INSTITUTO POLITECNICO NACIONAL

Escuela Superior de Ingeniería Química e Industrias Extractivas

PROGRAMA TUTORIAL DE METODOS NUMERICOS, PARA LA SOLUCION DE ECUACIONES NO LINEALES Y SISTEMAS DE ECUACIONES LINEALES Y NO LINEALES

PROYECTO DE INVESTIGACION

Que para obtener el Título de

INGENIERO QUIMICO INDUSTRIAL

p r e s e n t a n

GLORIA CATALINA VALADEZ BARRON

JESUS GARCIA MANRIQUEZ

México, D. F. 1992

INSTITUTO POLITECNICO NACIONAL

ESCUELA SUPERIOR DE INGENIERIA QUIMICA E INDUSTRIAS EXTRACTIVAS

DIVISION DE SISTEMAS DE TITULACION T-082-92

SECRETARIA
DE
EDUCACION PUBLICA

México, D F, a 25 de Mayo de 1992.

Al(los) C Pasante(s).	Carrera:	Generación:
GLORIA CATALINA VALADEZ BARRON	I.Q.I.	1986-1991
JESUS GARCIA MANRIQUEZ	I.Q.I	1986-1991

Mediante la presente se hace de su conocimiento que esta División acepta que el C. Ing. M. en C. FEDERICO DOMINGUEZ SANCHEZ sea orientador en el Tema de Tesis que propone(n) usted(es) desarrollar como prueba escrita en la opción PROYECTO DE INVESTIGACION(COLECTIVA 2 PASANTES) bajo el título y contenido siguientes.

"PROGRAMA TUTORIAL DE METODOS NUMERICOS, PARA LA SOLUCION DE ECUACIONES NO LINEALES Y SISTEMAS DE ECUACIONES LINEALES Y NO LINEALES"

RESUMEN
INTRODUCCION
I REVISION DE LOS METODOS NUMERICOS
II TECNICAS PARA LA ELABORACION DE PROGRAMAS TUTORIALES, INTERACTIVOS, Y DE SOLUCION DIRECTA
III ELABORACION DE PROGRAMAS TUTORIALES
IV ELABORACION DE PROGRAMAS INTERACTIVOS
V ELABORACION DE PROGRAMAS DE SOLUCION DIRECTA
CONCLUSIONES
BIBLIOGRAFIA
ANEXOS

Se concede plazo máximo de un año para presentarlo a revisión por el Jurado

LIC. FERNANDO FLORES BENITEZ
VOCAL DE CARRERA

M. en C. FEDERICO DOMINGUEZ SANCHEZ.
EL PROFESOR ORIENTADOR
CED. PROF. 462988

M. en C. NANCY P. MARTINEZ CRUZ.
EL JEFE DE LA DIVISION DE SISTEMAS DE TITULACION

ING. NESTOR L. DIAZ RAMIREZ
EL SUBDIRECTOR ACADEMICO

dvv*

SECRETARIA
DE
EDUCACION PUBLICA

INSTITUTO POLITECNICO NACIONAL

ESCUELA SUPERIOR DE INGENIERIA QUIMICA E INDUSTRIAS EXTRACTIVAS

MEXICO, D. F., 26 de noviembre de 1992

C. GLORIA CATALINA VALADEZ BARRON
JESUS GARCIA MANRIQUEZ
Pasante de Ingeniero QUIMICO INDUSTRIAL

Presente:

Los suscritos tenemos el agrado de informar a usted que, habiendo procedido a revisar el borrador de la modalidad de titulación correspondiente, denominado "PROGRAMA TUTORIAL DE METODOS NUMERICOS, PARA LA SOLUCION DE ECUACIONES NO LINEALES Y SISTEMAS DE ECUACIONES LINEALES Y NO LINEALES".

encontramos que el citado trabajo y/o proyecto de tesis, reúne los requisitos para autorizar el Examen Profesional y proceder a su impresión según el caso, debiendo tomar en consideración las indicaciones y correcciones que al respecto se le hicieron.

Atentamente

JURADO

C. M. en C. FEDERICO DOMINGUEZ SANCHEZ
(ORIENTADOR)

C. M. en C. JOSE ANGEL GARCIA M.

C. DR. GUILLERMO MARROQUIN SUAREZ

c.c.p.—**Expediente.**

shr.

AGRADECIMIENTOS

AGRADEZCO A DIOS POR LOS PADRES QUE ME DIO, Y A ELLOS POR LA EDUCACION, EJEMPLO Y APOYO QUE ME HAN BRINDADO.

AGRADEZCO A MIS HERMANOS POR SER MIS AMIGOS

GLORIA CATALINA

AGRADEZCO A DIOS POR TODAS LAS BONDADES QUE HA TENIDO PARA MI.

AGRADEZCO A MI MADRE POR QUE GRACIAS A ELLA SOY QUIEN SOY.

AGRADEZCO A MI PADRE POR BRINDARME TODO EL APOYO QUE SIEMPRE TUVE.

AGRADEZCO A MIS HERMANOS POR TODO LO QUE HAN HECHO POR MI.

JESUS

"Programa tutorial en métodos numéricos, para la solución de ecuaciones no lineales, sistemas de ecuaciones lineales y no lineales

Contenido: Pag.

Contenido: Pag.

Resumen

Se realizó una revisión bibliográfica que permitió seleccionar los métodos numéricos para la solución de ecuaciones no lineales y solución de sistemas de ecuaciones lineales y no lineales. Los métodos numéricos se seleccionaron de acuerdo al grado de exactitud y el tiempo de C.P.U. que emplean.

Se mencionan las técnicas empleadas para la elaboración de los programas interactivos y de solución directa. Tales técnicas son empleadas, tanto para agilizar la programación como para facilitar la programación interactiva, algunas técnicas empleadas son:

1. Programación estructurada
2. Bibliotecas de subrutinas (TPU)
3. Pilas, colas, árboles
4. Notación polaca

Con la ayuda de estas técnicas se reducen tiempos de operación y se facilita la programación de los mismos.

La parte medular de este trabajo es la elaboración de programas tutoriales cuya finalidad es enseñar los métodos numéricos al usuario sin necesidad de que exista un profesor que lo oriente. Si un usuario estándar no es capaz de aprender, el programa no tiene la cualidad de ser tutor. La escritura de los programas tutoriales, por tanto debe cuidar todos los aspectos involucrados en la enseñanza.

Para que el usuario pueda evaluarse en cuanto a los conocimientos asimilados en la etapa tutorial del trabajo se cuenta con programas interactivos en los cuales el usuario participa en la resolución de los problemas planteados por él mismo y el programa

lo evalúa, no permitiéndole que avance a la siguiente etapa hasta que no haya resuelto correctamente la actual. Como es necesario que durante las sesiones usuario-programa se realicen operaciones aritméticas el programa proporciona una calculadora integrada para que el alumno realice sus operaciones.

Una vez que el alumno haya aprendido la resolución paso a paso de los diversos métodos numéricos se plantea la opción de que pueda pedir la solución directa a problemas planteados por el programa, con el fin de que el alumno lo resuelva por su cuenta y pueda comparar su resultado con el de la computadora y de esta forma evaluar su aprendizaje. Y en caso necesario el usuario puede regresar a la sección tutorial.

Introducción

El Ingeniero Químico durante su formación académica se enfrenta a problemas de termodinámica, economía industrial, diseño de equipo, ingeniería de procesos, los cuales requieren para su solución resolver una ecuación lineal o no lineal; problemas de operaciones unitarias, cinética química, ingeniería de reactores en los que para llegar a un resultado es preciso resolver un sistema de ecuaciones lineales o no lineales. La solución de este tipo de problemas exige un conocimiento profundo de métodos numéricos y programación que no son del dominio, en general, de profesores y alumnos.

Actualmente existe una gran variedad de programas que realizan cálculos para la solución de ecuaciones no lineales, sistemas de ecuaciones lineales y no lineales e integración numérica, sin embargo estos programas no enseñan al usuario cómo se realizan estos cálculos -parte fundamental en el aprendizaje-, por lo que se hace necesario crear software tutorial para este tipo de cálculo que enseñe al alumno paso a paso la solución de problemas de ingeniería y ciencias que involucran métodos numéricos.

Hoy en día gracias al avance de tecnología en computación es posible realizar un gran número de cálculos en tiempos muy breves y resolver así problemas complejos que en otras epocas empleaban muchos días para realizarlos.

El presente trabajo tiene la finalidad de enseñar métodos numéricos a los estudiantes de las escuelas de Ingeniería del País mediante programas tutoriales, así como ayudar a los investigadores en las tareas de cálculos que les ocupe mucho tiempo mediante programas de cálculos eficientes y de fácil empleo.

Capitulo I
Revisión de los métodos numéricos

1. Ecuaciones no lineales.

1.1. Método de búsqueda exhaustiva

El Método de Búsqueda Exhaustiva, o también llamado de Fuerza Bruta, es empleado para encontrar la raiz de ecuaciones de la forma:

$$f(x) = 0$$

donde f(x) es una función de una variable x, como por ejemplo polinomios o bien funciones en las que se encuentran términos logaritmicos, exponenciales y/o trigonométricos:

$$f(x) = a + bx + cx^2 = 0$$
$$f(x) = e^x - \ln(3x) + \operatorname{sen}(2x) - 3x$$

Este método encuentra una solución de f(x)=0 dada la función continua f en el intervalo [a,b] donde f(a) y f(b) tienen signos opuestos.

Para encontrar la raiz el método sigue los siguientes pasos:

1. Se escoge un valor inicial x_0 donde:

$$a \leq x_0 \leq b$$

2. Se evalúa $f(x_0)$.
3. Se elige un valor para el ancho del salto (h), el cual debe ser mayor a cero y menor a la diferencia de $|b-a|$.

$$0 < h < |b-a|$$

4. Se elige un criterio de exactitud (ξ) muy pequeño.
5. Se propone un número máximo de iteraciones.
6. Se evalúa un nuevo valor de x sumando h a x_0, así

obtenemos x_1.

7. Evaluar $f(x_1)$.
8. Se tiene el siguiente criterio de conversión:

   Si $\left| x_1 - x_0 \right| \leq \xi$ entonces la raiz está entre x_0 y x_1 y termina el método.

   de modo contrario

   Si $f(x_0) * f(x_1) < 0$ entonces $h = -\frac{h}{2}$ y el valor de x_1 es ahora el valor inicial x_0 y el valor de $f(x_1)$ es ahora $f(x_0)$.
9. Se regresa al paso 6 hasta que se cumpla el criterio de convergencia o se llegue al número máximo de iteraciones propuesto.

Una representación gráfica del método se muestra en la figura 1.1.a.

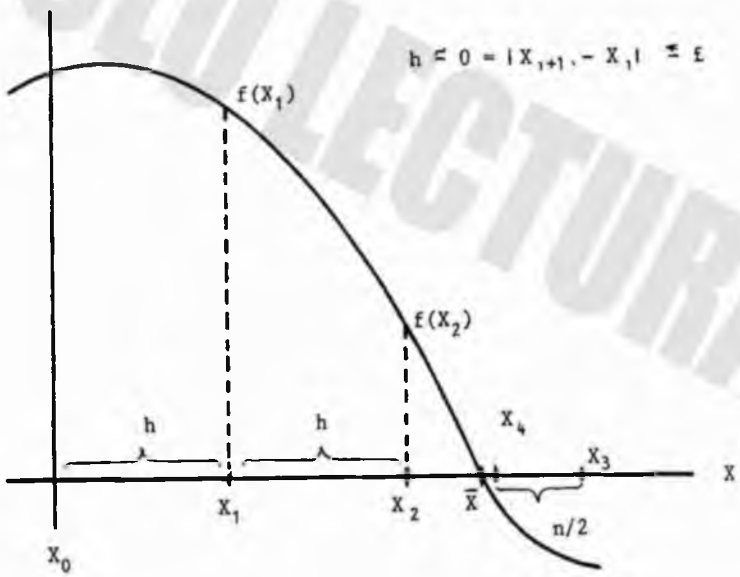


FIG. 1.1.a

1.2. Método de Newton Raphson

Uno de los problemas que se presenta con frecuencia en Ingeniería es encontrar las raíces de ecuaciones de la forma:

$$f(x) = 0$$

donde f(x) es una función real de una variable x, como por ejemplo un polinomio en x:

$$f(x) = 3x^3 - 2x^2 + 6x$$

o una función en la cual se encuentren términos trigonométricos, exponenciales y/o logarítmicos:

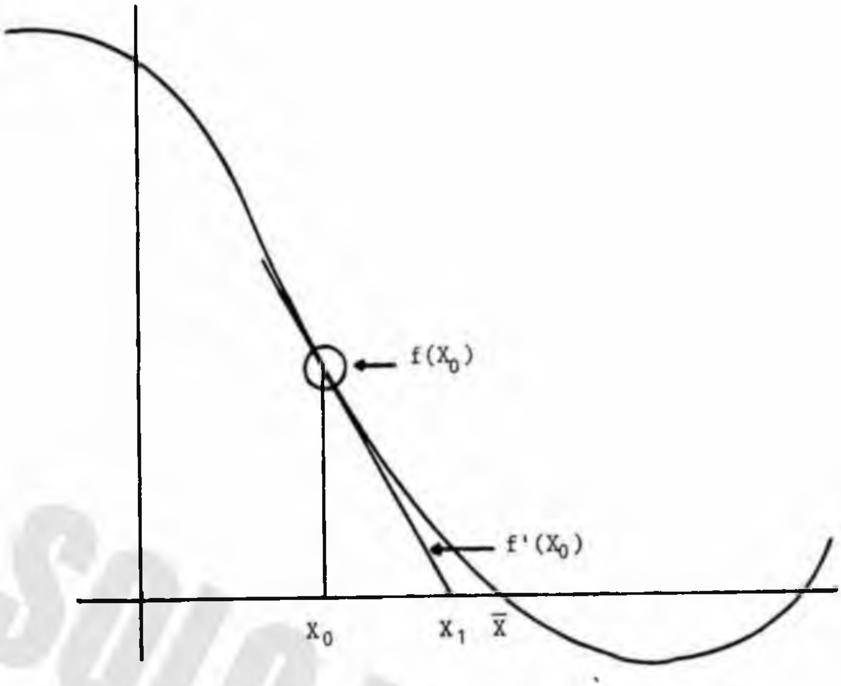
$$f(x) = \ln(3x) - e^x + 2\cos(x)$$

El método de Newton Raphson es de segundo orden de convergencia cuando se trata de raíces reales no repetidas.

Para encontrar la raíz se procede de la manera siguiente:

1. Se escoge un valor inicial x_0.
2. Se evalúa $f(x_0)$.
3. Se traza una tangente en el punto $f(x_0)$.
4. El punto donde intersectan la tangente y el eje de las x es el nuevo punto x_1.
5. Se repiten los pasos del 2 al 4 hasta encontrar que la diferencia entre el punto nuevo y el anterior sea menor o igual a un criterio de exactitud.

La deducción del método se muestra en la Figura 1.2.a.



$$f'(X_0) = \frac{f(X_0) - 0}{X_0 - X_1}$$

$$X_1 = X_0 - \frac{f(X_0)}{f'(X_0)}$$

$$X_{i+1} = X_i - \frac{f(X_i)}{f'(X_i)}$$

Fig. 1.2.a

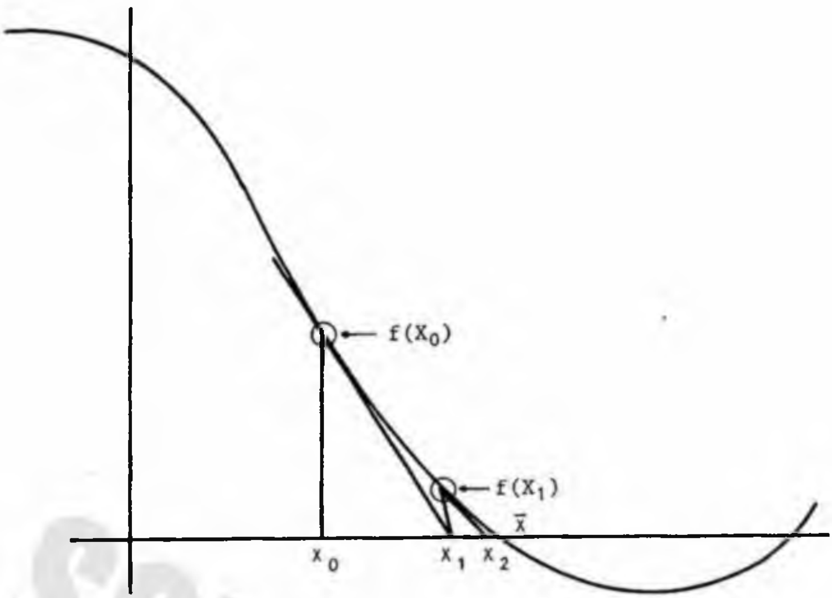


Fig. 1.2.c

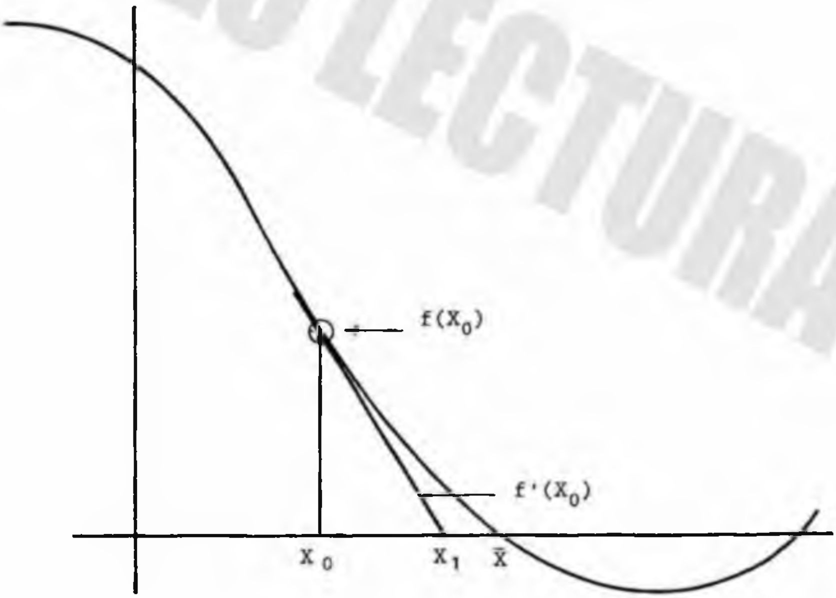


Fig. 1.2.b

El método sigue un proceso iterativo que dependiendo del valor inicial x_0 va a acercarse o a alejarse rápida o lentamente hacia la raíz.

Un ejemplo gráfico del procedimiento descrito anteriormente se muestra en las Figuras 1.2.b. y 1.2.c.

Se debe cuidar que los valores iniciales propuestos no conduzcan a la indeterminación de la función. esto es debido a que algunas funciones presentan intervalos en los cuales son continuas e intervalos en los que no son continuas. En la figura 1.2.d. se presentan los casos en donde el método tiene fallas.

Dependiendo del criterio de exactitud, es mayor o menor el tiempo empleado de C.P.U. y por tanto el número de iteraciones. esto no quiere decir que a menor tolerancia sea más exacto el resultado debido a que entre más cifras decimales se empleen mayor será el número de cifras perdidas durante el proceso.

EL METODO ES ATRAPADO POR UNA RAIZ IMAGINARIA

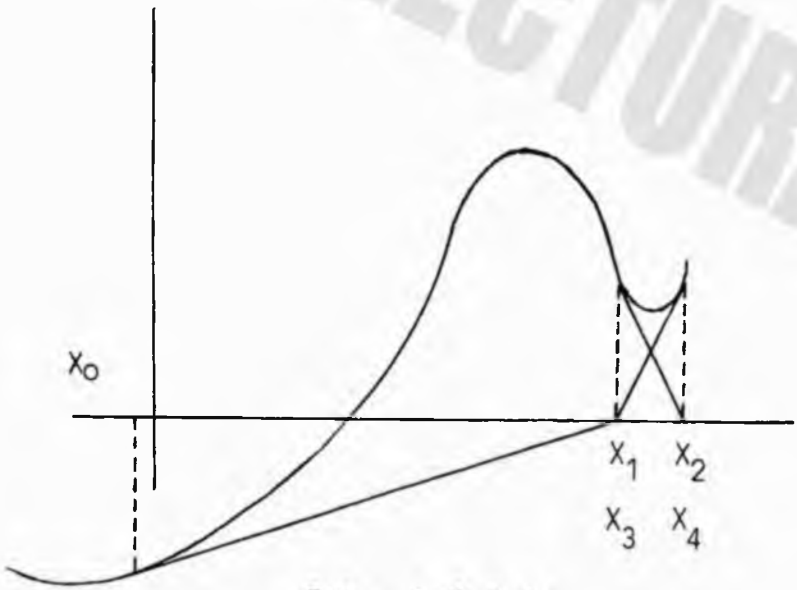


FIG. 1.2.d.1.

FALLAS DEL METODO

LA RAIZ ES UN PUNTO DE INFLEXION

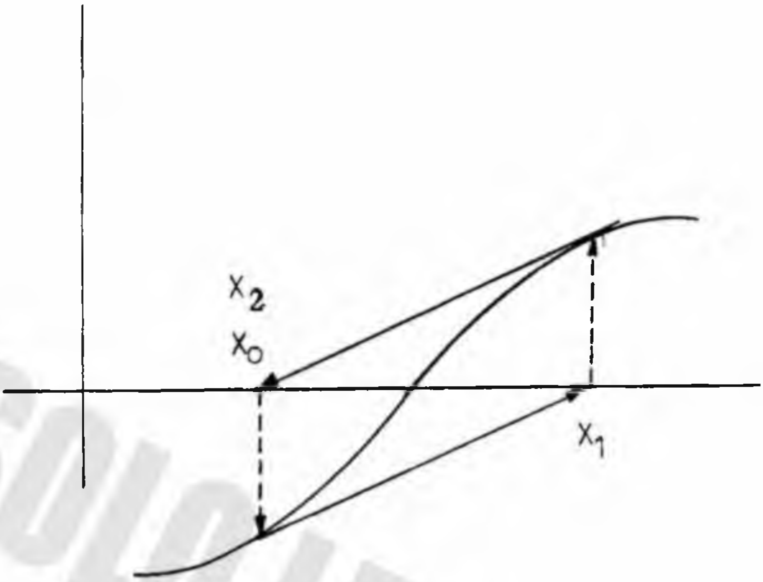


FIG. 1.2.d.2

EL METODO ES ATRAPADO POR UN MAXIMO O UN MINIMO

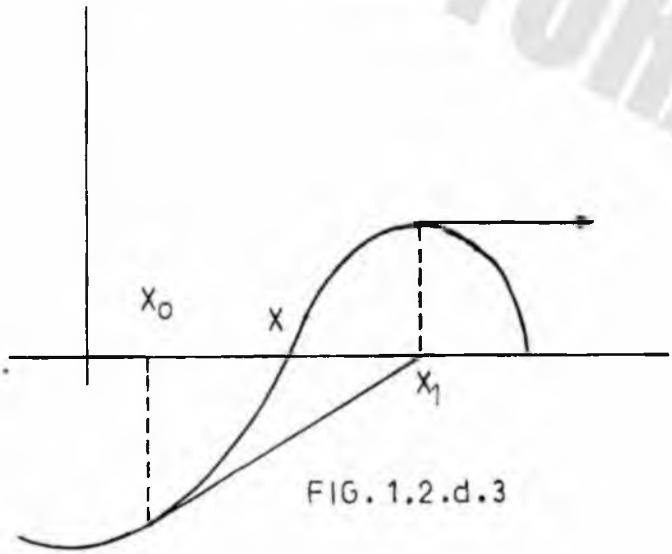


FIG. 1.2.d.3

1.3. Mètodo de Bisección.

El Mètodo de Bisección es empleado para encontrar la raìz de ecuaciones de la forma:

$$f(x) = 0$$

Para encontrar una solución de $f(x)=0$ dada la función continua f en el intervalo [a.b] donde $f(a)$ y $f(b)$ tienen signos opuestos se procede de la manera siguiente:

1.- Se elige un valor x_I (x a la izquierda) y otro valor x_D (x a la derecha). siempre y cuando $f(x_I) * f(x_D) < 0$. o sea que $x_I \le \bar{x} \le x_D$.
2.- Se elige un criterio de exactitud (ξ).
3.- Se evalùan $f(x_I)$ y $f(x_D)$.
4.- Se evalúa x_M de acuerdo a:

$$x_M = \frac{x_D + x_I}{2}$$

5.- Se evalúa $f(x_M)$.
6.- Si $f(x_I) * f(x_M) > 0$ entonces $x_I = x_M$.
7.- Si $f(x_I) * f(x_M) < 0$ entonces $x_D = x_M$.
8.- Se repite la secuencia a partir del paso 2 hasta que $\left| x_D - x_I \right|$ sea $< \xi$.

Una representación gràfica del mètodo se muestra en la figura 1.3.a.

Condición necesaria

$f(X_I)f(X_D) < 0$

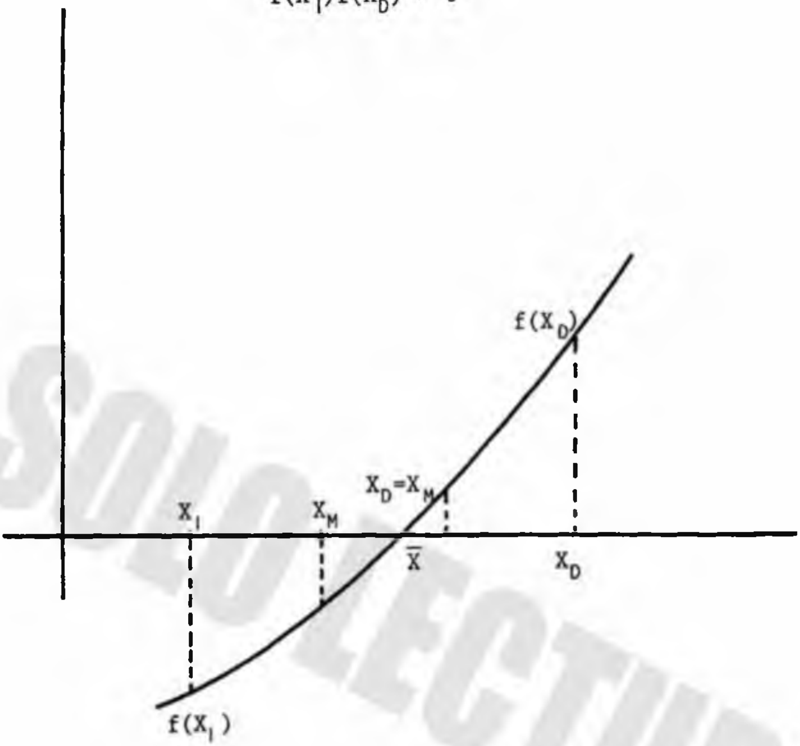


Fig. 1.3.a

2. Solución de Sistemas de Ecuaciones lineales

2.1 Eliminación de Gauss

La evaluación de la tasa de retorno para alternativas multiples nos conduce al planteamiento de una ecuación lineal por cada alternativa, originando así un sistema de ecuaciones de la forma siguiente:

$$\begin{aligned} 3x + 2y + 6z &= 20 \\ x - 4y + 2z &= 5 \\ 9x + y - 3z &= -4 \end{aligned}$$

donde x,y,z. representan valores presentes. futuros o anualidades y el término independiente el valor del rescate del bien.

Generalizando lo anterior tenemos:

$$\begin{array}{l} A_{11}x_1 + A_{12}x_2 + \dots + A_{1n}x_n = A_{1n+1} \\ A_{21}x_1 + A_{22}x_2 + \dots + A_{2n}x_n = A_{2n+1} \\ \quad\vdots \qquad\quad \vdots \qquad\qquad\quad \vdots \qquad\quad \vdots \\ A_{n1}x_1 + A_{n2}x_2 + \dots + A_{nn}x_n = A_{nn+1} \end{array}$$

El mètodo realiza una transformación lineal de fila y columna para obtener finalmente:

$$\begin{array}{l} A_{11}x_1 + A_{12}x_2 + \dots + A_{1n}x_n = b_1 \\ 0 \quad + A_{22}x_2 + \dots + A_{2n}x_n = b_2 \\ \vdots \qquad \vdots \qquad\qquad \vdots \qquad \vdots \\ 0 \quad + 0 \quad + \dots + A_{nn}x_n = b_{n+1} \end{array}$$

En donde:

$$x_n = \frac{b_{n+1}}{A_{nn}}$$

$$x_i = \frac{b_i - \sum_{j=i+1}^{n} A_{ij}\, x_j}{A_{ii}}$$

El procedimiento de Gauss es el siguiente :

Se elige a la primera ecuación como fila pivote y se procede a realizar las siguientes operaciones:

$$a'_{ij} = \frac{-A_{kk} * A_{ij}}{A_{ik}} + A_{kj}$$

$$b'_i = \frac{-A_{kk} * b_i}{A_{ik}} + b_k$$

De acuerdo a lo siguiente:

$k = 1, 2, 3, \ldots, n-1$

$i = k+1, k+2, \ldots, n$

$j = k+1, k+2, \ldots, n$

Al terminar este procedimiento se realiza una sustitución regresiva de acuerdo a:

$$x_n = \frac{A'_{nn+1}}{A_{nn}}$$

$$x_i = \frac{A'_i - \sum_{j=i+1}^{n} A_{ij}\, x_j}{A_{ii}}$$

en donde x_i es la raiz de la variable x.

2.2. Eliminación de Gauss-Jordan

Con frecuencia el cálculo de la cinética de una reacción de polimerización o una reacción de pirólisis nos conduce a resolver un sistema de ecuaciones lineales de la forma:

$$\begin{aligned} 3x + 2y + 6z &= 20 \\ x - 4y + 2z &= 5 \\ 9x + y - 3z &= -4 \end{aligned}$$

donde x,y,z, representan a alguna variable de la reacción que puede ser concentración, densidad, presión, temperatura, etc.

Generalizando lo anterior tenemos:

$$\begin{array}{ccccccccc} A_{11}x_1 & + & A_{12}x_2 & + & \cdots & + & A_{1n}x_n & = & A_{1n+1} \\ A_{21}x_1 & + & A_{22}x_2 & + & \cdots & + & A_{2n}x_n & = & A_{2n+1} \\ \vdots & & \vdots & & & & \vdots & & \vdots \\ A_{n1}x_1 & + & A_{n2}x_2 & + & \cdots & + & A_{nn}x_n & = & A_{nn+1} \end{array}$$

El método realiza una transformación lineal de fila y columna para obtener finalmente:

$$\begin{array}{ccccccccc} x_1 & + & 0 & + & \cdots & + & 0 & = & A'_{1n+1} \\ 0 & + & x_2 & + & \cdots & + & 0 & = & A'_{2n+1} \\ \vdots & & \vdots & & & & \vdots & & \vdots \\ 0 & + & 0 & + & \cdots & + & x_n & = & A'_{nn+1} \end{array}$$

En donde:

$$\begin{aligned} x_1 &= A'_{1n+1} \\ x_2 &= A'_{2n+1} \\ \vdots & \quad \vdots \\ x_n &= A'_{nn+1} \end{aligned}$$

El procedimiento de Gauss-Jordan consta de tres etapas :

1.- Normalización de la fila pivote i. La normalización se efectúa mediante el cociente:

$$A_{ij} = \frac{A_{ij}}{A_{ii}} \qquad . \qquad j = i+1, \; \ldots., \; n+1$$

2.- Eliminación de los elementos pivote arriba y abajo del elemento de la diagonal principal:

$$A'_{kj} = A_{kj} - A_{ij} \, A_{ki}$$

$j = i+1, \; \ldots, \; n+1$

$k = 1, \; \ldots, \; n$ para $k<>i$

$i = 1, \; \ldots, \; n$

3.- Asignación de la solución.

$x_i = A_{in+1}$ $\qquad i = 1, \; \ldots., \; n$

Ejemplo: resolver el sistema de ecuaciones lineales siguiente:

$$\begin{aligned} 3x_1 + 2x_2 + x_3 &= 6 \\ 5x_1 - 4x_2 - x_3 &= 0 \\ 2x_1 - 3x_2 + 5x_3 &= 4 \end{aligned}$$

La matriz coeficiente es:

$$\begin{bmatrix} 3 & +2 & +1 & 6 \\ 5 & -4 & -1 & 0 \\ 2 & -3 & +5 & 4 \end{bmatrix}$$

Lo primero que se debe verificar es que el número de ecuaciones sea como mínimo mayor o igual al número de incógnitas.

Etapa de normalizaciòn.

Para i=1. Aplicando la fòrmula tenemos:

$$A_{12} = \frac{A_{12}}{A_{11}} = \frac{2}{3}$$

$$A_{13} = \frac{A_{13}}{A_{11}} = \frac{1}{3}$$

$$A_{14} = \frac{A_{14}}{A_{11}} = \frac{6}{3} = 2$$

La matriz inicial se modifica a:

$$\begin{bmatrix} 1 & +2/3 & +1/3 & 2 \\ 5 & -4 & -1 & 0 \\ 2 & -3 & +5 & 4 \end{bmatrix}$$

Segunda etapa. Eliminaciòn.

Para k=2, j=2. Aplicando la fòrmula tenemos:

$$A'_{22} = A_{22} - A_{12}\,A_{21} = -4 - (2/3)(5) = -22/3$$

$$A'_{23} = A_{23} - A_{13}\,A_{21} = -1 - (1/3)(5) = -8/3$$

$$A'_{24} = A_{24} - A_{14}\,A_{21} = 0 - (2)(5) = -10$$

Para k=3, j=2.

$$A'_{32} = A_{32} - A_{12}\,A_{31} = -3 - (2/3)(2) = -13/3$$

$$A'_{33} = A_{33} - A_{13}\,A_{31} = 5 - (1/3)(2) = 13/3$$

$$A'_{34} = A_{34} - A_{14}\,A_{31} = 4 - (2)(2) = 0$$

La matriz anterior se modifica a:

$$\begin{bmatrix} 1 & +2/3 & +1/3 & 2 \\ 0 & -22/3 & -8/3 & -10 \\ 0 & -13/3 & +13/3 & 0 \end{bmatrix}$$

Y así se realizan los pasos de normalización y eliminación hasta llegar a la matriz siguiente:

$$\begin{bmatrix} 1 & +0 & +0 & 1 \\ 0 & +1 & +0 & 1 \\ 0 & +0 & +1 & 1 \end{bmatrix}$$

Asignando soluciones:

$$x_1 = 1$$
$$x_2 = 1$$
$$x_3 = 1$$

Al resolver un sistema de ecuaciones lineales por eliminación de Gauss-Jordan, la memoria de computadora requerida es proporcional al cuadrado del orden de la matriz y el trabajo de la computadora es proporcional al cubo del orden de la matriz. Debido a esto, la solución de sistemas lineales grandes con matrices con pocos ceros como elementos se vuelve practicamente imposible de realizar, la memoria se satura y el número de operaciones se incrementa enormemente.

2.3.- Jacobi

Las Leyes de Kirchhoff para circuitos elèctricos establecen que el flujo neto de corriente a travès de cada uniòn de un circuito es cero y que la caìda neta de voltaje alrededor de una secciòn cerrada de circuito es cero. Supongamos que se aplica un potencial de (V) volts entre los puntos A y G en el circuito de la figura 2.3.a. y que Vb, Vc, Vd, Ve y Vf son los potenciales en los puntos B, C, D, E y F respectivamente. Usando G como punto de referencia las leyes de Kirchhoff implican que estos potenciales satisfacen el siguiente sistema de ecuaciones lineales:

$$
\begin{array}{rrrrrl}
31Vb & -\ 10Vc & & & -\ 6Vf & = 15V \\
2Vb & -\ 8Vc & +\ 3Vd & +\ 3Ve & & = 0 \\
 & Vc & -\ 3Vd & +\ 2Ve & & = 0 \\
 & 2Vc & +\ 4Vd & -\ 7Ve & +\ Vf & = 0 \\
12Vb & & +\ 15Vd & & -\ 47Vf & = 0
\end{array}
$$

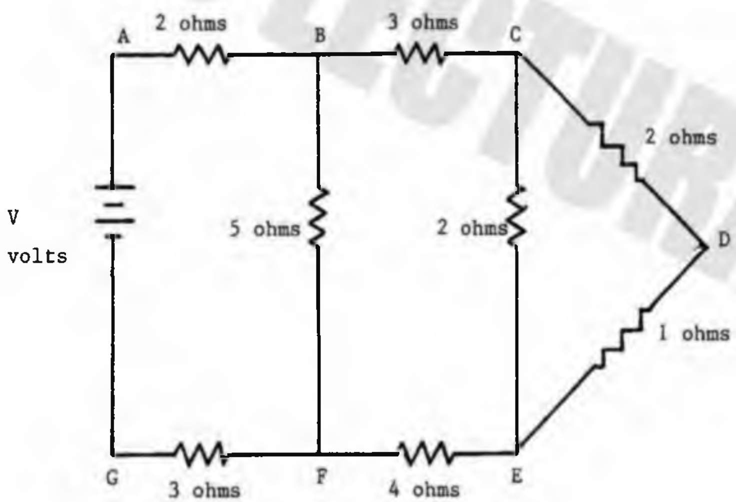


Fig. 2.3.1

Las ecuaciones anteriores se comportan de acuerdo al modelo siguiente:

$$a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1$$

$$a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n = b_2$$

$$\vdots \qquad \vdots \qquad \qquad \vdots \qquad \vdots$$

$$a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n = b_n$$

El método de Jacobi es de tipo iterativo y para la solución de un sistema de ecuaciones lineales, se aplica la siguiente secuencia:

1.- Se elige un criterio de exactitud (ξ) muy pequeño.

2.- Se despeja x_1 de la ecuación 1:

$$x_1 = \frac{1}{a_{11}} (b_1 - a_{12}x_2 - a_{13}x_3 - \dots - a_{1n}x_n)$$

$$x_2 = \frac{1}{a_{22}} (b_2 - a_{21}x_2 - a_{23}x_3 - \dots - a_{2n}x_n)$$

$$\vdots$$

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

En forma general:

$$x_i^{k+1} = \frac{1}{a_{ii}} \left[b_i - \sum_{\substack{j=1 \\ j \neq i}}^{n} a_{ij} x_j^k \right] \qquad i = 1,2,3,\dots,n$$

3.- Se elige un vector de valores iniciales $\bar{x}_0$, que puede ser:

$$\bar{x}_0 = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

4.- Se calculan nuevos valores para cada x_i :

$$x_1^1 = \frac{1}{a_{11}} (b_1 - a_{12}x_2 - a_{13}x_3 - \dots - a_{1n}x_n)$$

$$x_2^1 = \frac{1}{a_{22}} (b_2 - a_{21}x_2 - a_{23}x_3 - \dots - a_{2n}x_n)$$

$$\vdots$$

$$x_n^1 = \frac{1}{a_{nn}} (b_n - a_{n1}x_1 - a_{n2}x_2 - \dots - a_{nn-1}x_{n-1})$$

5.- Se repite el procedimiento hasta que:

$$\left| x_i^{k+1} - x_i^k \right| \leq \xi$$

para $1 \leq i \leq n$

o aplicando la norma euclideana:

$$\left| x_i^{k+1} - x_i^k \right| = \left[(x_1^{k+1}-x_1^k)^2 + (x_2^{k+1}-x_2^k)^2 + \dots + (x_n^{k+1}-x_n^k)^2 \right]^{\frac{1}{2}}$$

3. Solución de Sistemas de Ecuaciones no lineales.

3.1. Método iterativo de punto fijo

El resolver problemas de Ingeniería de reactores, optimización en Ingeniería de procesos y Economía Industrial en muchas ocasiones nos conduce a sistemas de ecuaciones no lineales que son difíciles de resolver algebraicamente, por lo tanto se debe recurrir a un método numérico que resuelva estos sistemas.

Es muy difícil, casi imposible, graficar las superficies multidimensionales definidas por estas ecuaciones; métodos como el de bisección y el de posición falsa, que siempre convergen en el caso de una dimensión, son inaplicables en varias dimensiones. Uno de los métodos tradicionales para la solución de estos sistemas es el método iterativo de punto fijo, en el cual se siguen los siguientes pasos:

Sea el sistema de ecuaciones:

$$f_1(x,y) = x^2 - 10x + y^2 + 8 = 0$$
$$f_2(x,y) = xy^2 + x - 10y + 8 = 0$$

1.- Despejamos a x de la función $f_1(x,y)$ y a y de la función $f_2(x,y)$.

$$x = \frac{x^2 + y^2 + 8}{10} = g_1(x,y)$$

$$y = \frac{xy^2 + x + 8}{10} = g_2(x,y)$$

2.- Se eligen valores iniciales para x y y, además de un criterio de convergencia (ξ) muy pequeño.

$$x_0 = 0$$

$$y_0 = 0$$

3.- Evaluar $g_1(x_o,y_o)$ para asignar su valor a x_1, y $g_2(x_o,y_o)$ para asignar su valor a y_1.

$$x_1 = g_1(x_o,y_o) = \frac{0^2 + 0^2 + 8}{10} = 0.8$$

$$y_1 = g_2(x_o,y_o) = \frac{0^2 + 0^2 + 8}{10} = 0.8$$

4.- Se repiten los pasos anteriores hasta que se llegue al criterio de convergencia:

$$\left| x_n - x_{n-1} \right| \leq \xi$$

$$\left| y_n - y_{n-1} \right| \leq \xi$$

También se puede conocer al iniciar si la función converge d acuerdo a la siguiente condición necesaria:

$$\left| \frac{\partial g_1(\bar{x},\bar{y})}{\partial x} \right| + \left| \frac{\partial g_1(\bar{x},\bar{y})}{\partial y} \right| < 1$$

$$\left| \frac{\partial g_2(\bar{x},\bar{y})}{\partial x} \right| + \left| \frac{\partial g_2(\bar{x},\bar{y})}{\partial y} \right| < 1$$

$$\frac{\partial g_1(\bar{x},\bar{y})}{\partial x} = \frac{2x}{10} \qquad \frac{\partial g_1(\bar{x},\bar{y})}{\partial y} = \frac{2y}{10}$$

$$\left| \frac{2}{10} \right| + \left| \frac{2}{10} \right| < 1$$

$$\frac{\partial g_2(\bar{x},\bar{y})}{\partial x} = \frac{y^2+1}{10} \qquad \frac{\partial g_2(\bar{x},\bar{y})}{\partial y} = \frac{2xy}{10}$$

$$\left| \frac{2}{10} \right| + \left| \frac{2}{10} \right| < 1$$

3.2. Mètodo de Newton Raphson

Este mètodo se obtiene de la expansión en serie de Taylor de la función alrededor de un punto.

$$f(x) = f(x_0) + f'(x_0)(x-x_0) + f''(x_0)\frac{(x-x_0)^2}{2!} + \ldots$$

Tomando los dos primeros tèrminos

$$0 = f(x) \cong f(x_0) + f'(x_0)(x-x_0)$$

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

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

Dadas las raíces de las funciones f_1 y f_2 y empleando la serie de Taylor se tiene:

$$f_1(x,y) = 0$$

$$f_2(x,y) = 0$$

$$0 = f_1(x,y) \cong f_1(x_0,y_0) + \frac{\partial f_1(x_0,y_0)}{\partial x}(x-x_0) + \frac{\partial f_1(x_0,y_0)}{\partial y}(y-y_0)$$

$$= f_2(x,y) \cong f_2(x_0,y_0) + \frac{\partial f_2(x_0,y_0)}{\partial x}(x-x_0) + \frac{\partial f_2(x_0,y_0)}{\partial y}(y-y_0)$$

Si $\quad x - x_0 = r$

$\quad\quad y - y_0 = s$

entonces:

$$\frac{\partial f_1(x_0,y_0)}{\partial x} r + \frac{\partial f_1(x_0,y_0)}{\partial y} s = -f_1(x_0,y_0)$$

$$\frac{\partial f_2(x_0,y_0)}{\partial x} r + \frac{\partial f_2(x_0,y_0)}{\partial y} s = -f_2(x_0,y_0)$$

Se resuelve este sistema de ecuaciones lineales y entonces:

$$x = r + x_0$$
$$y = s + y_0$$

Se repite el procedimiento hasta que:

$$r \cong 0$$
$$s \cong 0$$

para estas condiciones

$$f_1(\bar{x},\bar{y}) = 0$$
$$f_2(\bar{x},\bar{y}) = 0$$

Ejemplo:

$$f_1(x,y) = x^2 + y^2 - 10x + 8 = 0$$
$$f_2(x,y) = xy^2 + x - 10y + 8 = 0$$

$$\frac{\partial f_1}{\partial x} = 2x-10 \qquad \frac{\partial f_1}{\partial y} = 2y$$

$$\frac{\partial f_2}{\partial x} = y^2+1 \qquad \frac{\partial f_2}{\partial y} = 2xy-10$$

$$x_0 = 0$$
$$y_0 = 0$$

$$-10r + 0s = -8$$
$$r - 10s = -8$$

Resolviendo el sistema

$$r = 0.8$$
$$s = 0.88$$

Entonces:

$$x_1 = 0.8 + 0 = 0.8$$
$$y_1 = 0.88 + 0 = 0.88$$

Sustituyendo valores:

$$\frac{\partial f_1}{\partial x} = 2x - 10 = -8.4 \qquad \frac{\partial f_1}{\partial y} = 2y = 1.76$$

$$\frac{\partial f_2}{\partial x} = y^2 + 1 = 1.7744 \qquad \frac{\partial f_2}{\partial y} = 2xy - 10 = -8.592$$

$$f_1(x,y) = x^2 + y^2 - 10x + 8 = 1.4144$$
$$f_2(x,y) = xy^2 + x - 10y + 8 = 0.6191$$

Para la segunda iteración tenemos:

$$-8.4\ r + 1.76\ s = -1.4144$$
$$1.7744r - 8.592s = -0.6191$$

Resolviendo:

$$r = 0.1918$$
$$s = 0.1116$$

Como

$$r \not\cong 0$$
$$s \not\cong 0$$

Procedemos a la siguiente iteración hasta que

$$r \cong \xi$$
$$s \cong \xi$$

3.3. Método de Newton Raphson modificado.

La deducción de la siguiente ecuación se obtiene del método de Newton Raphson para la solución de una ecuación no lineal, pero si lo aplicamos para un sistema de n ecuaciones no lineales con n variables se tiene:

$$x_i^{k+1} = x_i^k - \frac{f_i(x_1^k, x_2^k, \ldots, x_n^k)}{\dfrac{\partial f_i}{\partial x_i}(x_1^k, x_2^k, \ldots, x_n^k)}$$

De donde:

$i = 1, 2, \ldots, n$

Ejemplo:

Si se tiene el siguiente sistema de ecuaciones no lineales:

$$f_1(x,y) = x^2 - 10x + y^2 + 8 = 0$$

$$f_2(x,y) = xy^2 + x - 10y + 8 = 0$$

y aplicando la ecuación anterior se tiene:

$$x^{k+1} = x^k - \frac{f_1(x^k, y^k)}{\dfrac{\partial f_1}{\partial x}(x^k, y^k)}$$

$$y^{k+1} = y^k - \frac{f_1(x^k, y^k)}{\dfrac{\partial f_1}{\partial y}(x^k, y^k)}$$

Para encontrar las raíces, se procede de la manera guiente:

1. Se escoge un valor inicial x_0, y_0 y un criterio de exactitud muy pequeño (ξ)

2. Se evalúan en este punto las funciónes f_1 y f_2.

3. Se evalúan en este punto las derivadas de las funciones f_1 y f_2 con respecto a x y y.

4. Se sustituyen los valores calculados en los puntos 2 y 3 en las ecuaciones anteriores para conocer los nuevos valores.

5. Se repiten los pasos del 2 al 4 hasta que la diferencia entre el punto nuevo y el anterior sea menor o igual al criterio de exactitud.

Capítulo II.

Técnicas para la elaboración de programas tutoriales, interactivos, y de solución directa.

1. Programación estructurada.

La característica que distingue a un lenguaje estructurado es su capacidad para compartamentalizar programa y datos. Esto significa que puede separar y almacenar del resto del programa toda la información e instrucciones necesarias para realizar una tarea específica. Generalmente la compartamentalización se realiza mediante subrutinas, llamadas frecuentemente subprogramas, con variables locales, las cuales son temporales. En este sentido es posible escribir subrutinas tales que lo que sucede en ellas no causará efectos en otras partes del programa. El excesivo uso de variables globales (variables conocidas por la totalidad del programa) puede dar lugar a arrastrar errores en un programa, al no poderse detectar por sus efectos locales. En una programación estructurada, todos los subprogramas son funciones concretas o procedimientos.

Las funciones y procedimientos son los bloques que constituyen una programación estructurada. Una tarea específica en un programa se puede definir y programar separadamente en funciones y procedimientos. Después de depurar la función o procedimiento, un programador puede contar con él para trabajar sin crear efectos secundarios en otras partes del programa. Todas las variables declaradas en esa función o procedimiento son conocidas sólo por esa subrutina.

En un lenguaje de programación estructurada al usar bloques de programa se crea un programa estructurado. Un bloque de código, es un grupo de sentencias del programa conectado lógicamente que se puede procesar como una unidad y están delimitadas por palabras clave dependiendo el lenguaje. El usar bloques de código crea programas legibles con una lógica fácil de seguir. Los bloques de código también ayudan a escribir programas mejores y libres de errores, porque el significado de cada sentencia está claro.

Un lenguaje de programación estructurada permite crear y mantener una biblioteca personal de funciones a la medida del usuario y permite separar archivos en código fuente.

2.- Bibliotecas de Procedimientos (TPU).

El elaborar un programa demasiado complejo significa que el programador empleará un sinnúmero de sentencias, ocasionando que el programa vaya creciendo. En la mayoría de estos programas algunas de las sentencias se repiten con frecuencia. Esto origina la creación de subrutinas, una subrutina permite estructurar el programa y disminuir el número de sentencias compactando así el programa. El llamado a una subrutina, hace que el compilador vaya de un punto a otro del programa y cada vez que se hace un llamado se tendrá que compilar la subrutina dando como resultado que el programa sea lento.

Para solucionar este problema los lenguajes de programación estructurada permiten compilar una o más subrutinas en un archivo, al cual, se le da el nombre de unidad o bilioteca de procedimientos. Una vez compilada esta unidad queda residente en memoria y puede ser llamada en un sinnumero de ocasiones desde cualquier otro programa sin la necesiadad de compilarla nuevamente.

Las ventajas que ofrece una biblioteca de procedimientos o unidad son:

1.- Programas más compactos
2.- Fácil identificación de posibles errores
3.- Programas más entendibles
4.- Empleo de programación estrucurada.

En otras palabras, la creación de bibliotecas de procedimientos o unidades permite seccionar nuestro programa en bloques y despues unirlos como si estuviéramos armando un rompecabezas.

3.- Pilas, colas y árboles.

Los programas están formados por algoritmos y estructuras de datos. Un buen programa es una mezcla de ambos. Escoger e implementar una estructura de datos es tan importante como las rutinas que manejan esos datos. Cómo está organizada y se accede a la información se determina normalmente por el problema a programar. Por tanto, se deduce que, como programador, debe tener en su "saco de artimañas" los métodos correctos de almacenamiento y recuperación para cualquier situación.

Las colas, las pilas y los árboles son el nivel final en la secuencia en que los datos se almacenarán y recuperarán. Cada uno de estos métodos proporcionará una solución a una clase de problema; cada uno es en esencia un dispositivo que representa una operación específica de almacenamiento y recuperación con la información dada y requerida. Los métodos comparten dos operaciones: almacenar un elemento y recuperar un elemento, en el que el elemento es una unidad de información.

Colas.- Una cola es una lista lineal de información a la que se accede en el orden "el primero que entra es el primero que sale". El primer elemento situado en la cola es el primero que se saca. el segundo que se pone en la cola es el segundo elemento que se saca y así sucesivamente. Este orden es la única manera de almacenar y recuperar; no se permite acceder a un determinado elemento de la cola de forma aleatoria. Las colas son comunes en la vida diaria. Por ejemplo una cola en un banco o en un restaurante de comida rápida (excepto cuando se cuelan).

Para entender como trabaja una cola veamos el siguiente ejemplo:

Acción	Contenido de la cola
Almacenar en cola (A)	A
Almacenar en cola (B)	A B
Almacenar en cola (C)	A B C
Sacar de cola, devuelve (A)	B C
Almacenar en cola (D)	B C D
Sacar de cola, devuelve (B)	C D
Sacar de cola, devuelve (C)	D

Pilas.- Una pila es lo opuesto a una cola, porque utiliza el acceso de "último en entrar primero en salir". Imaginemos una pila de platos: el plato de abajo de la pila será el último en usarse y el plato de arriba (el último plato puesto en la pila) es el primero en utilizarse. Las pilas se emplean en gran cantidad de sistemas de compiladores e intèrpretes.

Para entender como trabaja una pila veamos el siguiente ejemplo:

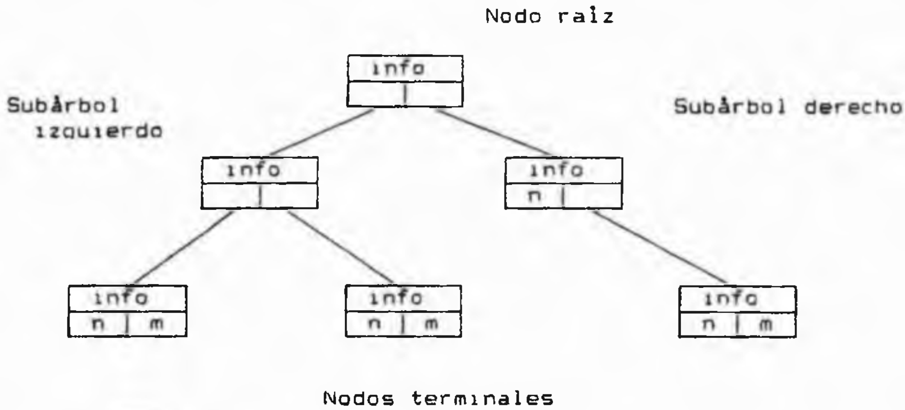
Acción	Contenido de la pila
Almacenar en pila (A)	A
Almacenar en pila (B)	B A
Almacenar en pila (C)	C B A
Sacar de pila, devuelve (C)	B A
Almacenar en pila (D)	D B A
Sacar de pila, devuelve (D)	B A
Sacar de pila, devuelve (B)	A

Arboles.- Aunque puede haber muchos tipos de árboles, los árboles binarios son especiales, porque cuando están ordenados se prestan a búsquedas, inserciones y borrados rápidos. Cada elemento en un árbol binario, consta de una información con un enlace al miembro izquierdo y otro al derecho.

La terminología especial necesaria para discutir los árboles es un caso clásico de mezcla de metáforas. La raíz es el primer elemento del árbol. A cada elemento de datos se le llama un nodo (o a veces una hoja del árbol), y a cualquier trozo del árbol se le llama un subárbol. Un nodo que no tiene subárboles conectados a él se le llama nodo terminal. La altura del árbol es igual al número de capas por debajo que crecen de su raíz. A través de esta discursión se piensa que los árboles binarios aparecen en la memoria de la misma manera que sobre papel, pero recordando que un árbol es solo una forma de estructurar datos en memoria y la memoria es de forma lineal.

En un árbol se puede insertar, borrar y acceder los elementos en cualquier orden, además de que la operación de recuperar no es destructiva y aunque son fáciles de visualizar, los árboles presentan problemas de programación difíciles. La mayoría de las funciones que utilizan los árboles son recursivas, porque el árbol en sí mismo es una estructura de datos recursivos; es decir, cada subárbol es un árbol.

Para poder entender esto veamos el siguiente ejemplo:



4. Notación polaca

El análisis de expresiones constituye la parte medular de todos los compiladores e intérpretes de lenguajes, debido a que en la actualidad la mayoría de las calculadoras emplean un analizador de expresiones.

Es sorprendente que al introducir una expresión (cadena de caracteres) en la calculadora nos de una respuesta inmediata. La velocidad de respuesta depende de que tan eficaz sea el analizador de expresiones.

Existen algunos analizadores que emplean la notación infija para elaborar el algoritmo de análisis, esta notación emplea la siguiente lógica:

Operando	Operador	Operando
32	+	27

Algunos otros emplean la notación postfija, tambien llamada polaca, que tiene la siguiente lógica:

Operando	Operando	Operador
32	27	+

Las ventajas de la notación postfija son las siguientes:

1. Una vez introducida la expresión en notación infija, el procedimiento de análisis (conversión) separa y almacena en una pila los operadores y los operandos.

2. El almacenar en una pila los operadores y en otra los operandos nos permite ordenar de acuerdo a su prioridad a los operadores y así el último operador en la pila es el de mayor prioridad.

3. Al término del análisis (conversión) la evaluación de la expresión es relativamente sencilla ya que se tiene ordenados a los operandos y operadores.

4. Se emplean pocas variables debido a que un nuevo valor parcial, se almacena donde el anterior.

Desventajas de la notación infija :

1. El análisis de la expresión origina la creación de un programa recursivo, ya que al analizar término por término de la expresión tiene que recorrer diversos niveles de procedimientos para identificación y almacenamiento de resultados parciales.

2. La expresión tiene que ser varias veces recorrida (examinada) debido a que se comparan los términos para determinar cuál es el de mayor prioridad.

3. Se utilizan muchas variables, para las cuales se debe asegurar, que cada vez que se emplee este procedimiento deben ser inicializadas correctamente.

Ejemplo:

Dada la expresión en notación infija :

(A+B)/(C-D)

realizar la conversión a notación postfija:

El programa arrojaría el siguiente resultado:

AB+CD-/

Como se puede observar la división (/) en este caso, tiene una prioridad secundaria, debido a que se deben realizar las operaciones de suma (+) y resta (-) en primer orden, o sea que para evaluar la expresión unicamente tenemos que ir de izquierda a derecha.

III. Elaboración de programas tutoriales

1. Características y diseño

Hace algunos años, cuando en las escuelas de Ingeniería del país se impartían los cursos de métodos numéricos, los alumnos empleaban demasiado tiempo en encontrar la solución de algún problema específico, debido a que con la regla de cálculo era casi imposible terminar un examen de fin de semestre.

Con el avance tecnológico empezaron a surgir computadoras con mayor velocidad en el procesamiento de datos, esto trajo como consecuencia la ampliación de los programas de estudio y la posibilidad de revisar varios métodos para cada tema en particular y así poder determinar cual método es más rápido, cuál tiene mayor grado de exactitud, cuál converge más rápido a la solución y en fin una serie de parámetros que nos guían hacia la selección correcta del método de acuerdo al problema que se tenga.

Actualmente se pretende explotar al máximo los beneficios que ofrece una computadora con una alta velocidad de procesamiento, y una de las formas de lograr esto, es la creación de programas tutoriales. Un programa tutorial es aquel que tiene como fin , valga la redundancia, ser el tutor del usuario en el aprendizaje de algún tema en particular de una forma amena, ligera y cordial.

Al diseñar un programa tutorial, se debe considerar la problemática existente en la relación profesor-alumno, esto es, no siempre es válido que un profesor que domine el tema, pueda trasmitir de una forma clara sus conocimiemtos hacia el alumno, esto se debe a dos causas principales: que el profesor carezca de una correcta metodología de enseñanza y que el alumno tenga poca capacidad de aprendizaje. Tomando en cuenta lo anterior podemos visualizar que se debe tener cuidado en el diseño de este tipo de programas, ya que estos deben de al menos igualar al profesor. Lo cual nos lleva a revisar una metodología de enseñanza aplicable a un programa.

Si consideramos que el programa tutorial debe imitar al profesor entonces, este tendrá que cumplir con las siguientes características:

Así como el profesor se apoya en materiales didácticos para impartir su cátedra, para este caso el programa tutorial deberá presentar efectos gráficos, visuales, de sonido, cambios de color e intensidad, diferentes tipos de letra, etc., en fin hará uso de aquello que permita mantener la atención del usuario fija en el tema.

Cuando un profesor se extiende demasiado en un tema ocasiona que el alumno pierda interés, por tanto el programa tutorial deberá presentar el tema lo más conciso y exacto que sea posible teniendo cuidado en remarcar los puntos importantes del mismo.

El programa tutorial debe ser de fácil acceso, presentado menús que lo lleven hacia donde el usuario necesite ir, tomando en cuenta, que existe la posibilidad de cometer errores en el momento de tener acceso al programa, debido a esto, se debe invalidar cualquier posibilidad de que el alumno incurra en errores contando con secuencias de escape

Generalmente el alumno se siente presionado por el profesor y por este motivo se guarda las dudas que pueda tener por temor a verse ridiculizado, en el caso de un programa tutorial el alumno no tendrá porque verse intimidado, debido a que el programa no lo regaña, ya que es su amigo.

Por último se contempla que el usuario debe tener ciertos co..imientos preliminares al tema de interés, en el caso particular de los métodos numéricos, se supone que el alumno ha asistido a cursos de álgebra, geometría y trigonometría, ya que sin estos ni el mejor profesor, ni el mejor programa tutorial podrán asisitir en el proceso de la enseñanza.

2. Validación y pruebas

La validación de un programa tutorial es un proceso que va de la mano con la escritura del mismo, por tanto es un proceso continuo, si no se validara al tiempo que se escriben las rutinas, esto ocasionaría que al validar el programa completo,se encontrarían demasiados errores y al tratar de identificarlos emplearíamos mucho tiempo.

El trabajar con gráficos da como consecuencia que se tenga que validar frecuentemente el programa debido a que se emplean coordenadas en pantalla (pixels) y en general los gráficos deben de presentar una distribución correcta en la pantalla. Otro aspecto importante es cuando se emplean pausas para que el usuario del programa tenga el tiempo necesario para poder leer y asimilar lo presentado. Cabe mencionar que se debe repetir cuantas veces sea necesario el proceso escritura-validación hasta que se tenga seguridad de que no existen fallas.

Lo anterior nos conduce a establecer, que quizá la característica más importante de un programa tutorial, es su diseño a prueba de errores.

Una vez concluida las etapas de esritura-validación del programa, este debe someterse a pruebas con diferentes tipos de usuarios con la finalidad de registrar información de las posibles anomalías que presente el programa durante su corrida. La información obtenida nos dirá si el programa está calificado o no para fungir como tutor. En el peor de los casos es necesario volver a la etapa de diseño, auxiliándose de la información recabada.

3. Presentación de un programa tutorial pantalla a pantalla

A continuación se muestra una corrida completa de un programa tutorial pantalla a pantalla, en este caso sobre el tema de solución de ecuaciones no lineales por el método de Newton Raphson. Para esta sección se emplearán párrafos descriptivos cuando se requiera hacer notar algún elemento importante.

La metodología que sigue el programa tutorial es:

1.- Presentar las bases teóricas del método.

2.- Realizar un ejemplo sencillo, tanto gráfica (cuando aplique), como analíticamente.

TUTORIAL PARA ECUACIONES NO LINEALES
POR EL METODO DE NEWTON RAPHSON

Este metodo encuentra una nueva aproximación a la raíz x de la ecuación :

$$f(x) = 0$$

a partir de un valor inicial x_0 y la derivada de la función: $f'(x)$ con la fórmula :

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

A continuación se mostrará un ejemplo

<Enter> Continuar <Esc> Salir

Identificación

$f(x)=2-x^2$

$f'(x)=-2*x$

Ejemplo : Sea la ecuacion

$$2-x^2 = 0$$

Usando el valor inicial x_0=0.5 y aplicando la ecuacion (1) se obtiene :

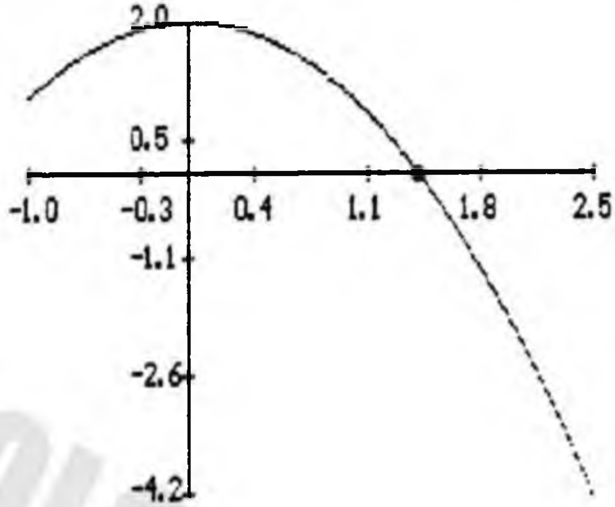
$$x_1 = 0.5 - \frac{2-(0.5)^2}{-2*(0.5)} = 2.25$$

que es la nueva aproximación a la raíz.

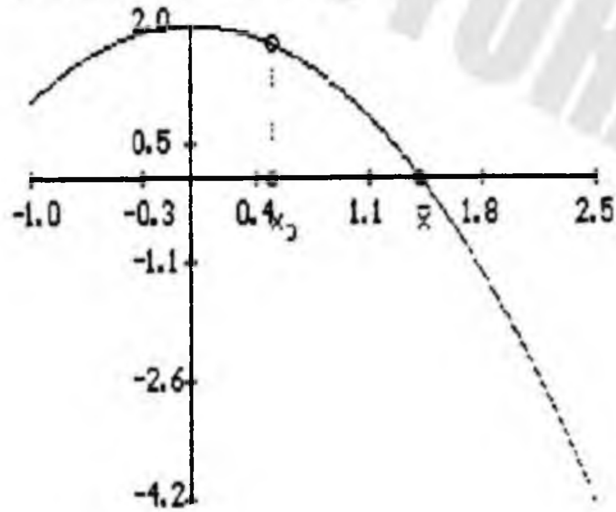
Una interpretación gráfica se muestra a continuación

<Enter> Continuar <Esc> Salir

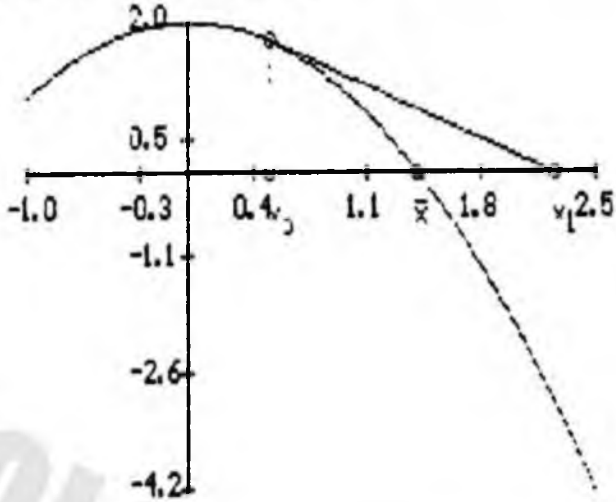
La gráfica de la funcion $f(x)=2-x^2$ en el intervalo [-1,2.5] es:



El valor inicial es $x_0=0.5$
La función evaluada en este punto es:



La interseccion de la tangente con el eje x
es la nueva aproximación x_1



<Enter> Continuar   <Esc> Salir

Usando el valor $x_1=2.25$ y aplicando la

ecuación (1) se obtiene :

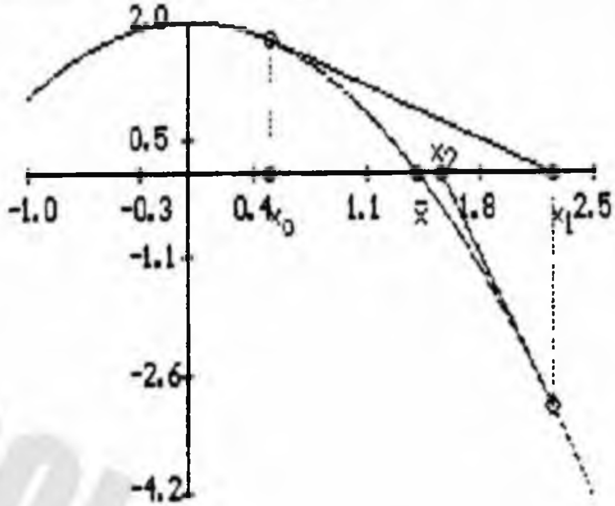
$$x_2 = 2.25 - \frac{2-(2.25)^2}{-2*(2.25)} = 1.67$$

que es la nueva aproximación a la raíz.

Una interpretación gráfica se muestra a continuación

<Enter> Continuar   <Esc> Salir

La interseccion de la tangente con el eje x
es la nueva aproximación x_2



<Enter> Continuar   <Esc> Salir

Usando el valor $x_2=1.57$ y aplicando la ecuación (1) se obtiene :

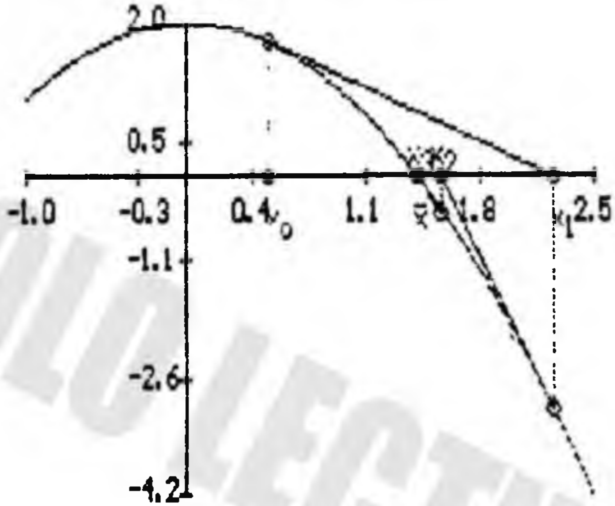
$$x_3 = 1.57 - \frac{2-(1.57)^2}{-2*(1.57)} = 1.42$$

que es la nueva aproximación a la raíz.

Una interpretación gráfica se muestra a continuación

<Enter> Continuar   <Esc> Salir

La interseccion de la tangente con el eje x es la nueva aproximación x_3 ; que es la raiz



<Enter> Continuar <Esc> Salir

IV.- Elaboración de programas interactivos

1. Características y diseño.

Una vez concluida la etapa de desarrollo de programas tutoriales, con el fin de que el usuario pueda darse cuenta del grado de conocimientos adquiridos, es necesario crear programas interactivos en los cuales la participación del usuario sea fundamental.

Un programa interactivo se caracteriza porque hace que el usuario proporcione resultados de las operaciones que se efectúan durante el seguimiento del mètodo y evalùa las respuestas proporcionadas por el alumno, asì como le niega el acceso a una parte posterior del programa sin haber cumplido satisfactoriamente con la parte actual. Y de este modo si el alumno no ha adquirido los conocimientos necesarios pueda optar por regresar a la sección tutorial. De esta forma, se logra un modelo de enseñanza-aprendizaje màs optimo.

Los programas interactivos, por tanto, deben contener procedimientos de evaluación de resultados eficientes, ya que cualquier error en el programa ocasiona que el alumno tome un pobre conocimiento del tema.

Para apoyar al usuario en la parte de operaciones aritméticas, el programa cuenta con una calculadora para este fin, y así impide que el usuario se distraiga en conseguir una. De lo anterior podemos presumir que el requisito indispensable para el usuario son GANAS DE APRENDER.

La metodología que presentan los programas interactivos es la siguiente:

1. Se presenta la(s) ecuacion(es), que se emplean para la resolución del problema específico.

2. Se marcan los pasos del método tal como se expusieron en la parte tutorial para que el usuario pueda ubicarse.

3. El programa sustituye los valores en la(s) ecuacion(es) por resolver de modo que el usuario aprenderá a realizar las sustituciones correctamente.

4. El alumno proporcionará los valores que le sean solicitados y podrá hacer uso de la calculadora para facilitarle las operaciones.

5. Si el alumno llegara a cometer algún error, el programa se lo hará saber para que intente nuevamente.

6. Se emplean cuadros de identificación como ayuda al usuario para reconocer las funciones y derivadas.

2. Presentación de un programa interactivo pantalla a pantalla.

A continuación,se ilustra paso a paso las diferentes pantallas que se presentan en el programa tutorial para la solución de sistemas de ecuaciones lineales por el método de Gauss Jordan.

INTERACTIVO PARA SISTEMAS DE ECUACIONES LINEALES
POR EL METODO DE ELIMINACION DE GAUSS-JORDAN

Hagamos ahora un ejercicio.

Se tiene el sistema:

$$5x_1 + 2x_2 + 1x_3 = 8$$

$$2x_1 + 4x_2 - 1x_3 = 5$$

$$3x_1 - 1x_2 - 4x_3 = -2$$

La matriz aumentada es entonces:

$$5 + 2 + 1 = 8$$

$$2 + 4 - 1 = 5$$

$$3 - 1 - 4 = -2$$

INTERACTIVO PARA SISTEMAS DE ECUACIONES LINEALES
POR EL METODO DE ELIMINACION DE GAUSS-JORDAN

La matriz aumentada es entonces:

$$5 + 2 + 1 = 8$$

$$2 + 4 - 1 = 5$$

$$3 - 1 - 4 = -2$$

FRIMERA ETAPA DE NORMALIZACION (Fila Pivote:1)

Dividamos ahora la primera fila por 5 :

$a_{1,1} = 1$

$a_{1,2} =$

Anota aqui tu respuesta

2/5
0.40000

<SB> Limpiar
<Esc> Salir

INTERACTIVO PARA SISTEMAS DE ECUACIONES LINEALES
POR EL METODO DE ELIMINACION DE GAUSS-JORDAN

La matriz aumentada es entonces:

$$5 + 2 + 1 = 8$$

$$2 + 4 - 1 = 5$$

$$3 - 1 - 4 = -2$$

PRIMERA ETAPA DE NORMALIZACION (Fila Pivote:1)

Dividamos ahora la primera fila por 5 :

$a_{1,1} = 1$

$a_{1,2} = 0.4$

$a_{1,3} = 0.2$

$a_{1,4} = 1.6$

<Enter> Continuar <Esc> Salir

Entonces tenemos:

$$\begin{array}{l} 1 + 0.4 + 0.2 = 1.6 \\ \boxed{2} + 4 - 1 = 5 \\ 3 - 1 - 4 = -2 \end{array}$$

PRIMERA ETAPA DE ELIMINACION Elemento Pivote: [2,1]

Ahora tomemos la primera fila multiplicada por
el negativo de 2 y sumemosle la segunda fila

$$\begin{array}{r} -2*(1 + 0.4 + 0.2 = 1.6) \\ + \quad (2 + 4 - 1 = 5) \\ \hline \end{array}$$

$a_{2,1} =$

Anota aqui tu respuesta

<F1>Calculadora <F2>Iniciar

Entonces tenemos:

$$\begin{array}{ccccccc} 1 & + & 0.4 & + & 0.2 & = & 1.6 \\ \boxed{2} & + & 4 & - & 1 & = & 5 \\ 3 & - & 1 & - & 4 & = & -2 \end{array}$$

PRIMERA ETAPA DE ELIMINACION Elemento Pivote: [2,1]

Ahora tomemos la primera fila multiplicada por el negativo de 2 y sumemosle la segunda fila

$$\begin{array}{r} -2*(1 \;+\; 0.4 \;+\; 0.2 \;=\; 1.6) \\ +\quad (2 \;+\; 4 \;-\; 1 \;=\; 5\;) \\ \hline \end{array}$$

$a_{2,1}$ = 0 0

$a_{2,2}$ = 4.8 ERROR

Anota aqui tu respuesta

<F1>Calculadora <F2>Iniciar

Entonces tenemos:

$$\begin{array}{ccccccc} 1 & + & 0.4 & + & 0.2 & = & 1.6 \\ \boxed{2} & + & 4 & - & 1 & = & 5 \\ 3 & - & 1 & - & 4 & = & -2 \end{array}$$

PRIMERA ETAPA DE ELIMINACION Elemento Pivote: [2,1]

Ahora tomemos la primera fila multiplicada por el negativo de 2 y sumemosle la segunda fila

$$\begin{array}{r} -2*(1 \;+\; 0.4 \;+\; 0.2 \;=\; 1.6) \\ +\quad (2 \;+\; 4 \;-\; 1 \;=\; 5\;) \\ \hline \end{array}$$

$a_{2,1}$ = 0 0 3.2 -1.4 1.8

$a_{2,2}$ = 3.2

$a_{2,3}$ = -1.4

$a_{2,4}$ = 1.8

<Enter> Continuar <Esc> Salir

Entonces tenemos:

$$\begin{array}{lcl} 1 + 0.4 + 0.2 & = & 1.6 \\ 0 + 3.2 - 1.4 & = & 1.8 \\ \boxed{3} - 1 - 4 & = & -2 \end{array}$$

SEGUNDA ETAPA DE ELIMINACION Elemento Pivote: [3,1]

Ahora tomemos la primera fila multiplicada por el negativo de 3 y sumemosle la tercera fila

$$\begin{array}{rl} & -3*(1 + 0.4 + 0.2 = 1.6) \\ + & (3 - 1 - 4 = -2) \\ \hline & 0 \quad -2.2 \quad -4.6 \quad -6.8 \end{array}$$

$a_{3,1} = 0$

$a_{3,2} = -2.2$

$a_{3,3} = -4.6$

$a_{3,4} = -6.8$

<Enter> Continuar <Esc> Salir

Entonces tenemos:

$$\begin{array}{lcl} 1 + 0.4 + 0.2 & = & 1.6 \\ 0 + 3.2 - 1.4 & = & 1.8 \\ 0 - 2.2 - 4.6 & = & -6.8 \end{array}$$

SEGUNDA ETAPA DE NORMALIZACION (Fila Pivote:2)

Dividamos ahora la segunda fila por 3.2:

$a_{2,1} = 0$

$a_{2,2} = 1$

$a_{2,3} = -0.4375$

$a_{2,4} = 0.5625$

<Enter> Continuar <Esc> Salir

Entonces tenemos:

1	$\boxed{+\ \ 0.4}$		+	0.2	=	1.6
0	+	1	-	0.4375	=	0.5625
0	-	2.2	-	4.6	=	-6.8

PRIMERA ETAPA DE ELIMINACION Elemento Pivote: [1,2]

Ahora tomemos la segunda fila multiplicada por
el negativo de 0.4 y sumemosle la primera fila

$$\begin{array}{rrrrrrrr} & -0.4*(0 & + 1 & - 0.4375 & = & 0.5625) \\ + & (1 & + 0.4 & + 0.2 & = & 1.6\ \ ) \\ \hline & 1 & 0 & 0.375 & & 1.375 \end{array}$$

$a_{1,1}$ = 1

$a_{1,2}$ = 0

$a_{1,3}$ = 0.375

$a_{1,4}$ = 1.375

<Enter> Continuar <Esc> Salir

Entonces tenemos:

1	+	0	+	0.375	=	1.375
0	+	1	-	0.4375	=	0.5625
0	$\boxed{-\ \ 2.2}$		-	4.6	=	-6.8

SEGUNDA ETAPA DE ELIMINACION Elemento Pivote: [3,2]

Ahora tomemos la segunda fila multiplicada por
el negativo de -2.2 y sumemosle la tercera fila

$$\begin{array}{rrrrrrrr} & 2.2*(0 & + 1 & - 0.4375 & = & 0.5625) \\ + & (0 & - 2.2 & - 4.6 & = & -6.8\ \ ) \\ \hline & 0 & 0 & -5.5625 & & -5.5625 \end{array}$$

$a_{3,1}$ = 0

$a_{3,2}$ = 0

$a_{3,3}$ = -5.5625

$a_{3,4}$ = -5.5625

<Enter> Continuar <Esc> Salir

```
Entonces tenemos:
          1   +   0      +  0.375  =  1.375
          0   +   1      -  0.4375 =  0.5625
          0   +   0      -  5.5625 = -5.5625


   TERCERA ETAPA DE NORMALIZACION (Fila Pivote:3)

   Dividamos ahora la tercera fila por -5.5625:


          a3,1  =  0

          a3,2  =  0

          a3,3  =  1

          a3,4  =  1


   <Enter> Continuar        <Esc> Salir
```

```
Entonces tenemos:
          1   +   0      + 0.375  =  1.375
          0   +   1      - 0.4375 =  0.5625
          0   +   0      + 1      =  1
PRIMERA ETAPA DE ELIMINACION Elemento Pivote: [1,3]
Ahora tomemos la tercera fila multiplicada por
el negativo de 0.375 y sumemosle la primera fila
               -0.375*(0  + 0   + 1      =  1     )
               +      (1  + 0   + 0.375  =  1.375 )
              ------------------------------------
  a1,1  =  1          1     0     0         1

  a1,2  =  0

  a1,3  =  0

  a1,4  =  1


     <Enter> Continuar        <Esc> Salir
```

```
Entonces tenemos:
            1  +  0     +  0        =  1
            0  +  1     -  0.4375   =  0.5625
            0  +  0     +  1        =  1
```

SEGUNDA ETAPA DE ELIMINACION Elemento Pivote: [2,3]

Ahora tomemos la tercera fila multiplicada por
el negativo de -0.4375 y sumemosle la segunda fila

```
            0.4375*(0  + 0   + 1        =  1      )
          +        (0  + 1   - 0.4375   =  0.5625)
          ------------------------------------------
a2,1 = 0            0    1     0           1
a2,2 = 1
a2,3 = 0
a2,4 = 1
```

<Enter> Continuar   <Esc> Salir


```
Entonces tenemos:
            1  +  0     +  0        =  1
            0  +  1     +  0        =  1
            0  +  0     +  1        =  1
```

Luego, la solucion del sistema propuesto es:

$X_1 = 1$

$X_2 = 1$

$X_3 = 1$

<Enter> Continuar   <Esc> Salir

V.- Elaboración de Programas de Solución Directa

1. Características y diseño.

Esta parte está dedicada a investigadores e ingenieros que en su rutina diaria tengan necesidad de resolver ecuaciones no lineales, sistemas de ecuaciones lineales y no lineales y no dispongan de mucho tiempo para poder resolverlas. La labor del usuario solo consiste en capturar la o las funciones a resolver y la labor del programa es proporcionar los resultados en un tiempo muy corto y con gran exactitud.

Un programa de solución directa debe de contar con un analizador de expresiones eficaz, el cual debe presentar las siguientes cualidades:

1.- Identificar los posibles errores de sintáxis que ocurran durante la captura de la información. Los principales errores de sintáxis ocurren cuando tecleamos dos operandos sin operador, al olvidar cerrar o abrir algún paréntesis, al poner dos operandos juntos.

2.- Prohibir el empleo de teclas que no sean necesarias para la captura de información, esto es: si en alguna parte del programa el usuario tiene que introducir algún valor numérico, se debe invalidar el acceso de cualquier otra tecla.

3.- Identificar los posibles errores matemáticos que surjan al dar valor a las variables. Los principales errores matemáticos que puden ocurrir son: la división entre cero y la raiz cuadrada de un número negativo

4.- Al capturar la expresión debe permitir al usuario poder corregir los errores que cometa, esto se hace presionando la tecla de retroceso y permitiendo corregir el error.

5.- Permitir el uso de asignaciones, se pueden realizar asignaciones, o sea, proporcionar valores a variables para después emplearlos en operaciones posteriores del modo siguiente:

Asignando:

A=25
2.50000

Trabajando con la variable:

3*A
7.50000

Un programa de solución directa para un método numérico culquiera debe realizar lo siguiente:

a. Pedir al usuario la(s) funcion(es), esto es, si vamos a resolver una ecuación no lineal por el método de Newton Raphson, el usuario tendrá que proporcionar una ecuación no lineal en función de una sola variable, de la forma:

$$f(x)=0$$

b. Pedir al usuario la derivada de cada función, esto es, si estamos resolviendo una ecuació no lineal por el método de Newton Raphson, tendremos que proporcionar la derivada.

c. Evaluar con el criterio de exactitud requerido, esto depende de la raíz de la variable que estemos buscando, por ejemplo: si estamos buscando una presión de vapor en psia tendremos que emplear un criterio de exactitud de $\xi=0.0001$, pero si estamos buscando una temperatura en grados centígrados bastará con u criterio de exactitud de $\xi=0.1$. La fórmula para evaluar el criterio de exactitud que utiliza el programa es la siguiente:

$$|x_n - x_{n-1}| \leq \xi$$

d. Pedir el vector de valores iniciales. Para algunos métodos, es importante seleccionar un buen valor inicial, ya que esto, ayuda a encontrar o no con mayor rapidez la solución.

e. Proporcionar los valores de las raíces cuando existan.

El programa de solución directa debe contar con subrutinas para analizar si con los valores iniciales y el criterio de exactitud requerido se puede llegar a la solución. De modo contrario deberá emitir un mensaje el cual explique porqué no se encontró la solución, esto es:

Si el valor inicial fué mal elegido.
Si llegado a un número determinado de iteraciones no converge el método.
Si existen errores matemáticos (indeterminaciones).

Como puede observarse la lógica del programa de solución directa es muy sencilla y casi no se presentan pantallas descriptivas, pero exige de la habilidad de un buen programador.

Con este tipo de programas el usuario que haya concluido con las etapas tutorial e interactiva podrá comprobar los resultados obtenidos en éstas.

2. Presentación de un programa de solución directa pantalla a pantalla.

A continuación se muestra una corrida pantalla a pantalla del programa de solución directa de ecuaciones no lineales por el método de Newton Raphson.

SOLUCION DIRECTA PARA ECUACIONES NO LINEALES
POR EL METODO DE NEWTON RAPHSON

Introduce la Funcion:

<F1> Iniciar <Esc> Salir

SOLUCION DIRECTA PARA ECUACIONES NO LINEALES
POR EL METODO DE NEWTON RAPHSON

Introduce la Funcion:

2*X+10*E-X
Dos o mas variables

<SB> Nuevo intento <Esc> Salir

```
SOLUCION DIRECTA PARA ECUACIONES NO LINEALES
      POR EL METODO DE NEWTON RAPHSON

Para la funcion:
  X*X*X+2*X*X+10*X-20
 Introduce la Derivada:





 <F1> Iniciar             <Esc> Salir
```

```
SOLUCION DIRECTA PARA ECUACIONES NO LINEALES
      POR EL METODO DE NEWTON RAPHSON

Para la funcion:
  X*X*X+2*X*X+10*X-20
 Introduce la Derivada:

3*X+5

<SB> Nuevo intento    <Esc> Salir
```

```
SOLUCION DIRECTA PARA ECUACIONES NO LINEALES
POR EL METODO DE NEWTON RAPHSON

Para la funcion:
  X*X*X+2*X*X+10*X-20


La derivada es:
  3*X*X-1[illegible]

Valor inicial Xo  =  1

Con una Tolerancia = 0.0001


     Error en la derivada o raiz imaginaria
 <F1> Nuevo Intento      <Esc> Salir
```



```
SOLUCION DIRECTA PARA ECUACIONES NO LINEALES
POR EL METODO DE NEWTON RAPHSON

Para la funcion:
  X*X*X+2*X*X+10*X-20


La derivada es:
  3*X*X+4*X+10

Valor inicial Xo  =  1

Con una Tolerancia = 0.0001

   La raiz  1.36881

     <Esc> Salir
```

Conclusiones

Consideramos que el presente trabajo será de gran utilidad para el aprendizaje de los métodos numéricos. Cada sección ha sido valuada y sometida a pruebas para lograr este objetivo.

Los fundamentos sobre métodos numéricos expuestos han sido cuidadosamente revisados por la Academia de Matemáticas y reconocidos autores en la materia.

Este programa tiene un alto grado de originalidad y llevó años poder concebirlo. Las críticas serán de gran ayuda para el mejoramiento del mismo.

El programa actualmente cuenta para la sección de solución de solución directa con el método de Newton Raphson. y se espera se integren algunos otros métodos.

En el semestre 1993-1993 el programa será sometido a prueba por estudiantes de los cursos de matemáticas de 4o. y 5o. semestre de la carrera de Ingeniero Químico Industrial.

Se busca que con este tipo de trabajos se motive a los alumnos a crear otros nuevos programas que les ayuden en las distintas ramas del conocimiento.

Bibliografía

Computer Science (A First Course)
Forsythe, A. I.
Keenan, T. A.
John Wiley and Sons, Inc.
Second Edition

Numerical Analysis
Burden, Richard L.
Faires, J. Douglas
PWS Boston
Third Edition

Advanced Turbo Pascal: Programming and Techniques
Schildt, Herbert
Mc Graw Hill, Inc.
First Edition

Métodos Numéricos para Ingenieros
Steven C. Chapra
Mc Graw Hill.
Primera Edición

Anexo

Manual del Usuario

Características y requerimiento del programa.

1. Características.

Este programa está elaborado en lenguaje de programación Turbo Pascal. versión 5.5. Todo el programa está hecho en modo gráfico para tarjeta de video MCGA.

Consta de una rutina que le permite correr con cualquier tipo de tarjeta de video a excepción de la tarjeta Hércules.

Fué elaborado para trabajar en monitores a color, pero puede correr en monitores monocromaticos. Aunque para poder apreciarlo en toda su magnitud es mejor correrlo a color.

Este programa se elaboró en una IBM PS/2 modelo 25 y ya que el I.P.N. cuenta con este tipo de computadoras no se tendrán problemas al correrlo.

Cuenta con secuencias de escape, las cuales se presentan en las etapas que consideramos se requiere y para su buen funcionamiento deben de respetarse.

Cuenta con un analizador de expresiones recursivo, un graficador de alta resolución y una calculadora integrada.

2. Requerimientos

Se requiere como mínimo 350 Kbytes en memoria RAM, ya que el programa consta de 25 archivos que ocupan 344 Kbytes.

Puede correr en cualquier tipo de tarjeta de video a excepció de la tarjeta Hércules.

Excelente para IBM PS/2 modelo 25 monitor a color, pero corre en cualquier PC compatible, ya sea monocromática o a colores.

3. Organización

El programa ejecutable se llama TMN.EXE y cuenta con bibliotecas de procedimientos, así como drivers para las distintas tarjetas de video y fonts para distintos tipos de letra.

Cuando se ejecuta el programa TMN hace uso de las bibliotecas de procedimientos de acuerdo a la opción del usuario.

4. Procedimiento para ejcutar el programa.

1.- Todos los archivos que utiliza el programa están en el directorio raíz del diskette que se le proporciona, así que para poder ejecutarlo solo basta teclear en la unidad A:> lo siguiente:

A:>TMN

y presionar la tecla <Return>.

2.- Si se quiere ejecutar desde un disco duro hay que crear un subdirectorio con el nombre de METODOS en la unidad C:> y realizar las siguientes instrucciones:

C:>MD METODOS <Return>

C:>CD METODOS <Return>

C:>COPY A:*.* <Return>

C:>TMN

A continuación se muestran las pantallas de entrada al programa, en los capítulos posteriores se ilustra como trabajan los métodos.

Nota. Es importante que se sigan las instrucciones al pie de la letra de los mensajes que aparecen durante su ejecución, ya que de modo contrario el programa continuará con su ejecución sin el correcto seguimiento del usuario.

Cuando se emplee la calculadora integrada y se empleen números decimales se debe anteceder el 0 (cero) al punto decimal.

5.- Instrucciones para emplear la calculadora durante los programas interactivos:

a). Cuando el usuario tenga la necesidad de realizar operaciones matemáticas en alguna parte del programa aparecerá un mensaje como se muestra a continuación:

<F1> Calculadora **<F2> Iniciar**

<F1> Calculadora Quiere decir que presionando la tecla F1 se tendrá acceso a la calculadora.

<F2> Iniciar Quiere decir que presionando la tecla F2 se podrá introducir el resultado solicitado por el programa.

b) Al presionar F1 aparece un recuadro que simula un display de una calculadora y aparecen los siguientes mensajes:

	<SB> Limpiar **<Esc> Salir**

<SB> Limpiar Quiere decir que presionando la barra espaciadora se limpia la pantalla y cada vez que hagamos una operación y la calculadora proporcione el resultado se deberá limpiar la pantalla antes de efectuar la siguiente operación, de modo contrario se encimará la nueva operación con la actual.

<Esc> Salir Quiere decir que presionando la tecla Esc podremos salir de la calculadora y continuar en el programa.

c) Se pueden realizar asignaciones, o sea, proporcionar valores a variables, para después emplear estas en operaciones posteriores, para entender esto veamos el siguiente ejemplo:

Asignando:

A=2.5 2.50000	**<SB> Limpiar** **<Esc> Salir**

Trabajando con la variable:

3*A 7 50000	**<SB> Limpiar** **<Esc> Salir**

d) Todos los puntos anteriores, a excepción del punto c), también aplican para la captura de funciones y derivadas para el programa de solución directa, con la restricción de que se debe de trabajar con la variable X para ambos casos.

e) El paquete cuenta en la sección de programas tutoriales e interactivos con los siguientes métodos:

- Newton Raphson, para la solución de ecuaciones no lineales.
- Eliminación de Gauss-Jordan, para la solución de sistemas de ecuaciones lineales.
- Newton Raphson, para la solución de sistemas de ecuaciones no lineales.

En la sección de programas de solución directa :

- Newton Raphson, para la solución de ecuaciones no lineales.

A continuación se muestran las pantallas del menú principal y de los submenús para la elección del tipo de problema y método:

TUTORIAL EN METODOS NUMERICOS

1.-Ecuaciones No Lineales

2.-Sistemas de Ecuaciones Lineales

3.-Sistemas de Ecuaciones No Lineales

4.-Salida

Selecciona tu opcion : 1

ECUACIONES NO LINEALES

1.-Tutorial

2.-Interactivo

3.-Solucion Directa

4.-Menu Principal

Selecciona tu opcion : 4

SISTEMAS DE ECUACIONES LINEALES

1.-Tutorial

2.-Interactivo

3.-Menu Principal

Selecciona tu opcion :   3

SISTEMAS DE ECUACIONES NO LINEALES

1.-Tutorial

2.-Interactivo

3.-Menu Principal

Selecciona tu opcion :   3