UNIVERSIDAD DE MONTERREY

DIVISION DE CIENCIAS NATURALES
Y EXACTAS

UNIVERSIDAD
DE MONTERREY

900693

BIBLIOTECA DE RUTINAS
PARA FORTRAN

REPORTE DEL PROGRAMA DE
EVALUACION FINAL
PRESENTADO POR

LOURDES MARIA MOLINA NORIEGA

EN OPCION AL TITULO DE:
INGENIERO EN COMPUTACION
ADMINISTRATIVA Y DE PRODUCCION

MONTERREY, N. L., DICIEMBRE DE 1986.

BIBLIOTECA
UNIVERSIDAD DE MONTERREY

A DIOS, MI SEÑOR, por permitirme concluir una de mis más grandes metas, llegar a ser profesionista. Y por llenarme siempre con su Amor, poniendo en mi camino lo que necesito.

A MIS PAPAS, JORGE Y ELBA, con todo mi amor y respeto por todo lo que me han brindado siempre, por hacer de mí lo que ahora soy, y por ser como son. LOS QUIERO MUCHO.

A MIS HERMANOS, Elba, Connie, July, Maye, Jorge y Yola, por soportarme y apoyarme durante el desarrollo de mi proyecto y siempre.

A MI ASESOR, Ing. Daniel Contreras, por su apoyo, confianza y ayuda brin-- dada para lograr la rea- lización de éste proyecto.

A mis amigos, compañeros, que de una manera desin- teresada me ofrecieron su amistad durante mi vida de estudiante.

. . . A TODOS, ¡MUCHAS GRACIAS!

QUE DIOS LOS BENDIGA.

I N D I C E

	PAGINA
INTRODUCCION.....	1
OBJETIVO.....	3
ALCANCES.....	5
BENEFICIOS.....	7
LIMITACIONES.....	8
DESCRIPCION GENERAL DEL PROYECTO.....	10
DESCRIPCION DETALLADA DE LOS METODOS UTILIZADOS PARA EL DESARROLLO DE LAS RUTINAS.....	13
- Multiplicación de Matrices.....	13
- Inversa de una Matriz.....	15
- Determánte de una Matriz.....	17
- Números al azar.....	19
- Ruta Crítica.....	20
- Graficador de barras.....	24
- Graficador de puntos.....	25
- Derivadas.....	26
- Raíces de una Ecuación.....	29
- Solución de Ecuaciones Simultáneas.....	31
- Factorial de un número.....	33
METODO DE ACCESO.....	34
SISTEMAS DE AYUDA.....	36

MANUAL DE USUARIO..... 38

- Cómo usar cualquier rutina..... 39

- Cómo usar el sistema de ayuda..... 40

- Alcances y limitaciones de cada rutina..... 41

MANUAL DEL ANALISTA..... 50

- Estructura General..... 51

- Diseño de rutinas..... 52

- Diagramas de flujo..... 53

- Listados de programas fuentes..... 72

- Procedimientos de Actualización..... 89

CONCLUSIONES..... 97

BIBLIOGRAFIA..... 99

ANEXO 1 - Listado del sistema de ayuda.

INTRODUCCION

Una biblioteca es un archivo que consiste de un conjunto de uno o más modulos objeto; es un grupo de rutinas estándares. La mayoría de las veces se encuentra almacenada en un código simbólico o en programas compilados y subdivididos en varios volúmenes de acuerdo al tipo de técnicas aritméticas o computacionales usadas.

El presente proyecto muestra el desarrollo de una biblioteca de rutinas generales para Fortran, que permitirá al usuario, en el momento de realizar algún programa o sistema que involucre cualquier rutina incluida, obtener resultados más fácil y rápidamente, ya que no tendrá necesidad de desarrollar los procedimientos, sino sólo --

llamar a la rutina correspondiente.

Como parte de proyecto se presenta un sistema de ayuda, un catálogo de rutinas, manual del usuario y del analista.

Su desarrollo se realizó en el equipo de cómputo Sistema Eclipse MV 4000, que es con el que cuenta la Universidad de Monterrey, utilizando el lenguaje FORTRAN 77.

OBJETIVOS

* Objetivo General.-

El propósito de éste proyecto es implantar y mantener una biblioteca de las rutinas específicas más usadas, para usuarios del lenguaje de alto nivel Fortran 77.

* Objetivos Particulares.-

- Permitir al usuario los accesos a la biblioteca y el uso de sus rutinas.

- Sustentar un directorio ó catálogo de la biblioteca.

- Permitir al usuario el acceso, solamente de lec--

tura, a las rutinas fuentes.

- Diseñar un sistema de ayuda, conocido en el lenguaje computacional como Help, para orientar el manejo de las rutinas.

- Diseñar un procedimiento de actualización y mantenimiento de la biblioteca de rutinas.

ALCANCES

* Las rutinas desarrolladas para formar la biblioteca son:

  - Multiplicación de dos matrices.
  - La inversa de una matriz de N x N [1]
  - Determinante de una matriz de N x N.
  - Números de azar.
  - Camino crítico.
  - Graficador de barras.
  - Graficador de puntos.
  - Derivadas.
  - Raíces de una ecuación polinomial.
  - Solución de Ecuaciones Simultáneas.
  - Obtener el factorial de un número.

* Desarrollo de forma de acceso a la biblioteca.
* Desarrollo de el procedimiento de actualización considerando un catálogo de rutinas.
* Desarrollo de un sistema de ayudas para la biblioteca y sus rutinas.
* El proyecto está abierto a la implantación de nuevas rutinas.

1.- La notación N x N, significa la dimensión de la matriz, N regiones, y N Columnas, es decir, una matriz cuadrada.

BENEFICIOS

- Reducir el tiempo necesitado para codificar algún programa que involucre los procedimientos seguidos por cualquier rutina incluída en la biblioteca, ya que el programador - usuario no tendrá que desarrollar su procedimiento, solamente llamar a la rutina correspondiente.

- Minimizar la redundancia de instrucciones cuando la misma rutina es necesitada varias veces en el programa que se está realizando.

- Aumentar el uso de la computadora para la resolución de problemas que involucren métodos numéricos, ya que las rutinas desarrolladas, mencionadas en el punto anterior, tienen como tema común la aplicación de estos métodos.

LIMITACIONES

- Para la implantación del presente proyecto, se tomarán en cuenta las Limitaciones que existen a causa de la máquina y el lenguaje utilizado.

- El usuario deberá cuidar que los datos solicitados por cualquier rutina que se utilice, sean los correctos, ya que sólo en algunas de ellas es posible validar algunos de los datos.

- La biblioteca está limitada a un número específico de rutinas, el cual puede agrandarse con la implantación de nuevas rutinas.

- El usuario tendrá acceso a las rutinas fuentes sólo para leerlas; no podrá escribir en ellas.

DESCRIPCION GENERAL DEL PROYECTO:

El objetivo general del proyecto es implantar una biblioteca de rutinas para Fortran 77. Por lo que se han -- desarrollado un número específico de ellas; todas éstas rutinas tienen como tema común la solución de problemas relacionados con la aplicación de métodos numéricos.

El nombre de cada rutina comienza con la letra " F ", letra que hace referencia al lenguaje para el cual han sido desarrolladas:

FMULMAT: Esta rutina obtiene la multiplicación de dos matrices.

FINVMAT: Esta rutina obtiene la matriz inversa de una

matriz de N x N.

FDETMAT: Esta rutina obtiene el determinante de una matriz de N x N.

FNUMAZA: Esta rutina obtiene un número al azar.

FRUTCRI: Esta rutina obtiene la ruta critica y la duración de un proyecto.

FDERIVA: Esta rutina obtiene la derivada numérica de un función con respecto a X.

FRAICES: Esta rutina obtiene las raíces de una ecuación polinomial de grado N.

FECUSIM: Esta rutina obtiene la solución de N e-- cuaciones con N incógnitas, lo que conoc- mos como ecuaciones simultáneas.

FFACTOR: Esta rutina obtiene el factor de un núme- ro N.

FGRAPUN: Esta rutina grafica los puntos de una e- cuación polinomial de grado N.

FGRABAR: Esta rutina grafica barras a partir de la tabulación de alguna estadística.

Para el uso correcto de la biblioteca es necesario desarrollar los puntos importantes que se deben conocer:

- El método de acceso a la biblioteca; contiene los métodos existentes, con sus ventajas y desventa--jas, indicando cual es el más conveniente y ade--cuado para usar.

- El desarrollo de un sistema de ayudas, tanto para el uso de la biblioteca y sus rutinas, como de los programas fuente de las mismas, ayudando de ésta manera al usuario a obtener los resultados deseados.

- El desarrollo de un manual para el usuario, que contiene la forma de usar cada rutina, así como los alcances y limitaciones de cada una de ellas. También la manera de llamar al sistema de ayuda.

- Y por último el desarrollo del manual de sistemas, que contiene la estructura y el procedimiento de actualización y mantenimiento de la biblioteca; así como el diagrama y listado de los programas.

DESCRIPCION DETALLADA DE LOS METODOS UTILIZADOS PARA EL DESARROLLO DE LAS RUTINAS.

En éste punto se dará una breve explicación de los criterios usados para construir cada rutina, y los pasos a seguir para programarlas.

Multiplicación de Matrices:

Una matriz es un arreglo que se forma con los coeficientes de un sistema de ecuaciones lineales.

Las operaciones básicas que se pueden realizar con matrices son; la suma, la multiplicación de un matriz por un número real y la multiplicación de una matriz con otra.

En éste proyecto se desarrolla la última de éstas operaciones: multiplicar dos matrices.

El producto R= A x B de dos matrices A y B es definido, si el número de columnas de A es igual al Número de renglones de B.

El producto de A (IxJ)* y B (JxK) es la matriz C (IxK) con elementos Cik definidos por:

$$Cik = \sum_{j=1}^{J} aij\ bjk$$

Esta fórmula se utiliza para programar la rutina.

* La notación A (IxJ) denota una matriz A de I renglones y J columnas.

Inversa de una matriz cuadrada:

Sea A una matriz cuadrada (NxN) no singular, es decir, que su determinante sea diferente de cero, por definición de matriz inversa, se tiene que A^{-1} es la inversa de A, si

$$A.\ A^{-1} = I,$$ donde I es la matriz identidad.

Es posible determinar la inversa de una matriz con el método de Gauss - Jordan de eliminación completa. Para lograrlo, bastará con aplicar las operaciones elementales sobre los renglones de la matriz ampliada (A,I) de manera de transformar A en I. Cuando se haya hecho, se obtendrá la matriz ampliada (I,A^{-1}), con lo que se tendrá la inversa buscada.

El procedimiento del método de eliminación está compuesto por N estados de computación, cada estado de 2 pasos; el k ésimo estado (k=1,2,...N) del proceso de eliminación es:

Paso 1.- Normalización del elemento k, k multiplicando el renglón k por el recíproco del elemento k, k si éste existe, si el elemento k, k es cero, entonces su recíproco

no está definido, y el renglón k deberá ser cambiado con algún renglón, en el cual el elemento i, k no sea cero.

Paso 2.- Hacer cero los elementos que se encuentran bajo la diagonal de la columna k reemplazando el renglón i (i=K) por la combinación lineal situada del renglón i y el renglón k.

Determinante de una matriz Cuadrada:

Toda matriz cuadrada A(NxN) tiene una cantidad escalar asociada llamada determinante, la cual se denota por $|A|$.

Cuando la matriz A es de 2 x 2, la forma de obtener el determinante es mediante la fórmula.

$$|A| = a_{11} a_{22} - a_{21} a_{12}.$$

Para calcular determinantes de orden N, se utiliza el método de Laplace, el cual se define con el siguiente teorema:

Teorema:

Si A es una matriz de n x n,entonces se cumple lo siguinete:

1) Si A' es la matriz que se obtiene de intercambiar dos renglones de A, entonces $|A'| = -|A|$

2) Si A' es la matriz que se obtiene al sumar un múltiplo de un renglón a otro, entonces $|A'| = |A|$

3) Si A' es la matriz que se obtiene al multiplicar un renglón de A por una constante $k \neq o$, entonces

$$|A'| = k\,|A|$$

BIBLIOTECA
UNIVERSIDAD DE MONTERREY

Este método es utilizado para programar esta rutina.

Números al azar:

En cualquier experimento de simulación o de muestreo, existe la necesidad de un mecanismo que proporcione números al azar. La forma de obtener éstos números puede ser muy variada y entre los primeros mecanismos empleados se tenían las ruletas, dados, etc.

Recientemente se han desarrollado algunas nuevas técnicas para obtener numeros al azar, entre los que destacan las tablas y las computadoras.

El método utilizado es por medio del uso de operaciones básicas con un número o semilla dado y números claves que permiten la obtención de estos numeros al azar.

Ruta Crítica:

La planeación de un proyecto se puede dividir en 3 etapas:

1.- Análisis de las actividades: se definen las actividades del proyecto.

2.- Construcción del diagrama de flechas: Se basa en la lista de actividades, la cual deberá estar completa y verificada. Este diagrama es una representación esquemática del conjunto de tareas que constituyen un proyecto.

3.- Numeración de los eventos: La red se numerará de acuerdo a las siguientes reglas:

- Hay que asignar un número entero positivo diferente a cada evento. Este hace referencia a las actividades, las que se designan con la letra i del evento origen y con j del evento destino.

- Para roda pareja (i,j) asociada a una actividad el número j debe ser mayor que i.

- Hay que agregar actividades ficticias cuando la

misma pareja de números identifique dos o más actividades.

Hacer que el valor de j sea mayor que el de i, per-- mite detectar si dentro del diagrama existe alguna secuencia cerrada de actividades.

Habiendo analizando lo anterior, se define la ruta crítica como la secuencia de actividades que establecen el tiempo mínimo en que se puede concluir un proyecto. Por lo tanto, es necesario conocer el tiempo que consume cada actividad.

El algoritmo de Ford - Fulkerson es el utilizado para el desarrollo de la rutina en cuestión.

Se determinan los tiempos de iniciación más próximos y los de terminación más lejanos de cada actividad, y a partir de estos valores se calcula la holgura total.

Este algoritmo se divide en dos frases:

1.- Se definen los tiempos más próximos de iniciación TPi igual a cero y después, junto con las duraciones ($d_{i,j}$) de cada actividad que parte del evento i y llega al evento j, se procede con la forma siguiente:

Si $d_{i,j} > TPj - TPi$

Se reemplaza el valor anterior de TPj por

$$TPj = Tpi + d_{i,j}$$

y si

$$d_{i,j} \leq TPj - TPi$$

Se conserva el valor anterior de TPj.

La lista total de las actividades se recurre aplicando las ec. anteriores todas las veces que sea necesario hasta que ningún TPi sea modificado.

Si después de un número de recorridos igual al número de actividades se sigue modificando algún TPi, es señal de que la red contiene cuando menos una cadena de actividades cerrada.

En la segunda etapa del procedimiento se definen Tli igual al valor máximo de los TPi para toda i y se sigue un proceso idéntico al de la primera etapa, aplicando:

$$d_{i,j} > TLj - TL$$

Se sustituye el valor de TL por:

$$TLi = TLj - d_{i,j}$$

y si

$$d_{i,j} \leq TLj - TLi$$

Se conserva el valor anterior de TLi

Con la información obtenida se calculan los tiempos de terminación más próximos, los de iniciación más lejanos y las holguras totales por medio de:

$$TTP_{i,j} = TPi + d_{i,j}$$

$$TIL_{i,j} = TLj - d_{i,j}$$

$$HT_{i,j} = TLj - (TPi + d_{i,j})$$

Cuando la holgura total de una actividad sea cero, entonces esa actividad es crítica y forma parte de la ruta.

Graficador de barras:

El graficador de barras es la simulación de un histograma, ya que se grafica el comportamiento de algún producto o situación durante un período de tiempo dado, con la diferencia de que en lugar de ser representado por cuadros es representado por barras.

Es necesario tener como información el número de períodos y los valores de aumento o disminución de cada período. Al igual que el graficador de puntos, si el tanaño de estos valores es mayor al permitido por el de la pantalla se hace escala, de diez, tantas veces sea nesesario.

Graficador de puntos:

Dada una función polinomial, se obtiene su gráfica por medio de puntos, que proporciona visualmente un diagrama del comportamiento de esa función dada de una variable.

Se establece que la variable 'y' depende de la variable independiente 'X'. Esto significa que para cada valor asignado a X, pueden ser determinados los valores correspondientes a Y. Cada par de valores correspondientesde X y Y satisfacen a la función dada. Tomando cada uno de estos pares de valores como coordenadas (x,y) de un punto en un sistema de coordenadas rectangulares. Este conjunto de puntos se llama gráfica. Si los valores que toma Y son mayores a los permitidos por el tamaño de la pantalla, se hace una escala de 10 multiplicado por el número de veces que se necesite reducir el tamaño.

Derivadas:

Se denomina ecuación diferencial a toda ecuación en la que aparecen una ovarias derivadas de una función. Se entiende por solución de una ecuación diferencial toda relación entre las variables dependiente e independiente en la que no aparece ninguna derivada y que sustituída en la ecuación dada la reduce a una identidad.

Para lograr esto, se determina algún procedimiento mediante el cuál se pueda encontrar una solución aproximada a la requerida con un grado de exactitud satisfactorio.

En general tienen que resolverse por métodos aproximados numéricos, entre los más importantes se encuentra el método de diferencias finitas.

Suponiendo que se tiene una función y=f(x), dada en forma tabular por una sucesión de valores de X; dados con un intérvalo constante de separación:

X	F(X)
X_0	$F(X_0)$
X_1	$F(X_1)$
X_2	$F(X_2)$
.	.
:	:

entonces las primeras diferencias de f se definen como:

$$\Delta f_k = f_{k+1} - f_k \text{ ,donde } k=0,\ 1,\ 2,\ldots$$

en general, las diferencias de orden superior se definen por:

$$\Delta^n f_k = \Delta(\Delta^{n-1} f_{k+1}) = \Delta^{n-1} f_{k+1} - \Delta^{n-1} f_k \quad \text{donde } n=$$

valor entero positivo.

Una de las aplicaciones más importantes de las diferencias finitas es el problema de la interpolación, como la función es dada en forma tabular con un intérvalo constante entre los sucesivos valores de X, la fórmula de interploración utilizada es:

$$f(x)= f(x_o+ rh)$$

$$f(x_o+ rh)=f_o+ r\Delta f_o + \frac{r(r-1)}{2!}\Delta^2 f + \frac{r(r-1)\ (r-2)}{3!}\Delta^3 f + \ldots$$

que se conoce como fórmula de interpolación progresiva de Gregory - Newton.

Esta fórmula se utiliza para calcular la derivada de una función tabular, derivándola con relación a r, entonces encontramos:

$$hf'(x_o+ rh)=\Delta f_o + \frac{2r-1}{2}\Delta^2 f_o + \frac{3r^2 -6r +2}{6}\Delta^3 f_o$$

$$+ \frac{2r^3 - 9r^2 + ilr - 3}{12} \Delta^4 f_o + \ldots$$

$$h^2 f''(x_o + rh) = \Delta^2 f_o + (r-1)\Delta^3 f_o + \frac{6r^2 - 18r + 11}{12} \Delta^4 f_o + \ldots$$

$$h^3 f''(x_o + rh) = \Delta^3 f_o + \frac{2r - 3}{2} \Delta^4 f_o + \ldots$$

Si se hace r=o, entonces; se obtiene:

$$f'(x_o) = \frac{1}{h} \left(\Delta f_o - \frac{1}{2}\Delta^2 f_o + \frac{1}{3}\Delta^3 f_o - \frac{1}{4}\Delta^4 f_o + \frac{1}{5}\Delta^5 f_o\right)$$

$$f''(x_o) = \frac{1}{h^2} \left(\Delta^2 f_o - \Delta^3 f + \frac{11}{12}\Delta^4 f_o - \frac{5}{6}\Delta^5 f_o\right)$$

$$f''(x_o) = \frac{1}{h^3} \left(\Delta^3 f_o - \frac{3}{2}\Delta^4 f_o + \frac{87}{60}\Delta^5 f_o\right)$$

Estas son las fórmulas utilizadas para encontrar la primera, segunda y la tercera derivada numérica de una función tabular.

Hay que tomar en cuenta que la exactitud de los resultados depende del tipo de función que sea, ya que si es del tipo polinomial, entre mayor sea el grado de la ecuación, menor será la exactitud.

Raíces de una ecuación:

La determinación de las raíces de una ecuación es uno de los problemas más antiguos de las matemáticas que se presenta con frecuencia. Para resolverlos. existe una gran cantidad de métodos que se diferencían por la necesidad de obtener todas las raíces de una ecuación o sólo algunas, por la determinación de las raíces reales o complejas, simples o múltiples, etc.

En el caso más general se tendrá una función f(x) y se tratará de encontrar los valores de X para los que f (x) = 0; por tratarse de resolver por medio de una computadora se usará el método de solución aproximada que consta de:

1.- La determinación de un valor aproximado a la raiz que se busca y,

2.- Mejoramiento de la solución hasta un grado de precisón preestablecido.

Para la obtención de las raíces es necesario saber que:

Toda ecuación algebraíca de grado n admite n raíces reales o complejas.

El método utilizado para encontrar las raíces de una

ecuación es el de Newton - Raphson, obteniendo todas las raíces, ya sean reales o complejas, aplicando la división sintética.

El procedimiento a seguir es:

Teniendo el grado n de la ecuación y los coefi -- cientes de la misma almacenados en un vector A. Se procederá a realizar la divisón sintética, aplicando:

$$b_1 = a_1$$

$$bi = ai + xbi-1 \quad , \quad i = 2,3,4,...n + 1$$

B(I), donde, I=1,...n, contendrá los coeficientes del polinomio cociente, B(M), donde M=n+1, será el residuo o sea, el valor del polinomio buscado.

Solución de Ecuaciones Simultáneas:

Una ecuación algebraíca lineal es aquella en donde en cada término la ecuación aparece únicamente una variable o incógnita elevada a la primera potencia.

Un sistema de ecuaciones es un conjunto de ecuaciones algebraícas que deben resolverse simultáneamente, de la forma:

$$a_{11}x_1 + a_{12}X_2 + a_{13}X_3 + \dots + a_{1n}X_n = b_1$$

$$a_{21}X_1 + a_{22}X_2 + a_{23}X_3 + \dots + a_{2n}X_n = b_2$$

$$\vdots$$

$$a_{n1}X_1 + a_{n2}X_2 + a_{n3}X_3 + \dots + a_{nn}X_n = b_n$$

Aplicando la definición de productos entre matrices, este sistema de n ecuaciones con n incógnitas puede escribirse en forma matricial.

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n} \\ \vdots & & & & \\ a_{n1} & a_{n2} & a_{n3} & \dots & a_{n} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

esto se escribe simbólicamente como: $A\,x = b$

La solución de un sistema de ecuaciones es un conjunto de valores de las incógnitas que verifican simultáneamente a todos y cada una de las ecuaciones del sistema.

De acuerdo con su solución el sistema puede ser compatible determinado, si la solución es única; o indeterminado, si la solución no es única.

Existen dos métodos para solucionar un sistema de ecuaciones lineales:

1.- Método de eliminación completo: Consiste en ir sustituyendo ecuaciones del sistema porcombinaciones lineales del mismo en las que se hayan eliminando incógnitas, hasta que en el caso más general, que de una sola incógnita en cada ecuación. Con esto podrá despejarse el valor de una incógnita de cada ecuación y con ello obtener la solución del sistema.

2.- Método de Gauss - Jordan: Puede decirse que es un método más exacto; para aplicar éste método deben estar ordenados, es decir, en una primera columna los coeficientes de la primera incógnita, en la segunda, losde la segunda, y así sucesiva-

mente, también los términos independientes deben estar ordenados, quitando las incógnitas de los coeficientes queda una matriz. Se tratará de convertir en ceros los elementos de la matriz representativos de los coeficientes del sistema, dejando la diagonal en unos, sin tomar en cuenta la columna de los términos independientes. De ésta manera se procede a interpretar los renglones de la última matriz obtenida, con lo que se obtendrá la solución del sistema.

Esta rutina utiliza el segundo método, ya que es más exacto y tiene mejor aplicación en la computadora.

Factorial de un número:

Factorial es el producto de los términos de una progresión aritmética.

Dado un número, se obtendrá su factorial, con ésta rutina.

METODOS DE ACCESO

Existen dos métodos de acceso a la biblioteca:

1.- Se puede accesar llamando a la biblioteca completa por medio del siguiente comando:

FORTRAN (Nombre del programa) LINK (Nombre de la biblioteca)

Con éste comando se hace referencia a la biblioteca cargando todas las rutinas almacenadas en ésta; si el programa no necesita alguna rutina en particular, el comando Link la carga sólo por un momento, y después la deja.

La ventaja de usar éste método es que el usuario no tendrá necesidad de llamar a cada una de las rutinas que solicita, simplemente llama a la biblioteca.

La posible desventaja de éste método es para la computadora, ya que carga todas las rutinas y ocupa más espacio en memoria, pero al momento de no usarlas todas, descarga las que no se necesiten.

2.- Se puede accesar llamando a cada una de las rutinas que se necesitan, usando el siguiente comando.

FORTRAN (NOmbre del programa) LINK (Nombre de las rutinas...)

Con éste comando se llama a cada una de las rutinas que se usan en el programa.

La Ventaja es que sólo se almacenan las rutinas necesarias al momento de llamarlas.

La desventaja es que el usuario tendrá que escribir todas y cada una de las rutinas que necesite.

El método que se escogió para accesar la biblioteca METNUM es el primero, ya que lo que se busca es facilitar su uso para quién la maneja, y porque la desventaja que tiene no es de mucho peso.

SISTEMAS DE AYUDA

Como su nombre lo dice, estos sistemas son desarrollados para ayudar al usuario a utilizar las rutinas de la -- mejor manera posible, y que el margen de equivocación por parte de éste, sea mínimo.

Estos sistemas se almacenarán en forma de marcos, para que el usuario sólo los llame por su nombre y pueda consultarlos.

Se desarrollaron dos sistemas de ayuda:

1.- El sistema: AYUDA. Se diseñó éste sistema para que el usuario pueda consultar la explicación del manejo

y de los parámetros utilizados en cada rutina. La forma de usar éste sistema se dá en el Manual del Usuario.

En el ANEXO 1, se puede observar la forma que tiene la explicación de cada rutina.

2.- El sistema: FUENTES. Se diseñó éste sistema para que el usuario pueda consultar los programas fuente de cada rutina. La forma de usar éste sistema se dá en el Manual de Usuario.

En el Manual del Analista, se encuentra el listado de cada programa fuente.

MANUAL DEL USUARIO

COMO USAR CUALQUIER RUTINA?

Al estar desarrollando un programa en el que se llegara a necesitar cualquiera de las rutinas incluídas en la biblioteca METNUM, lo único que se debe hacer, es llamarlas por medio del nombre que se les asignó y dando los parámetros que se solicitan. En el capítulo de Sistema de Ayuda, se especifica el formato a seguir para cada rutina incluída y una breve explicación de los parámetros a seguir.

Antes de ejecutar el programa que solicitó la(s)

rutina(s), se tiene que ligar con el programa fuente de la biblioteca, usando el siguiente comando.

FORTRAN (Nombre del programa) LINK (Nombre de la biblioteca).

COMO USAR EL SISTEMA DE AYUDA?

La biblioteca METNUM de rutinas cuenta con un sistema de ayudas, el cual, como su nombre lo dice, ayuda al usuario a conocer la forma en que se deben usar las rutinas incluídas.

Para llamar a éste sistema se debe utilizar el siguiente comando:

A Y U D A

La cual mostrará un catálogo con los nombres que recibe la ayuda de cada rutina, dándose la opción a escoger el requerido.

También cuenta con un HELP para los programas fuentes de las rutinas, para llamarlo, se debe utilizar la siguiente instrucción:

F U E N T E S

La cual mostrará un catálogo con los nombres que recibe la ayuda de los programas fuente de cada rutina, dándose la operación a escoger el requerido.

ALCANCES Y LIMITACIONES DE CADA RUTINA

1.- Multiplicación de dos matrices.

Esta rutina obtiene la multiplicación de dos matrices. La máxima dimensión que pueden tener estas matrices es de 20 renglones y 20 columnas cada una. Si excede a éste número, se mandará un mensaje de error a la pantalla, y se tendrá que dar otra vez los datos.

El número de renglones de la primera matriz debe ser igual al número de columnas de la segunda. Si no es así, se mandará un mensaje de error a la pantalla, y se tendrá que dar los datos otra vez.

Los datos necesitados para ésta rutina son:

- La dimensión de las matrices a multiplicar, debe ser número entero positivo.
- Los valores de las matrices, pueden ser reales, positivos o negativos.

Se debe cuidar que los valores de las matrices sean dados en forma correcta, ya que no hay forma de validarlos.

BIBLIOTECA
UNIVERSIDAD DE MONTERREY

2.- Inversa de una matriz cuadrada:

Esta rutina obtiene la matriz inversa de una matriz de N x N, donde N debe tener como valor máximo a 20, es decír, su máxima dimensión es de 20 renglones y 20 columnas. Si se excede éste número, se mandará un mensaje de error a pantalla, y se tendrá que capturar los datos otra vez.

Los datos necesitados por ésta rutina son:

- La dimensión de la matriz, debe ser número entero positivo.
- Los valores de la matriz, puede ser reales positivos o negativos.

Se debe cuidar que los valores de la matriz sean dados en forma correcta, ya que no hay forma de validar.

3.- Determinante de una matriz cuadrada:

Esta rutina obtiene el determinante de una matriz -- cuadrada, es decir, debe tener el mismo número de renglones y de columnas. La máxima dimensión de la matriz es de 20, tanto para renglones, como para columnas. Si se excede

éste número, se manda un mensaje de error y se tendrá -- que dar los datos una vez más.

Los datos necesitados son:

- La dimensión de la matriz, debe ser número entero positivo.
- Los valores de la matriz, pueden ser reales positivos o negativos.

Se debe checar que los datos sean dados en forma correcta, ya que no hay forma de validar.

4.- Números al azar:

Con ésta rutina se obtiene un número al azar, a partir de un número dado.

Como dato, entonces, sólo se pide un número real positivo a partir del cual se obtendrá el resultado.

5.- Camino Crítico:

Con ésta rutina se obtiene el camino o ruta crítica de un proyecto y su duración total. El máximo número de actividades que se pueden usar es 100.

Los datos que se necesitan son:

- Los prerequisitos de cada actividad, debe ser entero positivo.
- Los postrequisitos de cada actividad, debe ser entero positivo.
- La duración de cada actividad, puede ser real positiva.
- Nombre de cada actividad, puede ser alfanumérica.

Se deben checar los datos, que sean dados de forma correcta, ya que no hay forma de validar.

La ruta crítica se guarda en un vector que contiene en forma sucesiva el nombre de las actividades involucradas en ésta, y en una variable la duración total.

6.- Graficador de Barras:

Grafica un histograma usando barras. El número de periódos a graficar debe ser menor que 20.

Los datos que se necesitan son:

- El número de períodos a manejar. Debe ser entero positivo.
- Los valores que toma el comportamiento de lo que se desea analizar. Debe ser entero positivo.

Debe checarse los datos, al capturarse, ya que no hay forma de validar.

El histograma se guarda en una matriz, y la escala en una variable.

7.- Graficador de Puntos:

Grafica los puntos de una función polinomial. Esta rutina está limitada a graficar sólo 12 puntos para el eje de las X, desde 6 hasta 6 dadas por el programa, con esos puntos se obtienen los valores para Y, que estarán limitados por el tamaño de la pantalla, haciendo una es-- cala cuando sean mayores. También está limitada a grafi-

car sólo funciones de forma polinomial con un grado máximo de 20, cuando el grado exceda éste valor se mandará un mensaje de error.

Los datos necesitados son:

- El grado de la ecuación polinomial, debe ser entero positivo.
- Los coeficientes de la ecuación y el valor independiente, pueden ser reales positivos o negativos.

Dará también la escala realizada. La gráfica se guarda en una matriz.

Se debe checar que los datos sean dados en forma correcta, ya que no hay forma de validar.

8.- Derivadas:

Se obtiene la primera, segunda y tercera derivada numérica de cualquier función tabulada dando los valores tanto para X como para Y, en el primer valor dado para X. Sólo son necesarias 6 tabulaciones, para obtener los resultados esperados.

Los datos que se necesitan son:

- Seis valores para X. Deben ser reales positivos.
- Seis valores respectivos para Y. Deben ser reales positivos o negativos.

Se deben checar los datos, que sean dados en forma correcta, ya que no hay forma de validar.

Los resultados se dan en un vector, que contiene en la primera posición, la primera derivada; en la segunda, la segunda derivada y en la tercera, la tercera derivada.

9.- Raíces de una ecuación:

Esta rutina obtiene las raíces de una ecuación polinomial, tanto las reales como las complejas, por lo cual se deberán dar los datos de la siguiente manera:

- El grado de la ecuación, deberá ser un número menor a 20 y entero positivo.
- Los coeficientes de la ecuación deben ser dados en -- forma real y compleja, es decir, para cada coeficiente real (positivo o negativo) se dará también el coeficiente complejo (que será siempre cero).

Ejemplo: $3x^3 - 2x^2 + 6x - 3 = 0$

Los coeficientes serán: 3, 0,- 2, 0,6,0, - 3, 0.

El resultado se dará en un vector, el cual contendrá las raíces reales y complejas, por lo cual, al mandarlas imprimir, ya sea por pantalla o impresora, se deberá definir el formato como 2 veces el resultado.

Ejemplo: FORMAT (2(F9.2))

para que aparezcan los dos tipos de raíces.

10.- Solución de n ecuaciones con n incógnitas:

Obtiene la solución de ecuaciones simultáneas. El máximo número de ecuaciones que puede resolver es 20.

Los datos necesitados son:

- El número de ecuaciones. Debe ser entero positivo.
- Los coeficientes de cada ecuación y su valor independiente, dados en forma de matriz. Pueden ser reales positivos o negativos. En caso de que algún coeficientesea cero, se deberá poner.

El resultado se dá en un vector. El cual contendrá los valores de las incógnitas en forma sucesiva, es decir, X_1, X_2, X_3,...

11.- Factorial de un número:

Se obtiene el factorial de un número. Esta rutina está limitada a obtener el factorial de números positivos y menores que 69.

El dato que se pide es el número del cual se necesita su factorial.

M A N U A L

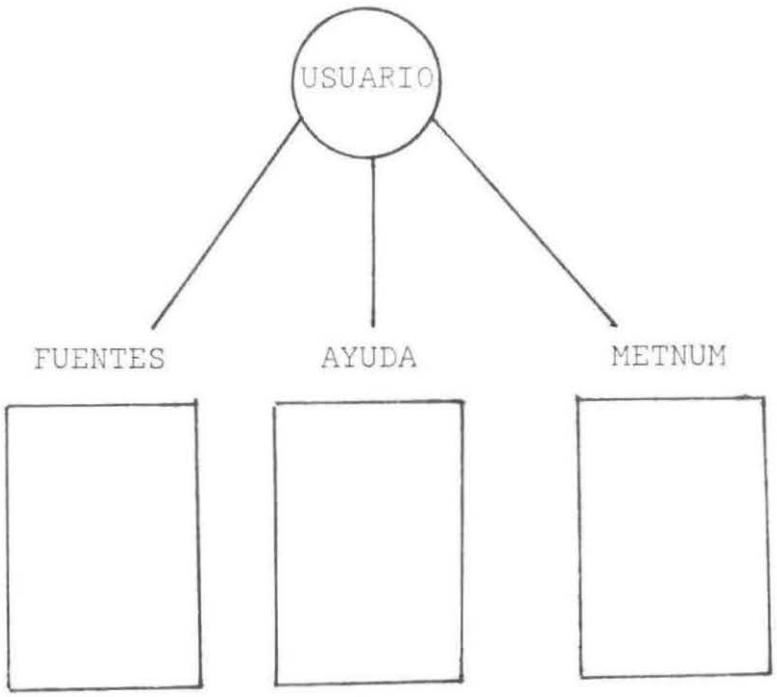
D E L

A N A L I S T A

Estructura General:

El usuario tendrá acceso a los programas fuentes (de lectura solamente) por medio del Sistema de Ayuda FUENTES; al sistema AYUDA, que le permitirá conocer la manera en que se maneja cada rutina de la biblioteca; y a la biblioteca METNUM.

La estructura general del sistema es la siguiente:



Diseño de rutinas

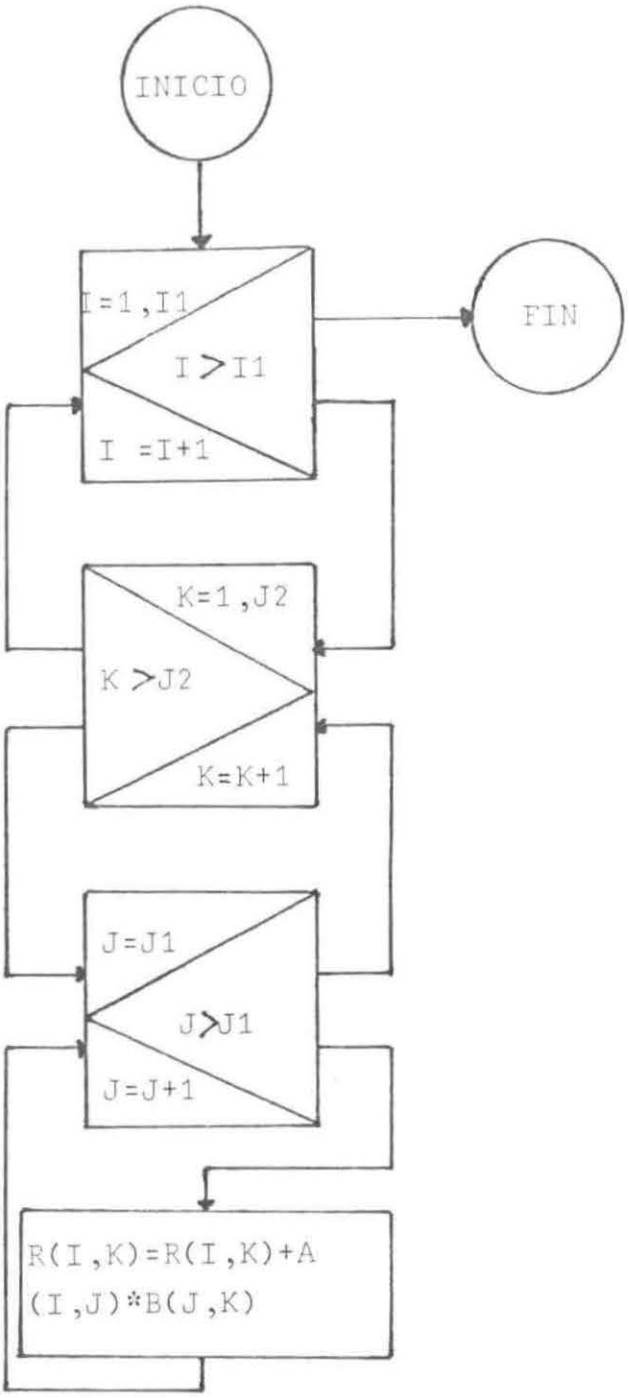
--Algoritmos--

Los pasos a seguir para la programación de cada rutina están dados en el capítulo: "Descripción detallada de rutinas desarrolladas", para cualquier duda o aclaración, en cuanto al tema en cuestión, pase a consultar dicho capítulo.

-- Diagramas de Flujo --

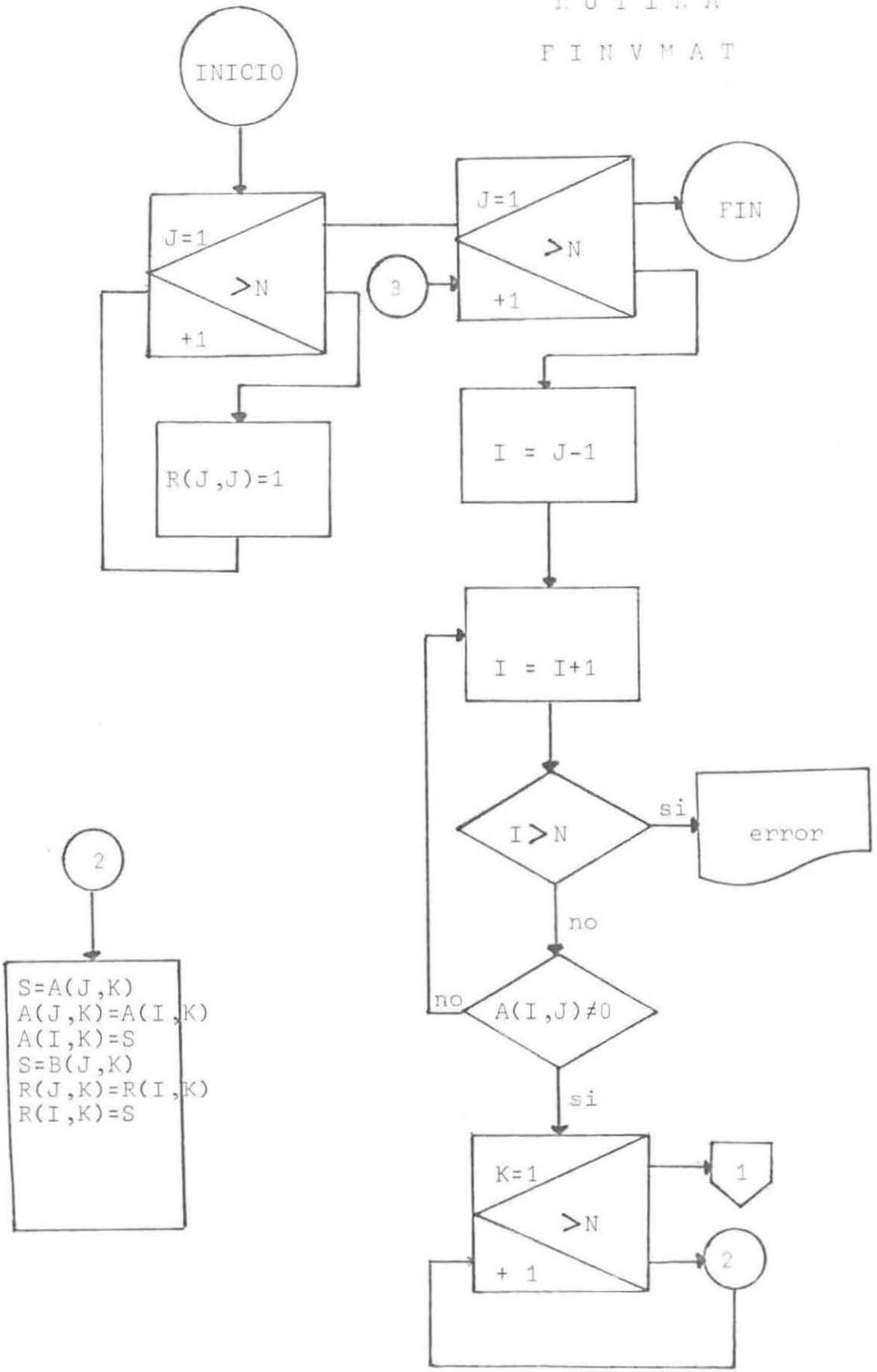
R U T I N A

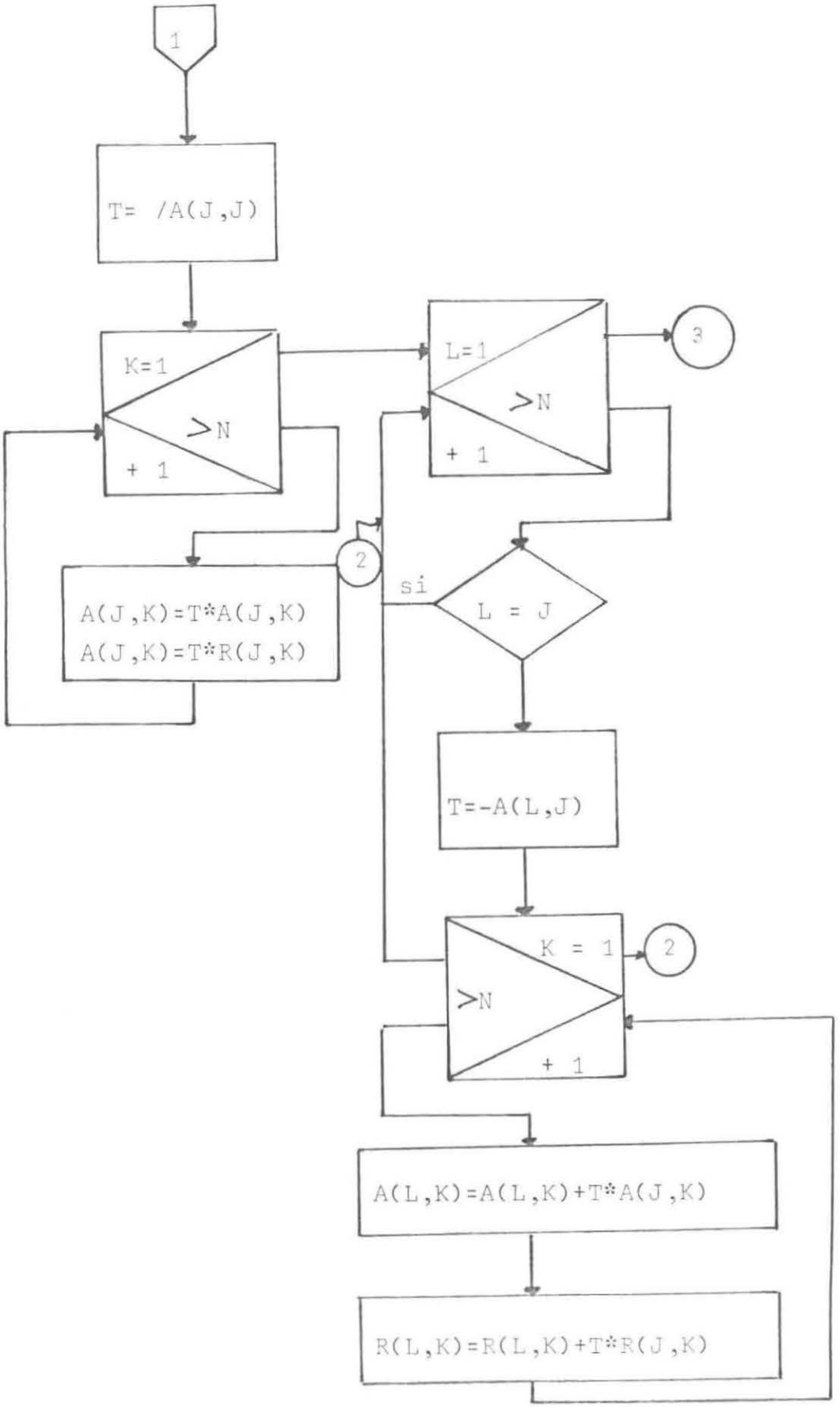
F M U L M A T



R U T I N A

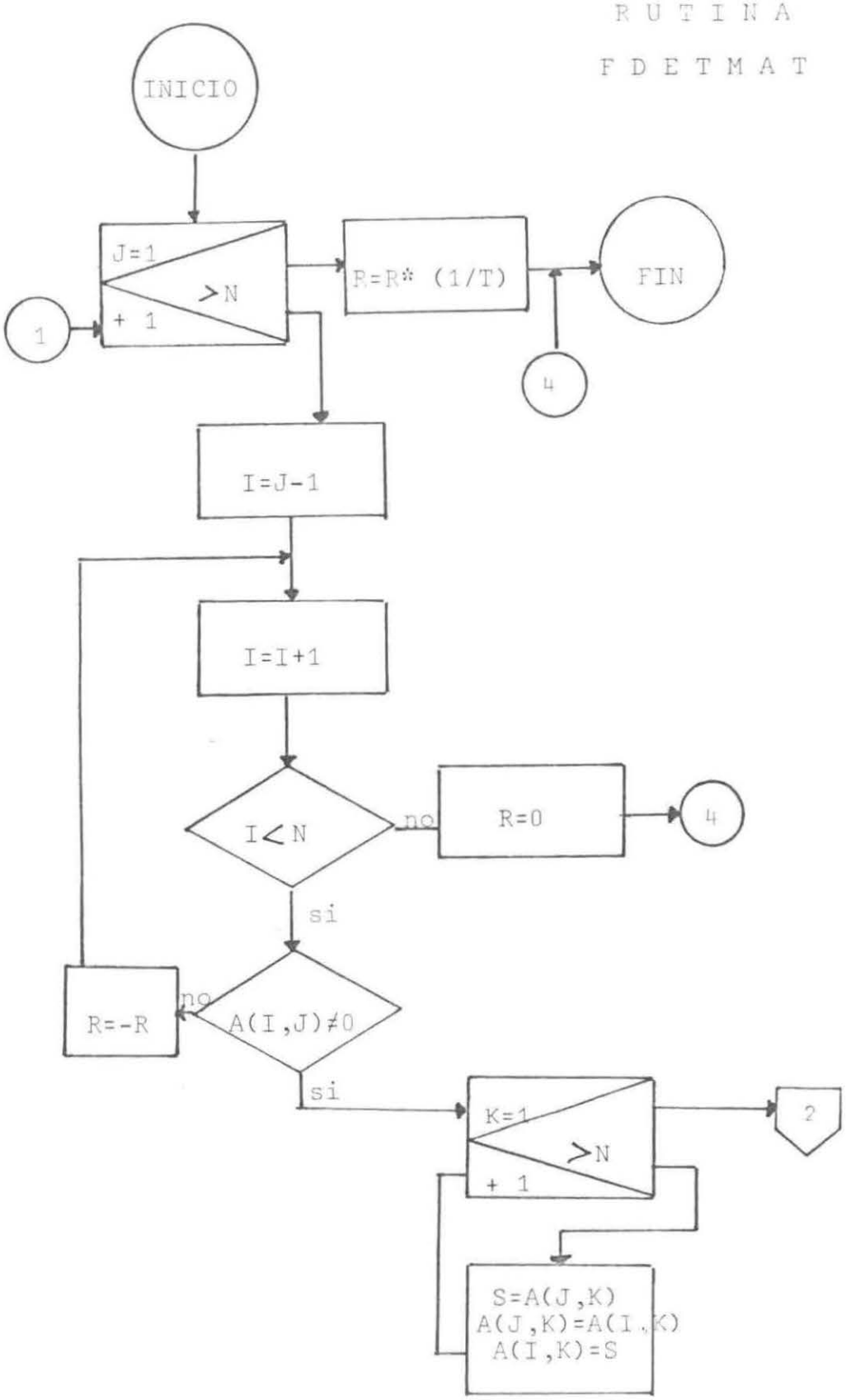
F I N V M A T

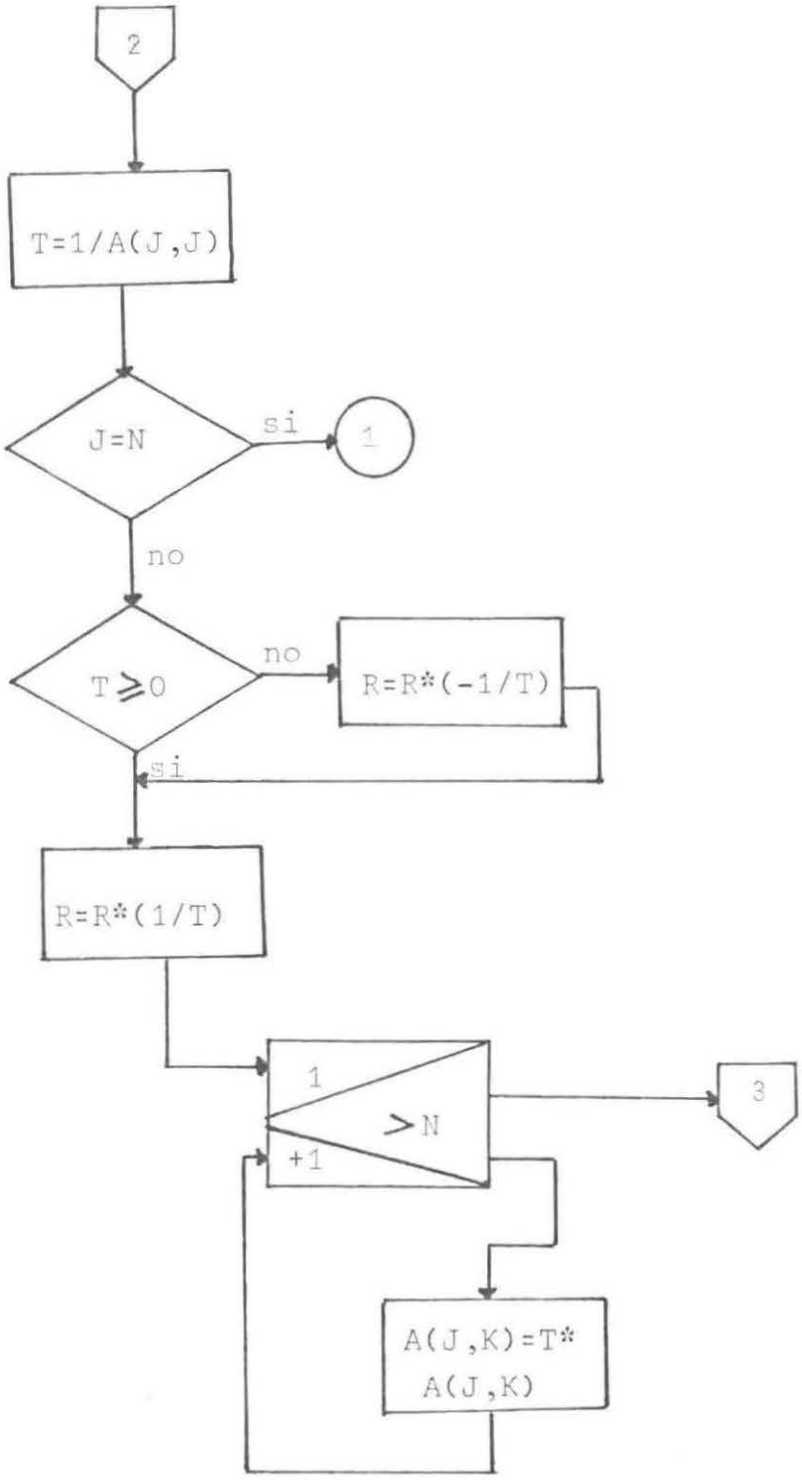


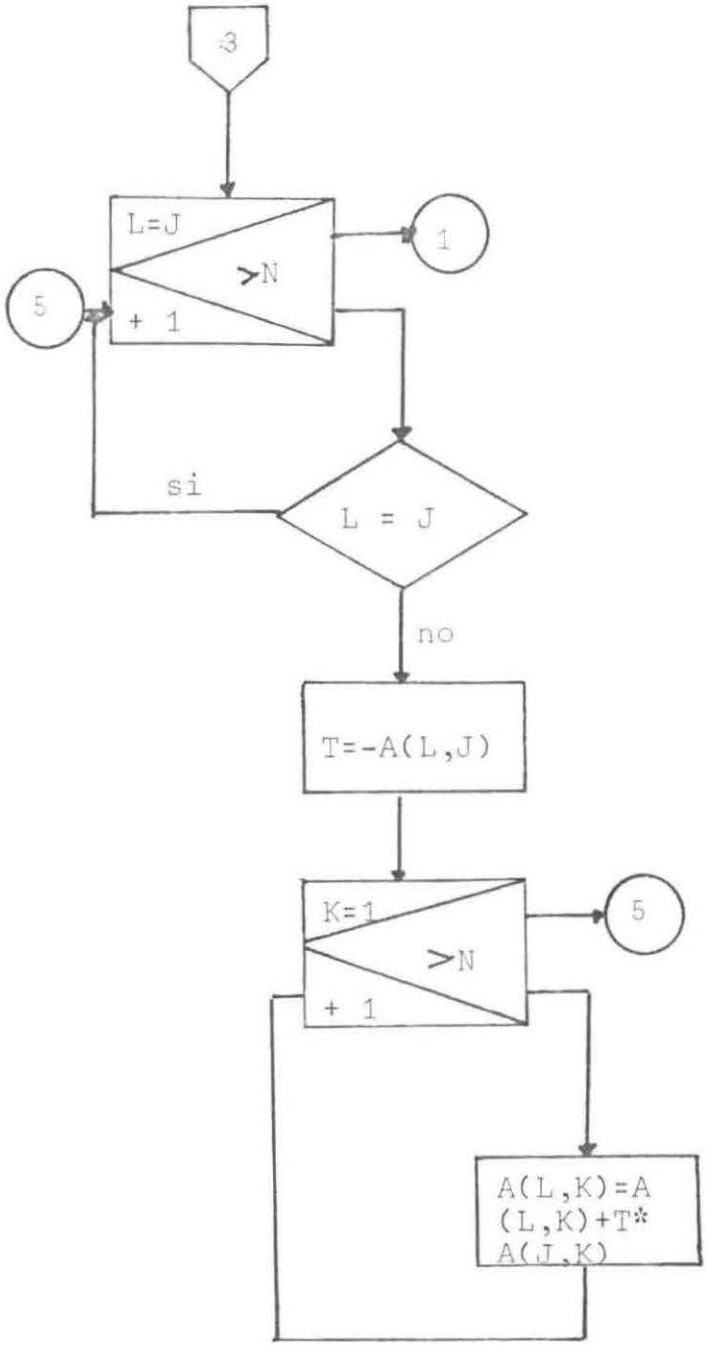

1
T= /A(J,J)
K=1
>N
+ 1
L=1
>N
+ 1
3
2
A(J,K)=T*A(J,K)
A(J,K)=T*R(J,K)
si
L = J
T=-A(L,J)
K = 1
2
>N
+ 1
A(L,K)=A(L,K)+T*A(J,K)
R(L,K)=R(L,K)+T*R(J,K)

R U T I N A

F D E T M A T

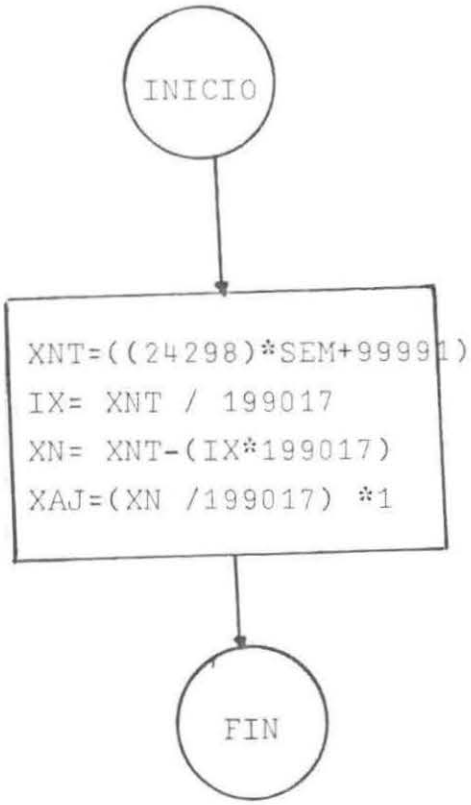



2
T=1/A(J,J)
J=N
si
1
no
T ≥ 0
no
R=R*(-1/T)
si
R=R*(1/T)
1
> N
+1
3
A(J,K)=T*
A(J,K)


3
L=J
>N
+ 1
1
5
si
L = J
no
T=-A(L,J)
K=1
>N
+ 1
5
A(L,K)=A
(L,K)+T*
A(J,K)

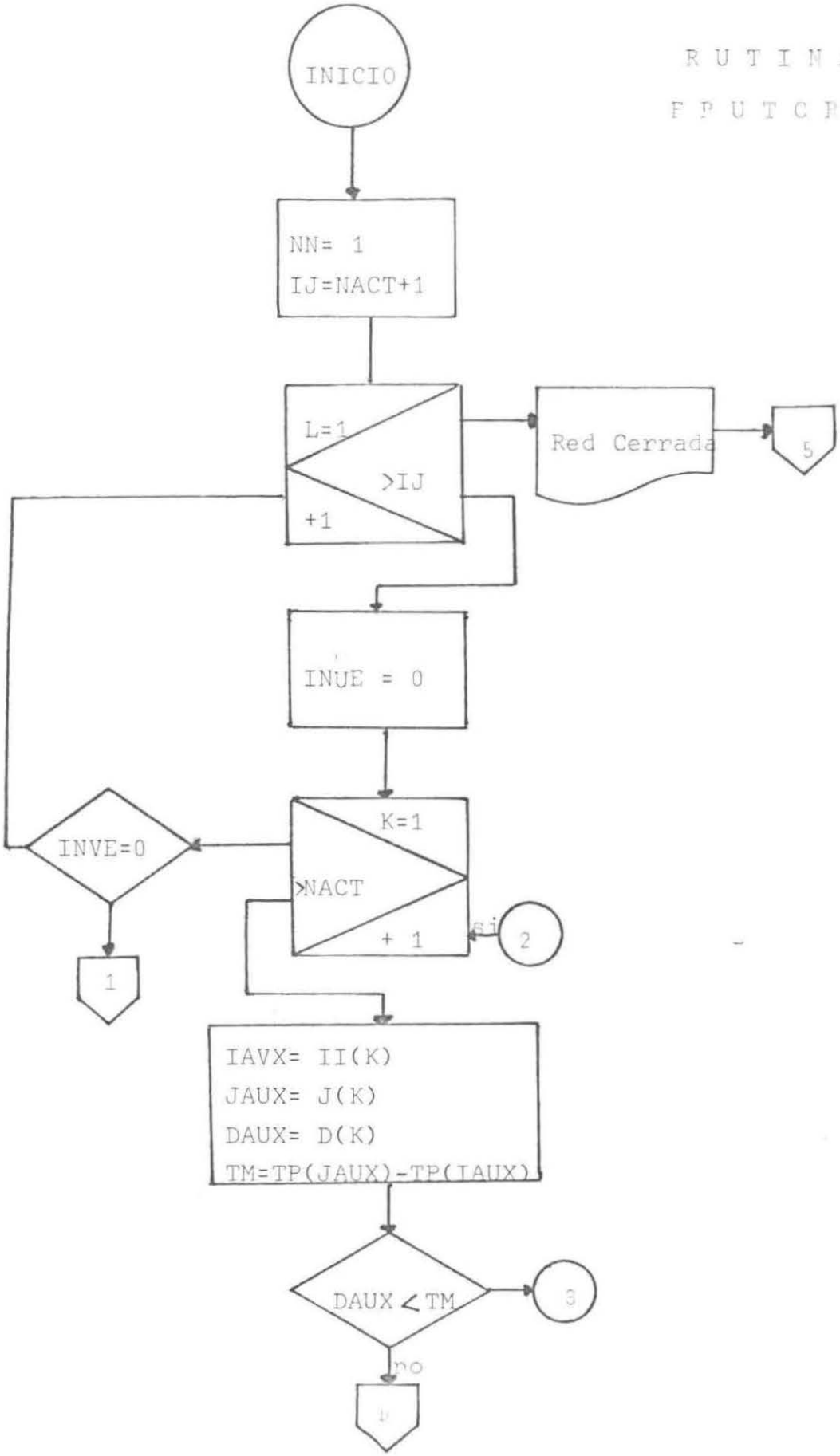
F U N C I O N

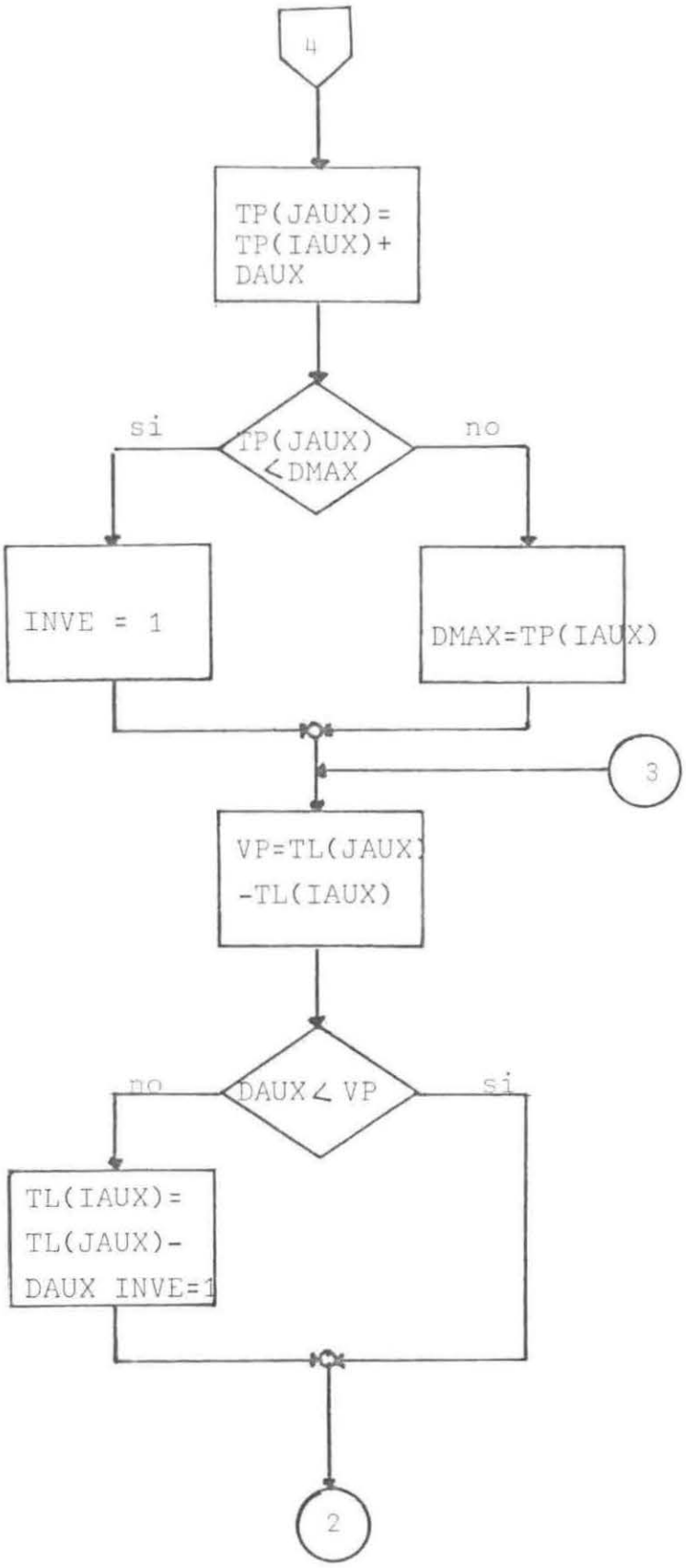
F N U M A Z A

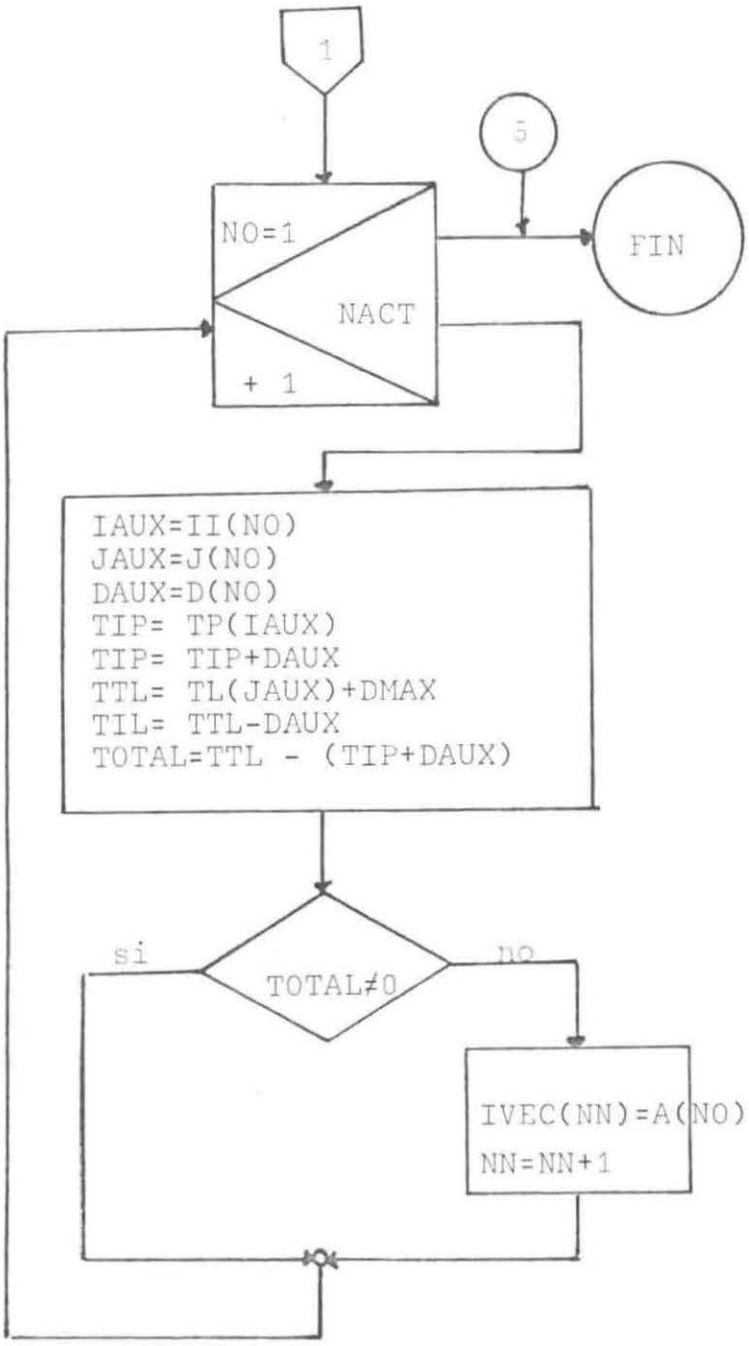


R U T I N A

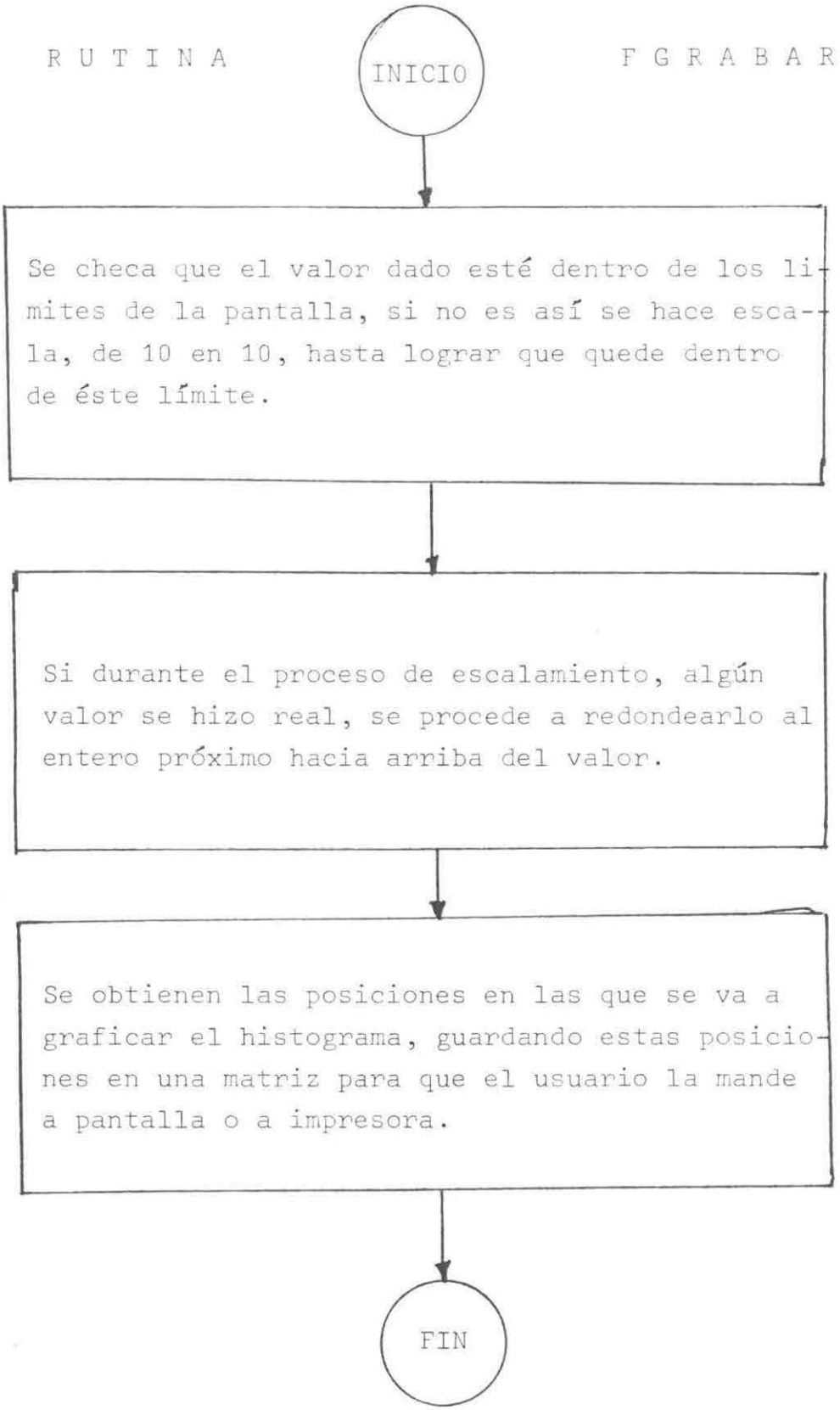
F P U T C P I




4
TP(JAUX)=
TP(IAUX)+
DAUX
si
TP(JAUX)
<DMAX
no
INVE = 1
DMAX=TP(IAUX)
3
VP=TL(JAUX)
-TL(IAUX)
no
DAUX < VP
si
TL(IAUX)=
TL(JAUX)-
DAUX INVE=1
2

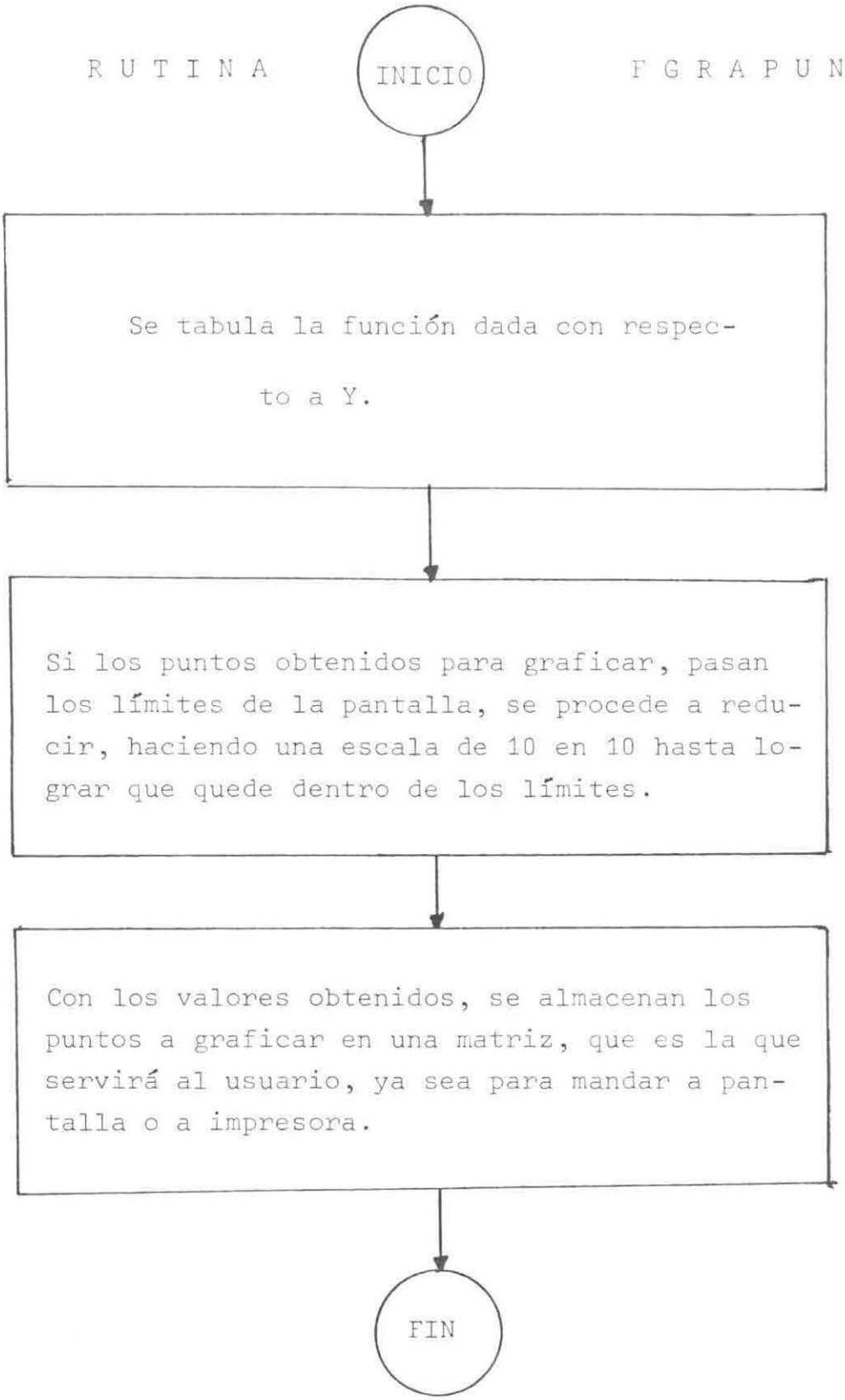

1
5
NO=1
NACT
+ 1
FIN
IAUX=II(NO)
JAUX=J(NO)
DAUX=D(NO)
TIP= TP(IAUX)
TIP= TIP+DAUX
TTL= TL(JAUX)+DMAX
TIL= TTL-DAUX
TOTAL=TTL - (TIP+DAUX)
si
TOTAL≠0
no
IVEC(NN)=A(NO)
NN=NN+1

R U T I N A F G R A B A R



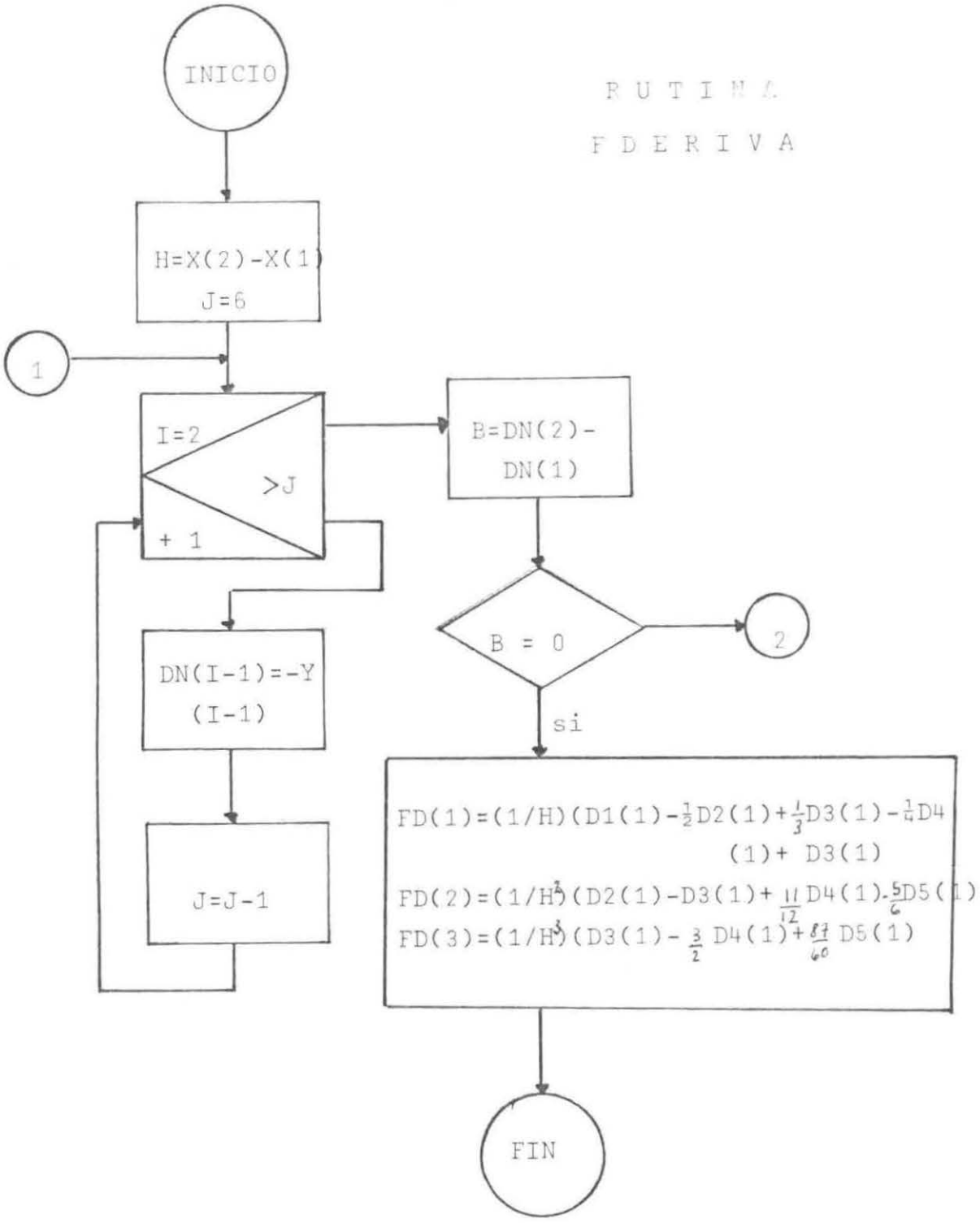
R U T I N A

F G R A P U N



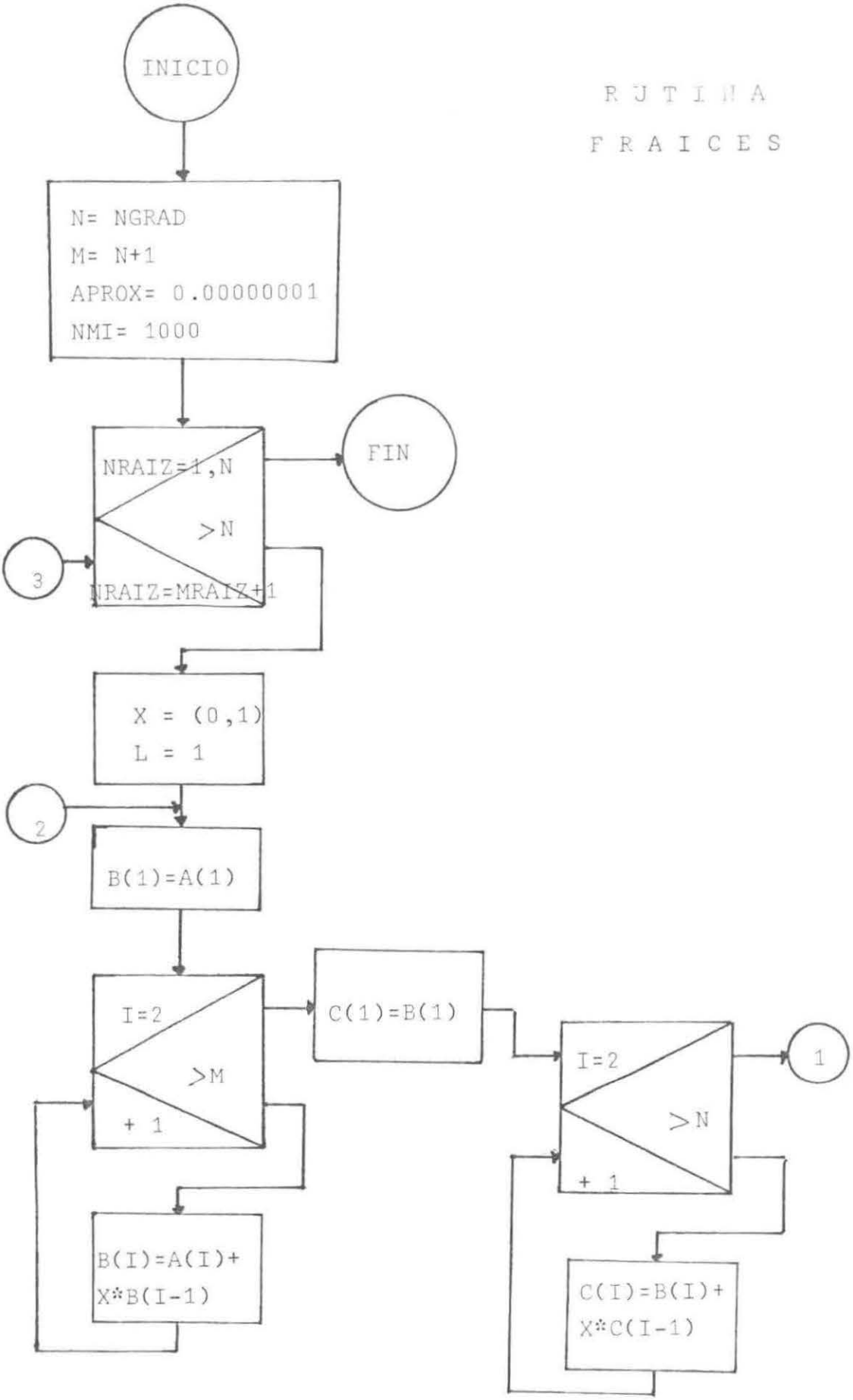
R U T I N A

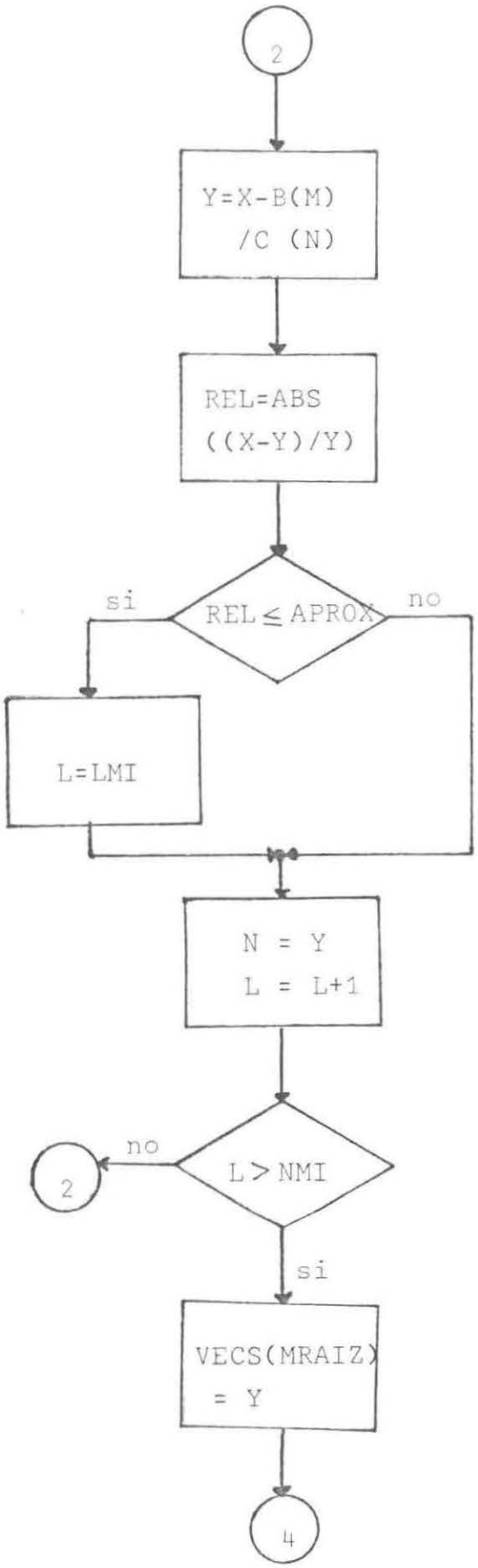
F D E R I V A

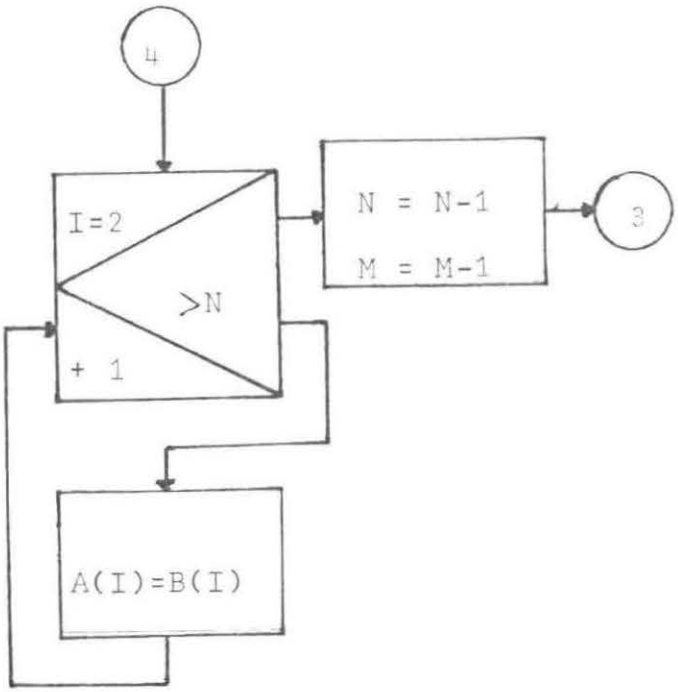


R U T I N A

F R A I C E S

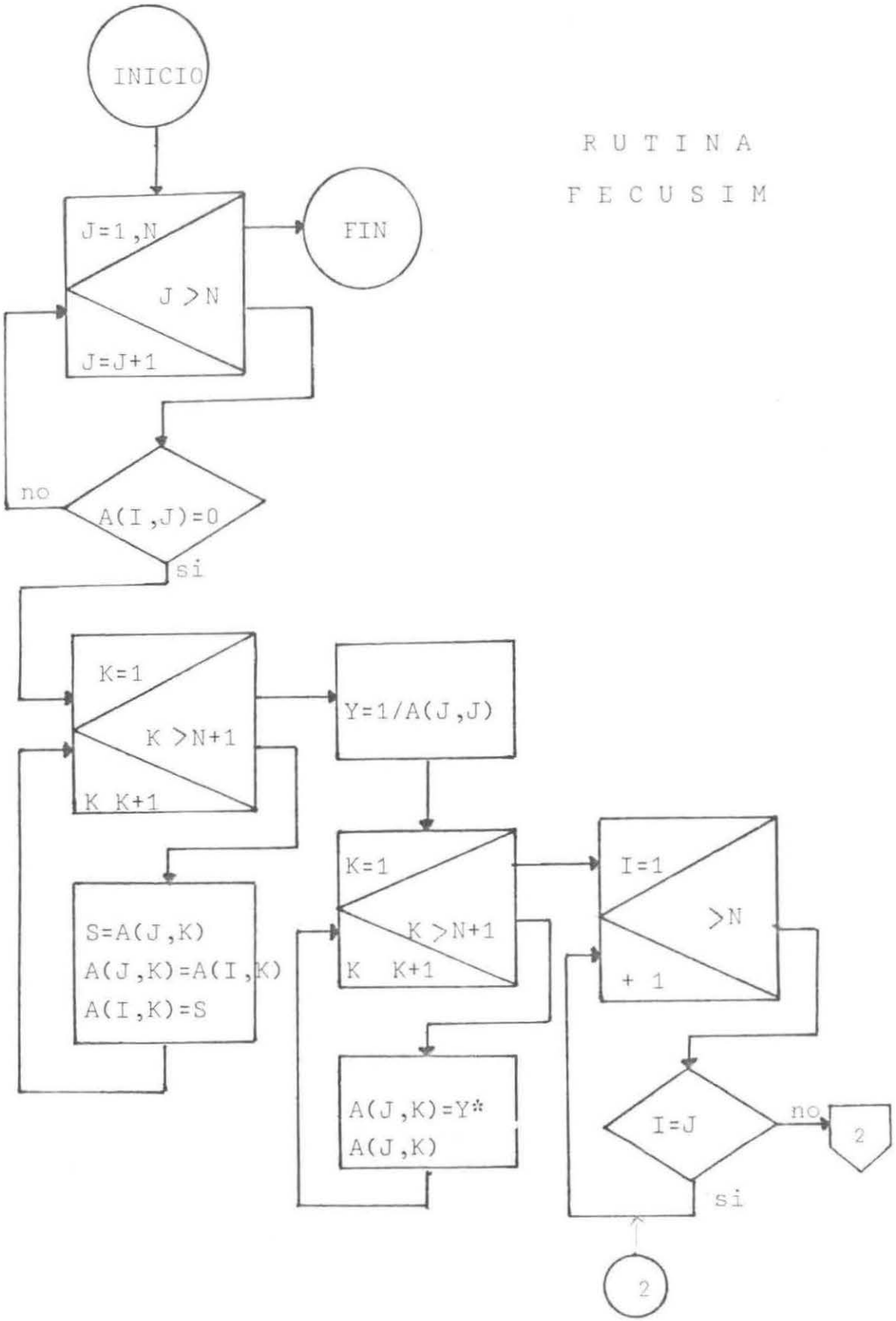


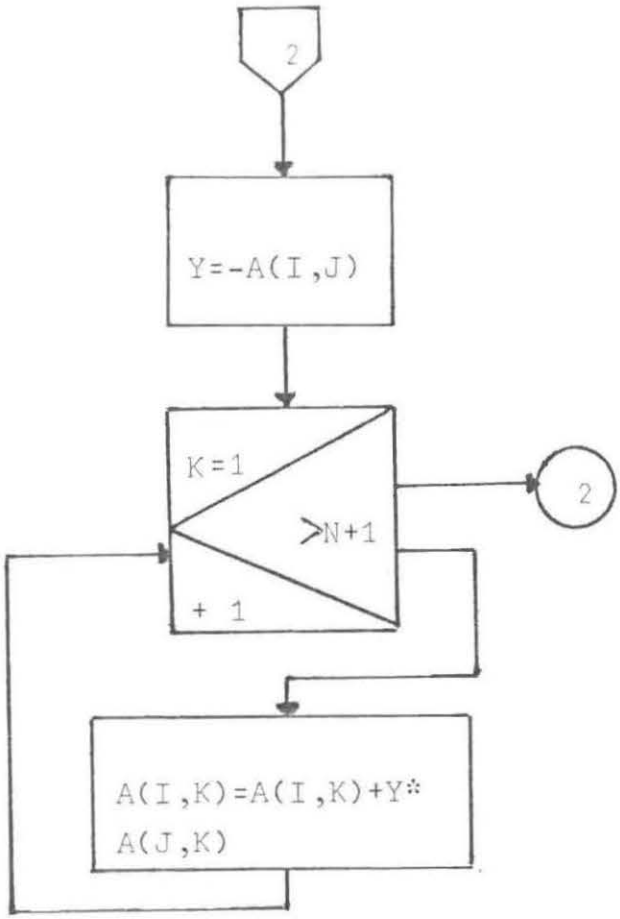

2
Y=X-B(M)
/C (N)
REL=ABS
((X-Y)/Y)
REL ≤ APROX
si
no
L=LMI
N = Y
L = L+1
L > NMI
no
2
si
VECS(MRAIZ)
= Y
4


4
I=2
>N
+ 1
N = N-1
M = M-1
3
A(I)=B(I)

R U T I N A

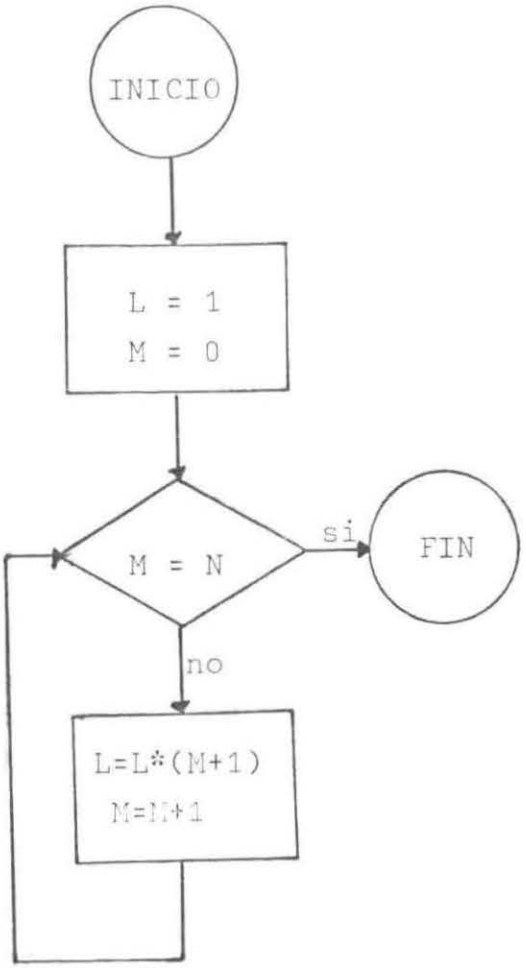
F E C U S I M




2
Y=-A(I,J)
K=1
>N+1
+ 1
2
A(I,K)=A(I,K)+Y*
A(J,K)

F U N C I O N

F F A C T O R



-- Listados de Programas Fuente --

```
      PROGRAM MULTMAT
      DIMENSION A(20,20),B(20,20),R(20,20)
      IF(J1.NE.I2) GOTO 99 !SI EL # DE COLUMNAS DE A ES DIFERENTE AL #
                           DE RENGLONES DE B, SE MANDA UN MENSAJE DE
                           ERROR, SI NO CONTINUA

      EL SIGUIENTE CICLO SE USA PARA LLENAR LA MATRIZ DE RESULTADOS
      CON CEROS.

      DO 15 I=1,I1
      DO 13 K=1,J2
      R(I,K)=0.0

      EL SIGUIENTE CICLO SE UTILIZA PARA REALIZAR LA MULTIPLICACION
      DE LAS MATRICES A Y B DEJANDO EL RESULTADO EN LA MATRIZ R.

      DO 11 J=1,J1
      R(I,K)=R(I,K)+A(I,J)*B(J,K)
   11 CONTINUE
   13 CONTINUE
   15 CONTINUE
   99 WRITE(*,100)
  100 FORMAT(2X,"ERROR: EL # DE COLUMNAS DE -A- DEBE SER IGUAL AL # DE RENGLONES
      STOP
      END
```

```
      PROGRAM MATINV
      DIMENSION A(20,20),R(20,20)

      CICLO UTILIZADO PARA FORMAR LA MATRIZ IDENTIDAD

      DO 3 J=1,N
      R(J,J)=1
    3 CONTINUE

      CICLO UTILIZADO PARA OBTENER LA MATRIZ INVERSA

      DO 4 J=1,N

      CICLO UTILIZADO PARA CHECAR SI EL ELEMENTO A(I,J) ES CERO; SI SI,
      SE MANDA AL CICLO DE INTERCAMBIO DE RENGLONES HASTA ENCONTRAR UNO
      DIFERENTE DE CERO

      I=J-1
   13 I=I+1
      IF(I.GT.N) GOTO 14
      IF(A(I,J).NE.0) GOTO 5
      GOTO 13

      CICLO UTILIZADO PARA INTERCAMBIAR RENGLONES, SI ESTO ES NECESARIO,
      SI NO, SOLO REACOMODA LOS NUMEROS EN SU POSICION ORIGINAL EN EL -
      RENGLON QUE SE PROCESA

    5 DO 6 K=1,N
      S=A(J,K)
      A(J,K)=A(I,K)
      A(I,K)=S
      S=R(J,K)
      R(J,K)=R(I,K)
      R(I,K)=S
    6 CONTINUE
      T=1/A(J,J)          ! RECIPROCO DEL ELEMENTO A(J,J)

      CICLO UTILIZADO PARA NORMALIZAR EL ELEMENTO A(J,J), MULTIPLICANDO
      EL RENGLON J POR EL RECIPROCO OBTENIDO, EMPEZANDO A FORMAR LA MA-
      TRIZ RESULTANTE

      DO 7 K=1,N
      A(J,K)=T*A(J,K)
      R(J,K)=T*R(J,K)
    7 CONTINUE

      CICLO UTILIZADO PARA HACER CERO LOS ELEMENTOS QUE SE ENCUENTRAN
      BAJO LA DIAGONAL DE LA COLUMNA K, Y OBTENCION DE LA MATRIZ INVER-
      SA, MULTIPLICANDO LA MATRIZ IDENTIDAD POR LA MATRIZ ELEMENTAL

      DO 8 L=1,N
      IF(L.EQ.J) GOTO 8
      T=-A(L,J)
      DO 9 K=1,N
      A(L,K)=A(L,K)+T*A(J,K)
      R(L,K)=R(L,K)+T*R(J,K)
    9 CONTINUE
    8 CONTINUE
    4 CONTINUE

      CICLO PARA MEJORAR LA EXACTITUD DE LA MATRIZ RESULTANTE R

      DO 20 I=1,N
```

```
 22 CONTINUE
 20 CONTINUE
 14 WRITE(*,100)
100 FORMAT(2X,"ERROR:")
    STOP
    END
```

```
      PROGRAMA DETERMINANTE
      DIMENSION A(20,20)
      R=1     ! VARIABLE UTILIZADA PARA ALMACENAR EL RESULTADO
      IF (N.EQ.2) GOTO 12    ! SI LA DIMENSION DE LA MATRIZ ES 2, LA DE-
                               TERMINANTE SE OBTIENE POR MEDIO DE UNA FOR
                               MULA

      CICLO UTILIZADO PARA CHECAR SI EL ELEMENTO A(I,J) ES CERO; SI SI,
      MANDA AL CICLO DE INTERCAMBIO DE RENGLONES, HASTA ENCONTRAR UNO -
      DIFERENTE DE CERO, DESPUES SE HACEN LAS OPERACIONES NECESARIAS PA-
      OBTENER EL DETERMINANTE.

      DO 10 J=1,N
      I=J-1
    3 I=I+1
      IF (I.LE.N) GOTO 15
      R=0
      GOTO 13
   15 IF (A(I,J).NE.0) GOTO 4
      R=-R
      GOTO 3

      CICLO UTILIZADO PARA INTERCAMBIAR RENGLONES, SI ESTO ES NECESARIO,
      SI NO, SOLO ACOMODA LOS NUMEROS EN SU POSICION ORIGINAL EN EL REN-
      GLON QUE SE PROCESA

    4 DO 5 K=1,N
      S=A(J,K)
      A(J,K)=A(I,K)
      A(I,K)=S
    5 CONTINUE
      T=1/A(J,J)   ! RECIPROCO DEL ELEMENTO A(I,J)
      IF (J.EQ.N) GOTO 10
      IF (T.GE.0) GOTO 7
      R=R*(-1/T)   ! SI EL RECIPROCO ES CERO O NEGATIVO, EL DETERMINANTE
                     SE MULTIPLICA POR EL RECIPROCO DEL RECIPROCO NEGATIVO
      GOTO 11
    7 R=R*(1/T)    ! SI EL RECIPROCO ES POSITIVO, EL DETERMINANTE SE MUL-
                     TIPLICA POR EL RECIPROCO DEL RECIPROCO.

      CICLO UTILIZADO PARA FORMAR LA MATRIZ DADA EN MATRIZ TRIANGULAR.

   11 DO 6 K=1,N
      A(J,K)=T*A(J,K)
    6 CONTINUE
      DO 9 L=J,N
      IF(L.EQ.J) GOTO 9
      T=-A(L,J)
      DO 8 K=1,N
      A(L,K)=A(L,K)+T*A(J,K)
    8 CONTINUE
    9 CONTINUE
   10 CONTINUE
      R=R*(1/T)   ! VALOR FINAL DEL DETERMINANTE DE LA MATRIZ DADA.
      GOTO 13

      FORMULA PARA OBTENER EL DETERMINANTE, CUANDO LA MATRIZ ES DE DOS
      RENGLONES Y DOS COLUMNAS.

   12 R=R*((A(1,1)*A(2,2))-(A(1,2)*A(2,1)))
   13 STOP
      END
```

PROGRAMA NAZAR

CALCULOS PARA OBTENER UN NUMERO AL AZAR, A PARTIR DE LA SEMILLA DADA, POR MEDIO DE NUMEROS CLAVES Y OPERACIONES ESPECIFICAS.

```
AN=SEM
XNT=((24298)*AN+99991)
IX=XNT/199017
XN=XNT-(IX*199017)
XAJ=(XN/199017)*1+0  !VALOR FINAL DEL NUMERO AL AZAR
STOP
END
```

```
      PROGRAM CAMCRI2
      CHARACTER *1 A(100), IVEC(100),R(100)
      DIMENSION II(100),J(100),D(100),TP(100),TL(100)
      NN=1
      IJ=NACT+1

      CICLO USADO PARA DEFINIR COMO CERO LOS TIEMPOS MAS PROXIMOS DE INI
      CIACION Y MAS LEJANOS DE TERMINACION, ASI COMO LA DURACION MAXIMA.

      DO 1 I=1,NACT
      TP(I)=0
      TL(I)=0
    1 CONTINUE
      DMAX=0

      CICLO USADO PARA CALCULAR LOS TIEMPOS MAS PROXIMOS DE INICIACION
      Y MAS LEJANOS DE TERMINACION.

      DO 6 L=1,IJ
      INUE=0
      DO 5 K=1,NACT
      IAUX=II(K)
      JAUX=J(K)
      DAUX=D(K)
      TM=TP(JAUX)-TP(IAUX)
      IF(DAUX.LE.TM) GOTO 4
      TP(JAUX)=TP(IAUX)+DAUX
      IF(TP(JAUX).LE.DMAX) GOTO 3
      DMAX=TP(JAUX)
    3 INUE=1
    4 VP=TL(JAUX)-TL(IAUX)
      IF(DAUX.LE.VP) GOTO 5
      TL(IAUX)=TL(JAUX)-DAUX
      INUE=1
    5 CONTINUE
      IF(INUE.EQ.0) GOTO 7
    6 CONTINUE
      WRITE(*,100)
      GOTO 101

      CICLO USADO PARA CALCULAR LOS TIEMPOS MAS LEJANOS DE INICIACION Y
      MAS PROXIMO DE TERMINACION, ASI COMO SU HOLGURA TOTAL.

    7 DO 8 NO=1,NACT
      IAUX=II(NO)
      JAUX=J(NO)
      DAUX=D(NO)
      TIP=TP(IAUX)
      TTP=TIP+DAUX
      TTL=TL(JAUX)+DMAX
      TIL=TTL-DAUX
      TOTAL=TTL-(TIP+DAUX)
      IF(TOTAL.NE.0) GOTO 8
      IVEC(NN)=A(NO)
      NN=NN+1
    8 CONTINUE
      DO 9 I=1,NN-1
      R(I)=IVEC(I)        ! VECTOR QUE ALMACENA EL CAMINO CRITICO.
    9 CONTINUE
      DT=DMAX             ! DURACION TOTAL DEL PROYECTO
  100 FORMAT(2X,"LA RED DE ACTIVIDADES ESTA CERRADA")
  101 STOP
      END
```

```
      PROGRAM GRAFBAR
      CHARACTER *1 MATB(20,70)
      DIMENSION ICY(70),OPI(70),IYR(70),KXY(70)
      IF(NCO.GT.35) GOTO 17 !VALIDA QUE ESTE DENTRO DEL RANGO
      DO 50 I=1,NCO
      OPI(I)=ICY(I)
   50 CONTINUE

      CICLO PARA CHECAR SI ES NECESARIO HACER ESCALA CON LOS VALORES DA-
      DOS.

      ICO=0
      IEY=1
      K=0
    1 SW=0
      DO 2 I=1,NCO
      IF(OPI(I).GT.18.0) THEN
      SW=1
      ENDIF
    2 CONTINUE
      IF(SW.EQ.0) GOTO 30

      CICLO QUE REALIZA LA ESCALA NECESITADA

      DO 3 I=1,NCO
      OPI(I)=OPI(I)/10
    3 CONTINUE
      ICO=ICO+1
      GOTO 1
   30 DO 35 I=1,ICO
      IEY=IEY+1
   35 CONTINUE
      DO 5 I=1,NCO
      VPA2=OPI(I)
      IYR(I)=ANINT(VPA2)
    5 CONTINUE
      DO 6 I=1,NCO
      KXY(I)=IYR(I)
    6 CONTINUE

      BUSQUEDA DEL NUMERO MAYOR Y MENOR DADOS, PARA CONOCER EL ESPACIO
      QUE SE USARA EN LA MATRIZ DE SALIDA.

      DO 8 I=1,NCO-1
      KL=I+1
      DO 7 J=KL,NCO
      IF(KXY(I).GE.KXY(J)) GOTO 7
      MA=KXY(I)
      KXY(I)=KXY(J)
      KXY(J)=MA
    7 CONTINUE
    8 CONTINUE

      CICLO PARA ENCONTRAR LOS VALORES MAXIMOS DE LOS EJES.

      MAY=KXY(1)
      MEY=KXY(NCO)
      MTY=MEY+MAY

      CICLO PARA ENCONTRAR EL EJE DE LOS PERIODOS Y EL EJE DEL COMPORTA-
      MIENTO DURANTE ESOS PERIODOS.

      DO 11 I=1,20
```

```
 10 CONTINUE
 11 CONTINUE
    MTO=MAY+1
    NQ=2*(NCO+1)
    DO 12 I=2,NQ
    MATB(MTO,I)="-"
 12 CONTINUE
    DO 13 I=1,MAY
    MATB(I,1)="¦"
 13 CONTINUE
    MATB(MTO,1)="+"
    JY=3

    CICLO USADO PARA FORMAR EL HISTROGRAMA Y GUARDARLO EN LA MATRIZ DE
    SALIDA.

    DO 15 I=1,NCO
    JY=JY+1
    NC=JY
    JY=JY+1
    NR=(MAY-IYR(I))+1
 14 MATB(NR,NC)="¦"
    NR=NR+1
    IF(NR.GT.MAY) GOTO 15
    GOTO 14
 15 CONTINUE
    GOTO 101
 17 WRITE(*,100)
100 FORMAT(2X,"ERROR:NUMERO DE PERIODOS NO PERMITIDO")
101 STOP
    END
```

```
      PROGRAM GRAFICO
      CHARACTER *1 MATG(20,60)
      DIMENSION A9(20),VEC(20,5),IVXY(20,5),IVP(20)
      IF(NGR.LT.20) GOTO 18
      WRITE(*,100)
      GOTO 101
   18 K=1
      K1=2
      M=NGR+1

      CICLO UTILIZADO PARA TABULAR LA ECUACION POLINOMIAL DADA, A PARTIR
      DE VALORES DADOS A X, DESDE 6 NEGATIVO HASTA 6 POSITIVO.

      DO 2 I=1,6
      L=-I
      NE=NGR
      Y=0.0
      Y1=0.0
      DO 1 J=1,NGR
      Y=Y+(A9(J)*(L**NE))
      Y1=Y1+(A9(J)*(I**NE))
      NE=NE-1
    1 CONTINUE
      Y=Y+A9(M)
      Y1=Y1+A9(M)
      S=I+0.0
      S1=L+0.0

      LOS VALORES DE X DADOS Y LOS DE Y OBTENIDOS, SE ALMACENAN EN UNA
      MATRIZ, LA PRIMER COLUMNA CONTIENE LOS VALORES DE X, Y LA SEGUNDA
      LOS VALORES DE Y.

      VEC(K,1)=S
      VEC(K1,1)=S1
      VEC(K,2)=Y1
      VEC(K1,2)=Y
      K=K+2
      K1=K1+2
    2 CONTINUE
      ICE=0
      IES=1
    3 SW=0

      CICLO UTILIZADO PARA OBTENER UNA ESCALA, SI ESTA ES NECESARIA, EN
      CUANTO A LOS VALORES OBTENIDOS DE Y.

      DO 4 I=1,12
      DK=-10.0
      IF(VEC(I,2).GT.10.0.OR.VEC(I,2).LT.DK) THEN
      SW=1
      ENDIF
    4 CONTINUE

      SI EL SWITCH UTILIZADO PARA SABER SI ES NECESARIO HACER ESCALA ES
      UNO, SE REALIZA EL PROCEDIMIENTO, SI NO, CALCULA LA ESCALA ACUMU-
      LADA.

      IF(SW.EQ.0) GOTO 40
      DO 5 I=1,12
      VEC(I,2)=VEC(I,2)/10
    5 CONTINUE
      ICE=ICE+1
      GOTO 3
```

```
 6 CONTINUE

   CICLO UTILIZADO PARA REDONDEAR LOS VALORES DE LA MATRIZ DE TABULA-
   CION AL ENTERO CORRESPONDIENTE.

   DO 7 I=1,12
   VP0=VEC(I,1)
   IVXY(I,1)=ANINT(VP0)
   VP2=VEC(I,2)
   IVXY(I,2)=ANINT(VP2)
   IVP(I)=IVXY(I,2)
 7 CONTINUE
   IH2=14

   CICLO UTILIZADO PARA OBTENER EL MAYOR Y MENOR VALOR DE LA COLUMNA
   DE LAS Y'S

   DO 9 I=1,11
   KR=I+1
   DO 8 J=KR,12
   IF(IVP(I).GE.IVP(J)) GOTO 8
   ME=IVP(I)
   IVP(I)=IVP(J)
   IVP(J)=ME
 8 CONTINUE
 9 CONTINUE

   RUTINA UTILIZADA PARA OBTENER LOS VALORES NECESARIOS PARA FORMAR
   EL EJE 'X', Y EL EJE 'Y'.

   IV1=IVP(12)
   IV2=IVP(1)
   IF(IV2.LT.0) THEN
   IV2=-IV2
   ENDIF
   IF(IV1.LT.0) THEN
   IV1=-IV1
   ENDIF
   IVT=IV1+IV2
   LVT=(IVT-IV1)+1

   CICLO PARA LIMPIAR LA MATRIZ DE SALIDA

   DO 14 I=1,20
   DO 13 J=1,60
   MATG(I,J)=" "
13 CONTINUE
14 CONTINUE

   CICLO UTILIZADO PARA FORMAR EL EJE 'X'.

   DO 10 I=1,14
   MATG(LVT,I)="_"
10 CONTINUE

   CICLO UTILIZADO PARA FORMAR EL EJE 'Y'.

   DO 11 I=1,LVT+1
   MATG(I,7)="|"
11 CONTINUE

   CICLO UTILIZADO PARA ALMACENAR LAS COORDENADAS(X,Y) EN LA MATRIZ
   DE SALIDA.
```

```
      NB=IVXY(I,1)
      LU=IVXY(I,2)
      LU=LU*(-1)
      NM=LU+LVT
      MATG(NM,NB)=".."
   12 CONTINUE
  100 FORMAT(2X,"ERROR: GRADO DE LA ECUACION NO ADECUADO")
  101 STOP
      END
```

```
PROGRAMA DERIVADAS
DIMENSION X(20),Y(20),D1(10),D2(10),D3(10),D4(10),FD(3)
H=X(2)-X(1)    ! INTERVALO CONSTANTE ENTRE LOS SUCESIVOS VALORES
                 QUE TOMA LA X, EN LA FUNCION TABULADA

CICLO UTILIZADO PARA LIMPIAR LOS VECTORES QUE SE USARAN DURANTE
EL PROCESO.

DO 9 I=1,10
D1(I)=0.0
D2(I)=0.0
D3(I)=0.0
D4(I)=0.0
9 CONTINUE

LOS SIGUIENTES CUATRO CICLOS SE UTILIZAN PARA OBTENER LAS DIFE-
RENCIAS DE ORDEN SUPERIOR, PARA APLICARLAS EN LAS FORMULAS CO--
RESPONDIENTES.

DO 2 I=2,6
D1(I-1)=Y(I)-Y(I-1)
2 CONTINUE
B=D1(2)-D1(1)
IF(B.EQ.0) GOTO 7      ! SI AL OBTENER CUALQUIER DIFERENCIA, ESTA
                         ES IGUAL A CERO, SE CORTA EL PROCESO HA-
                         CIENDO CERO TODAS LAS DIFERENCIAS Y SE -
                         PROCEDE A APLICAR LAS FORMULAS.
DO 3 I=2,5
D2(I-1)=D1(I)-D1(I-1)
3 CONTINUE
BC=D2(2)-D2(1)
IF(BC.EQ.0) GOTO 7
DO 4 I=2,4
D3(I-1)=D2(I)-D2(I-1)
4 CONTINUE
BD=D3(2)-D3(1)
IF(BD.EQ.0) GOTO 7
DO 5 I=2,3
D4(I-1)=D3(I)-D3(I-1)
5 CONTINUE
BE=D4(2)-D4(1)
IF(BE.EQ.0) GOTO 7
D5=BE
GOTO 7
7 DO 6 I=1,3
FD(I)=0.0
6 CONTINUE

LAS SIGUIENTES TRES LINEAS OBTIENEN LA PRIMERA, LA SEGUNDA Y LA -
TERCERA DERIVIDA RESPECTIVAMENTE, DE LA FUNCION TABULADA QUE SE -
DIO, CUYOS RESULTADOS SE ALMACENAN EN UN VECTOR.

FD(1)=(1/H)*(D1(1)-(0.5*D2(1))+(1/3*D3(1))-(0.25*D4(1))+(0.2*D5))
FD(2)=(1/(H**2))*(D2(1)-D3(1)+(11/12*D4(1))-(5/6*D5))
FD(3)=(1/(H**3))*(D3(1)-(3/2*D4(1))+(87/60*D5))
STOP
END
```

```
      PROGRAM RAICES
      DIMENSION A(20),B(20),C(20),VECS(20)
      COMPLEX A,B,C,VECS,X,Y      ! SE DEFINEN LAS MATRICES Y LAS VARIA--
                                    BLES QUE SE USARAN EN FORMA COMPLEJA
                                    YA QUE SE CALCULAN TAMBIEN LAS RAICES
                                    COMPLEJAS.

      VERIFICACION DE QUE EL GRADO DE LA ECUACION SEA EL ADECUADO, ENTRE
      DOS Y VEINTE, SI NO ES ASI, MANDA UN MENSAJE DE ERROR

      IF(NGRAD.LT.2.OR.NGRAD.GT.20) GOTO 100
      N=NGRAD
      M=N+1
      NRAIZ=1
      APROX=0.00000001    ! VALOR MAXIMO ACEPTABLE DE APROXIMACION.
      NMI=1000            ! VALOR MAXIMO DE ITERACIONES PERMITIDAS.

      CICLO QUE EFECTUA LA DIVISION SINTETICA PARA REDUCIR EL GRADO DE -
      LA ECUACION E IR OBTENIENDO LAS RAICES.

      DO 8 NRAIZ=1,NGRAD
      X=(0.,1.)     ! VALOR INICIAL DEL DIVISOR.
      L=1
   20 B(1)=A(1)     ! VECTOR QUE CONTIENE LOS COEFICIENTES DEL POLINOMIO
                      COCIENTE.

      LOS SIGUIENTES CICLOS REALIZAN LAS OPERACIONES BASICAS DE LA DI-
      VISION PARA OBTENER LA REDUCCION DEL POLINOMIO.

      DO 3 I=2,M
      B(I)=A(I)+X*B(I-1)
    3 CONTINUE
      C(1)=B(1)
      DO 4 I=2,N
      C(I)=B(I)+X*C(I-1)
    4 CONTINUE
      Y=X-B(M)/C(N)
      REL=ABS((X-Y)/Y)   ! APROXIMACION RELATIVA
      IF(REL.LE.APROX) THEN    ! SE VERIFICA QUE SE HAYA LLEGADO AL VALOR
                                 MAXIMO ACEPTADO DE LA APROXIMACION RELA-
                                 TIVA
      L=NMI
      ENDIF
      X=Y
      L=L+1
      IF(L.GT.NMI) GOTO 6    ! SE VERIFICA QUE EL CICLO LLEGA AL NUMERO
                               MAXIMO DE ITERACIONES PERMITIDO PARA EN-
                               CONTRAR LAS RAICES DE LA ECUACION.
      GOTO 20
    6 VECS(NRAIZ)=Y
      DO 7 I=2,N
      A(I)=B(I)
    7 CONTINUE
      N=N-1
      M=M-1
    8 CONTINUE
      GOTO 103
  100 WRITE(*,101)
  101 FORMAT(2X,"ERROR: GRADO DE LA ECUACION NO ADECUADO")
  103 STOP
      END
```

```
      DIMENSION A(20,20),VSOL(20)
      MN=N+1
      IF(N.GT.20) GOTO 20  ! SOLO SE ACEPTAN 20 ECUACIONES Y 20 INCOG-
                           NITAS, SI ES MAYOR A ESTE NUMERO MANDA MEN-
                           SAJE DE ERROR.

      CICLO PARA CHECAR QUE LA PRIMERA ECUACION NO TENGA COEFICIENTE -
      CERO PARA LA COLUMNA QUE SE PROCESA, SI ES CERO SE MANDA AL CICLO
      DE INTERCAMBIO DE RENGLONES.

      DO 7 J=1,N
      DO 8 I=J,N
      IF (A(I,J).NE.0) GOTO 9
    8 CONTINUE
      GOTO 25

      CICLO PARA INTERCAMBIAR RENGLONES, SI ES NECESARIO, SI NO SOLO REA
      COMODA LOS VALORES EN SU POSICION ORIGINAL.

    9 DO 11 K=1,MN
      S=A(J,K)
      A(J,K)=A(I,K)
      A(I,K)=S
   11 CONTINUE

      CICLO UTILIZADO PARA OBTENER UN COEFICIENTE DE UNO EN LA PRIMER -
      COLUMNA QUE NO TENGA CERO.

      Y=1/A(J,J)
      DO 12 K=1,MN
      A(J,K)=Y*A(J,K)
   12 CONTINUE

      CICLO UTILIZADO PARA RESTAR LA ECUACION EN PROCESO, DE LOS OTROS
      RENGLONES.

      DO 13 I=1,N
      IF (I.EQ.J) GOTO 13
      Y=-A(I,J)
      DO 15 K=1,MN
      A(I,K)=A(I,K)+Y*A(J,K)
   15 CONTINUE
   13 CONTINUE
    7 CONTINUE     ! ESTE PROCESO ES REPETIDO PARA TODAS LAS ECUACIONES.

      CICLO UTILIZADO PARA MEJORAR LA EXACTITUD DE LOS RESULTADOS.

      DO 14 I=1,N
      A(I,MN)=(A(I,MN)*1000+0.5)/1000
      VSOL(I)=A(I,MN)
   14 CONTINUE
      GOTO 50
   20 WRITE(*,100)
      GOTO 50
   25 WRITE(*,101)
  100 FORMAT(2X,"ERROR:NUMERO DE ECUACIONES E INCOGNITAS MAYOR AL PERMITIDO")
  101 FORMAT(2X,"ERROR:SOLUCION NO UNICA")
   50 STOP
      END
```

```
      PROGRAM FACTORIAL
      IF(N.LT.0) GOTO 3    ! SOLO SE ACEPTAN ENTEROS POSITIVOS
      L=1
      M=0
    2 IF(M.EQ.N) GOTO 4
      L=L*(M+1)            ! ACUMULACION DEL PRODUCTO PARA OBTENER EL RE-
                             SULTADO.
      M=M+1
      GOTO 2
    3 WRITE(*,100)
  100 FORMATO(2X,"ERROR: NO SE ACEPTAN NUMEROS NEGATIVOS")
    4 STOP
      END
```

Procedimientos de Actualización

El procedimiento de actualización tiene como objeto mostrar al analista la manera en que se puede dar mantenimiento a la biblioteca, es decir, como insertar, borrar, o reemplazar una rutina de la misma.

El equipo de cómputo utilizado, ECLIPSE MV4000, cuenta con un editor de bibliotecas llamado AOS Liberary File Editor (LFE. como se llamará en lo sucesivo), se usa para crear, editar y analizar librerias.

LFE trabaja en paralelo con el comando Link. Sus -- funciones son:

- Analizar el contenido de una biblioteca.
* Borrar, insertar o reemplazar los módulos objetos o -- rutinas en una biblioteca.
- Extraer módulos objeto de una biblioteca.
- Listar los títulos de todos los módulos objeto de una biblioteca.
- Acomodar dos o más bibliotecas en una nueva.

Para llamar a LFE, se necesita escribir el comando XEQ (abreviado X) del AOS CLI, seguido por LFE y la fun-- ción-letra apropiada, switches y archivos objetos. Una

función-letra es un comando de LFE.

Para cumplir con el objetivo establecido para éste punto, se analizará solamente las funciones de borrar, insertar y reemplazar, cuyas función-letra son:

D (Borrar)

I (Insertar)

R (Reemplazar)

1.- Borrar una rutina de la biblioteca. (D)

Formatos:

X LFE/I =biblioteca ⟨/DELETE⟩ D título...

X LFE ⟨/DELETE⟩ D biblioteca/I biblioteca/O título...

X LFE ⟨/DELETE⟩ D biblioteca biblioteca/O título...

donde:

biblioteca/I o /I=biblioteca	Es el nombre de la biblioteca de entrada. Si se usa el 3er. formato, la /I es opcional.
biblioteca/O o /O=biblioteca	Es el nombre de la biblioteca de salida.
/DELETE	Es un switch especial que hace que LFE borre la biblioteca de salida antes de crear una nueva. Se usa cuando se quiere que la biblioteca de salida tenga el mismo nombre que la de entrada.

D Es la función-letra.

TITULO Es el nombre o nombres de una ó más rutinas en la biblioteca que desea borrar.

Ejemplos:

1) X LFE D METNUM.LB/I NMETNU.LB/O MULMAT

Se crea una nueva biblioteca NMETNU a partir de METNUM, conteniendo todas las rutinas excepto -- MULMAT.

2) X LFE/I=METNUM.LB/O=METNUM.LB/DELETE D MULMAT

Se crea una biblioteca con el mismo nombre de la entrada, pero antes de crear la que no contenga la rutina MULMAT, el switch/DELETE la borra para no causar error.

2.- Insertar una ó más rutinas en una libreria. (I)

Formatos:

X LFE/I=biblioteca/O=biblioteca /DELETE I título/A ó título/B archivo objeto...

X LFE /DELETE I biblioteca/I biblioteca/O título/A ó título/B archivo objeto...

X LFE /DELETE I biblioteca biblioteca/O título /A ó título /B archivo objeto...

donde:

biblioteca/I o /I=biblioteca	Es el nombre de la biblioteca de entrada, es opcional, si se usa el 3er. formato.
biblioteca/O o /O=biblioteca	Es el nombre de la biblioteca de salida.
/DELETE	Es un switch opcional, hace que LFE borre la biblioteca de salida antes de crear una nueva. Se usa cuando el nombre de la biblioteca de entrada sea el mismo que el de salida.

I Es la función letra.

TITULO Es el nombre de una rutina en la biblioteca de entrada, el título puede estar seguido por /A ó /B. Si es /A, LFE inserta el archivo (S) objeto en la biblioteca inmediatamente después del nombre de la rutina dado. Si es /B, LFE inserta el archivo(s), objeto en la biblioteca

Archivo Objeto El archivo objeto insertado.

Se inserta uno ó más rutinas en la biblioteca, para formar una nueva de salida.

Nótese la diferencia entre título y archivo objeto. un título es el nombre de módulo objetivo o rutina, y un archivo objeto es el nombre del archivo a insertar.

Ejemplo:

1) X LFE/I=METNUM.LB/O=NMETNU.LB I MULMAT/A SUMMAT.OB

Se inserta la rutina SUMMAT en la biblioteca METNUM después de la rutina MULMAT, obteniendo la biblioteca NMETNU.

3.- Reemplaza una rutina de una biblioteca con una rutina. (R)

Formatos:

X LFE/I=biblioteca/O=biblioteca /DELETE R título archivo objeto...

X LFE /DELETE R biblioteca/I biblioteca/O título archuvo objeto...

X LFE /DELETE R biblioteca biblioteca/O título archivo objeto...

donde:

biblioteca/I o /I=biblioteca	Es el nombre de la biblioteca de entrada. Si se usa el 3er. formato la /I es opcional.
biblioteca/O o /O=biblioteca	Es el nombre de la biblioteca de salida (Una versión modificado de el de entrada).
/DELETE	Es un switch opcional, que hace que LFE borre la biblioteca de salida antes de crear una nueva.

R	Es la función-letra.
TITULO	Es el nombre de la rutina que se quiere reemplazar.
Archivo Objeto	Es el nombre de uno ó más archivos objeto con los que se desea reemplazar la rutina.

Ejemplo:

```
X LFE  R METNUM.LB/I NMETNU.LB/O  MULMAT  MULM2.OB
```

Se desea reemplazar la rutina MULMAT de la biblioteca METNUM, por MULM2.OB obteniendo la biblioteca NMETNU que contendrá el cambio realizado.

Nota: Para cada comando I ó R, debe existir una correspondencia de uno a uno cuando se trate de varias rutinas a insertar o reemplazar.

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

Al término del presente proyecto, se ven cumplidos los objetivos particulares, como el general. Su realización servirá a aquellas personas que desarrollen programas relacionados con el uso de métodos numéricos o graficadores, ya que facilitará su codificación, acortará el número de instrucciones, y el tiempo de desarrollo de éstos programas.

Esta biblioteca servirá también para el desarrollo futuro de nuevas rutinas que se vayan requiriendo y estén relacionadas con el tema que tienen en común las ya existentes, pués cuenta con ésta opción, y son mínimas las que se han implantado en éste proyecto.

Dá también la pauta a seguir para la realización de nuevas bibliotecas, necesitadas con diferentes objetivos, en cuanto al tema que se requiere desarrollar.

Todo esto para aprovechar más los beneficios que la computadora nos pueda brindar, poniéndolos al servicio de las personas que lo requieran.

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

- Lehman, Charles H. Algebra.
  Editorial Limusa - Wiley. México, 1964.

- Luthe, Rodolfo; Olivera, Antonio; Schutz, Fernando.
  Métodos Numéricos.
  Editorial Limusa. México, 1984.

- McCalla, Thomas R. Introduction to Numerical Methods and FORTRAN programming.
  Editorial John Wiley & Sons, Inc. U.S., 1967.

- Manual de Usuario (AOS Y LFE) para Data General.

- Riggs James L. Sistemas de Producción: Planeación, Análisis y Control.
  Editorial Limusa. México, 1976.

- Sippl, Charles J. Computer Dictionary & Hand Book.
  Editorial Howard W. Sams & Co. Inc., 3o. Edición, U.S., 1980.

- Wylie, C.R. Matemáticas Superiores para Ingeniería.
  Editorial Libros McGraw - hill. España. 1976.

A N E X O 1

RUTINA FMULMAT

FORMATO:

FMULMAT(I1,J1,I2,J2,A,B,R)

Esta rutina se utiliza para obtener la multiplicacion de dos matrices. A continuacion se define cada uno de los parametros a utilizar:

I1 Es el numero de renglones de la primer matriz. Puede ser variable o constante.

J1 Es el numero de columnas de la primer matriz. Puede ser variable o constante.

I2 Es el numero de renglones de la segunda matriz. Puede ser variable o constante.

J2 Es el numero de columnas de la segunda matriz. Puede ser variable o constante.

A Es el nombre de la primer matriz.

B Es el nombre de la segunda matriz.

R Es el nombre de la matriz resultante.

La maxima dimension de una matriz es de 20 renglones y 20 columnas.

EJEMPLOS:

FMULMAT(5,8,8,6,X,Y,Z)

Se define la multiplicacion de la matriz X, con cinco renglones y ocho columnas; y la matriz Y, con ocho renglones y seis co-- columnas; dejando la matriz resultante en Z.

FMULMAT(K,L,M,N,F,G,H)

Se define la multiplicacion de la matriz F, con K renglones y - L columnas; y la matriz G, con M renglones y F columnas; dejando la matriz resultante en H.

RUTINA FINVMAT

FORMATO:

FINVMAT(N,A,R)

Esta rutina se utiliza para obtener la inversion de una matriz cuadrada. A continuacion se define cada uno de los parametros que se utilizan:

N Es el numero de renglones y de columnas de la matriz cuya inversion se desea obtener. Puede ser variable o constante. Se pide solamente un valor, ya que la matriz debe ser cuadrada.

A Es el nombre de la matriz.

R Es el nombre de la matriz resultante.

La maxima dimension de la matriz es de 20 renglones y 20 columnas.

EJEMPLOS:

FINVMAT(4,X,Z)

Se define la inversion de la matriz X, con cuatro renglones y cuatro columnas; dejando la matriz resultante en Z.

FINVMAT(J,Y,R)

Se define la inversion de la matriz Y, con J renglones y J columnas; dejando la matriz resultante en R.

RUTINA FDETMAT

FORMATO:

FDETMAT(N,A,R)

Esta rutina se utiliza para obtener el determinante de una matriz cuadrada. A continuacion se define cada uno de los parametros que se utilizan:

N Es el numero de renglones y de columnas de la matriz cuya determinante se desea obtener. Puede ser variable o constante. Se pide solamente un valor ya que la matriz debe ser cuadrada.

A Es el nombre de la matriz.

R Es el nombre de la variable donde se almacena el valor de la determinante.

La maxima dimension de la matriz es de 20 renglpones y 20 columnas.

EJEMPLOS:

FDETMAT(12,X,D)

Se define la determinante de la matriz X con doce renglones y doce columnas; dejando el valor resultante en D.

FDETMAT(M,AM,R)

Se define la determinante de la matriz AM con M renglones y M columnas; dejando el valor resultante en R.

FUNCION FNUMAZA

FORMATO:

R=FNUMAZA(SEM)

Esta funcion se utiliza para obtener un numero al azar a partir de una semilla dada, con la que se realizan ciertas operaciones, dando como resultado un numero que se encuentra dentro del rango (0,1). A continuacion se define cada parametro a utilizar:

SEM — Es el numero a partir del cual se genera un numero al azar Puede ser variable o constante entera.

R — Es el nombre de la variable donde se almacena el valor resultante.

EJEMPLOS:

R=FNUMAZA(S)

Se obtiene un numero al azar R, a partir del numero o semilla dado, definido como S.

R=FNUMAZA(456)

Se obtiene un numero al azar R, a partir del numero o semilla dado como 456.

RUTINA FRUTCRI

FORMATO:

FRUTCRI(NA,II,J,D,A,R,DT)

Esta rutina se utiliza para obtener la ruta critica de cualquier proyecto representado por medio de una red de flechas, ya que cada - actividad se representa como un flecha dentro de la red, la cual tiene un origen y un destino, que son llamados eventos, y tiene cierta duracion dentro del proyecto total; esta informacion es requerida para la obtencion de la ruta critica buscada. A continuacion se defi-- nen los parametros que se utilizan:

NA — Es el numero de actividades en el proyecto incluyendo las ficticias, si estas son requeridas en la red.

II — Nombre del vector donde se almacena la lista de eventos - desde donde sale cada actividad (ii,j).

J — Nombre del vector donde se almacena la lista de eventos a donde llega cada actividad (ii,j).

D — Nombre del vector donde se encuentra la lista de la duracion de cada actividad (ii,j).

A — Nombre del vector donde se encuentra la lista de actividades representadas por una letra que muestra la actividad - (ii,j).

R — Nombre del vector donde se almacena la ruta critica resultante del proyecto en cuestion.

DT — Nombre de la variable donde se almacena la duracion total resultante de todo el proyeto.

EJEMPLO:

FRUTCRI(L,KI,KJ,DA,AC,RC,DP)

Se define la ruta critica RC y la duracion total DP del proyecto, dado que se tienen como datos: KI, vector con L eventos de salida de actividad; KJ, vector con L eventos de llegada; DA, - vector con L valores de duracion; AC, vector con L actividades.

RUTINA FGRABAR

FORMATO:

FGRABAR(NCO,ICY,MATB,IEY)

Esta rutina se utiliza para obtener la grafica del comportamiento de algo durante un periodo de tiempo, esto es lo que conocemos como histrograma, que se grafica a partir de un numero de periodos y valores que tiene en cada periodo dados. A continuacion se definen los parometros utilizados:

NCO — Es el numero de periodos durante el cual se realizara el histograma. Puede ser variable o constante entera positiva menor que 35.

ICY — Es el nombre del vector donde se encuentra la informacion relacionada con los valores que toma la estadistica realizada. Deben ser valores enteros positivos.

MATB — Es el nombre de la matriz donde se almacena la grafica resultante, que si se desea se puede observar por pantalla o por impresora, dejando este procedimiento para que sea desarrollado por el usuario.

IEY — Nombre de la variable donde se almacena el valor de la escala que se realizo.

EJEMPLO:

FGRABAR(N,I,NG,ME)

Se define la matriz MG, como la grafica de un histograma, analizando N periodos, cada uno con sus valores estadisticos se especifica en el valor I, y su escala en 1:ME.

RUTINA FGRAPUN

FORMATO:

FGRAPUN(NGR,A9,MATG,IES)

Esta rutina se utiliza para obtener la grafica de una funcion del tipo polinomial, por medio de puntos. A continuacion se definen los parametros utilizados:

NGR — Es el grado de la funcion que se desea graficar, debe ser menor que 20. Puede ser variable o constante entera positiva.

A9 — Es el vector donde se encuentran los coeficientes de la funcion.

MATG — Es el nombre de la matriz donde se almacena la grafica resultante. Si el ususario la desea ver por pantalla o impresora, debera hacer el procedimiento necesario, imprimiendo la matriz por renglones.

IES — Es el nombre de la variable donde se almacena el valor de la escala realizada.

EJEMPLO:

FGRAPUN(N,C,MS,ME)

Se define la matriz MS, como la grafica de la funcion de grado N cuyos coeficientes estan definidos en el vector C. La escala realizada es 1:ME.

RUTINA FDERIVA

FORMATO:

FDERIVA(X,Y,FD)

Esta rutina se utiliza para obtener la primera, segunda y tercera derivada numerica de una funcion (x,y), a partir de su tabulacion dada, aplicando formulas de interpolacion. A continuacion se definen los parametros que se utilizan:

X — Nombre del vector que contiene la informacion dada, referente a los valores que toma X, debe existir un intervalo constante entre estos valores.

Y — Nombre del vector que contiene la informacion dada, referente a los valores que toma Y.

FD — Nombre del vector resultante, donde se almacena la primera, segunda y tercera derivada numerica obtenidas.

Tanto el vector X, como el Y, deben tener una dimension maxima de 6, por ser este el numero de periodos que se necesitan para obtener los resultados deseados.

EJEMPLO:

FDERIVA(TX,TY,DS)

Se define el vector DS como el resultado de la aplicacion de la formula de interpolacion, para derivar numericamente una funcion utilizando como datos de operacion la tabulacion de la misma, siendo TX el vector que contiene los valores que toma X, y TY el vector que contiene los valores de Y.

RUTINA FRAICES

FORMATO:

FRAICES(NG,A,R)

Esta rutina se utiliza para obtener todas las raices posibles de una ecuacion polinomial, tanto las reales como las complejas, a partir del grado de la ecuacion y los coeficientes de la misma. A continuacion se definen los parametros que se utilizan:

NG — Es el grado de la ecuacion. El maximo valor del grado es 20. Puede ser variable o constante entera positiva.

A — Es el nombre del vector donde se encuentran los coeficientes de la ecuacion, los valores deben ser reales. Deben contener tanto los coeficientese reales como los complejos, estos seran siempre cero, pero para efectos de ejecucion son necesarios.

R — Es el nombre del vector donde se almacena el resultado de las raices reales y complejas, su acomodo es de una real y una compleja, igual que como deben estar acomodados los coeficientes.

EJEMPLO: $2x^2+3x+6=0$

FRAICES(N,VC,R)

Se define el vector R como las raices reales y complejas de una ecuacion polinomial de grado N, cuyos coeficientes estan dados en el vector VC, que ademas de tener los coeficientes reales, tiene tambien las complejas (2,0,3,0,6,0).

RUTINA FECUSIM

FORMATO:

FECUSIM(N,A,VSOL)

Esta rutina se utiliza para obtener la solucion de un numero dado de ecuaciones, con el mismo numero de incognitas; esto es lo que conocemos como ecuaciones simultaneas, y como tal se resuelven. A continuacion se definen los parametros que se utilizan:

N — Es el numero de ecuaciones y de incognitas. Puede ser variable o constante. Debe ser entero positivo.

A — Es el nombre de la matriz que contiene los coeficientes de de las incognitas de las ecuaciones dadas.

VSOL — Es el nombre del vector resultante, donde se almacena el valor solucion para cada incognita de las ecuaciones dadas.

La maxima dimension de la matriz A, es de 20 renglones y 20 columnas, esto es, 20 ecuaciones con 20 incognitas. Por lo tanto el vector VSOL tambien tiene una dimension de 20 elementos.

EJEMPLO:

FECUSIM(J,DAT,R)

Se define el vector solucion R, con J elementos, para J ecuaciones y J incognitas, donde el valor de los coeficientes de las ecuaciones se almacena en DAT.

FUNCION FFACTOR

FORMATO:

R=FFACTOR(N)

Esta funcion se utiliza para obtener el factorial de un numero - entero menor a 13. A continuacion se definen los parametros que se - utilizan:

N Es el numero del cual se obtiene el factorial. Puede ser - variable o constante entera positiva.

R Es el nombre de la variable donde se almacena el valor del - resultado.

EJEMPLO:

R=FFACTOR(I)

Se define la variable R, como el factorial de I.