+hkd/2)~(let kt,fx)~(let x,x1+h)~(let y,y1+hkt)~
129: '(let kc,fx)~(let y1,y1+(ku+2*kd+2*kt+kc)*h/6)~(let x1,x1+h)~
130: '/rvy11~(left)/rvx11~(down)(right)(let x1,x11)~(let y1,y11)~
131: '(if i>=2)(let i,i+1)~(left)(branch 133)
132: '(let i,i+1)~(branch 127)
133: '(let x11,x1+h)/rvx11~
134: '(up 3)/c~xiu~(right)(up)/c~y11~(down)/c~yiu~(down)/c~yid~(left)/c~xid~(down)/c~xit~
135: '(right)/c~yit~(let x,xiu)~(let y,yiu)~(let fiu,fx)~(let x,xid)~(let y,yid)~(let fid,fx)~
136: '(let x,xit)~(let y,yit)~(let gi,(2*fx-fid+2*fiu)/3)~(let y1,y11+gih)~
137: '(down)/rvy11~(up 2)/c~yiu~(down)/c~y11~(left)/c~xiu~(down)/c~xid~(right)/c~yid~(down)/c~yit~
138: '(left)/c~xit~(let x,xiu)~(let y,yiu)~(let fiu,fx)~(let x,xid)~(let y,yid)~(let fid,fx)~
139: '(let x,xit)~(let y,yit)~(let gi,(fx+afid+fiu)/3)~
140: '(let y1,y11+gih)~(right)/rvy11~(down)(left)(let x1,x11)~(let y1,y11)~
141: '(if i>=n-1)(quit)
142: '(let i,i+1)(branch 133)

```

METODO DE MILNE DE 6o. ORDEN

```

121: '(paneloff)(?)`/rea3..e20`/wcr(goto)a3`/rncfxy`b3`/rncx0`ba`/rncy0`b5`
/rncxf`b6`/rncn`b7`
122: '/rnc`j1`/rncf`j2`/rncx1`j3`/rncx11`j4`/rncyl1`j5`/rncyl11`j6`/rncflu`k
17`/rncfid`k18`/rncfit`k19`/rncfic`k20`/rncyl11`l1`/rncgl`l2`
123: '/rncx`k1`/rncy`k2`/rncxk`k3`/rncxk`k4`/rncxk`k5`/rncxk`k6`/rncxlu`k7`
/rncxid`k8`/rncxit`k9`/rncxic`k10`/rncxid1`k11`/rncylu`k12`/rncyid`k
13`/rncyid`k14`/rncylic`k15`/rncylic1`k16`
124: '`f(x,y) = (down)`x0 = (down)`y0 = (down)`xt = (down)`N = (goto)b3` (pa
nelon)
125: '(?)` (down)(?)` (down)(?)` (down)(?)` (down)(?)` (down)(?)` (paneloff)(goto)d7` xi` (r
ight)` y1` (down)(left)` wth
126: '(let h,(xf-x0)/n)`/rvx0` (right)/rvy0` (down)(let xi,x0)` (let yi,y0)`
127: '(let x,x1)` (let y,y1)` (let ku,fx)` (let x,x1+h/2)` (let y,y1+h*ku/2)`
128: '(let kd,fx)` (let y,y1+h*kd/2)` (let kt,fx)` (let x,x1+h)` (let y,y1+n
kt)`
129: '(let kc,fx)` (let y1,y1+(ku+2*kd+2*kt+kc)*h/6)` (let x1,x1+h)`
130: '/rvy11` (left)/rvx11` (down)(right)(let xi,x11)` (let y1,y11)`
131: '(if i>=4)(let i,i+1)` (left)(branch 133)
132: '(let i,i+1)` (branch 127)
133: '(let x1,x1+h)`/rvx11`
134: '(up 5)/c`xiu` (right)(up)/c`y11` (down)/c`yiu` (down)/c`yid` (left)/c`xid
` (down)/c`xit`
135: '(right)/c`yit` (down)/c`ylic` (left)/c`xic` (down)/c`xic1` (right)/c`yic1`
136: '(let x,x1u)` (let y,y1u)` (let fiu,fx)` (let x,xid)` (let y,yid)` (let fi
d,fx)`
137: '(let x,xit)` (let y,yit)` (let fit,fx)` (let x,xic)` (let y,yic)` (let fi
c,fx)`
138: '(let x,xic1)` (let y,yic1)` (let g1,(11*fx-14*fc+26*fit-14*fid+11*fiu
)+3/10)` (let y1,y11+g1*h)`
139: '(down)/rvy11` (up 4)/c`yiu`/c`y11` (left)/c`xiu` (down)/c`xid` (right)/c
`yid` (down)/c`yit`
140: '(left)/c`xit` (down)/c`xic` (right)/c`ylic` (down)/c`yic1` (left)/c`xic1`
141: '(let x,x1u)` (let y,y1u)` (let fiu,fx)` (let x,xid)` (let y,yid)` (let fi
d,fx)`
142: '(let x,xit)` (let y,yit)` (let fit,fx)` (let x,xic)` (let y,yic)` (let fi
c,fx)`
143: '(let x,xic1)` (let y,yic1)` (let g1,(7*fx+32*fc+12*fit+32*fid-7*fiu)+
2/45)`
144: '(let y1,y11+g1*h)` (right)/rvy11` (down)(left)(let xi,x11)` (let y1,y1
11)`
145: '(if i>=n-1)(quit)
146: '(let i,i+1)(branch 133)

```

METODO DE ADAMS - MOULTON

```

121: '(paneloff)(?)~rea3..e20~/wcr(goto)a3~/rncfxy~b3~/rncx0~ba~/rncy0~b5~/
  /rncxf~b6~/rncn~b7~
R21: '(paneloff)/fradomoult.wk1~
122: '/rnc~j1~/rnci~j2~/rncxi~j3~/rncxil~j4~/rncyl~j5~/rncyli~j6~/rncfliu~k
  17~/rncfid~k18~/rncfit~k19~/rncyl1~l1~/rncgl~l2~
123: '/rncx~k1~/rncy~k2~/rncku~k3~/rnckd~ka~/rnckt~k5~/rncck~k6~/rncxlu~k7~/
  /rncxid~k8~/rncxil~k9~/rncxic~k10~/rncylu~k12~/rncyid~k13~/rncylt~k14~/
  /rncylic~k15~
124: '"f(x,y) = ~(down)"x0 = ~(down)"y0 = ~(down)"xf = ~(down)"N = ~(goto)b3"(pa
  nelon)
125: '(?)~(down)(?)~(down)(?)~(down)(?)~(down)(?)~(paneloff)(goto)d7~"xl"(r
  ight)yl"(down)(left)/wth
126: '(let h,(xf-x0)/n)~/rvx0~"(right)/rvy0~"(down)(let xl,x0)~(let yl,y0)~
  127: '(let x,xl)~(let y,yl)~(let ku,fx)~(let x,xl+h/2)~(let y,yl+h*ku/2)~
  128: '(let kd,fx)~(let y,yl+h*kd/2)~(let kt,fx)~(let x,xl+h)~(let y,yl+h*
  kt)~
129: '(let kc,fx)~(let yl,yl+(ku+2*kd+2*kt+kc)*h/6)~(let xil,xl+h)~
130: '/rvyl1~"(left)/rvxl1~"(down)(right)(let xl,xil)~(let yl,yl1)~
131: '(if l>=3)(let l,l+1)~(left)(branch 133)
132: '(let l,l+1)~(branch 127)
133: '(let xil,xl+h)~/rvxl1~
134: '(up 4)/c~xiu"(right)/c~yiu"(down)/c~yid"(left)/c~xid"(down)/c~xit"
  135: '(right)/c~yit"(down)/c~yic"/c~yl1"(left)/c~xic"(down)
136: '(let x,xiu)~(let y,yiu)~(let fl,fx)~(let x,xid)~(let y,yid)~(let fl
  d,fx)~
137: '(let x,xit)~(let y,yit)~(let flt,fx)~(let x,xic)~(let y,yic)~
138: '(let gl,(55*xy-59*fit+37*fid-9*flu)/24)~(let yil,yil+gl*h)~
139: '(right)/rvyl1~"/c~yic"(up 3)/c~yiu"(left)/c~xiu"(down)/c~xid"(right)/
  c~yid"(down)/c~yit"
140: '/c~yil"(left)/c~xit"(down)/c~xic"
141: '(let x,xiu)~(let y,yiu)~(let fl,fx)~(let x,xid)~(let y,yid)~(let fl
  d,fx)~
142: '(let x,xit)~(let y,yit)~(let flt,fx)~(let x,xic)~(let y,yic)~
143: '(let gl,(9*xy+18*fit-5*fid+flu)/24)~
144: '(let yil,yil+gl*h)~(right)/rvyl1~"(down)(left)(let xil,xil)~(let yl,yi
  l)~
145: '(if l>=n-1)(quit)
146: '(let l,l+1)(branch 133)

```



Capítulo: VIII

Aplicaciones

GENERALIDADES.

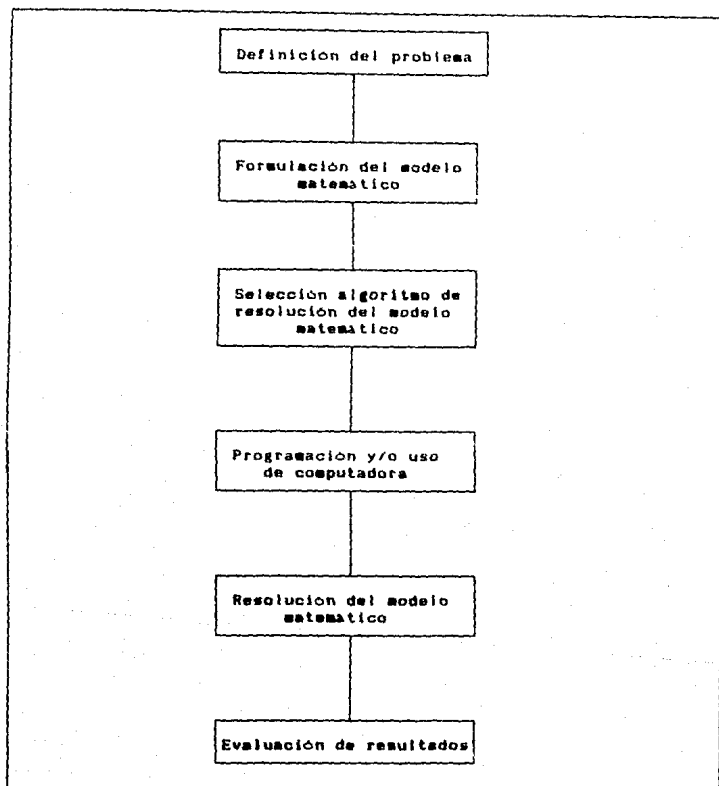
La finalidad de este capítulo es la de aplicar los procedimientos numéricos analizados en el capítulo II para resolver problemas de Ingeniería. Los métodos numéricos son importantes en la práctica ya que frecuentemente se encuentran problemas que no se pueden plantear desde un punto de vista analítico, y de esta manera mostrar como una computadora y las técnicas numéricas, especifican alternativas de solución de problemas, las cuales ahorran tiempo y esfuerzo al Ingeniero químico.

Dentro del contexto general de análisis, desarrollo y diseño de procesos, en los cuales la materia se transforma química y/o físicamente, se puede presentar el diagrama de bloques de la figura 1.

Para la definición del problema deberán tenerse conocimientos sobre las diferentes operaciones y procesos unitarios, para poder identificar que fenómenos ocurren dentro del proceso y de esta manera regirse por las leyes de la naturaleza características de tales fenómenos. Leyes formuladas en diferentes ramas de la física y la química, tales como: la termodinámica, fenómenos de transporte, etc.

Para la formulación del modelo es necesario tener conocimiento acerca del análisis dimensional y de balances de materia y energía.

En análisis dimensional se aplican los teoremas de Buckingham y Raleigh, los cuales establecen que un fenómeno físico puede representarse por las variables físicas relevantes aisladas y combinadas de tal manera que la ecuación del modelo sea dimensionalmente homogénea.



Los balances totales y parciales apoyándose en las leyes de conservación de la masa y la energía cuantifican la cantidad de materia y energía que se esté procesando. Dichos balances están representados por la siguiente expresión:

$$\text{Entrada} + \text{Generación} = \text{Salida} + \text{Acumulación}$$

El término Entrada corresponde a la cantidad de materia o energía que entra al sistema de análisis.

El término Salida corresponde a la cantidad de materia o energía que sale del sistema de análisis.

El término Acumulación corresponde a la rapidez con que cambia la masa o la energía dentro del sistema de análisis (si los balances no se hacen por unidad de tiempo, la acumulación será la diferencia de masa o energía con respecto a dos estados diferentes o a dos tiempos diferentes).

La Generación corresponde a la masa o energía que se libere o se consuma dentro del sistema de análisis (aplicable a los casos de sistemas con reacción química o casos de mezclado exotérmico o endotérmico).

Sistema : Parte del universo que se encuentra bajo observación y/o experimentación.

Como ejemplos de aplicación se resolverán problemas correspondientes a los siguientes temas :

- 1) Propiedades termodinámicas y de transporte.
- 2) Balances de materia y energía.
- 3) Transferencia de calor.
- 4) Flujo de fluidos.

En los casos que así se requiera se formularán los modelos

correspondientes mediante balances de materia y energía, dependiendo del caso.

Un mismo problema será resuelto con las diversas técnicas aplicables para el caso, con el objeto de compararlas y hacer recomendaciones en los casos en que sea necesario.

PROBLEMAS.

1) La viscosidad de un gas es aproximadamente proporcional a la temperatura absoluta evaluada a cierta potencia.

Determine el exponente de T y la constante de proporcionalidad para los siguientes datos del bióxido de carbono a 1 atm. de presión:

$$\mu = kT^N \quad \dots\dots\dots \text{Ecuación 1}$$

usando logaritmos, tendremos:

$$\text{Log } \mu = \text{Log } k + N \text{ Log } T$$

El modelo tiene la forma: $Y = A_0 + A_1 X_1$

Donde $Y = \text{Log } \mu$; $A_1 = N$; $X = \text{Log } T$

Si utilizamos la técnica de mínimos cuadrados para modelos lineales, truncando a partir del término $A_2 T_2$ y alimentamos los datos se obtiene:

T K	μ cp	Log T	Log μ
288	0.01457	2.459392	-1.83654
293	0.0148	2.468867	-1.82937
303	0.0153	2.481442	-1.81530
313	0.157	2.49554	-1.80410
372	0.01861	2.570542	-1.73025
455	0.02221	2.658011	-1.65345
575	0.0268	2.758667	-1.57186
763	0.033	2.882524	-1.48148
958	0.038	2.981365	-1.42021
1123	0.0436	3.050379	-1.38051
1325	0.0479	3.122215	-1.31866

Oblenemos :

$$A_0 = -3.76681$$

$$A_1 = 0.788953$$

de donde:

$$\text{Antilog } A_0 = k = 0.000171$$

$$N = 0.788953$$

$$\mu = 0.000171 \times 0.788953$$

2). Calcule el volumen molar del metano a una presión de 200 bars y a una temperatura de 400 C.

utilizando :

- a) La ecuación del gas ideal
- b) La ecuación de Van der Waals
- c) La ecuación de Redlich-Kwong
- d) La ecuación de Beattie-Bridgeman
- e) La ecuación de Benedict-Webb-Rubin.

a) Ecuación de gas ideal

$$PV = nRT$$

$$\text{Datos: } T = 1211.7 \text{ R, } P = 197.38 \text{ atm.}$$

$$\bar{V} = RT/P$$

$$R = 0.7302 \frac{\text{atm pie}^3}{\text{lb mol R}}$$

$$\text{Solución: } \bar{V} = 4.4826 \frac{\text{pie}^3}{\text{lb mol R}}$$

b) Ecuación de Van der Waals

$$\left(P - \frac{a}{\bar{V}^2}\right)(\bar{V} - b) - RT = 0$$

$$\text{Datos: } T = 673 \text{ K, } P = 197.38 \text{ atm.}$$

$$R = 1.314 \frac{\text{atm pie}^3}{\text{lb mol K}}$$

Esta ecuación es resuelta con diferentes técnicas y los resultados se presentan en la tabla 1.1

La ecuación de Van der Waals se puede modificar de la siguiente

manera:

$$\bar{V}^3 - \frac{Pb}{P} - \frac{RT}{P} \bar{V}^2 + \frac{a}{P} \bar{V} - \frac{ab}{P} = 0$$

la cual puede ser resuelta con el algoritmo de la ecuación cúbica, de donde se obtienen las siguientes raíces:

$$X_1 = 4.62372526$$

$$X_2 = 0.270268 + 0.60071i$$

$$X_3 = 0.270268 - 0.60071i$$

c) Ecuación de Redlich-Kwong.

$$\frac{RT}{\bar{V} - b} - \frac{a}{T^{0.5} \bar{V}(\bar{V} + b)} - P = 0$$

Datos: $T_c = 180.5 \text{ K}$, $P_c = 45.8 \text{ atm.}$, $a = 8071.95$, $b = 0.473625$

$$R = \frac{1.314 \text{ atm pie}^3}{\text{lb mol K}}$$

Esta ecuación puede ser resuelta con diferentes técnicas numéricas y los resultados se presentan en la tabla 1.2

d) Ecuación de Beattie-Bridgeman.

$$RT \left(\bar{V} + B_0 \left(1 - \frac{b}{\bar{V}} \right) \right) \left(1 - \frac{c}{\bar{V}T} \right) - A_0 \left(1 - \frac{a}{\bar{V}} \right) - P\bar{V}^2 = 0$$

Datos: $T = 1211.7 \text{ K}$, R , $P = 197.38 \text{ atm.}$

$$R = \frac{0.082 \text{ atm Lt}}{\text{lb mol K}}$$

$$A_0 = 2.2769$$

$$B_0 = 0.5587$$

$$a = 0.01855$$

$$b = -0.01587$$

$$c = 12300$$

Esta ecuación puede ser resuelta con diferentes técnicas numericas y los resultados se presentan en la tabla 1.3

e) Ecuación de Benedict-Webb-Rubin.

$$\frac{RT}{V} + \frac{1}{V^2} (RTB_0 - A_0 - \frac{C}{T^2}) + \frac{1}{V^3} (RTB - a) + \frac{ac}{V^2} + \frac{c}{T^2 V^3} (1 + \frac{\gamma}{V^2}) \cdot \exp(-\frac{\gamma}{V^2}) - P = 0$$

Datos: T = 673 K, P = 197.38 atm.

$$R = \frac{0.082 \text{ atm Lt}}{\text{lb mol K}}$$

$$A_0 = 1.855$$

$$\alpha = 0.000124359$$

$$B_0 = 0.0426$$

$$\gamma = 0.006$$

$$C_0 = 22570$$

$$a = 0.01855$$

$$b = -0.01587$$

$$c = 12300$$

Esta ecuación puede ser resuelta con diferentes técnicas numericas y los resultados se presentan en la tabla 1.4

CALCULO DEL VOLUMEN MOLAR DEL METANO RESOLVIENDO LA ECUACION DE VAN DER WAALS, CON DIFERENTES TECNICAS NUMERICAS.

TECNICA	RAIZ	VALORES INICIALES		
SECANTE	4.623725	1	10	
REGULA FALSI	4.623725	1	10	
WENSTEIN	4.623725	1	10	
MULLER	4.623725	-1	5	10
SUSTITUCION DIRECTA	4.623715	1		
MEDIO INTERVALO	4.623725	1	10	
HOOK-KEEVES	4.623724	1	$\Delta X = 0.5$	
NEWTON-RAPHSON	4.623725	1		

TABLE 1.1

CALCULO DEL VOLUMEN MOLAR DEL METANO RESOLVIENDO LA ECUACION DE REDLBRICH-KWONG, CON DIFERENTES TECNICAS NUMERICAS.

TECNICA	RAIZ	VALORES INICIALES	
SECANTE	4.612823	1	3
REGULA FALSI	4.612823	1	3
WEINSTEIN	4.612823	1	3
REGLER	NO CONV.		
SUSTITUCION DIRECTA	4.612823	1	
MEDIO INTERVALO	4.612823	1	3
HOOK-HEEVES	4.612823	1	$\Delta X = 0.5$
NEWTON-RAPHSON	4.612823	1	

TABLA 1.2

CALCULO DEL VOLUMEN MOLAR DEL METANO RESOLVIENDO LA ECUACION DE BEATTIE-BRIDGEMAN, CON DIFERENTES TECNICAS NUMERICAS.

TECNICA	RAIZ	VALORES INICIALES	
SECANTE	0.296826	0.1	1
REGULA FALSI	0.296826	0.1	1
NEWSTEIN	0.296826	0.1	1
PULLER	NO CONV.		
SUSTITUCION DIRECTA	0.297973	0.1	
MEDIO INTERVALO	0.296828	0.1	1
HOOK-HEEVES	0.296827	0.1	$\Delta X = 0.5$
NEWTON-RAPHSON	0.296827	1	

Tabla 1.3

NOTA:
EL VOLUMEN MOLAR ESTA DADO EN L³/gmol.

CALCULO DEL VOLUMEN MOLAR DEL METANO RESOLVIENDO LA ECUACION DE BENEDICT-WEBB-REID, CON DIFERENTES TECNICAS NUMERICAS.

TECNICA	RAIZ	VALORES INICIALES	
SECANTE	0.267994	1	0.1
REGULA FALSI	0.267994	1	0.1
WEGSTEIN	0.267994	1	0.1
MULLER	NO CONV.		
SUSTITUCION DIRECTA	0.266913	0.1	
MEDIO INTERVALO	0.267994	1	0.1
HOOK-HELVES	0.267994	0.1	$\Delta X = 0.5$
NEWTON-RAPHSON	0.266889	0.1	

TAULA 1.4

NOTA:
EL VOLUMEN MOLAR ESTA DADO EN $L/gmol$.

3). Calcule las temperaturas de burbuja y de rocío para la siguiente mezcla: 20% en mol de Etano, 40% en mol de Propano, 40% en mol de Butano, a una presión de 760 mmHg.

a) Utilizando la ley de Raoult y la ecuación de Antoine.

b) Utilizando el factor k de equilibrio (constante de equilibrio, reportado en el diagrama de Depriester).

Solución.

Para un sistema de equilibrio liquido-vapor se cumple que:

$y_i = k_i x_i$, en donde y_i , x_i , k_i son la fracción mol en el vapor, fracción mol en el líquido y constante de equilibrio para el componente i respectivamente.

Punto de Burbuja:

$$\sum_{i=1}^N k_i x_i - 1 = 0$$

Punto de rocío:

$$\sum_{i=1}^N \frac{y_i}{k_i} - 1 = 0$$

N = número de componentes químicos.

a) Ley de Raoult.

$$k_i = \frac{P_i}{P_T} = \frac{10^{(A_i - \frac{B_i}{C_i + T})}}{P_T}$$

P_i = Presión de vapor del componente i

A_i, B_i, C_i = constantes de Antoine del componente i

T = temperatura de equilibrio para la presión total P_T

Punto de Burbuja

$$\sum_{i=1}^N \frac{10^{(A_i - \frac{B_i}{C_i + T})}}{P_T} \cdot x_i - 1 = 0$$

Punto de Rocío

$$\sum_{i=1}^N \frac{P_T \cdot y_i}{10^{(A_i - \frac{B_i}{C_i + T})}}$$

componente	Constantes de Antoine		
	A	B	C
Etano	6.8026	656.4	256
n-Propano	6.8297	813.2	248
n-Butano	6.8302	845.9	240

De la aplicación de las diferentes técnicas numéricas se obtiene la siguiente tabla:

b) Utilizando el factor k de equilibrio

Temperatura	Etano	n-Propano	n-Butano
-100	2.3	0.17	0.13
-80	3.4	0.36	0.03
-60	4.8	0.68	0.07
-40	6.8	1.15	0.16
-20	9.2	1.8	0.3
0	12.0	2.6	0.52
20	15.5	3.7	0.85
40	19.0	5.0	1.28
60	23.5	6.8	1.85

haciendo un ajuste por mínimos cuadrados, encontramos el modelo y los valores de las constantes para evaluar la k de equilibrio.

$$k_i = \frac{\text{Exp}(A_i - B_i / (C_i + T))}{P_r}$$

constantes.

componente	A	B	C
Etano	14.7108243	-2441.6943	436.398167
n-Propano	13.2861863	-1747.35481	307.524278
n-Butano	13.1398735	-1994.37739	278.659715

Punto de Burbuja

$$\sum_{i=1}^N \frac{10 \left(A_i - \frac{B_i}{C_i + T} \right)}{P_T} = x_i - 1 = 0$$

Punto de Rocío

$$\sum_{i=1}^N \frac{P_T = y_i}{10 \left(A_i - \frac{B_i}{C_i + T} \right)}$$

De la aplicación de las diferentes técnicas numéricas se obtiene la tabla 1.5.

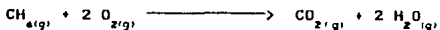
CALCULO DE LA TEMPERATURA DE BURBUJA Y LA TEMPERATURA DE
 ROCIO UTILIZANDO EL FACTOR k DE EQUILIBRIO.

TECNICA	T. ROCIO °C	VALORES INICIALES	T. BURBUJA	VALORES INICIALES
SECANTE	-19.3847	-50 -18	-57.6655	-180 -30
REGULA FALSI	-19.3847	-50 -18	-57.6655	-180 -30
WENSTEIN	-19.3847	-50 -18	-57.6655	-180 -30
PULLER	NO CONVERGE		-57.6647	-180 -50 -30
MEDIO INTERVALO	-19.3822	-50 18	-57.6628	-180 -30
HOOKI-JEEVES	-19.3855	-30 $\Delta X = 0.5$	-57.6648	-70 $\Delta X = 0.5$

TABLA 1.5

4). Calcule la máxima temperatura de los productos si se quema Metano con 50% de exceso de aire y si ambos reactivos se alimentan al quemador a 77 F.

reacción química (base 1 mol de CH_4)



para máxima temperatura el proceso se considera adiabático y la eficiencia de la reacción 100%, implicando que en el producto no habrá CH_4 y que todo el Carbono se convierta en CO_2 .

Si se alimenta Oxígeno en exceso entonces en el producto habrá Oxígeno además del Nitrógeno alimentado con el aire.

Producto.

CO_2	1 mol
H_2O	2 moles
O_2	1 mol
N_2	11.2857 moles (alimentadas con 3 moles de Oxígeno).

15.2857 moles totales del producto

Balance de energía.

considerando proceso a régimen estacionario:

$$H_{(\text{entrada})} + \text{Generación} = H_{(\text{salida})}$$

H = Entalpia.

$T = 298 \text{ K}$ (entrada)

$$H_{(\text{entrada})} = \text{moles mezcla reactiva} \cdot \int_{298}^{T_f} c_p(\text{reactivos}) \cdot dT = 0$$

$$H_{\text{salida}} = \text{moles producto} \cdot \int_{298}^{T_f} c_{p(\text{producto})} \cdot dT = 0$$

$$\text{Generación} = -\Delta H_r$$

se calcula el calor de la reacción a 298 K (Temperatura a la que se reporta la información de calores de formación y combustión).

$$\Delta H_r = \sum H_f (\text{productos}) - \sum H_f (\text{reactivos})$$

$$\Delta H_r = H_f (\text{CO}_2) + 2H_f (\text{H}_2\text{O}) - H_f (\text{CH}_4) - 0$$

$$\Delta H_r = -94051.8 + 2(-57798) - (-17889) - 0 = 191758.8$$

Componente	Fracción mol	A	B (10 ³)	C (10 ⁵)
CO ₂	0.06542	6.214	10.396	-3.545
H ₂ O	0.13084	7.256	2.298	0.283
O ₂	0.03642	6.148	3.102	-0.923
N ₂	0.7383	6.524	1.250	-0.001
Mezcla producto		6.574	2.106	0.683

$$H (\text{entrada}) = 0$$

$$\therefore \text{Generación} = H (\text{salida})$$

el modelo matemático que se obtendrá será:

$$1758.8 = 15.2857 \cdot \int_{298}^{T_f} (A + BT + CT^2) dT$$

$$= 15.2857 [A(T_f - 298) + B((T_f^2 - 298^2)/2) + C((T_f^3 - 298^3)/3)] - 191758.8 = 0$$

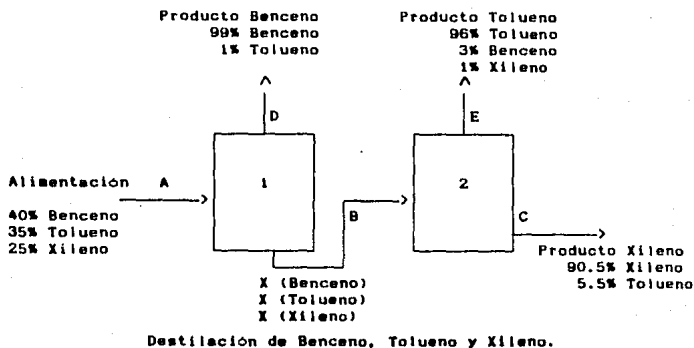
Esta ecuación es resuelta con diferentes técnicas y los resultados se presentan en la tabla 1.6.

TECNICA	DATE	VALORES INICIALES
SECANTE	1641.9968	350 1000
REGULA FALSI	1641.9968	350 1000
NEWSTEIN	1641.9968	350 1000
PULLER	1641.9968	350 700 1000
SUSTITUCION DIRECTA	NO CONVERGE	
PEDIO INTERVALO	1641.9968	350 1000
HOOKZ-HELVES	1641.9968	1700 $\Delta x = 0.5$
NEWTON-RAPHSON	0.266889	350

TABLA 1.6

5). En el proceso a regimen estacionario que se muestra en la figura de abajo todas las composiciones se dan en porcentaje en mol. Calcule:

- El porcentaje de recuperaci3n de cada componente en su respectiva corriente de producto.
- La composici3n de la corriente B



- % de recuperaci3n Benceno = $(0.99D)/(0.40A) = 100$
- % de recuperaci3n Tolueno = $(0.96E)/(0.35A) = 100$
- % de recuperaci3n Xileno = $(0.905)/(0.25A) = 100$

Las propiedades son intensivas y son independientes de la cantidad de A alimentada, esto es cierto desde el punto de vista de conservaci3n de la masa.

Por balances de materia se pueden calcular D, E y C.

b) los valores X_b , X_t , X_c (propiedades intensivas), también serán independientes del valor de A. Esto implica que los dos incisos podrán ser resueltos simultáneamente, al resolver las ecuaciones de balances de materia.

a) Balances de materia globales.

$$A = D + C + E$$

Balances parciales.

$$0.4A = 0.99D + 0.03E$$

$$0.35A = 0.01D + 0.96E + 0.095C$$

Base de cálculo A = 100

Se tiene entonces un sistema de tres ecuaciones con tres incógnitas

Matriz Aumentada.

$$\left| \begin{array}{cccc} 1 & 1 & 1 & 100 \\ 0.99 & 0.03 & 0 & 40 \\ 0.01 & 0.96 & 0.095 & 35 \end{array} \right|$$

De la aplicación de las diversas técnicas, se obtiene la tabla 1.7.

METODO	RAICES
INVERSION MULTIPLICACION	X(1) = 39.39341 X(2) = 33.35879 X(3) = 27.25579
JACOBI	NO CONVERGE
GAUSS-SEIDEL	NO CONVERGE

Tabla 1.7

COMENTARIOS:

X(1) = B

X(2) = B

X(3) = C

b) Balances parciales (Destilador 1)

$$A = B + D$$

$$0.4A = X_b B + 0.99D$$

$$0.35A = X_t B + 0.01D$$

D fue resuelto para A = 100 en el inciso anterior por lo tanto:

$$B = A - D$$

$$X_x - 1 = X_b - X_t$$

$$X_b = (0.4A - 0.99D)/B$$

Porcentaje de recuperación de cada componente.

$$\% \text{ Rec. Benceno} = (0.99 \cdot 39.3934)/(0.4 \cdot 100) \cdot 100 = 97.4986$$

$$\% \text{ Rec. Tolueno} = (0.96 \cdot 33.3507)/(0.35 \cdot 100) \cdot 100 = 91.4762$$

$$\% \text{ Rec. Xileno} = (0.905 \cdot 27.2557)/(0.25 \cdot 100) \cdot 100 = 98.6656$$

Composición de la corriente B.

$$B = A - D$$

$$B = 100 - 39.3934 = 60.6066$$

$$X_b = (0.4A - 0.99D)/B = (0.4 \cdot 100 - 0.99 \cdot 39.3934)/60.6066$$

$$X_b = 0.0165$$

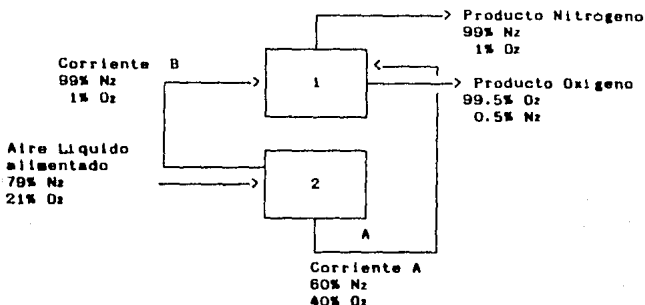
$$X_t = (0.35A - 0.01D)/B = (0.35 \cdot 100 - 0.01 \cdot 39.3934)/60.6066$$

$$X_t = 0.5709$$

$$X_x = 1 - X_b - X_t = 1 - 0.0165 - 0.5709$$

$$X_x = 0.4126$$

6). Se produce oxígeno puro mediante la destilación de aire líquido a bajas temperaturas. Para separar el Nitrógeno y el Oxígeno, se utilizan dos columnas de destilación tal y como se ilustra en la siguiente figura:



Si el proceso es a régimen estacionario

- Calcule el porcentaje de oxígeno alimentado en la corriente de aire, que es recuperado en la corriente de producto Oxígeno.
- Calcule el flujo de la corriente A para 100 lb mol/Hr de producto Oxígeno.

a) Porcentaje de recuperación de Oxígeno

$$\% \text{ Rec. Oxígeno} = (0.995C)/(0.21F) \times 100$$

El porcentaje del N₂ es una propiedad intensiva e independiente de la base de cálculo.

Para 100 lb mol/Hr de C cuánto vale F.

Para resolver en forma conjunta los dos incisos, y tomando como base C = 100. Se resolverán las ecuaciones de balances de materia correspondientes.

Balances de Materia.

$$\text{Entrada} = \text{Salida}$$

$$F = D + C$$

$$0.79F = 0.99D + 0.005C$$

$$F = B + A$$

$$0.79F = 0.6A + 0.99B$$

$$C = 100$$

Sistema de ecuaciones.

$$F - D = 100$$

$$0.7F - 0.99D = 0.5$$

$$F - B - A = 0$$

$$0.79F - 0.6A - 0.99B = 0$$

Matriz Aumentada.

1	-1	0	0	100
0.79	-0.99	0	0	0.5
1	0	-1	-1	0
0.79	0	-0.6	-0.99	0

De la aplicación de las diversas técnicas numéricas se obtiene la tabla 1.8.

Pociento de oxígeno recuperado

$$\% \text{ Rec. Oxígeno} = (0.995C)/(0.21F) \times 100$$

$$\% \text{ Rec. Oxígeno} = (0.995 \times 100)/(0.21 \times 429.49) \times 100$$

$$\% \text{ Rec. Oxígeno} = 98.2069$$

Flujo de la corriente para 100 lb mol/Hr de producto de oxígeno, de la tabla de resultados obtuvimos:

$$A = 252.5641 \text{ lb mol/Hr.}$$

METODO	VALORES INICIALES	RAICES
INVERSION MULTIPLICACION		X(1) = 492.5 X(2) = 392.5 X(3) = 252.5641 X(4) = 239.9358
JACOBI	X(1) = 128 X(2) = 28 X(3) = 68 X(4) = 68	X(1) = 492.499 X(2) = 392.499 X(3) = 252.564 X(4) = 239.935
GAUSS-SEIDEL	X(1) = 128 X(2) = 28 X(3) = 68 X(4) = 68	X(1) = 492.496 X(2) = 392.497 X(3) = 252.564 X(4) = 239.932

TABLE 1.8

COMENTARIOS

X(1) = P

X(2) = D

X(3) = A

X(4) = B

7). Calcule el calor requerido para incrementar la temperatura de 1g mol de Etileno desde 100 hasta 700 c a 1 atm. de presión, utilizando los siguientes datos:

Temperatura K	Cp cal/gmol K
298	10.41
500	15.16
700	18.76
1000	22.57

- a) Desarrollando previamente un modelo de Cp vs. Temperatura e integración analítica.
 b) Desarrollando previamente un modelo de Cp vs. Temperatura e integración numérica.

Calculo del calor requerido.

$$Q = M \int_{T_0}^{T_f} C_p dT$$

Q = Calor requerido

M = moles del compuesto

Cp = Capacidad calorifica del compuesto

To = Temperatura inicial

Tf = Temperatura final

La integración se podrá efectuar en forma numérica o analítica.

a) Desarrollo del modelo Cp vs. Temperatura. Utilizando la técnica del polinomio de interpolación obtenemos:

Ao = 0.800624317

$$A_1 = 0.0379998569$$

$$A_2 = -2.0883432E-5$$

$$A_3 = 4.64244173E-9$$

El modelo tiene la forma siguiente:

$$Y = A_0 + A_1X + A_2X^2 + A_3X^3$$

Integración analítica.

$$Q = M \int_{T_0}^{T_f} (A_0 + A_1T + A_2T^2 + A_3T^3) dT$$

$$Q = M [A_0(T_f - T_0) + A_1(T_f^2 - T_0^2)/2 + A_2(T_f^3 - T_0^3)/3 + A_3(T_f^4 - T_0^4)/4]$$

$$Q = 10791.3723 \text{ cal.}$$

b) El modelo del C_p será el mismo que el obtenido en el inciso (a).

Aplicando las distintas técnicas numéricas se obtiene:

Técnica	Q (cal)
Regla del Trapecio	10745.3309
Regla de Simpson	10779.7391
Regla de Simpson 3/8	10791.3723
Integración de Romberg	10791.3723
Integración Gaussiana	10791.3589

8). Por una tubería horizontal de acero comercial cedula 40 de 1" de diámetro nominal, fluye Argón, el cual entra a la tubería a una presión de 7 atm. Abs., a una temperatura de 100 C y a una velocidad de 30 m/s. Si el proceso es isotérmico, qué presión y velocidad tendrá el Argón a 70 m. dentro de la tubería, con respecto a la entrada.

Solución:

$$P_1 = 7 \text{ atm.}$$

$$T_1 = 100 \text{ C}$$

$$V_1 = 30 \text{ m/s}$$

$$D \text{ nominal} = 1''$$

$$D \text{ interior} = 1.049'' \text{ (Ap. C-6 Foust)}$$

Balance de energía por unidad de masa.

$$\frac{dP}{\rho} + \frac{V}{2gc} dV = -dH_f$$

considerando comportamiento de gas ideal

$$\rho = \frac{P}{RT} = PM$$

$$\frac{RT}{PM} \frac{dP}{P} + \frac{V}{2gc} dV = -\frac{1}{2} \frac{f}{gc} V^2 \frac{dL}{Di}$$

Balance de materia.

$$Q_1 \rho_1 = Q_2 \rho_2 = Q \rho$$

para área de flujo constante:

$$AV\rho = AV_1\rho_1$$

$$V \frac{P^+}{RT} = V_1 \frac{P_1^+}{RT} \quad \therefore \quad V = \frac{V_1 P_1^+}{P}$$

De donde:

$$dV = -V_1 P_1^+ \frac{dP^+}{P^2}$$

Finalmente

$$\frac{dP}{dL} = \frac{\frac{1}{2D_1} \frac{f}{gc} \left(\frac{V_1 P_1^+}{P} \right)^2}{\frac{RT}{P M} - \frac{1}{2gc} \frac{V_1^2 P_1^2}{P}}$$

Esta ecuación deberá resolverse con la siguiente condición inicial para $L = 0$, $P = 14817.6$

Datos transformados al sistema de unidades FPS.

$$V_1 = 98.4252 \text{ ft/seg.}$$

$$P_1 = 14817.6 \text{ Lbf/ft}^2$$

$$T = 672 \text{ R}$$

$$R = 1545.2 \text{ Lbf ft/Lb mol R}$$

$$gc = 32.2 \text{ ft/seg}^2$$

$$D_1 = 0.0874167 \text{ ft}$$

$$PM = 40 \text{ Lb/Lb mol}$$

$$L = 229.6588 \text{ ft}$$

$$NRe = (Di V \rho) / \mu = (Di G) / \mu$$

Para área constante y régimen estacionario, G = constante

$$G = V_1 \rho_1 = 98.4252 * (14817.6 * 40) / (1545.2 * 672)$$

$$G = 56.181094 \text{ Lb/ft}^2 \text{ seg.}$$

$$\mu = 1.74712E-5 \text{ Lb/ft seg} \quad (\text{se supondrá constante en toda la trayectoria del flujo})$$

$$NRe = 281100.659$$

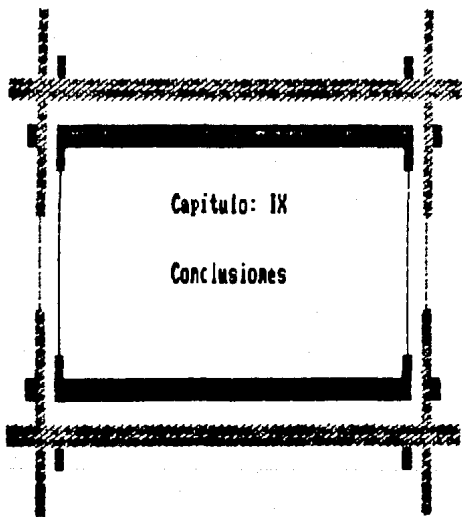
$$\text{Rugosidad relativa} = \epsilon / D = 0.00175 \text{ (Ap. C Foust)}$$

$$f = 0.023 \text{ (Ap. C Foust)}$$

De la aplicación de las diferentes técnicas numéricas, se obtiene la tabla 1.9

METODO	SOLUCION (P = 1676)	% ERROR CON RESPECTO A LA SOL. ANALITICA
EULER	8113.6835	-1.825
EULER MODIFICADO	8815.8286	-6.6215E-5
PUNTO MEDIO	5816.2282	-8.8122
HEUN	8815.8267	-8.222E-3
RUNGE-KUTTA 3er. ORDEN 5 CONSTANTES	8815.8348	-8.42E-5
RUNGE-KUTTA 3er. ORDEN 6 CONSTANTES	8815.8163	5.7972E-5
RUNGE-KUTTA 4o. ORDEN 7 CONSTANTES	8815.8228	9.236E-7
RUNGE-KUTTA 4o. ORDEN 10 CONSTANTES	8815.8221	4.874E-7
METODOS DE MULTIPASOS		
DOS PASOS	8828.5915	-8.8569
TRES PASOS	8815.6294	-6.196E-3
CUATRO PASOS	8815.1286	-1.886E-3
CINCO PASOS	8815.8432	-2.153E-4
METODOS PREDICTORES CORRECTORES		
RUNGE 4o. ORDEN	8815.8284	1.886E-5
RUNGE 6o. ORDEN	8815.8221	5.9E-7
ADAMS-ROULTON	8815.8143	7.98E-5

TABLA 1.9 CONDICIONES INICIALES: $X_{INICIAL} = 0$, $X_{FINAL} = 229.65588$, $V_{INICIAL} = 14817.6$, No. INTERVALOS = 30



Capitulo: IX

Conclusiones

CONCLUSIONES

De toda la información dentro del área del análisis numérico, se seleccionaron los métodos de uso más frecuente en la resolución de problemas de Ingeniería Química.

En algunos casos, la solución de problemas exigirá la combinación de dos o más métodos, sin embargo, el material presentado será suficiente para obtener un resultado satisfactorio.

Es importante señalar que los diagramas de flujo que acompañan a cada método numérico, le dan versatilidad para el uso de este material, haciéndolo independiente del tipo de lenguaje o paquete que se desee utilizar, pero es aun más importante la codificación de los métodos, pues con esto se ahorra una gran cantidad de tiempo en hacerlo. Además que los capítulos dedicados a la descripción del paquete de programación Lotus 1-2-3, ofrecen al lector una guía para poder entender el "lenguaje" de programación de Lotus 1-2-3, y de esta manera, no solo poder programar los métodos numéricos sino también poder acondicionarlos para necesidades particulares y de trabajo.

Análisis de las diferentes técnicas numéricas presentadas

En la actualidad está más generalizado el uso de matemáticos que el uso de las tablas, nomogramas o gráficas; esto está desplazando el uso de las técnicas de interpolación en los cálculos. Aun cuando estas tiendan a desaparecer, se tomó la decisión

de incluirlas en esta tesis, fundamentalmente por el soporte que dan al desarrollo de las diversas técnicas numéricas. Por lo anterior, la interpolación no se incluye en el siguiente resumen, obtenido de la aplicación de las técnicas numéricas en la resolución de problemas.

Resumen

Resolución de una ecuación algebraica no lineal.

Técnicas posibles a utilizar:

- | | |
|---------------|-----------------------|
| -Secante | -Substitucion directa |
| -Regula Falsi | -Medio intervalo |
| -Wengstein | -Hooke - Jeeves |
| -Muller | -Newton - Raphson |

Recomendación:

Newton - Raphson

Integración:

Técnicas posibles a utilizar:

- | | |
|---------------------|-----------------------|
| -Regla del trapecio | -Regla de Simpson |
| -Romberg | -Regla de Simpson 3/8 |
| -Gauss | |

Recomendación:

Romberg

Regla de Simpson 3/8

Resolución de una ecuación diferencial ordinaria

Técnicas posibles a utilizar:

-Euler

-Euler modificado

-Punto medio

-Heun

-Runge - Kutta 3er. orden 5 constantes

-Runge - Kutta 3er. orden 6 constantes

-Runge - Kutta 4o. orden 7 constantes

-Runge - Kutta 4o. orden 10 constantes

-Métodos de multipasos:

1) Dos pasos

2) Tres pasos

3) Cuatro pasos

4) Cinco pasos

-Predictores correctores:

1) Milne de 4o. orden

2) Milne de 6o. orden

3) Adams - Moulton

Recomendación:

Todos los métodos acumulan un error de truncación o redondeo de un paso anterior, razón por la cual las resoluciones de las ecuaciones diferenciales tienen como criterio prioritario la exactitud; en caso de dos o más técnicas con un mismo resultado, se recomienda la de menor tiempo de proceso.

Sistemas de ecuaciones algebraicas lineales

Técnicas posibles a utilizar:

-Jacobi

-Gauss - Seidel

-Inversión - Multiplicación

Recomendación:

Si la matriz aumentada es tridiagonal se recomienda el método de Gauss - Seidel.

BIBLIOGRAFIA

- o Mc. Cabe, Smith
Unit operations of chemical engineering
Kogakusha, 1976

- o Foust, Wenzel, Clup, Maus, Andersen
Principios de operaciones unitarias
Compañia editorial continental, S. A. 1974

- o Treybal
Mass transfer operations
Kogakusha, 1968

- o Perry, Chilton
Chemical engineer's handbook (fifth edition)
Kogakusha, 1973

- o Forman
Numerical Methods that work
Harper international, 1970

- o Jacques
A first course in computing and numerical methods
Addison - Wesley, 1970

o Conte, de Boor

Analisis numerico

Mc. Graw Hill, 1974

o Collatz

The numerical treatment of differential equations

Springer - Verlag New York inc., 1966

o Carnahan, Luther, Wilkes

Applied numerical methods

John Wiley & sons, inc., 1969

o Spiegel

Manual de formulas y tablas matematicas

Prindle Weber & Smith, 1981

o Kreyszig

Advanced engineering mathematics

John Wiley, 1972

o Franks

Modeling and simulation in chemical engineering

John Wiley, 1972

o Sherwood, Reed

Applied mathematics in chemical engineering

Mc. Graw Hill, 1957

e Jenson, Jeffreys

Mathematical methods in chemical engineering
Academic press, 1969

e Chapra, Canale

Metodos numericos para ingenieros
Mc. Graw Hill, 1988

e Geoffrey T. Leblond, Duglas Ford Cobb

Como usar Lotus 1-2-3
Macrobit editores, S. A. de C. V., 1990

e Tesis. Técnicas numericas aplicadas a la ingenieria

José Antonio Hernandez Morales
Sergio Alvarez Navarro, Facultad de Quimica, 1986