

136



**UNIVERSIDAD NACIONAL AUTONOMA DE MÉXICO**

**FACULTAD DE INGENIERIA**

**GENERACIÓN DE UN ALGORITMO  
PARA DISEÑO DE OBSERVADORES  
LINEALES MULTIVARIABLES**

**T E S I S**

**PARA OBTENER EL GRADO DE :**

**DOCTOR EN INGENIERIA**

**P R E S E N T A**

**JOSE HORACIO SANDOVAL RODRIGUEZ**

**MÉXICO**

**TESIS CON  
FALLA DE ORIGEN**



Universidad Nacional  
Autónoma de México



**UNAM – Dirección General de Bibliotecas**  
**Tesis Digitales**  
**Restricciones de uso**

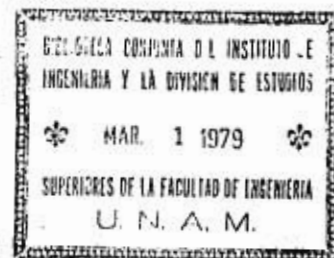
**DERECHOS RESERVADOS ©**  
**PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

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

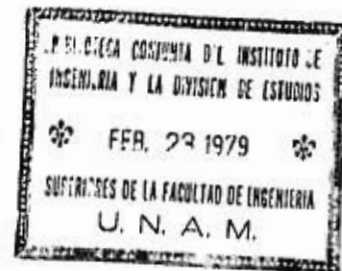
El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición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

GENERACION DE UN ALGORITMO PARA DISEÑO DE OBSERVADORES

LINEALES MULTIVARIABLES



EXAMEN DE GRADO DE MAESTRIA



Propuso: Ismael Espinosa E.

Realizó: J. Horacio Sandoval R.

ENERO DE 1978



UNIVERSIDAD NACIONAL  
AUTÓNOMA

DIVISION DE ESTUDIOS SUPERIORES  
FACULTAD DE INGENIERIA

CIUDAD UNIVERSITARIA  
APDO. POSTAL 70-256

MEXICO 20, D. F.  
TEL. 548.58-77

0096

01149  
136

MEMORANDO

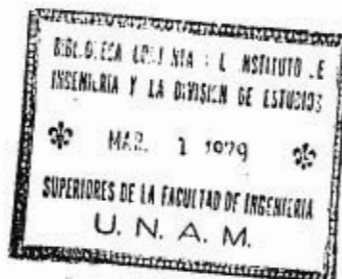
Para: M. en C. Luis Marcial Hernández Ortega  
De: Ismael Espinosa E.

El tema de examen de grado de maestría en control para el ingeniero

JOSE HORACIO SANDOVAL RODRIGUEZ

es el siguiente:

GENERACION DE UN ALGORITMO PARA DISEÑO  
DE OBSERVADORES LINEALES MULTIVARIABLES.



Condiciones:

- i) Lenguaje: FORTRAN IV Versión B6700
- ii) Que acepte sistemas hasta de orden 10 con n entradas y m salidas
- iii) Que verifique controlabilidad y observabilidad
- iv) Que utilice un método sencillo (por ej. el de Fallside) para asignar los polos al observador
- v) Que permita cambiar fácilmente la matriz de ubicación de polos, para así poder escoger el mejor observador
- vi) Que grafique las variables de estado, tanto las disponibles como las estimadas
- vii) El programa deberá estar disponible en disco y verificado su funcionamiento con algunos ejemplos.

El plazo máximo considerado adecuado es de un mes.

Atentamente

Cd. Universitaria, a 4 de Noviembre de 1977

1. INTRODUCCION	1
2. CONTROLABILIDAD Y OBSERVABILIDAD	3
3. ALGORITMO DE LUENBERGER PARA LA CONSTRUCCION DEL OBSERVADOR	8
4. BASE TEORICA DEL PROGRAMA (METODOS NUMERICOS)	16
5. BIBLIOGRAFIA	26
A.1 SOLUCION DE LA ECUACION DE ESTADO (SISTEMA Y OBSERVADOR)	29
A.2 LISTADO DEL PROGRAMA	
A.3 EJEMPLOS	

## 1. INTRODUCCION

En el análisis de sistemas dinámicos uno de los objetivos principales es lograr que el sistema responda de alguna forma determinada. Para un sistema dado esto puede lograrse en ocasiones. La respuesta a cuando es posible llevar un sistema o a las variables de estado que lo definen a ciertos valores predeterminados, se encuentra en el concepto de *Controlabilidad*, el cual fue introducido por Kalman en la década de los 60.

Una vez que la condición de Controlabilidad Absoluta se ha satisfecho, surge la pregunta de cómo lograr el control necesario para poder llevar el sistema a un cierto estado. Una de las técnicas más comunes es la realimentación de estado, es decir, a la excitación original se agrega el vector de estado.

Esta técnica presenta un nuevo problema cuando una o más de

las variables de estado no puede medirse directamente, y por tanto no puede realimentarse.

Esta dificultad implica la necesidad de contar al menos con alguna estimación de las variables de estado que faltan, o en general, de todo el estado.

El concepto de *Observabilidad* resuelve cualitativamente esta dificultad, ya que permite saber si dado el sistema y sus salidas es posible recuperar la información de todo el vector de estado.

Debe notarse que los conceptos de Controlabilidad y Observabilidad indican si el sistema se puede controlar y observar, pero no indican cómo lograr ese control y esa observación, respectivamente.

En el presente escrito se tratará únicamente de los sistemas necesarios para estimar el estado, también llamados, *Observadores*. Se analizará un método en particular, el de Luenberger. También se describirá el programa para computadora digital que genera los observadores con el método indicado además de resolver la ecuación de estado del sistema y del observador.

tico de su matriz de transferencia  $G(s)$ , donde

$$G(s) \triangleq c(sI - A)^{-1} B + D$$

o incluyendo la definición de la matriz de transición en el dominio complejo  $s$

$$G(s) \triangleq c \phi(s) B + D$$

Este resultado se obtiene de tomar la transformada de Laplace de las ecs 1 y 2, y eliminar el vector de estado  $x$  para llegar a la expresión

$$y(s) = G(s) u(s)$$

## II) Criterio de Wolovich

Si se representa un sistema dinámico con operadores diferenciales de la siguiente manera

$$P(D) x(t) = Q(D) u(t)$$

$$y(t) = R(D) z(t) + T(D) u(t)$$

será controlable, si y solo si,  $P(D)$  y  $Q(D)$  son matrices primo izquierda relativas, y además será observable, si y solo si,  $P(D)$  y  $R(D)$  son primo derecho relativas.

## III) Criterio de Gilbert

Para aplicar este criterio se requiere llevar las ecs 1 y 2 del sistema a la forma normal



## 2. CONTROLABILIDAD Y OBSERVABILIDAD

2.1 Los planteamientos formales de controlabilidad y observabilidad son los siguientes:

Controlabilidad.- Si un sistema se caracteriza por la ecuación de estado

$$\dot{x}(t) = Ax(t) + Bu(t) \quad (1)$$

donde

- A matriz ( $n \times n$ ), llamada matriz de coeficiente
- B matriz ( $n \times r$ ), llamada matriz de distribución
- $u(t)$  vector de excitaciones ( $r \times 1$ )
- $x(t)$  vector de estado ( $n \times 1$ )

se dice que es completamente controlable si el estado  $x(t)$  para  $t = t_0$  puede llevarse mediante alguna entrada  $u(t)$  a un estado final cualquiera  $x(t_f)$  en un tiempo finito  $(t_f - t_0) \geq 0$ .

Observabilidad.- Si además de la ec (1) se tiene la ecuación de salida

$$y(t) = cx(t) + Du(t) \quad (2)$$

donde

- C matriz ( $m \times n$ ), llamada matriz de salida
- D matriz ( $m \times r$ ), llamada matriz de transmisión
- $y(t)$  vector de salida ( $m \times 1$ )

se dice que el estado  $x(t_0)$  es completamente observable si para cualquier entrada  $u(t)$  existe un tiempo finito  $t_f \geq t_0$  tal que es posible deducir  $x(t_0)$  a partir de la información dada por  $y(t)$  en ese intervalo.

2.2 Existen varias formas para definir la controlabilidad y observabilidad de un sistema a partir de las ecs (1) y (2), aquí se indicaran solo cuatro de ellas

I) Análisis de la Observabilidad y Controlabilidad empleando la matriz de transferencia. (Método de Chen)

El sistema lineal e invariante en el tiempo definido por las ecuaciones (1) y (2) será controlable y observable, si y solo si, el orden de la matriz A es igual al grado de la matriz de transferencia  $G(s)$ , lo anterior equivale a decir que el sistema será controlable y observable, si y solo si, el polinomio característico de la matriz A es igual al polinomio caracterís

tico de su matriz de transferencia  $G(s)$ , donde

$$G(s) \triangleq c(sI - A)^{-1} B + D$$

o incluyendo la definición de la matriz de transición en el dominio complejo  $s$

$$G(s) \triangleq c \phi(s) B + D$$

Este resultado se obtiene de tomar la transformada de Laplace de las ecs 1 y 2, y eliminar el vector de estado  $x$  para llegar a la expresión

$$y(s) = G(s) u(s)$$

## II) Criterio de Wolovich

Si se representa un sistema dinámico con operadores diferenciales de la siguiente manera

$$P(D) x(t) = Q(D) u(t)$$

$$y(t) = R(D) z(t) + T(D) u(t)$$

será controlable, si y solo si,  $P(D)$  y  $Q(D)$  son matrices primo izquierda relativas, y además será observable, si y solo si,  $P(D)$  y  $R(D)$  son primo derecho relativas.

## III) Criterio de Gilbert

Para aplicar este criterio se requiere llevar las ecs 1 y 2 del sistema a la forma normal

$$\dot{z}(t) = \Delta z(t) + \beta u(t)$$

$$y(t) = \gamma z(t) + D u(t)$$

al transformar de coordenadas mediante la matriz modal asociada a la matriz A, entonces el sistema será controlable si  $\beta$  no tiene renglones nulos y será observable si  $\gamma$  no tiene columnas nulas.

#### IV) Criterio de Kalman

Para la aplicación de este criterio se requiere la ecuación de estado y la de salida con las cuales se forman las siguientes matrices

$$P \triangleq [B \ AB \ A^2B \ \dots \ A^{n-1}B]$$

$$Q \triangleq [C^T \ A^T C^T \ (A^T)^2 C^T \ \dots \ (A^T)^{n-1} C^T]$$

donde

P matriz ( $n \times nr$ ) y es llamada matriz de controlabilidad

Q matriz ( $n \times mn$ ) y es llamada matriz de observabilidad

Este criterio indica que el sistema será completamente controlable, si y solo si, el rango de la matriz P es igual al orden del sistema, en este caso n.

Así mismo, el sistema será completamente observable, si y solo si, el rango de la matriz Q es igual al orden del sistema, en este caso n.

Como se ve el criterio más simple de aplicar es el de Kalman, aunque el más claro conceptualmente es el de Gilbert. Debe mencionarse que estos son algunos de los criterios existentes solamente, y que el criterio de Kalman es el que se usa con mayor frecuencia, y es el que se incluyó en el programa.

### 3. ALGORITMO DE LUENBERGER PARA LA CONSTRUCCION DEL OBSERVADOR

Existen en la literatura bastantes algoritmos para la generación de los observadores, cada uno de ellos tiene sus propias características y en la mayoría de los casos son de aplicación particular, es decir, han sido desarrollados para una necesidad específica.

Aquí se describirá el algoritmo desarrollado por Luenberger el cual no es, necesariamente el óptimo en procesamiento o en resultados, aunque si es el más divulgado.

#### *3.1 Algoritmo de Luenberger*

Este algoritmo requiere que el sistema analizado sea completamente observable, y a partir de las entradas y salidas estima

el vector de estado.

De lo anterior se tiene que el observador es un sistema lineal e invariante en el tiempo cuyos controles son las entradas y salidas del sistema que observa y cuya salida es la estimación del vector de estado, tal como se muestra en la figura 1.

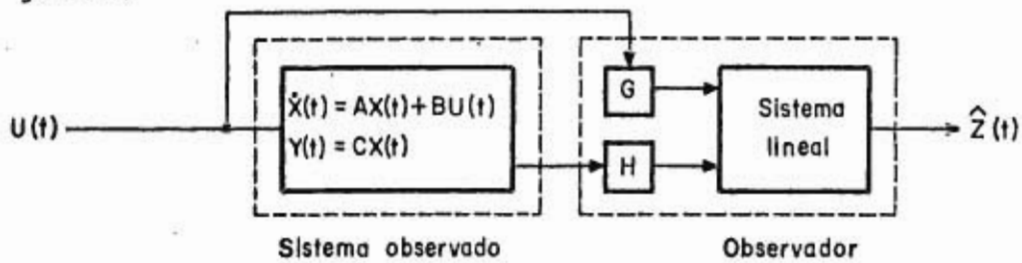


Figura 1

Dado que el observador es un sistema dinámico lineal, se puede escribir su ecuación de estado como

$$\frac{d\hat{z}(t)}{dt} = \tilde{A}\hat{z}(t) + \tilde{B}\tilde{u}(t) \quad (4)$$

donde

$\tilde{A}$  matriz  $(n \times n)$

$\tilde{B}$  matriz  $[n \times (r + m)]$

$\tilde{u}(t)$  vector de  $(r + m) \times 1$

$\hat{z}$  vector  $n \times 1$

$$\tilde{u}(t) = \begin{bmatrix} u(t) \\ y(t) \end{bmatrix}$$

$$\tilde{B} = [G \ H]$$

o de manera equivalente

$$\frac{d\hat{z}(t)}{dt} = \tilde{A}\hat{z}(t) + Gu(t) + Hcx(t) \quad (5)$$

donde

G matriz  $n \times r$

H matriz  $n \times m$

Se desea llegar a un vector  $z(t)$  cuyos elementos sean combinaciones lineales de las variables de estado del sistema, de manera que

$$z(t) = T x(t) \quad (6)$$

Por tanto,  $\hat{z}(t)$  es la estimación de  $z(t)$ , lo cual involucra un error asociado en la estimación  $\Delta z(t)$  tal que

$$\Delta z(t) = \hat{z}(t) - z(t) \quad (7)$$

Ahora el problema consiste en definir las matrices  $\tilde{A}$ , G y H que minimicen el error  $\Delta z(t)$ . Luenberger formó el siguiente procedimiento:

Derivando la ec 6

$$\begin{aligned} \frac{dz(t)}{dt} &= T \frac{dx(t)}{dx} \\ &= T [Ax(t) + Bu(t)] \\ &= T Ax(t) + T Bu(t) \end{aligned}$$



restando esta última igualdad a la ec 5, se tiene

$$\frac{d\hat{z}(t)}{dt} - \frac{dz(t)}{dt} = \hat{A}\hat{z} - Ax(t) + [G - TB]u(t) + [HC - TA]x(t) \quad (8)$$

recordando la ec 7 se tiene

$$\begin{aligned} \tilde{A}\Delta z(t) &= \tilde{A}\hat{z} - \tilde{A}z \\ \tilde{A}\Delta z(t) &= \tilde{A}\hat{x}(t) - \tilde{A}x(t) \\ \therefore \tilde{A}\hat{z}(t) &= \tilde{A}\Delta z(t) + \tilde{A}z(t) \\ &= \tilde{A}\Delta z(t) + \tilde{A}Tx(t) \end{aligned}$$

sustituyendo en la ec 8

$$\frac{d}{dt} \Delta z(t) = \tilde{A}\Delta z(t) + [G - TB]u(t) + [HC - TA + \tilde{A}T]x(t) \quad (9)$$

si en esta última ecuación se obliga que

$$\begin{aligned} \tilde{A}T + HC &= TA \\ G &= TB \end{aligned} \quad (10)$$

resulta  $\frac{d}{dt} \Delta z(t) = \tilde{A}\Delta z(t)$

que es un sistema de ecuaciones diferenciales de primer orden en  $\Delta z(t)$ , y cuya solución esta dada por

$$\Delta z(t) = e^{\tilde{A}t} \Delta z(t = t_0)$$

de manera que si la parte real de todos los valores propios de  $\tilde{A}$  es negativa se tiene

$$\Delta z(t) \rightarrow 0 \text{ cuando } t \rightarrow \infty$$

lo que implica que

$$\tilde{z}(t) \rightarrow z(t) = T x(t) \quad (11)$$

Dado que los polos del observador deben tener parte real negativa surge la pregunta ¿qué tan negativa? Analizando el comportamiento del observador y del sistema en el dominio del tiempo, vemos que es conveniente seleccionar los polos del observador a la izquierda de la región donde están los polos del sistema, de manera que el transitorio provocado por el observador, no se confunda o afecte apreciablemente el transitorio del sistema observado.

La limitación para no ir más hacia la izquierda, usualmente es de índole física, en el programa solo se considera la posición relativa de los polos del sistema y los del observador.

### 3.2 Elección de los polos del observador

La ecuación característica del observador es

$$\det (\lambda I - \tilde{A}) = |\lambda I - \tilde{A}| = 0 \quad (12)$$

y haciendo la matriz T igual a la identidad, las ecuaciones 10 resultan

$$\tilde{A} = A - HC \quad (13)$$

$$G = B$$

sustituyendo en la ecuación 12

$$|\lambda I - A + HC| = 0$$

y tomando como factor común la matriz  $(\lambda I - A)$  se tiene

$$|(\lambda I - A) [I + (\lambda I - A)^{-1}HC]| = 0$$

$$|(\lambda I - A)| |I + (\lambda I - A)^{-1}HC| = 0$$

haciendo  $\phi(\lambda) \triangleq (\lambda I - A)^{-1}$

$$\text{resulta } |\lambda I - A| |I_n + \phi(\lambda) HC| = 0$$

y empleando la siguiente igualdad de determinantes

$$|I_n + KG| = |I_m + GK|$$

donde

$I_n$  identidad de orden  $n$

$I_m$  identidad de orden  $m$

$K$  matriz  $n \times m$

$G$  matriz  $m \times n$

se llega a

$$|\lambda I_n - A| |I_m + c\phi(\lambda) H| = 0$$

o en su lugar

$$|\lambda I_n - A| |I_m + s(\lambda) H| = 0 \quad (14)$$

donde

$$s(\lambda) \triangleq c\phi(\lambda)$$

Dado que los polos del observador son diferentes a los del sistema, entonces

$$|\lambda I_n - A| \neq 0$$

por tanto la única posibilidad para que se anule la ecuación es que el segundo factor sea cero es decir

$$|I_m + s(\lambda) H| = 0 \quad (15)$$

para cuando  $\lambda$  adquiere el valor de cualquier polo del observador.

Una manera de lograr que el determinante sea nulo, es mediante una columna o un renglón de ceros obtenidos a partir de la matriz  $H$ .

Sea el renglón  $j$ -ésimo el que se desea anular, y llamando

$\phi_j$  al  $j$ -ésimo renglón de la matriz identidad

$s_j(\lambda)$  al  $j$ -ésimo renglón de la matriz  $s(\lambda)$

entonces el  $j$ -ésimo renglón de la ec 15 queda

$$\phi_j - s_j(\lambda) H = 0$$

Esta expresión no es suficiente para determinar  $H$ , que posee  $n$  renglones, pero es posible obtener a partir de los polos del observador.

n renglones linealmente independientes uno para cada uno. Esto es una consecuencia de la observabilidad del par  $(A, C)$ , es decir de que sus modos linealmente independientes.

De esta manera se obtiene la matriz regular  $W$ .

$$W \triangleq [s_j(\lambda_1) \ s_j(\lambda_2) \ \dots \ s_j(\lambda_n)]^T$$

en donde  $s_j(\lambda_i)$  es el  $j$ -ésimo renglón de  $s(\lambda)$  asociado con  $\lambda_i$

Luego para el determinante 15 tenemos

$$[\phi_{j1} \ \phi_{j2} \ \dots \ \phi_{jn}] - [s_j(\lambda_1) \ s_j(\lambda_2) \ \dots \ s_j(\lambda_n)] = 0$$

donde  $\phi_{ji}$  es el  $j$ -ésimo renglón de  $I_r$  asociado con  $\lambda_i$

llamando 
$$\phi \triangleq [\phi_{j1} \ \phi_{j2} \ \dots \ \phi_{jn}] \quad (n \times m)$$

queda 
$$\phi + W H = 0$$

y dado que  $W$  es invertible (a menos que la estructura de  $s_j(\lambda_i)$  tenga elementos nulos y este renglón se use para todos los polos)

$$H = -W^{-1} \phi$$

Una vez definida la matriz  $H$ , se pueden obtener las matrices  $\tilde{A}$  para formar la ecuación de estado del observador y verificar si los polos del observador quedaron bien ubicados.

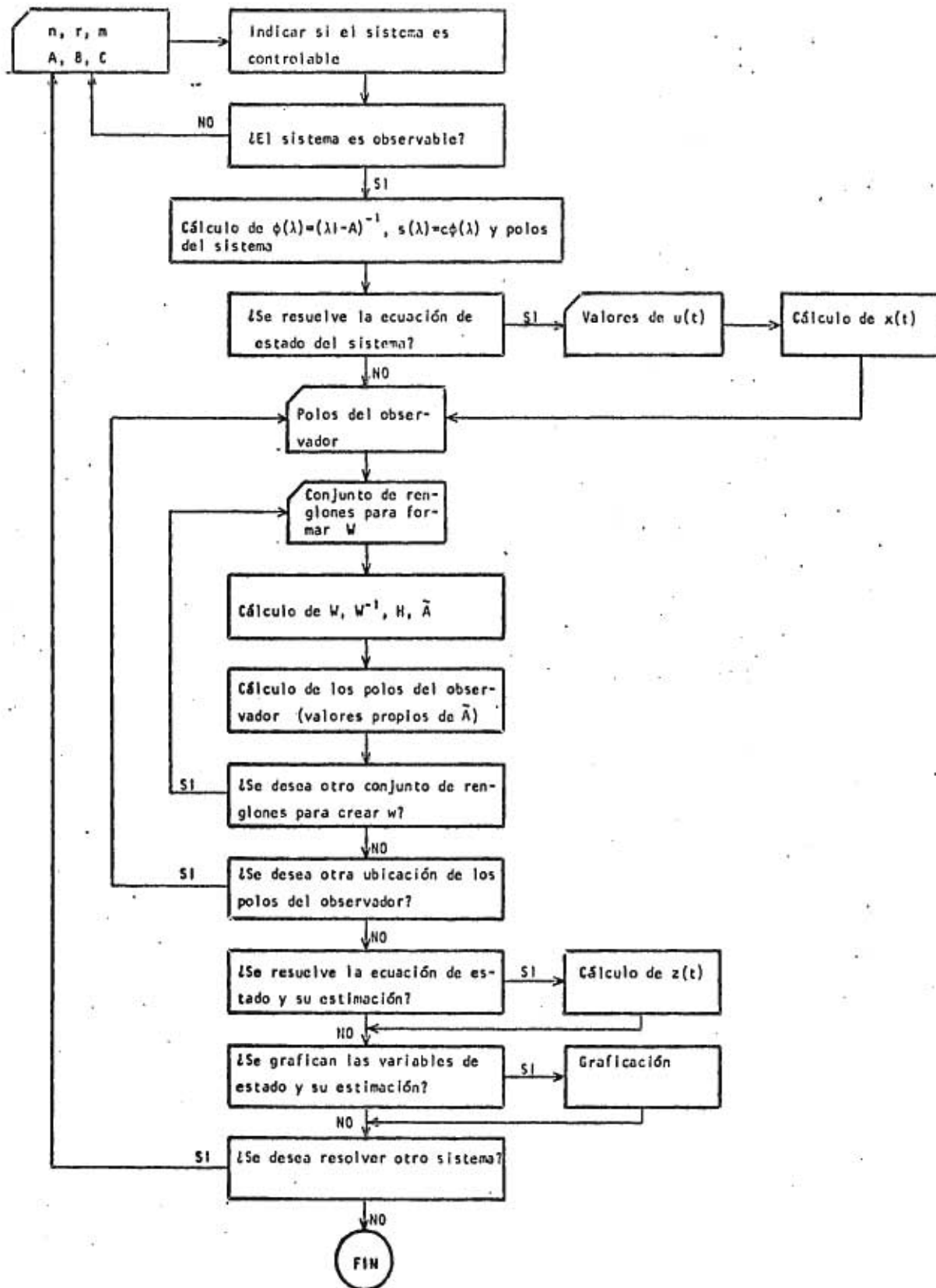


Fig 2. Diagrama de bloques del programa

#### 4. BASE TEORICA DEL PROGRAMA (METODOS NUMERICOS)

El programa OBSERVADOR/UNO que efectúa todas las operaciones indicadas en la fig 2, se creó tomando algunas partes ya hechas, desarrollando totalmente otras e integrando todas ellas.

Dada la magnitud de OBSERVADOR/UNO, 2300 proposiciones, no se describirán los diversos artificios empleados para reducir el tiempo de procesamiento o las localidades de memoria, ya que eso resultaría muy largo y tedioso. En su lugar, se indicará brevemente en que consiste la mayoría de los subprogramas y en ocasiones el método numérico empleado. Así mismo las entradas necesarias, aunque resulte redundante con las indicaciones que da el propio programa en el momento de su ejecución. Esto último se debe a que el programa se desarrolló para usarse en forma interactiva con la computadora a través de algún teletipo.

#### 4.1 Controlabilidad y Observabilidad

Siguiendo el diagrama de bloques encontramos como primer etapa el análisis de la controlabilidad y observabilidad del sistema. Este análisis se hizo de acuerdo con el criterio de Kalman y se efectúa en la subrutina RANTOT la cual sirve para controlabilidad y observabilidad indistintamente.

A RANTOT se alimentan las matrices A y B o las transpuestas de A y C, así como sus dimensiones y se obtiene el rango de la matriz ampliada de controlabilidad u observabilidad.

RANTOT emplea las subrutinas MULMAT, LLAMA y MFGR.

La subrutina MULMAT multiplica dos matrices reales, se usa en bastantes puntos del programa por lo que no se volverá a describir.

La subrutina LLAMA reordena las localidades de memoria para hacer compatibles las subrutinas que funcionan en el sistema IBM, con el sistema Burroughs.

La subrutina MFGR es una parte primordial en esta etapa y también cuando se analiza la degeneración de raíces múltiples. Fue tomada del paquete de subrutinas científicas de IBM citada en las referencias.



El método numérico en que se basa es eliminación Gaussiana. Acepta matrices rectangulares, y aun cuando se analice una matriz cuadrada singular, la respuesta es, entre otros datos, el rango. Maneja un error interno en función de la tolerancia del subprograma que la llamó.

Dado que para construir el observador no se requiere que el sistema sea controlable, solo se imprime si el sistema satisface esta condición o no, sin tener ninguna consecuencia en el resto del programa.

#### *4.2 Inversión de la matriz polinomial $(\lambda I - A)$*

Esta etapa consta de un conjunto de subrutinas tomadas de H. Elliot citado en las referencias. Este paquete requiere como entrada la matriz polinomial almacenada en un arreglo tridimensional y regresa como resultado de matriz polinomial ajunta (la transpuesta de la matriz de cofactores) y un polinomio, que en esta aplicación es el determinante. En otras aplicaciones será el determinante afectado por un escalar.

De este paquete se tomaron las ideas para construir la subrutina MUMCMP que multiplica una matriz de elementos constantes por otra de elementos polinomiales.

Del determinante de  $(S I - A)$  se obtienen los valores propios

del sistema mediante la subrutina POLRT tomada del conjunto de IBM. Esta subrutina recibe como dato el polinomio característico y calcula las raíces del mismo por el método de Newton-Raphson. Debe mencionarse la eficiencia la eficiencia de este subprograma, ya que no importa que existan raíces múltiples, complejas o nulas, en la gran mayoría de los casos las localiza sin problemas.

Antes de hacer los cálculos necesarios para formar la matriz W, se revisa si los renglones solicitados no son mayores que los disponibles (el límite está dado por el número de salidas M), y si la ordenación que se pide en ese momento no se ha empleado anteriormente.

Después se sustituyen los valores de los polos con los renglones indicados y se valúan los polinomios de  $s(\lambda)$  para formar W.

La evaluación de los polinomios para un valor determinado de la variable independiente se hace mediante la función HOR, que emplea el método de Horner.

Para calcular la inversa de la matriz W, se emplean las subrutinas LLAMA (explicada antes) y MINV.

La subrutina MINV también fue tomada del grupo de subrutinas de IBM. Efectúa la inversión mediante el método de Gauss-Jordan.

El cálculo de  $H$  y  $\tilde{A}$ , resulta evidente.

#### 4.4 Cálculo de los polos del observador

En esta etapa se emplea la subrutina DETERM, que es parte del paquete de H. Elliot. Se alimenta a la subrutina la matriz polinomial  $(\lambda I - \tilde{A})$  y se obtiene como resultado los valores propios de  $\tilde{A}$ , estos valores se recuperan mediante la única proposición COMMON del programa.

DETERM contiene un llamado a la subrutina POLRT explicada anteriormente.

#### 4.5 Solución de la ecuación de estado del sistema y del observador

De los diferentes métodos que existen para resolver la ecuación

$$\dot{x}(t) = Ax(t) + Bu(t)$$

se decidió buscar algún método cerrado, es decir, no emplear los métodos aproximados de integración tipo Pickard. Con esta condición, fue necesario calcular la ecuación solución

$$x(t) = e^{At}x(0) + e^{At} \int_0^t e^{-A\tau} Bu(\tau) d\tau$$

es decir, fue necesario calcular  $e^{At}$ . De las diversas formas

que hay para calcular la matriz exponencial, se decidió efectuar la diagonalización de la matriz  $A$ . La causa principal para elegir este camino fue que en las etapas 4.2 y 4.4 ya se calculaban los valores propios, solo faltaba generar los vectores propios.

Pero un problema previo a la generación de los vectores propios es determinar si la matriz es diagonalizable o no. Esto se superó analizando la degeneración de la matriz al sustituir los polos repetidos (cuando existan) mediante la subrutina MFGR que ya se mencionó.

Afortunadamente los casos de polos repetidos son raros, y es más raro aún que al sustituirlos en la matriz  $(\lambda I - A)$  provoquen degeneración incompleta (matriz no diagonalizable).

Después de este análisis, se llama a la subrutina VECPRO, para generar los vectores propios asociados a un valor propio dado, de multiplicidad uno o mayor.

Esta subrutina se formó depurando y adaptando un programa publicado por James, Smith y Wolford.

Actualmente la subrutina VECPRO no acepta valores propios complejos, a pesar de que la base teórica y el método numérico es el mismo, esto se debe a que todo el programa se lleva en do-

ble precisión (salvo la etapa de graficación) y el sistema Burroughs 6700 no maneja variables complejas en doble precisión. Es posible sacrificar la doble precisión con el fin de trabajar en el campo de los complejos, y los cambios necesarios son pocos y sencillos.

Una vez localizados los vectores propios y formada la matriz modal, se procede a su inversión, inversa que teóricamente siempre existe y que puede no encontrarse debido a errores de redondeo principalmente en matrices mal condicionadas.

De aquí la necesidad de comprobar el proceso mediante la pre y postmultiplicación para encontrar la matriz diagonal,

Otro resultado importante de esta diagonalización es el orden de los valores propios sobre la diagonal.

Antes de continuar, es útil notar que aunque existen métodos que calculan simultáneamente los valores y vectores propios de una matriz usualmente requieren que la matriz sea, de alguna forma, particular (en banda, simétrica, positiva definida, etc) y el único método (que se encontró) aplicable a cualquier tipo de matriz fue el de las potencias o iteraciones. Pero este método falla cuando los valores propios son parecidos o iguales (independientemente del signo) y al invertir la matriz modal y multiplicar con la matriz original, se presentan términos no

nulos fuera de la diagonal, y mucho mayores que la tolerancia aceptada en el cálculo de los valores y vectores propios, aún con doble precisión.

Para mayor detalle ver los libros de Wilkinson y Faddeeva.

Una vez diagonalizada la matriz de coeficientes, se tiene que el vector de estado esta dado por

$$x(t) = Pe^{Dt} P^{-1} x(0) + Pe^{Dt} \int_0^t e^{-D\tau} P^{-1} Bu(\tau) d\tau$$

donde  $A = PDP^{-1}$

para la implantación de esta solución en el programa, se hacen dos hipótesis:

- a) Las condiciones inicial son nulas:  $x(t=0) = 0$
- b) Las excitaciones son conjuntos de funciones escalón de diferentes amplitudes:  $u(t) = K v(t)$  siendo  $K$  un vector y  $v(t)$  la función escalón.

de aquí se obtiene

$$x(t) = Pe^{Dt} P^{-1} A^{-1} BK - A^{-1} BK$$

Análogamente se tiene, de la ecuación de estado del observador

$$\dot{z}(t) = \tilde{A}x(t) + Gu(t) + Hc x(t)$$

la siguiente solución

$$z(t) = Rse^{Dt} H + Re^{\Delta t} (L - SH - M) + R(M - L)$$

donde  $\tilde{A} = R\Delta R^{-1}$

$\Delta$  matriz diagonal asociada con  $\tilde{A}$

$R$  y  $R^{-1}$  matriz modal y su inversa.

las demás matrices así como el desarrollo completo para obtener  $x(t)$  y  $z(t)$  se encuentra en los apéndices.

Una de las características más importantes del programa es que los vectores de estado, del sistema y del observador, se obtienen en expresiones *continuas*, de manera que para calcular el estado en un tiempo  $t$  solo hay que valuar las funciones correspondientes. Esto es una ventaja notable porque permite definir el comportamiento de las variables de estado para cualquier tiempo y con cualquier incremento, sin aumentar la memoria necesaria y con variaciones despreciables en el tiempo de procesamiento.

#### 4.6 Graficación

En esta última etapa se emplean las subrutinas FINAL, VALUA y GRAFIC. El subprograma FINAL requiere las matrices que contienen la solución del sistema y del observador, y los tiempos en los que hay que valuar y graficar los estados. Estos tiempos se dan a través del tiempo inicial, el número de puntos que se desea (51 o menos) y el espaciamiento entre ellos.

Estos datos son enviados al subprograma VALUA que calcula dichos valores para todos los tiempos solicitados en todas las variables de estado.

Por último estos datos son enviados a la subrutina GRAFIC, previa consulta con el usuario, para su presentación final a través de una gráfica con los valores numéricos del tiempo, variable del sistema y estimación del observador correspondiente en la margen izquierda.

El límite de 51 puntos para la gráfica, obedece a que la salida proyectada es por teletipo. Es posible, y también deseable, pedir menos de 51 puntos cuando se desea analizar solo un intervalo pequeño.

Al finalizar la etapa de graficación, se ofrece la posibilidad de analizar otro sistema. En caso afirmativo, el programa regresa a la primer etapa, en caso contrario la ejecución termina.



## 5. BIBLIOGRAFIA

Ogata Katsuhiko, "State space analysis of control systems",  
Prentice-Hall, New Jersey; 1967

"System/360 Scientific Subroutine Package", Versión III,  
Programmer's Manual

Elliot, H, "Implementation of computer algorithms related to  
multivariable system theory" NSF-ENG73-0384601/3

Espinosa, I., "Apuntes del curso. Dinámica de sistemas linea  
les", DESFI, UNAM, 1978

Faddeeva V. N., "Computational methods of linear algebra" Ed.  
Dover, 1959

Carnahan B, Luther H A, Wilkes J O, "Applied numerical methods",  
Ed. John Wiley, 1969

Apostol T M, "Calculus", Vol III, Xerox College Publishing, 1969

Wolovich W. A, "Linear multivariable systems", Ed. Springer-Verlag,  
1974

James M. L, Smith G M, Wolford J C, "Métodos numéricos aplica-  
dos a la computación digital con Fortran"; Ed. Iris, 1967

Frazer, Duncan, Collar, "Elementary matrices", Ed. Cambridge at  
the University Press, 1965

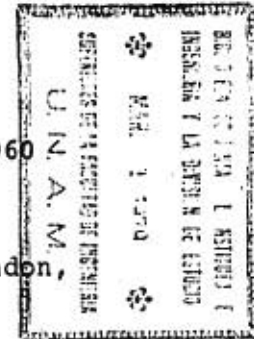
Chen C T, "Introduction to linear system theory", Ed. Holt,  
Rinehart, Winston, 1970

Gantmacher, F, "The theory of matrices", Ed. Chelsea, 1960

Wilkinson, J. H, "The algebraic eigenvalue problem", London,  
Osford, 1965

Acton F. S, "Numerical methods that work", Ed. Harper-Row, 1970

Luenberger, D G, "Observers for multivariable systems", IEEE  
Trans on Automatic Control, Vol AC-11, No 2, Abril 1966



Gilbert, E. G, "Controllability and observability in multivariable control systems", I.S.I.A.M. Control, Vol 2, No 1, 1963

## A1. SOLUCION DE LA ECUACION DE ESTADO (SISTEMA Y OBSERVADOR)

### A1.1 Ecuación de estado del sistema

Sea la ecuación

$$\dot{x}(t) = Ax(t) + Bu(t) \quad (\text{A.1})$$

cuya solución está dada por

$$x(t) = e^{At} x(0) + e^{At} \int_0^t e^{-A\tau} Bu(\tau) d\tau \quad (\text{A.2})$$

considerando las siguientes hipótesis

- a)  $x(t=0) = 0$
- b)  $u(t) = Kv(t)$ ;  $v(t) =$  función escalón  
 $K$  vector de constantes ( $n \times 1$ )
- c)  $e^{At} = Pe^{Dt}P^{-1}$ ;  $D$  matriz diagonal  
 $P, P^{-1}$  matriz modal y su inversa

se llega a

$$x(t) = Pe^{Dt} \int_0^t e^{-D\tau} d\tau P^{-1} BK \quad (A.3)$$

dado que

$$\int_0^t e^{-D\tau} d\tau = D^{-1}(I - e^{-Dt}) = (I - e^{-Dt}) D^{-1}$$

se tiene

$$\begin{aligned} x(t) &= Pe^{Dt} (I - e^{-Dt}) D^{-1} P^{-1} BK \\ x(t) &= Pe^{Dt} D^{-1} P^{-1} BK - PD^{-1} P^{-1} BK \\ x(t) &= Pe^{Dt} P^{-1} A^{-1} BK - A^{-1} BK \end{aligned} \quad (A.4)$$

### A1.2 Ecuación de estado del Observador

Sea la ecuación

$$\dot{z}(t) = \tilde{A} z(t) + Gu(t) + HC x(t) \quad (A.5)$$

cuya solución es

$$z(t) = e^{\tilde{A}t} x(0) + e^{\tilde{A}t} \int_0^t e^{-\tilde{A}\tau} [Gu(\tau) + Hcx(\tau)] d\tau \quad (A.6)$$

considerando las siguientes hipótesis

- $z(t=0) = 0$
- $u(t) = Kv(t)$  igual que antes
- $e^{\tilde{A}t} = R e^{\Delta t} R^{-1}$ ;  $\Delta$  matriz diagonal  
 $R, R^{-1}$  matriz modal y su inversa

se llega a

$$z(t) = \operatorname{Re}^{\Delta t} \int_0^t e^{-\Delta \tau} d\tau R^{-1} GK + \operatorname{Re}^{\Delta t} \int_0^t e^{-\Delta \tau} R^{-1} H c x(\tau) d\tau$$

$$z(t) = \alpha(t) + \beta(t) \quad (\text{A.7})$$

donde

$$\alpha(t) = \operatorname{Re}^{\Delta t} \int_0^t e^{-\Delta \tau} d\tau R^{-1} GK$$

que es una expresión igual a la ecuación A.3, y efectuando las sustituciones correspondientes

$$\begin{aligned} \alpha(t) &= \operatorname{Re}^{\Delta t} [I - e^{-\Delta t}] \Delta^{-1} R^{-1} GK \\ &= \operatorname{Re}^{\Delta t} \Delta^{-1} R^{-1} GK - R \Delta^{-1} R^{-1} GK \end{aligned} \quad (\text{A.8})$$

para  $\beta(t)$  se tiene:

$$\beta(t) = \operatorname{Re}^{\Delta t} \int_0^t e^{-\Delta \tau} R^{-1} H c x(\tau) d\tau$$

y sustituyendo la ecuación A.4

$$\begin{aligned} \beta(t) &= \operatorname{Re}^{\Delta t} \int_0^t e^{-\Delta \tau} R^{-1} H C P e^{D\tau} P^{-1} A^{-1} B K d\tau - \operatorname{Re}^{\Delta t} \int_0^t e^{-\Delta \tau} R^{-1} H C A^{-1} B K d\tau \\ &= \beta_1(t) + \beta_2(t) \end{aligned} \quad (\text{A.9})$$

por comodidad se define la matriz  $Q$  ( $n \times n$ )

$$\begin{aligned} Q &= R^{-1} H C P \\ \therefore \beta_1(t) &= \operatorname{Re}^{\Delta t} \int_0^t e^{-\Delta \tau} Q e^{D\tau} d\tau P^{-1} A^{-1} B K \\ &= \operatorname{Re}^{\Delta t} \cdot \gamma(t) \cdot P^{-1} A^{-1} B K \end{aligned} \quad (\text{A.10})$$

$$\text{donde } \gamma(t) = \int_0^t e^{-\Delta\tau} Q e^{D\tau} d\tau$$

En el integrando de gama se tiene el producto de dos matrices diagonales y una matriz cualquiera, esto se puede aprovechar recordando que cuando una matriz diagonal premultiplica a otra matriz, el renglón  $i$ -ésimo de la matriz producto es igual al renglón de la matriz que postmultiplica escalado por el elemento  $i$ -ésimo de la matriz diagonal. Análogamente, sucede lo mismo con las columnas cuando la matriz diagonal postmultiplica.

Tomando en cuenta estas propiedades y definiendo los elementos de cada matriz

$$\begin{aligned} [e^{-\Delta\tau}] &= (e^{-\delta_i\tau}) \text{ para } i = j \\ &= 0 \quad \text{para } i \neq j \end{aligned}$$

$$[Q] = (q_{ij})$$

$$\begin{aligned} [e^{-D\tau}] &= (e^{-d_i\tau}) \text{ para } i = j \\ &= 0 \quad \text{para } i \neq j \end{aligned}$$

se llega a que el término general del integrando es:

$$q_{ij} \cdot e^{-\delta_i\tau} \cdot e^{d_j\tau} = q_{ij} e^{-(\delta_i-d_j)\tau}$$

integrando este término general se tiene

$$\int_0^t q_{ij} e^{-(\delta_i - d_j)\tau} d\tau = \left[ q_{ij} \frac{e^{-(\delta_i - d_j)\tau}}{-(\delta_i - d_j)} \right]_0^t$$

$$= \frac{q_{ij}}{d_j - \delta_i} [e^{(d_j - \delta_i)t} - 1] \text{ para toda } i, j$$

y haciendo  $s_{ij} = \frac{q_{ij}}{d_j - \delta_i}$

se llega a

$$\gamma_{ij}(t) = \int_0^t q_{ij} e^{-(\delta_i - d_j)\tau} d\tau = s_{ij} e^{(d_j - \delta_i)t} - s_{ij} \quad (\text{A.11})$$

analizando el primer sumando de la ecuación A.11 se observa que es posible descomponerlo en un producto, es decir,

$$\gamma_{ij}(t) = e^{-\delta_i t} s_{ij} e^{d_j t} - s_{ij}$$

colocado en forma matricial se tiene

$$\gamma(t) = e^{-\Delta t} \cdot S \cdot e^{Dt} - S$$

donde  $S = (s_{ij})$

sustituyendo  $\gamma(t)$  en la ecuación A.10

$$\beta_1(t) = R S e^{Dt} P^{-1} A^{-1} B K - R e^{\Delta t} S P^{-1} A^{-1} B K$$

Para el segundo sumando de la ec A.9 se tiene

$$\beta_2(t) = - R e^{\Delta t} \int_0^t e^{-\Delta \tau} d\tau Q D^{-1} P^{-1} B K$$

donde se encuentra otra vez la estructura de la ec A.3



YR/SALID (10/09/78) <sup>3</sup>SL/OBSERV

1:12 PM TUESDAY, FEB  
27 Febrero 1979.

100	* RESET FILE	0000100
200	*PESF1 SINGLE	0000200
250	*SET LIST	0000250
300	FILE 0=SALL,UNIT=RENTC,RECFRD=22	0000300
400	INTEGR R	0000400
500	LOGICAL FLAG	0000500
600	LINAL * 0 F * I * I1 * T2 * DI * TDI * T3 * VALS * VECTS * CT * C * A * T4 * RR * NI * DET * AG	0000600
700	* I * H * I * P * RR * R * R * IO * FS * PHENOR * PHAYOR * AT * NA * H * NI * H * HC	0000700
800	* * T5 * AI * AD * MD * AGD * DIC * DIA * QS	0000800
900	* * AK * DE * X * Y * DIP * BK * XH * ZI * DZI	0000900
1000	DIMENSION A(10,10),R(10,10),C(10,10),CT(10,10),FI(10,10,20),T3(10)	0001000
1100	* I(10,10,20),T1(10,10,20),T2(10,10,20),DI(20),VALS(10),VECTS(10)	0001100
1200	* T4(20),RR(20),RI(20),IPEN(10),IT1(10),IT2(10),IH(10,10),AG(10,10)	0001200
1300	* I(10,10),NA(10),RRR(20),RIN(20),FS(10,10,20),XH(10),ZH(10)	0001300
1400	* AT(10,10),NA(10,10),HC(10,10),NI(10,10),H(10,10),HC(10,10)	0001400
1500	* * T5(10,10),AI(10,10),AD(10,10),HEI(10,10),AGD(10,10),DZH(10)	0001500
1600	* * AI(10),BK(10),X(10,10),Y(10,23),DIPRI(10),Q(10,10),S(10,10)	0001600
1700	COMPLI /POLLOS/RRR/RIO	0001700
1750	WRITE (6,97)	0001750
1752	97 FORMAT (///1X,130(" *")/" *",128X," *"/" *",41X,	0001752
1754	*"LISTA DE OBSERVADORES LINEALES MULTIVARIABLES"	0001754
1755	*41X," *"/" *",106X,"J. FRANCISCO SANDOVAL F *"/	0001755
1756	*" *",106X,"NOV 1977 - ENL 1978 *"/	0001756
1760	*" *",128X," *"/1X,130(" *")///	0001760
1762	*20X,"SL USA EL METODO DE LUENPERGER PARA CREAR EL OBSERVADOR."/	0001762
1764	*20X,"LOS POLOS DE ESTE SE ASIGNAN POR EL PROCEDIMIENTO DE PROGRAM."/	0001764
1766	*20X,"EL PROGRAMA ACEPTA SISTEMAS HASTA DE ORDEN 10 CON 10 ENTRADA	0001766
1767	*S Y 10 SALIDAS"///	0001767
1768	*20X,"ECUACION DE ESTADO: DX(T)/DT = A * X(T) + B * U(T)"/	0001768
1770	*20X,"EQUACION DE SALIDA: Y(T) = C * X(T)"/	0001770

1772		*33X>"DONDE: A ES MATRIZ (N*N)"/	00001772
1774		*42X>"B ES MATRIZ (N*R)"/	00001774
1776		*42X>"C ES MATRIZ (N*N)"/	00001776
1800	C		00001800
1900	C	LECTURA DE LAS DIMENSIONES DE LAS MATRICES A, B, Y C.	00001900
2000	C		00002000
2100		3 WRITE (6,125)	00002100
2200		125 FORMAT (/2X>"DAME EL ORDEN DEL SISTEMA (N), EL NUMERO DE EXCITACIONES (R) Y EL DE SALIDAS (H) Y LA APROXIMACION PARA EL PROCESO (TOL)"/)	00002200
2300		*15X>"(R) Y EL DE SALIDAS (H) Y LA APROXIMACION PARA EL PROCESO (TOL)"/)	00002300
2310		*15X>"(R) Y EL DE SALIDAS (H) Y LA APROXIMACION PARA EL PROCESO (TOL)"/)	00002310
2400		READ(5,/) N,R,H,TOL	00002400
2500		IF = 0	00002500
2600		IF (1. .LE. 0) CALL EXIT	00002600
2700		SI = 6H5I	00002700
2800	C		00002800
2900	C	REVISION DE LAS DIMENSIONES	00002900
3000	C		00003000
3100		IF (1. .GE. R .AND. H .GE. H) GO TO 4	00003100
3200		WRITE (6,101) H,R,H	00003200
3300		101 FORMAT (/10X>"MATRICES MAL DEFINIDAS"/	00003300
3400		*10X>"LOS VALORES DE H,R, Y H SON",3I5," RESPECTIVAMENTE"/	00003400
3500		*10X>"EJECUCION TERMINADA")	00003500
3600		GL TO 3	00003600
3700	C		00003700
3800	C	LECTURA (POR REGIONES) E IMPRESION DE LA MATRIZ "A"	00003800
3900	C		00003900
4000		4 WRITE (6,126)	00004000
4100		126 FORMAT (/2X>"DAME LOS ELEMENTOS DE LA MATRIZ DE COEFICIENTES DEL SISTEMA (MATRIZ A) POR REGIONES"/	00004100
4200		*15X>"(MATRIZ A) POR REGIONES"/	00004200
4300		READ(5,/) ((A(I,J)),J=1,H),I=1,R)	00004300
4400		WRITE (6,105)	00004400

4500	105	FORMAT (/20X,"MATRIZ A"//)	00004500
4600		CALL SACAM (A,M,N)	00004600
4700	106	FORMAT (2X,10F9.3)	00004700
4800	C		00004800
4900	C	LECTURA (POR RENGLONES) E IMPRESION DE LA MATRIZ "B"	00004900
5000	C		00005000
5100		WRITE (6,127)	00005100
5200	127	FORMAT (/2X,"DAME LOS ELEMENTOS DE LA MATRIZ DE DISTRIBUCION (MATRIZ D) POR RENGLONES"/)	00005200
5300		*IZ D) POR RENGLONES"/)	00005300
5400		II (I,LE,0) GO TO 5	00005400
5500		READ (5,/) ((B(I,J)),J=1,M),I=1,N)	00005500
5600		WRITE (6,102)	00005600
5700	102	FORMAT (/20X,"MATRIZ B"//)	00005700
5800		CALL SACAM (B,M,N)	00005800
5900	C		00005900
6000	C	LECTURA (POR RENGLONES) E IMPRESION DE LA MATRIZ "C"	00006000
6100	C		00006100
6200		5 WRITE (6,126)	00006200
6300	126	FORMAT (/2X,"DAME LOS ELEMENTOS DE LA MATRIZ DE SALIDA (MATRIZ C) POR RENGLONES"/)	00006300
6400		* POR RENGLONES"/)	00006400
6500		READ (5,/) ((C(I,J)),J=1,H),I=1,H)	00006500
6600		WRITE (6,103)	00006600
6700	103	FORMAT (/20X,"MATRIZ C"//)	00006700
6800		CALL SACAM (C,M,H)	00006800
6900	C		00006900
7000	C	ASIGNACION DE LA TOLERANCIA (TOL) PARA TODO EL PROCESO	00007000
7100	C		00007100
7200		II (TOL,EQ,0.0) TOL = 1.0E-6	00007200
7300	C		00007300
7400	C	ANALISIS DE LA CONTROLABILIDAD DEL SISTEMA	00007400
7500	C		00007500

7600	IF (IRANGO .EQ. 0) GO TO 31	00007600
7700	CALL PARTOT (A,B,N,P,TOL,IRANGC)	00007700
7800	IF (IRANGC .EQ. 0) GO TO 30	00007800
7900	WRITE (6,129) IRANGC,N	00007900
8000	129 FORMAT (//20X,"EL RANGO DE LA MATRIZ AMPLIADA (" ,I2," ) ES MENOR QU	00008000
8100	*E EL ORDEN DEL SISTEMA (" ,I2," )."/	00008100
8200	*20X,"EL SISTEMA NO ES COMPLETAMENTE CONTROLABLE"/)	00008200
8300	GO TO 31	00008300
8400	30 WRITE (6,130)	00008400
8500	130 FORMAT (//20X,"EL SISTEMA ES COMPLETAMENTE CONTROLABLE"/)	00008500
8600	C	00008600
8700	C ANALISIS DE LA OBSERVABILIDAD DEL SISTEMA	00008700
8800	C	00008800
8900	31 DO 11 I = 1 , N	00008900
9000	DO 11 J = 1 , N	00009000
9100	11 CT(J,I) = C (I,J)	00009100
9200	DO 29 I=1,N	00009200
9300	DO 29 J=1,N	00009300
9400	29 AT(I,J) = A(J,I)	00009400
9500	CALL PARTOT (A,CT,N,N,TOL,IRANGD)	00009500
9600	IF (IRANGD .EQ. 0) GO TO 14	00009600
9700	WRITE (6,106) IRANGD,N	00009700
9800	106 FORMAT (//20X,"EL RANGO DE LA MATRIZ AMPLIADA (" ,I2," ) ES MENOR QU	00009800
9900	*E EL ORDEN DEL SISTEMA (" ,I2," )."/	00009900
10000	*20X,"EL SISTEMA NO ES COMPLETAMENTE OBSERVABLE"/	00010000
10100	*20X," EJECUCION TERMINADA"/)	00010100
10200	CALL EXIT	00010200
10300	14 WRITE (6,104)	00010300
10400	104 FORMAT (//20X,"EL SISTEMA ES COMPLETAMENTE OBSERVABLE"/)	00010400
10500	C	00010500
10600	C FORMACION DE LA MATRIZ POLINOMIAL (SI=A) PARA SU USO EN EL	00010600

10700	C	PROCESO DE LUHPERCEP Y PARA OBTENER LOS VALORES PROPIOS DEL	00010700
10800	C	SISTEMA ORIGINAL	00010800
10900	C		00010900
11000		CALL LIMPYA(F,10,10,20)	00011000
11100		CALL LIMPYA (F,10,10,20)	00011100
11200		DL 12 J=1,N	00011200
11300		DL 12 I=1,N	00011300
11400		12 F(J,1,1) = -A(1,J)	00011400
11500		DL 13 I=1,N	00011500
11600		13 F(I,1,2) = 1.0	00011600
11700		WRITE (6,111)	00011700
11800		111 FORMAT(/10X,"ELEMENTOS DE LA MATRIZ CUADRADA (SI=A)")	00011800
11900	C		00011900
12000	C	IMPRESION DE LA MATRIZ POLINOMIAL (SI=A)	00012000
12100	C		00012100
12200		CALL PERT (F,N,10,20,TOL)	00012200
12300	C		00012300
12400	C	INVERSTON DE (SI=A)	00012400
12500	C		00012500
12600		CALL INVERI (F,F,I,T1,T2,DI,H,GRADIV,20,TOL,ISIN)	00012600
12700		IF (ISIN .NE. 1) GO TO 17	00012700
12800		WRITE (6,144)	00012800
12900		144 FORMAT (///30X,"LA MATRIZ (SI=A) ES SINGULAR"//)	00012900
13000		GL TC 3	00013000
13100		17 WRITE (6,112)	00013100
13200		112 FORMAT (/10X,"ELEMENTOS DE LA MATRIZ INVERSA DE (SI=A)")	00013200
13300	C		00013300
13400	C	IMPRESION DE LA MATRIZ INVERSA DE (SI=A) Y DE SU DETERMINANTE	00013400
13500	C		00013500
13600		CALL PERT (F,10,10,20,TOL)	00013600
13700		WRITE (6,113)	00013700

13800	113	FORMAT (/10X,"DETERMINANTE DE (SI=A)"/	00013800
13900		*10X,"(POLINOMIO DE GRADO N)"/	00013900
14000		CALL PLINT3 (DI,20,TOL)	00014000
14100	C		00014100
14200	C	VALORES PROPIOS DEL SISTEMA ORIGINAL O RAICES DEL DETERMINANTE	00014200
14300	C	LE (SI=A)	00014300
14400	C		00014400
14500	C	CALL POLR1 (DI,T4,RI,RI,IER)	00014500
14600	C	GO TO (1,50,51,52,50),IER + 1	00014600
14700	C	50 WRITE (6,108) N	00014700
14800	C	108 FORMAT (/20X,"EL VALOR DE N (">13," ) ES MENOR QUE 1"/	00014800
14900	C	*20X,"EJECUCION TERMINADA")	00014900
15000	C	CALL EXIT	00015000
15100	C	51 WRITE (6,109) N	00015100
15200	C	109 FORMAT (/20X,"EL VALOR DE N (">13," ) ES MAYOR QUE 36 (CAPACIDAD	00015200
15300	C	*DE PLINT)"/20X,"EJECUCION TERMINADA")	00015300
15400	C	CALL EXIT	00015400
15500	C	52 WRITE (6,110)	00015500
15600	C	110 FORMAT (/20X,"EL METODO PARA LOCALIZAR LAS RAICES NO CONVERGE"/	00015600
15700	C	*20X,"EJECUCION TERMINADA")	00015700
15800	C	CALL EXIT	00015800
15900	C	1 WRITE (6,107) (RR(I),RI(I),I=1,N)	00015900
16000	C	107 FORMAT (/20X,"VALORES PROPIOS DEL SISTEMA"/	00016000
16100	C	*20X,"PARTE REAL">10X,"PARTE IMAG."/(>6X>2E25.5))	00016100
16200	C	WRITE (6,145)	00016200
16300	C	FLAG5 =0H	00016300
16400	C	145 FORMAT (/10X,"DESEAS RESOLVER LA ECUACION DE ESTADO DEL SISTEMA (S	00016400
16500	C	*I) (L)"/)	00016500
16600	C	CALL (5,120) FLAG5	00016600
16700	C	IF (FLAG5 .NE. SI) GO TO 16	00016700
16800	C	CALL MODAL (A,RI,RI,TOL,PA,IND)	00016800

16900		IF (IIB .NE. 3) GO TO 3	00016900
17000	C		00017000
17100	C	HA ES LA MATRIZ HEDAL DEL SISTEMA (ASOCIADA CON A)	00017100
17200	C		00017200
17300		WRITE (6,133)	00017300
17400	133	FORMAT (//20X,"MATRIZ HEDAL DE A (HA)"/)	00017400
17500		CALL SACAM (MA,N,1)	00017500
17600		DO 8 I=1,N	00017600
17700		DO 8 J=1,N	00017700
17800	A	HAI ( J,I)=HA(I,J)	00017800
17900		CALL LLAMA (MA,I,N,10,10,1)	00017900
18000		CALL HINV (MAI,N,DHA,IT1,IT2)	00018000
18100		CALL LLAMA (MA,I,N,10,10,2)	00018100
18200		WRITE (6,134)	00018200
18300	134	FORMAT (//20X,"INVERSA DE HA (HAI)"/)	00018300
18400		CALL SACAM (MA,I,N,1)	00018400
18500		CALL HULMAT (MAI,A,T5,I,N,1)	00018500
18600		CALL HULMAT (T5,HA,AD,I,N,1)	00018600
18700		WRITE (6,135)	00018700
18800	135	FORMAT (//20X,"MATRIZ FIAGONAL ASOCIADA CON A (AD)"/)	00018800
18900		CALL SACAM (AD,I,N)	00018900
18910	3A	WRITE (6,160)	00018910
18915	160	FORMAT (//2X,"DAME LOS VALORES INICIALES DEL SISTEMA"/)	00018915
18920		READ (5,/) (XH(I),I=1,N)	00018920
18925		WRITE (6,161)	00018925
18930	161	FORMAT (//20X,"VALORES INICIALES DEL VECTOR DE ESTADO PL SISTEMA"/)	00018930
18935		CALL SACAM (XH,N,1)	00018935
18937	C		00018937
18938	C	P*(-1) *X(0) = 13	00018938
18939	C		00018939
18940		CALL HULMAT(MA,I,XI,T3,I,N,1)	00018940

18542	C	ASIGNACION DE LOS PUNOS DEL SISTEMA A LAS MATRICES SOLUCION	00018942
18545		DO 39 J=1,N	00018945
18547		X(J,11) = AD(J,J)	00018947
18548		Y(J,21) = AG (J,J)	00018948
18549	C	CALCULO DE LA MATRIZ DE COEFICIENTES DE LA SOLUCION HOMOGENEA	00018949
18550		DE 39 I=1,N	00018950
18555	30	T5(I,J) = MA(I,J) * T3(J)	00018955
18556		FLAG9 = 6H	00018958
18560		WRITE (6,142)	00018960
18562	162	FORMAT (/10X,"DESEAS IMPRIMIR LA SOLUCION HOMOGENEA DEL SISTEMA (	00018962
18563		*S1) (N0)"/)	00018963
18564		FLAD (5,120) FLAG9	00018964
18566		IF (FLAG9 .NE. S1) GO TO 37	00018966
18568		WRITE (6,143)	00018968
18570	163	FORMAT (/20X,"SOLUCION HOMOGENEA DE LA ECUACION DE ESTADO DEL SIS	00018970
18571		*TENA X(T) = X1 * EXP(Y2*T)"/)	00018971
18572		WRITE (6,149)	00018972
18574		CALL SACAM (T5,N,10)	00018974
18576		WRITE (6,150)	00018976
18578		CALL SACAM (X(1,11),N,1)	00018978
18582	37	WRITE (6,144)	00018982
18583	164	FORMAT (/10X,"DESEAS OTRO CONJUNTO DE VALORES INICIALES DEL SISTEN:	00018983
18584		*A (S1) (N0)"/)	00018984
18585		FLAG10 = 6H	00018985
18587		FLAD (5,120) FLAG10	00018987
18588		IF (FLAG10 .EQ. S1) GO TO 38	00018988
19000		FLAG7 = .FALSE.	00019000
19100	15	WRITE (6,146)	00019100
19200	147	FORMAT (/2X,"DAME LAS AMPLITUDES DE LAS FUNCIONES ESCALON DE LA EX	00019200
19300		*CITACION"/)	00019300
19400		FLAD (5,7) (AK,I),I=1,P)	00019400



19500		WRITE (6,147)	00019500
19600	147	FORMAT (/20X,"VECTOR DE AMPLITUDES DE LAS FUNCIONES ESCALON DE LA	00019600
19700		*EXCITACION"/)	00019700
19800		CALL SACAM (AK,R,1)	00019800
19900		CALL MULMAT (B,AK,BK,H,R,1)	00019900
19950	C	F**(-1) * D * K	00019950
20000		CALL MULMAT (MAI,DK,T3,H,R,1)	00020000
20100		IF (FLAG7) GO TO 53	00020100
20120	C	FLAG7=TRUE, INDICA QUE DEBE SALTAR PAPA CUANDO SE PIDE HAS	00020120
20121	C	DE UN CONJUNTO DE ENTRADAS (EVITA DOBLE INVERSION DE AD)	00020121
20200		FLAG7 = .TRUE.	00020200
20300		DO 7 I=1,N	00020300
20600		7 AL(I,1) = 1.0/AD(I,1)	00020600
20690	C	D**(-1) * F**(-1) * B * K	00020690
20700	53	CALL MULMAT (AD,T3,DIPBK,H,R,1)	00020700
20789	C		00020789
20790	C	TEJNING INDEPENDIENTE DE X(T): F * D**(-1) * P**(-1) * B * I	00020790
20791	C		00020791
20800		CALL MULMAT (MA,DIPBK,X(1,12),H,R,1)	00020800
20900		DO 6 J=1,N	00020900
20910		X(J,12) = - X(J,12)	00020910
21000		DO 6 I=1,N	00021000
21100		6 X(I,J) = MA(1,J) * DIPBK(J) + T5(I,J)	00021100
21200		WRITE (6,146)	00021200
21300	148	FORMAT (/20X,"SOLUCION GENERAL DE LA ECUACION DE ESTADO DEL SISTL	00021300
21400		*HA: X(T) = X1 * EXP(X2*T) + X3"/)	00021400
21500		WRITE (6,149)	00021500
21600	149	FORMAT(/20X,"MATRIZ DE COEFICIENTES DE LA SOLUCION X(T) (X1)"/)	00021600
21700		CALL SACAM (X,R,1)	00021700
21800		WRITE (6,150)	00021800
21900	150	FORMAT (/20X,"VECTOR DE COEFICIENTES DEL EXPONENTE DE LA SOLUCION	00021900

22000	* X(T) (X2)"/)	00022000
22100	CALL SACAM (X(1,11),N,1)	00022100
22200	MIITL (6,151)	00022200
22300	151 FLPMAT(//20X,"VECTOR DE TERMINOS CONSTANTES DE LA SOLUCION X(T) (	00022300
22310	*X3)"/)	00022310
22400	CALL SACAM (X(1,12),N,1)	00022400
22500	C	00022500
22600	C MULTIPPLICACION DE LA MATRIZ DE ELEMENTOS CONSTANTES C POR LA	00022600
22700	C INVERSA DE LA MATRIZ POLINOMIAL (SI=A)	00022700
22800	C	00022800
22900	MIITL (6,152)	00022900
23000	FLAG6 =6H	00023000
23100	152 FLPMAT (/10X,"DESEAS OTRO CONJUNTO DE ENTREFAS (SI) (NO)"/)	00023100
23200	FLAD (5,120) FLAG6	00023200
23300	IF (FLAG6 .EQ. 5) GO TO 15	00023300
23400	14 CALL RUBCHP (C,F,I,F,M,I,M,I,M,M,20,TDI)	00023400
23500	MIITL (6,114)	00023500
23600	110 FLPMAT (/10X,"ELEMENTOS DE LA MATRIZ POLINOMIAL S")	00023600
23700	CALL PERT (I,M,N,20,TDI)	00023700
23800	C	00023800
23900	C LECTURA DE LA POSICION DE LOS POLIS DEL OBSERVADOR (ESFERADA)	00023900
24000	C	00024000
24100	23 MIITL(6,115)	00024100
24200	NI. =0	00024200
24300	115 FLPMAT(/2X,"DAME LA UBICACION DE LOS POLIS DEL OBSERVADOR"/)	00024300
24400	FLAD(5,/) (FN(1),I=1,N)	00024400
24500	C	00024500
24600	C IMPRESION Y ANALISIS DE LOS POLIS DEL OBSERVADOR	00024600
24700	C	00024700
24800	PIENGE = RR (1)	00024800
24900	LL 26 I = 2,N	00024900

25000	26	PHENDR = DMIN1 (PHENDR,PR(I))	00025000
25100		PHAYDR = P <sub>N</sub> (I)	00025100
25200		LL 27 I=2,N	00025200
25300	27	PHAYDR = DMAX1 (PHAYDR,PR(I))	00025300
25400		IF (PHENDR .GT. PHAYDR) GO TO 22	00025400
25500		WRITE (6,125) PHENDR,PHAYDR	00025500
25600	123	FORMAT (/10X,"ALGEBRAICAMENTE EL MENOR POLO DEL SISTEMA ES",F8.4/	00025600
25700		*10X,"Y EL MAYOR DEL OBSERVADOR ES",F8.4/)	00025700
25800		GO TO 23	00025800
25900	C		00025900
26000	C	LECTURA DE LOS REACCIONES SELECCIONADAS PARA FORMAR LA MATRIZ	00026000
26100	C	MOLECULAR N. AL SUSTITUIR LOS VALORES DE LOS NUEVOS POLOS	00026100
26200	C		00026200
26300	29	WRITE (6,116)	00026300
26400	116	FORMAT (2X,"DAME EL ORDEN DE LOS REACCIONES PARA CREAR N"/)	00026400
26500		READ (5,/) (IREN(I),I=1,N)	00026500
26600		CALL LIMPIA (N,10,10,1)	00026600
26700		CALL LIMPIA (I,10,10,1)	00026700
26800		CALL CREAT (F,0,IREN,DI,IPEN,FR,W,IE,TDI,PK)	00026800
26900		LL 32 I=1,10	00026900
27000		LL 32 J=1,10	00027000
27100	32	W <sub>I</sub> (I,J)=W(I,J)	00027100
27200		GO TO (22,23,20),KK	00027200
27300	28	WRITE (6,132)	00027300
27400	132	FORMAT (/20X,"MATRIZ W"/)	00027400
27500		CALL SACAR (N,N,N)	00027500
27600	C		00027600
27700	C	CALCULO DE LA MATRIZ INVERSA DE W (VI).	00027700
27800	C		00027800
27900		CALL LLAMA (N,N,N,10,10,1)	00027900
28000		CALL INIV (N,N,DET,IT1,IT2)	00028000

28100	II (LAPS(DEI) .GT. 0. ) GC TO 24	00028100
28200	WRITE (6,117)	00028200
28300	117 FLEMAT (/10X,"LA MATRIZ W RESULTO SINGULAR"//)	00028300
28400	GC TO 25	00028400
28500	24 CALL LLAMA (WI,NI,10,10,2)	00028500
28600	WRITE (6,131)	00028600
28700	131 FLEMAT (/20X,"MATRIZ INVERSA DE W (WI)"//)	00028700
28800	CALL SACAM (WI,NI,1)	00028800
28900	CALL MULMAT (WI,II,HI,HI,HI,HI)	00028900
29000	CALL MULMAT (HI,CI,FI,FI,FI,FI)	00029000
29100	DL 2 I = 1.N	00029100
29200	DL 2 J = 1.N	00029200
29250	HC (I,J) = - HC (I,J)	00029250
29300	2 AG(I,J) = A(I,J) - HC (I,J)	00029300
29400	WRITE (6,141)	00029400
29500	141 FLEMAT (/30X,"ECUACION DE ESTADO DEL OBSERVADOR: DZ(T)/DT = AG *	00029500
29600	* (T) + G * U(T) + FC * X(T)"//)	00029600
29700	WRITE (6,136)	00029700
29800	136 FLEMAT (/20X,"MATRIZ DE COEFICIENTES DEL OBSERVADOR (AG)"//)	00029800
29900	CALL SACAM (AG,HI,HI)	00029900
30000	WRITE (6,142)	00030000
30100	142 FLEMAT (/20X,"MATRIZ DE DISTRIBUCION ASOCIADA CON U(T) (G)"//)	00030100
30200	CALL SACAM (G,NI,NI)	00030200
30300	WRITE (6,143)	00030300
30400	143 FLEMAT (/20X,"MATRIZ DE DISTRIBUCION ASOCIADA CON X(T) (FC)"//)	00030400
30500	CALL SACAM (FC,NI,NI)	00030500
30600	C	00030600
30700	C CALCULO DE LOS VALORES PROPIOS DE AG	00030700
30800	C	00030800
30900	CALL LIMPIA (F3,10,10,20)	00030900
31000	CALL LIMPIA (T2,10,10,20)	00031000

31100	CALL LIMPIA (RND,20,1,1)	00031100
31200	CALL LIMPIA (RIG,20,1,1)	00031200
31300	LL 20 I = 1,10	00031300
31400	LL 21 J = 1,10	00031400
31500	21 FS(J,1,1) = -A <sub>0</sub> (I,J)	00031500
31600	20 FS (1,1,2) = 1.0	00031600
31700	CALL DLTERM (FS,T2,N,20,TOL)	00031700
31800	WRITE (6,124) (RBC(I),RIO(I),I=1,N)	00031800
31900	12# FORMAT (/20X,"VALORES PROPIOS DEL OBSERVADOR"//	00031900
32000	* 30X,"PARTE REAL",10X,"PARTE IMAG."// (15X,2E25.5))	00032000
32100	25 WRITE (6,110)	00032100
32200	11# FORMAT(/10X,"SELECCIONAS UN NUEVO CONJUNTO DE RENGLONES (SI) (NO)"	00032200
32300	*//)	00032300
32400	FLAG (5,120) FLAG1	00032400
32500	120 FORMAT (A6)	00032500
32600	IF (FLAG1 .EQ. SI ) GO TO 22	00032600
32650	IF (FLAG5 .NE. SI ) GO TO 10	00032650
32700	WRITE (6,137)	00032700
32800	137 FORMAT(/10X,"DESLAS RESOLVER LA ECUACION DE ESTADO DEL OBSERVADOR	00032800
32900	* (SI) (NO)"//)	00032900
33000	FLAG (5,120) FLAG4	00033000
33100	IF (FLAG4 .NE. SI) GO TO 10	00033100
33200	C	00033200
33300	C        LD ES LA MATRIZ MODAL DEL OBSERVADOR (ASOCIADA CON A6)	00033300
33400	C	00033400
33500	CALL MODAL (AG,RRO,RIO,RTO,I,O,II,L)	00033500
33600	WRITE (6,138)	00033600
33700	13# FORMAT (/20X,"MATRIZ MODAL DE AG (LD)"//)	00033700
33800	CALL SACAM (MO,N,1)	00033800
33900	LL 9 I=1,N	00033900
34000	LL 9 J=1,N	00034000

34100	o HLI (J,I) = ND(J,I)	00034100
34200	CALL LLAMA (MO1,H,H,10,10,1)	00034200
34300	CALL MINV (MO1,H,H,IT1,IT2)	00034300
34400	CALL LLAMA (MO1,H,H,10,10,2)	00034400
34500	WRITE (6,139)	00034500
34600	130 FLEHAT (/20X,"INVERSA DE HG (HGI)"//)	00034600
34700	CALL SACAM (MO1,H,H)	00034700
34800	CALL BULHAT (MO1,AG,T5,H,H,N)	00034800
34900	CALL BULHAT (T5,H,AGD,H,H,N)	00034900
35000	WRITE (6,140)	00035000
35100	140 FLEHAT (/20X,"MATRIZ PIAGORAI ASOCIADA CON AG (AGD)"//)	00035100
35200	CALL SACAM (AGD,H,H)	00035200
35201	DL 40 I=1,N	00035201
35202	Y(I,23) = X(I,12)	00035202
35203	40 Y(I,22) = AGD(I,I)	00035203
35210	15 WRITE (6,165)	00035210
35215	165 FLEHAT (/2X,"DAME LOS VALORES INICIALES DEL OBSERVADOR"//)	00035215
35220	READ (5,/) (ZH(I),I=1,I)	00035220
35225	WRITE (6,166)	00035225
35230	166 FLEHAT (/20X,"VALORES INICIALES DEL VECTOR DE ESTADO DEL OBSERVADO	00035230
35231	+I."//)	00035231
35235	CALL SACAM (ZH,H,1)	00035235
35250	LL 41 I=1,N	00035250
35255	41 LZH(I) = ZH(I) - XH(I)	00035255
35260	CALL BULHAT (MO1,LZH,T3,H,H,1)	00035260
35265	DL 42 J=1,N	00035265
35275	DL 42 I=1,N	00035275
35280	Y(I,10+J) = ND (I,J) + T3(J)	00035280
35290	42 Y(I,J) = X(I,J)	00035290
37800	WRITE (6,154)	00037800
37900	154 FLEHAT (/20X,"SOLUCION DE LA ECUACION DE ESTADO DEL OBSERVADOR 2	00037900

3A000	* (T) = Z1 * EXP(Z2*T) + Z3 * EXP(Z4*T) + Z5**/)	00038000
3A100	WRITE (6,153)	00038100
3A200	153 FORTAT (/20X,"MATRIZ DE COEFICIENTES DE LA SOLUCION Z(T) (Z1)**/)	00038200
3A300	CALL SACAM (Y(1,1),N,1)	00038300
3A400	WRITE (6,155)	00038400
3A500	155 FORTAT (/20X,"VECTOR DE COEFICIENTES DEL EXPONENTE DE LA SOLUCION	00038500
3A600	* Z(T) (Z2)**/)	00038600
3A700	CALL SACAM (Y(1,21),N,1)	00038700
3A800	WRITE (6,156)	00038800
3A900	156 FORTAT (/20X,"MATRIZ DE COEFICIENTES DE LA SOLUCION Z(T) (Z3)**/)	00038900
39000	CALL SACAM (Y(1,11),N,1)	00039000
39100	WRITE (6,157)	00039100
39200	157 FORTAT (/20X,"VECTOR DE COEFICIENTES DE LA SOLUCION Z(T) (Z4)**/)	00039200
39300	CALL SACAM (Y(1,22),N,1)	00039300
39400	WRITE (6,158)	00039400
39500	158 FORTAT(/20X,"VECTOR DE TERMINOS CONSTANTES DE LA SOLUCION Z(T)**/)	00039500
39600	CALL SACAM (Y(1,23),N,1)	00039600
39610	FLAG11 = 6H	00039610
39620	WRITE (6,167)	00039620
39630	167 FORTAT (/10X,"DESEAS OTRO CONJUNTO DE VALORES INICIALES DEL OBSERV	00039630
39632	*ALOR (SI) (NO)**/)	00039632
39640	READ (5,120) FLAG11	00039640
39650	IF (FLAG11 .EQ. SI) GO TO 18	00039650
39700	FLAG9=6H	00039700
39800	WRITE (6,159)	00039800
39900	159 FORTAT (/10X,"DESEAS GRAFICAR LAS SOLUCIONES (SI) (NO)**/)	00039900
40000	READ (5,120) FLAG9	00040000
40100	IF (FLAG9 .EQ. SI) CALL FINAL (X,Y,1)	00040100
40200	10 WRITE (6,119)	00040200
40300	119 FORTAT (/10X,"SELECCIONAS OTRO CONJUNTO DE FOLDS (SI) (NO)**/)	00040300
40400	READ (5,120) FLAG2	00040400

40500	IF (FLAG2 .EQ. SI ) GO TO 23	00040500
40600	FLAG3 = 6H	00040600
40700	WRITE (6,121)	00040700
40800	121 FORMAT (/10X,"DESEAS RESOLVER OTRO SISTEMA (SI) (NO)"/)	00040800
40900	FLAG (5,120) FLAG3	00040900
41000	IF (FLAG3 .EQ. SI ) GO TO 3	00041000
41100	WRITE (6,122)	00041100
41200	122 FORMAT (20X,"EJECUCION TERMINADA"/)	00041200
41300	CALL EXIT	00041300
41400	END	00041400
41500	SUBROUTINE FINAL (X,Y,I)	00041500
41600	REAL * 8 X,Y,T1,DT	00041600
41700	DIMENSION X(10,12),Y(10,23),A(10,51),F(10,51),CT(51),V1(51),V2(51)	00041700
41800	SI = 6I:SI	00041800
41900	WRITE (6,100)	00041900
42000	100 FORMAT (/20X,"SE GRAFICAN LAS VARIABLES DE ESTADO POR PARES (LAS	00042000
42100	*ELL SISTEMA Y LA ESTIMACION"/	00042100
42150	*20X,"CORRESPONDIENTE DEL OBSERVADOR) A INTERVALO CONSTANTE"/	00042150
42152	*20X,"LAS GRAFICAS PUEDEN SER DE 51 PUNTOS O MENOS."/	00042152
42154	*20X,"EL EJE DE LOS TIEMPOS X(T)=Z(T)=0.0 ESTA INDICADO POR LA LET	00042154
42155	*RA I"/)	00042155
42300	1 WRITE (6,101)	00042300
42400	101 FORMAT (/10X,"DAME EL TIEMPO INICIAL *EL INCREMENTO Y EL NUMERO DE	00042400
42410	* PUNTOS QUE DESEAS"/)	00042410
42500	FLAG (5,/) TI=DT*I	00042500
42550	IF (TI .GE. 0.0 .AND. DT .GT. 0.0) GO TO 5	00042550
42552	WRITE (6,106)	00042552
42554	106 FORMAT (30X,"ERROR: EL TIEMPO INICIAL ES NEGATIVO O EL INCREMENTO	00042554
42555	*NO ES POSITIVO"/)	00042555
42556	GO TO 1	00042556
42600	* CALL VALUA (X,T,I,DT,A,F,CT,V1,V2)	00042600



42700	DL 2 I=1,N	00042700
42800	DL 3 J=1,NP	00042800
42900	3 V1(J) = A(I,J)	00042900
43000	DL 4 J=1,NP	00043000
43100	4 V2(J) = B(I,J)	00043100
43140	FLAG2 = 6H	00043148
43150	WRITE (6,103) I,I	00043150
43152	103 FORMAT (/10X,"DESEAS GRAFICAR LAS VARIABLES X",I2," Y Z"	00043152
43154	" ,I2," (SI) (NO)"/)	00043154
43160	FLAL (5,120) FLAG2	00043160
43165	IF (FLAG2 .NE. SI) GO TO 2	00043165
43166	WRITE (6,105) I,I	00043166
43167	105 FORMAT (/50X,"VARIABLES X",I2,"(*) DEL SISTEMA Y Z",I2,	00043167
43168	"(0) ALL OBSERVADOR")	00043168
43170	WRITE (6,104) I,I	00043170
43175	104 FORMAT (" TIEMPO Y",I2,8X,"Z",I2)	00043175
43200	CALL GRAFIC (V1,V2,CT,FP)	00043200
43250	2 CONTINUE	00043250
43300	FLAG1 = 6H	00043300
43400	WRITE (6,102)	00043400
43500	102 FORMAT (/10X,"DESEAS CALCULAR PARA OTROS TIEMPOS (SI) (NO)"/)	00043500
43600	FLAL (5,120) FLAG1	00043600
43700	120 FORMAT (A6)	00043700
43800	IF (FLAG1 .EQ. SI) GO TO 1	00043800
43900	RETURN	00043900
44000	END	00044000
44100	SUBROUTINE SACAR (A,NP,I)	00044100
44200	FLAL * 0 A(10,10)	00044200
44300	DL 1 I=1,N	00044300
44400	1 WRITE (6,10)(A(I,J),J=1,N)	00044400
44500	10 FORMAT (10F12.4)	00044500

44600	RETURN	00044600
44700	END	00044700
44800	SUBROUTINE VALUA (X,Y,T,DT,V,H,CT,P)	00044800
44900	REAL * 8 X,Y,T,DT,T,E1,E2,A	00044900
45000	DIMENSION X(10,12),Y(10,23),V(10,51),H(10,51),CT(51),F1(10),E2(10)	00045000
45100	T = T	00045100
45200	DO 1 I=1,N	00045200
45300	DO 2 I=1,N	00045300
45400	E1(I) = DEXP(Y(I,21)*T)	00045400
45500	2 E2(I) = DEXP(Y(I,22)*T)	00045500
45600	DO 3 I=1,N	00045600
45700	A = 0.0	00045700
45800	DO 4 J=1,N	00045800
45900	4 A = A + X(I,J) * E1(J)	00045900
46000	V(I,1) = SNGL(A+X(I,12))	00046000
46100	A = 0.0	00046100
46200	DO 5 J=1,N	00046200
46300	5 A = A + Y(I,J) * E1(J) + Y(I,10+J) * E2(J)	00046300
46400	3 H(I,1) = SNGL(A + Y(I,23))	00046400
46500	CT(I) = SNGL(A)	00046500
46600	1 T = T + DT	00046600
46700	RETURN	00046700
46800	END	00046800
46900	SUBROUTINE GRAPHIC (X,Y,Z,H)	00046900
47000	DIMENSION X(1),Y(1),Z(1),FILEA(101)	00047000
47100	C = X(1)	00047100
47200	G = C	00047200
47300	DO 1 I=1,N	00047300
47400	C = AMIN1(C,X(I),Y(I))	00047400
47500	1 G = AMAX1(G,X(I),Y(I))	00047500
47550	WRITE(6,101) C,G	00047550

47555	101	FORMAT (30X,"(MII:III)">F11.4>64X>F11.4,"(MAXIII)")	00047555
47600		ES = (G-C)/100.	00047600
47650		EJE = G*C	00047650
47700		L = ABS (C/ES) + 1.5	00047700
47800		ILS = 1.0/ES	00047800
47900		LL 3 I=1,101	00047900
48000	3	FILA(I) = 1n	00048000
48100		K=1	00048100
48200		J=1	00048200
48300	DL 4	I=1,N	00048300
48400		FILA (J) = 1n	00048400
48500		FILA (I) = 1n	00048500
48600		FILA (1) = 1n+	00048600
48650		FILA (101) = 1n+	00048650
48700		IF (LJE .LE. 0.0 ) FILA (L) = 1nI	00048700
48900		J = (X(I) -C) *RES +1.5	00048900
49000		K = (Y(I)-C) *RES +1.5	00049000
49100		FILA (J) = 1n+	00049100
49200		FILA (K) = 1n0	00049200
49300	4	WRITE (6,100) Z(I),X(I),Y(I),FILA	00049300
49400	100	FORMAT (1X,>F6.3>2F11.4>2X>101A1)	00049400
49500		RETURN	00049500
49600		END	00049600
49700		SUBROUTINE MODAL (A>VALR>VALI>N>TOL>Y>IPI)	00049700
49800		REAL *8 A>VALR>VALI>TOL>VEC>T1>T2>T3>T4>Y>X >AA	00049800
49900	C		00049900
50000	C	ESTA SUBROUTINE GENERA LA MATRIZ MODAL DE UNA MATRIZ A .	00050000
50100	C	REQUIERE LA MATRIZ A Y LOS VALORES PROPIOS DE LA MISMA.	00050100
50200	C	DETECTA SI HAY VALORES REALES, COMPLEJOS, Y SI ES POSIBLE	00050200
50300	C	GENERAR TODO EL ESPACIO DE N DIMENSIONES CUANDO HAY	00050300
50400	C	MULTIPLICIDAD EN LAS RAICES.	00050400

50500	r		00050500
50600		DIMEI, SIGN A(10,10), VALR(10), VALI(10), AA(10,10), T2(10), I3(10),	00050600
50700		SI(10), T4(10), X(10,10), Y(10,10)	00050700
50800		II.D = 0	00050800
50900	r		00050900
51000	r	LECTURA DE LA PARTE IMAGINARIA DE LAS RAICES.	00051000
51100	r		00051100
51200		DL 1 I=1,N	00051200
51300		IF (LABS(VALI(I)) * GE. TOL.) GO TO 10	00051300
51400		1 CONTINUE	00051400
51500	r		00051500
51600	r	ORDENACION DE LAS RAICES.	00051600
51700	r		00051700
51800		II. = I-1	00051800
51900		DL 2 I = 1, NN	00051900
52000		T1 = VALR(I)	00052000
52100		DL 3 J = I+1, N	00052100
52200		IF (T1 * LE. VALR(J)) GO TO 3	00052200
52300		VALR(I) = VALR(J)	00052300
52400		VALR(J) = T1	00052400
52500		T1 = VALR(I)	00052500
52600		3 CONTINUE	00052600
52700		2 CONTINUE	00052700
52800	r		00052800
52900	r	DEFINICION DE LA MULTIPLICIDAD DE CADA RAIZ Y ELIMINACION DE	00052900
53000	r	LOS VALORES REPETIDOS.	00053000
53100	r		00053100
53200		T2(I) = VALR(I)	00053200
53300		DL 5 I=1,N	00053300
53400		5 K(I) = 1	00053400
53500		K = 1	00053500

53600	DL 6 I = 2 * N	00053600
53700	IF (ABS(VALR(I-1)-VALR(I)) .LT. TOL) GO TO 4	00053700
53800	K = I + 1	00053800
53900	T2 (I) = VALR (I)	00053900
54000	GL TL 0	00054000
54100	4 M(K) = M(K) + 1	00054100
54200	6 CONTINUE	00054200
54300	C	00054300
54400	C SI K ES IGUAL A N, IMPLICA QUE NO HAY VALORES REPETIDOS.	00054400
54500	C	00054500
54600	IF (I .GE. N) GO TO 11	00054600
54700	DL 7 I=1,K	00054700
54800	IF (M(I) .LE. 1) GO TO 7	00054800
54900	C	00054900
55000	C ANALISIS DE LA DEGENERACION EN PUNOS REPETIDOS.	00055000
55100	C	00055100
55200	DL 8 J= 1,N	00055200
55300	DL 9 L=1,N	00055300
55400	0 AA(J,L) = -A(J,L)	00055400
55500	1 AA(J,J) = AA(J,J) + T2(I)	00055500
55600	CALL LLAMA (AA,N,1,10,10,1)	00055600
55700	CALL HGR (AA,N,1,TOL,IF,T3,T4)	00055700
55800	IF (IE + M(I) .EQ. 1) GO TO 7	00055800
55900	WRITE (6,100) I2(I),M(I),IF	00055900
56000	100 FLEAT (///40A,"MATRIZ NO DIAGONALIZABLE"/	00056000
56100	+45X,"EL VALOR PROPIO",F12.4," TIENE MULTIPLICIDAD",I4/	00056100
56200	845X,"Y LA MATRIZ (SI=A) ASOCIADA A ESTE VALOR ES DE RANGO",I5/	00056200
56300	845X,"LS DECIR, POSEE DEGENERACION INCOMPLETA."/)	00056300
56400	IFL = 1	00056400
56500	RETURN	00056500
56600	7 CONTINUE	00056600

56700	C		00056700
56800	C	GENERACION DE LOS VECTORES PROPIOS Y FORMACION DE LA MATRIZ	00056800
56900	C	MODAL Y.	00056900
57000	C		00057000
57100		11 J2 = 0	00057100
57200		J1 = 1	00057200
57300		LL 12 I = 1,N	00057300
57400		J2 = J2 + N(I)	00057400
57500		LL 13 J = 1,N	00057500
57600		LL 13 L = 1,N	00057600
57700		13 AA(J,L) = A(J,L)	00057700
57800	C		00057800
57900	C	X ES LA MATRIZ QUE TRANSPORTA LOS VECTORES PROPIOS DE VECPRD	00057900
58000	C	A MODAL.	00058000
58100	C		00058100
58200		CALL VECPRD (A,N,T2(I),I,TOL,X)	00058200
58300		J3 = 0	00058300
58400	C		00058400
58500	C	ALMACENAMIENTO DE LOS VECTORES PROPIOS EN LA MATRIZ MODAL Y.	00058500
58600	C		00058600
58700		LL 15 J = J1,J2	00058700
58800		J3 = J3 + 1	00058800
58900		LL 15 L = 1,N	00058900
59000		15 Y(L,J) = X(L,J3)	00059000
59100		12 J1 = J2 + 1	00059100
59200		IID = 3	00059200
59300		WRITE	00059300
59400		10 WRITE (6,101)	00059400
59500		101 FLEMAT (//40X,"VALORES PROPIOS COMPLEJOS"/	00059500
59600		"45X,"EL PROGRAMA AUN NO CONTEMPLA ESTA POSIBILIDAD"/)	00059600
59700		IID = 2	00059700

59800		RTULU	00059800
59900		END	00059900
60000		SUBROUTINE VECTOR (A,VAL,N,TOL,X)	00060000
60100	C		00060100
60200	C	DETERMINA LOS VECTORES PROPIOS LINEALMENTE INDEPENDIENTES ASOCIADOS A UN VALOR PROPIO DADO.	00060200
60300	C		00060300
60400	C	JAMES SPITZ, WOLFORD (PAG 278)	00060400
60500	C	MODIF E IMPLANT POR H. SANCHEVAL	00060500
60600	C	DIC 1977	00060600
60700	C		00060700
60800		REAL *8 A,VAL,TOL,GDE *X *R *AUX	00060800
60900		DIMENSION A(10,10),B(10,10),X(10,10),IX(10) ,AUX(11,10)	00060900
61000	C		00061000
61100	C	CALCULO DE LOS ELEMENTOS EN LA DIAGONAL DE LA MATRIZ DE COEFICIENT	00061100
61200	C		00061200
61300		DO 7 I=1,N	00061300
61400		7 A(I,I) = A(I,I) -VAL	00061400
61500	C		00061500
61600	C	REGISTRO DEL ORDEN INICIAL DE LAS COMPONENTES DEL VECTOR CARACTERISTICO.	00061600
61700	C		00061700
61800	C		00061800
61900		DO 8 I = 1,N	00061900
62000		* IX(I) = I	00062000
62100		K= K	00062100
62200		* K= K-1+1	00062200
62300	C		00062300
62400	C	LOCALIZACION DEL MAYOR ELEMENTO PIVOTE.	00062400
62500	C		00062500
62600		JJ= 1	00062600
62700		II=1	00062700
62800		GLE =ABS(A(I,I))	00062800

62900		DL 11 J=1,K	00062900
63000		DL 11 I=1,K	00063000
63100		IF (GDL*GE. DABS(A(I,J))) GO TO 11	00063100
63200		JJ= J	00063200
63300		II= 1	00063300
63400		GLE = LABS (A(I,J))	00063400
63500		11 CONTINUE	00063500
63600	C		00063600
63700	C	VERIFICACION Y EN SU CASO EJECUCION DEL INTERCAMBIO DE	00063700
63800	C	REGIONES Y/O COLUMNAS.	00063800
63900	C		00063900
64000		IF (EPS *EG. 0.0 ) EPS = TOL * GDL	00064000
64100		IF ( II *LE. 1 ) GO TO 14	00064100
64200		DL 13 J=1,K	00064200
64300		TEMP = A(I1,J)	00064300
64400		A(I1,J) = A(I,J)	00064400
64500		13 A(I,J) = TEMP	00064500
64600		14 IF (JJ*LE. 1) GO TO 17	00064600
64700	C	INTERCAMBIO DE COLUMNAS Y REGISTRO DEL MISMO.	00064700
64800		TEMP = IX(M)	00064800
64900		L = JJ + N - K	00064900
65000		IX(M) = IX(L)	00065000
65100		IX(L) = TEMP	00065100
65200		DL 16 I = 1,K	00065200
65300	C		00065300
65400	C	ELIMINACION DE GAUSS-JORDAN.	00065400
65500	C		00065500
65600		TEMP = A(I,JJ)	00065600
65700		A(I,JJ) = A(I,1)	00065700
65800		16 A(I,1) = TEMP	00065800
65900		17 IF (DABS(A(I,1)) *LE. EPS) GO TO 23	00065900



```

06000      TLIF = 1.0/A(1,1)
06100      1# LL 20 J=2,K
06200      DL 20 I=2,N
06300      E(I-1,J-1) = A(1,J) - A(1,J)*A(I,1) * TEMP
06400      IF (ABS(B(I-1,J-1)) .GE. EPS *DABS(A(I,J))) GO TO 20
06500      E(I-1,J-1) = 0.0
06600      2# CONTINUE
06700      LL 21 J = 2,K
06800      21 E(I,J-1) = A(1,J) *TEMP
06900      I=I-1
07000      IF (I.LE. 0) GO TO 31
07100      LL 22 J=1,K
07200      DL 22 I=1,N
07300      2# A(I,J) = B(I,J)
07400      GL TO 9
07500      C
07600      C      RETENCION DE LOS VECTORES LINEALMENTE INDEPENDIENTES A PARTIR
07700      C      DE LA MATRIZ A.
07800      C
07900      23 DL 24 J=1,K
08000      24 A(J,J) = -1.
08100      IF (I.LT. 10) GO TO 59
08200      LL 57 J = 1,10
08300      DL 57 I = 1,10
08400      57 ALX(I,J) = A(I,J)
08500      DL 60 I=1,K
08600      LL 60 J = 1,K
08700      ALX(I+1,J) = ALX(I,J)
08800      DL 60 I=1,N
08900      6# ALX(I,J) = AUX(I+1,J)
09000      LL 62 J=1,10

```

```

00066000
00066100
00066200
00066300
00066400
00066500
00066600
00066700
00066800
00066900
00067000
00067100
00067200
00067300
00067400
00067500
00067600
00067700
00067800
00067900
00068000
00068100
00068200
00068300
00068400
00068500
00068600
00068700
00068800
00068900
00069000

```



69100	DL 62 I=1,K	00069100
69200	62 A(I,J) = AUX(I,J)	00069200
69300	GL TL 61	00069300
69400	50 DL 25 KH= 1,K	00069400
69500	DL 25 J=1,K	00069500
69600	A(I+1,J) = A(I,J)	00069600
69700	DL 25 I=1,N	00069700
69800	25 A(I,J) = A(I+1,J)	00069800
69900	61 DL 27 J=1,K	00069900
70000	DL 27 I=1,K	00070000
70100	DL 27 LUM=1,N	00070100
70200	IF (IX(NUM) .NE.1) GO TO 27	00070200
70300	X(I,J) = A (NUM,J)	00070300
70400	27 CONTINUE	00070400
70500	C	00070500
70600	C IMPRESION DE LOS VECTORES PROPIOS LINEALMENTE INDEPENDIENTES.	00070600
70700	C	00070700
70800	C WRITE (6,100)	00070800
70900	100 FLMAT (/10X,"EL(LOS) VECTOR(ES) PROPIO(S) LINEALMENTE INDEPENDIENTES)	00070900
71000	*TE(S) ASOCIADO(S) CON",F10.4," ES(S)"/)	00071000
71100	C DL 29 I=1,N	00071100
71200	C 20 WRITE(6,101) (A(I,J),J=1,K)	00071200
71300	101 FLMAT (10X,10F12.4)	00071300
71400	RETURN	00071400
71500	31 WRITE(6,102)	00071500
71600	102 FLMAT (//20X,"EL RANGO DE (SI=A) ES IGUAL AL ORDEN DE A"/	00071600
71700	*20X,"L EJECUCION TERMINADA")	00071700
71800	CALL EXIT	00071800
71900	END	00071900
72000	FUNCTION HOR (A,N,X)	00072000
72100	C	00072100

72200	C	EVALUA EL POLINOMIO $A(1) + A(2)*X + \dots + A(N)*X^{*(N+1)}$	00072200
72300	C	POR EL METODO DE HORNER	00072300
72400	C		00072400
72500	C	A VECTOR DE COEFICIENTES. EL PRIMER ELEMENTO ES EL TERMINO	00072500
72600	C	INDEPENDIENTE.	00072600
72700	C	L GRADO DEL POLINOMIO HAS U.C.	00072700
72800	C	X VALOR DE LA VARIABLE INDEPENDIENTE	00072800
72900	C		00072900
73000	C	REAL *B A* X*NOF.	00073000
73100	C	DIMENSION A(1)	00073100
73200	C	IT = 1	00073200
73300	C	HCF = 0.0	00073300
73400	C	DO 1 I = 1,N	00073400
73500	C	HCF = HOR * X + A(IT)	00073500
73600	C	1 IT = IT + 1	00073600
73700	C	LLTUI.	00073700
73800	C	LLI.	00073800
73900	C	SUBROUTINE GREAT (F*H*DI*ID*IP*IT*IK)	00073900
74000	C		00074000
74100	C	I (10*10*20) MATRIZ POLINOMIAL DE I*N	00074100
74200	C	I. NUMERO DE RENGLONES DE F	00074200
74300	C	I. NUMERO DE COLUMNAS DE F	00074300
74400	C	II,LI(1) RENGLONES SELECCIONALES PARA FORMAR LA MATRIZ A	00074400
74410	C	II,CI(1) UBICACION DE LOS NUEVOS POLOS.	00074410
74415	C	T MATRIZ (I*N) DE ELEMENTOS CONSTANTES GENERADA AL SU	00074415
74418	C	BSTITUIR LOS POLOS DE F(1) EN LOS RENGLONES LLECI	00074418
74419	C	DES	00074419
74420	C	II MATRIZ IDENTIDAD EXPANDIDA DE ACUERDO CON LA SELECCION	00074420
74422	C	EMPLEADA EN T	00074422
74425	C	II=1 INDICA QUE ALGUN POLO NUEVO ES IGUAL A UNO VIEJO.	00074425
74500	C	REAL *B HOR*F*H*IT*DI*ID*IP*IT*IK	00074500

74600	DIMENSION T(10,10,20), IREN(10), N(10), T(10,10), IH(10,10), TEL(20),	00074600
74700	*ML (150), LIV(20)	00074700
74800	DATA HE/150*0/, IK/0/	00074800
74900	IF (IK .EQ. 0) IK = 0	00074900
75000	IL = 1	00075000
75100	IF (IK .LE. 0) GO TO 5	00075100
75200	DO 7 I = 1, IK	00075200
75300	IA = (I-1) * N + 1	00075300
75400	DO 6 J = 1, N	00075400
75500	IF (HE(IA) .NE. IREN(J)) GO TO 7	00075500
75600	IA = IA + 1	00075600
75700	CONTINUE	00075700
75800	WRITE (6,104) I	00075800
75900	104 FORMAT (/10X,"ESTA ORDENACION DE DIMENSIONES YA SE PROBO EN LA ITER-	00075900
76000	*ACION." ,13//)	00076000
76100	CONTINUE	00076100
76200	7 CONTINUE	00076200
76300	5 IIT = I*N	00076300
76400	IIT = MINO (10,IIT)	00076400
76500	IF (I .LE. IIT) GO TO 10	00076500
76600	WRITE (6,102) (I,I=1,N)	00076600
76700	102 FORMAT (/10X,"YA SE HICIERON DIEZ INTENTOS, QUE ES EL LIMITE DE	00076700
76800	*ITERACIONES"/10X,"SE EMPLEARON LAS SIGUIENTES ORDENACIONES:"/	00076800
76900	*5X,"ITER",10(" BFI",I?))	00076900
77000	DO 9 I = 1,15	00077000
77100	II = I*N	00077100
77200	IA = II-N+1	00077200
77300	6 WRITE (6,103) I, (PE(J), J=IA,IP)	00077300
77400	103 FORMAT (/2X,1117)	00077400
77500	WRITE (6,105)	00077500
77600	105 FORMAT (/10X,"ASIGNAR NUEVOS POLOS CO REPETIDOS EN CASO DE SER RL	00077600

77700	*CLSARIE)"/)	00077700
77800	II. = 2	00077800
77900	ELTUII.	00077900
78000	12 WRITE (6,106) 1,II*(J,IPEN(J),J=1,II)	00078000
78100	106 FLEMAT (/10X,"EL RENGLOH SOLICITADO PARA LA POSICION" ,I3," ES MAY	00078100
78200	*EL QUE" ,I3," (GRUPO DE SALIDAS)"/)	00078200
78300	*(10X,"PARA EL RENGLOH" ,I3," SE PIDE EL RENGLOH" ,I3//)	00078300
78400	ELTUII.	00078400
78500	10 IA = II * N	00078500
78600	II. = II + 1	00078600
78700	WRITE (6,107) 1K,(I,EN(J),J=1,II)	00078700
78800	107 FLEMAT (/ 2X,"INTENTO PODER" ,I3," PARA COPIAR LA MATRIZ N CON I	00078800
78900	*OS RELACIONES :"	00078900
79000	WRITE (6,121) (PH(J),J=1,II)	00079000
79100	121 FLEMAT (/37X,"I LOS PULOS DEL OBSERVADOR :"	00079100
79200	UL 6 I = 1,N	00079200
79300	IA = IA + 1	00079300
79400	IF (I,EN(1) .GT. 1) GO TO 12	00079400
79500	7 H(I,IA) = I,EN(I)	00079500
79600	LL 11 I = 1,N	00079600
79700	LL 11 J = 1,N	00079700
79800	11 II(I,J) = 0	00079800
79900	LL 1 I = 1,N	00079900
80000	II. = II,EN(1)	00080000
80100	II(I,II) = 1	00080100
80200	ULNEN = HOR(CDIV,II+1,PH(II))	00080200
80300	IF (LALS(DENOM) .GE. TPI) GO TO 3	00080300
80400	WRITE (6,100) EN(I)	00080400
80500	100 FLEMAT (/10X,"EL VALOR DE UN NUEVO PULO" ,F10.4," COINCIDE CON AL	00080500
80600	*GULO DE LOS ORIGINALES"/)	00080600
80700	II. = 2	00080700

80800	RETURN	00080800
80900	3 DL 2 J = 1,N	00080900
81000	RG = HZEL (I,J,IR,20,TCL)	00081000
81100	IF (RG .GT. 0) GO TO 13	00081100
81200	T(I,J) = 0.0	00081200
81300	GL TO 2	00081300
81400	13 DL 4 K = 1,NG	00081400
81500	4 TLR(K) = F(J,IR,K)	00081500
81600	T(I,J) = HOR(TLR,RG,PH(I))/DEFOP	00081600
81700	2 CONTINUE	00081700
81800	1 CONTINUE	00081800
81900	KI = 3	00081900
82000	RETURN	00082000
82100	END	00082100
82200	SUBROUTINE LIMFIA(A,N,I,K)	00082200
82300	REAL *8 A	00082300
82400	DIMENSION A(1)	00082400
82500	IT = I+N*K	00082500
82600	DL 1 I=1 ,N1	00082600
82700	4 A(I) = 0.0	00082700
82800	RETURN	00082800
82900	END	00082900
83000	SUBROUTINE NUMCHP (A1,A2,A3,L1,I1,L2,I2,I3,I3,TD1,TCL)	00083000
83100	IMPLICIT REAL*8 (A,T)	00083100
83200	DIMENSION A1(10,10),A2(10,10,20),A3(10,10,20),TEMP(20)	00083200
83300	IF (I1 .NE. L2) GO TO 100	00083300
83400	L3 = L1	00083400
83500	I3 = I2	00083500
83600	DL 50 I = 1, M3	00083600
83700	DL 50 J = 1, L3	00083700
83800	DL 25 I1 = 1, I01	00083800

83500	25	TEMP (I,1) = 0.0	00083900
84000		DO 20 I = 1, M1	00084000
84100		MA1 = 1	00084100
84200		IF (DABS (A1(J,N)) .LE. TOL ) MA1=0	00084200
84300		MA2 = IZEL (A2,I,N, ID1, TOL)	00084300
84400		IF ( MA1 .EQ. 0 .OR. MA2 .EQ. 0) GO TO 20	00084400
84500		TT = A1 (J,N)	00084500
84600		DO 10 I,2 = 1, MA2	00084600
84700	10	TEMP (I,2) = TEMP (K2) + TT * A2(I,I,K2)	00084700
84800	20	CONTINUE	00084800
84900		DO 30 K1 = 1, I01	00084900
85000		A3 (I,J,K1) = TEMP (K1)	00085000
85100	30	CONTINUE	00085100
85200	50	CONTINUE	00085200
85300		RETURN	00085300
85400	100	WRITE (6,900)	00085400
85500	900	FORMAT (/2X,"ILLEGAL MATRIX MULTIPLICATION (CONFORMABILITY)")	00085500
85600		RETURN	00085600
85700		END	00085700
85800		SUBROUTINE RANIGT(X,Y,I,J,EPS,IR)	00085800
85900	C		00085900
86000	C	DESCRIPCION :	00086000
86100	C		00086100
86200	C	SE DETERMINA EL RANGO DE LAS MATRICES (X,Y) DE ACUERDO CON:	00086200
86300	C	EL CRITERIO DE HALLAN, O SEA, T= (Y,XY,XXY,XXX,....)	00086300
86400	C	--- SI Y ES LA MATRIZ C DE LA ECUACION DE ESTADO, SE INFIERE	00086400
86500	C	ACERCA DE LA CONTROLABILIDAD DEL SISTEMA.	00086500
86600	C	--- SI Y ES LA TRANSUESTA DE LA MATRIZ C DE LA ECUACION DE	00086600
86700	C	SALIDA, SE INFIERE ACERCA DE LA OBSERVABILIDAD DEL SISTEMA	00086700
86800	C		00086800
86900	C	ALGUMENTOS	00086900

67000	C	X	MATRIZ DE M*N ES LA MATRIZ A EN AMBOS CASOS	00087000
67100	C	Y	MATRIZ DE M*N	00087100
67200	C	I:	NUMERO DE REPLICACIONES (VARIABLES DE ESTADO)	00087200
67300	C	I:	NUMERO DE COLUMNAS (EXCITACIONES O SALIDAS)	00087300
67400	C	LPS	TOLERANCIA RELATIVA DE LOS ELEMENTOS DE LAS MATRICES EN	00087400
67500	C		FUNCION DEL MAXIMO VALOR ABSOLUTO PIV.	00087500
67600	C		SI ABS(T(I,J)) ES MENOR QUE ABS(PIV)*EPS, IMPLICA T(I,J)	00087600
67700	C		IGUAL A CERO DURANTE EL PIVOTEO.	00087700
67800	C	IR:	RANGO DE LA MATRIZ T	00087800
67900	C		SI IR = -1 ERROR EN LAS DIMENSIONES DE LA MATRIZ	00087900
68000	C		SI IR = 0 LA MATRIZ ES NULA.	00088000
68100	C			00088100
68200	C			00088200
68300	C			00088300
68400	C			00088400
68500			REAL *B X,Y,XT,YT,T,TT,EPS	00088500
68600			(DIMENSION X(10,10),Y(10,10),XT(10),YT(100),T(10,100),TT(10,100))	00088600
68700			DL 11 I2 = 1,M	00088700
68800			LL 11 I1 = 1,N	00088800
68900		11	TT (I1,I2) = Y(I1,I2)	00088900
69000			CALL LLAMA (IT,M,I,EPS,IR,XT,YT)	00089000
69100			CALL HFOR (I1,M,M,EPS,IR,XT,YT)	00089100
69200			IF (IR) 1,2,3	00089200
69300		1	WRITE (6,100)	00089300
69400		100	FORMAT (//10X, "ERROR EN LAS DIMENSIONES DE LA MATRIZ Y"//)	00089400
69500			RETURN	00089500
69600		2	WRITE (6,101)	00089600
69700		101	FORMAT (//10X, "LA MATRIZ Y ES UNA MATRIZ NULA"//)	00089700
69800			RETURN	00089800
69900		3	IF (IR .LT. N) GOTO 4	00089900
90000			WRITE (6,102)	00090000

H. SANDOVAL R.  
 DICIEMBRE 1977



90100	102	FORMAT (//10X,"LA MATRIZ Y ES CUADRADA Y DE RANGO N"/)	00090100
90200		RETURN	00090200
90300	4	IF = N-1	00090300
90400	5	DO 6 J = 1,N	00090400
90500		DO 6 I = 1,N	00090500
90600	6	T(I,J) = Y(1,J)	00090600
90700		DO 10 I = 1,NP	00090700
90800		II = I+1	00090800
90900		III = II+M	00090900
91000		CALL MULMAT (X,Y,T(1,II+1),I,II,III)	00091000
91100		DO 7 II = 1,III	00091100
91200		DO 7 I2 = 1,N	00091200
91300	7	TT(I2,II) = T(I2,II)	00091300
91400		IF (III .GE. N) CALL LIAMA (TT,II,III,10,100,1)	00091400
91500		IF (III .GE. N) CALL HFGR(TT,I,III,EPS,IR,XT,YT)	00091500
91600	C	WRITE (6,104) IR,I	00091600
91700	104	FORMAT (//10X,"EL RANGO ES",I4," HASTA LA POTENCIA",I4,"A DE A"/)	00091700
91800		IF (II .LT. N) GO TO 8	00091800
91900	C	WRITE (6,105) IR,I	00091900
92000	105	FORMAT (//10X,"EL VALOR DEL RANGO",I4," ES IGUAL AL ORDEN",I4," DE	00092000
92100		"L. SISTEMA"/)	00092100
92200		RETURN	00092200
92300	8	JJ = 0	00092300
92400		DO 9 J = II + 1,III	00092400
92500		JJ = JJ + 1	00092500
92600		DO 9 II = 1,N	00092600
92700	9	Y(II,JJ) = T(II,J)	00092700
92800	10	CONTINUE	00092800
92900		WRITE (6,106)	00092900
93000	106	FORMAT (//10X,"EL RANGO DE LA MATRIZ ANEXIADA ES MENOR QUE EL ORDEN DE	00093000
93100		"L. SISTEMA"/)	00093100

93200	RETURN	00093200
93300	END	00093300
93400	SUBROUTINE MULMAT (X,Y,Z,M,L,N)	00093400
93500	REAL *8 X,Y,Z,I	00093500
93600	DIMENSION X(10,10),Y(10,10),Z(10,10)	00093600
93700	DO 2 I = 1,N	00093700
93800	DO 2 J = 1,M	00093800
93900	T = 0.0	00093900
94000	DO 1 K = 1,L	00094000
94100	1 T = T + X(1,K)*Y(K,J)	00094100
94200	2 Z(I,J) = T	00094200
94300	RETURN	00094300
94400	END	00094400
94500	SUBROUTINE LLAMA (A,M,N,IND,IRD)	00094500
94600	C	00094600
94700	C "LLAMA" ORGANIZA LOS ELEMENTOS DE UN ARREGLO DE ACUERDO CON EL MANE-	00094700
94800	C JO MATRICIAL DE IBM O BURROUGHS.	00094800
94900	C	00094900
95000	C A MATRIZ	00095000
95100	C I. ORDEN DE LA MATRIZ (DEBE SER MENOR O IGUAL A MD)	00095100
95200	C I. ORDEN DE LA MATRIZ (DEBE SER MENOR O IGUAL A MD)	00095200
95300	C I.F DIMENSION DECLARADA PARA LA MATRIZ A EN EL PROGRAMA QUE	00095300
95400	C I.D DIMENSION DECLARADA PARA LA MATRIZ A EN EL PROGRAMA QUE	00095400
95500	C LLAMA A "LLAMA"	00095500
95600	C IND = 1 TRANSFORMA UNA MATRIZ A UN VECTOR CONTINUO	00095600
95700	C IND = 2 TRANSFORMA UN VECTOR CONTINUO A UNA MATRIZ	00095700
95800	C	00095800
95900	C	00095900
96000	C	00096000
96100	C	00096100
96200	C REAL *8 A	00096200

RODRIGO SANDOVAL R.  
 INST. ING. UNAM, NOV 1977.

96300		DIMENSION A(1)	00096300
96400	C		00096400
96500	C	CUANDO N = 00 NO SE REQUIERE NINGUNA TRANSFORMACION	00096500
96600	C		00096600
96700		IF (L *GT* ND *OR* H *GT* ND) GO TO 5	00096700
96800		IF (H *ER* ND) RETURN	00096800
96900		GL TO (2*3)*IND	00096900
97000	C		00097000
97100	C	TRANSFORMACION DE MATRIZ A VECTOR CONTINUA	00097100
97200	C		00097200
97300		2 I = 0	00097300
97400		LL 1 IC = 1 * N	00097400
97500		I = (IC - 1) * ND	00097500
97600		LL 1 L = 1 * N	00097600
97700		I = I + 1	00097700
97800		1 A(I) = A (K+L)	00097800
97900		RTURN	00097900
98000	C		00098000
98100	C	TRANSFORMACION DE VECTOR CONTINUA A MATRIZ	00098100
98200	C		00098200
98300		3 III = 1 * H + 1	00098300
98400		LL 4 I = 1 * N	00098400
98500		IC = (III - 1) * ND + III + 1	00098500
98600		LL 4 J = 1 * N	00098600
98700		III = III - 1	00098700
98800		IC = IC - 1	00098800
98900		4 A(II) = A (KN)	00098900
99000		RTURN	00099000
99100		5 WRITE (6*100)	00099100
99200		100 FLPBAT (//100* "LA MATRIZ A EXCEDE LAS DIMENSIONES PREVISTAS")	00099200
99300		CALL EXIT	00099300

99400	EID	00099400
99500	SUBROUTINE MFGK (A, H, EPS, IRANK, IROW, ICOL)	00099500
99600	REAL *8 A, IROW, ICOL, PIV, HOLD, EPS, TOL, SAVE	00099600
99700	DIMENSION A(1), IROW(1), ICOL(1)	00099700
99800	IF (H) 2, 2, 1	00099800
99900	1 IF (I.) 2, 2, 4	00099900
100000	2 IIANK = -1	00100000
100100	3 RETURN	00100100
100200	4 IIANK = 0	00100200
100300	PIV = 0.0	00100300
100400	JJ = 0	00100400
100500	DO 6 J = 1, N	00100500
100600	ICOL(J) = J	00100600
100700	DO 6 I = 1, N	00100700
100800	JJ = JJ + 1	00100800
100900	HOLD = A (JJ)	00100900
101000	IF (ABS(PIV) - DABS(HOLD)) 5, 6, 6	00101000
101100	5 PIV = HOLD	00101100
101200	II = I	00101200
101300	IC = J	00101300
101400	6 CONTINUE	00101400
101500	DO 7 I = 1, N	00101500
101600	7 IIROW(I) = I	00101600
101700	TIL = ABS (EPS*PIV)	00101700
101800	II = I*H	00101800
101900	DO 19 ICOL = M, N, 1	00101900
102000	8 IF(DABS(PIV) - TOL) 20, 20, 9	00102000
102100	9 IIANK = IRANK + 1	00102100
102200	JJ = II - IRANK	00102200
102300	IF (JJ) 12, 12, 10	00102300
102400	10 DO 11 J = IRANK, II, H	00102400

102500	I = J + JJ	00102500
102600	SAVE = A(J)	00102600
102700	A(J) = A(I)	00102700
102800	11 A(I) = SAVE	00102800
102900	JJ = IPON (IR)	00102900
103000	IIOW (IR) = IRON(IRANK)	00103000
103100	IIOW (IRANK) = JJ	00103100
103200	12 JJ = (IC - IRANK) * M	00103200
103300	IF (JJ) 15,15,13	00103300
103400	13 KI = ICOL	00103400
103500	LL 14 J = 1, M	00103500
103600	I = II + JJ	00103600
103700	SAVE = A(KK)	00103700
103800	A(KK) = A(I)	00103800
103900	KI = II + 1	00103900
104000	14 A(I) = SAVE	00104000
104100	JJ = ICOL(II)	00104100
104200	ICOL(II) = ICOL(IRANK)	00104200
104300	ICOL(IRANK) = JJ	00104300
104400	15 II = IRANK + 1	00104400
104500	II = IRANK - M	00104500
104600	LL = ICOL + MM	00104600
104700	IF (II) 16,25,25	00104700
104800	16 JJ = LL	00104800
104900	SAVE = PIV	00104900
105000	PIV = 0.0	00105000
105100	LL 19 J = KK, M	00105100
105200	JJ = JJ + 1	00105200
105300	HOLD = A(JJ)/SAVE	00105300
105400	A(JJ) = HOLD	00105400
105500	L = J - IRANK	00105500

105600	II (IRANK -N) 17,19,19	00105600
105700	17 II=JJ	00105700
105800	DL 19 I = KK,N	00105800
105900	II = II +M	00105900
106000	II. = II -L	00106000
106100	A(II) = A(II) -HOLD + A(III)	00106100
106200	IF (LABS(A(II))-LAPS(PIV))19,19,18	00106200
106300	18 PIV = A(II)	00106300
106400	II. = J	00106400
106500	IC = I	00106500
106600	19 CONTINUE	00106600
106700	20 II (IRANK-1)3,25,21	00106700
106800	21 II. = LL	00106800
106900	LL 24 J = 2,IRANK	00106900
107000	II = J -1	00107000
107100	II. = II. * K	00107100
107200	JJ = LL	00107200
107300	LL 23 I = KK,M	00107300
107400	HLLI = 0.0	00107400
107500	JJ = JJ +1	00107500
107600	III. = JJ	00107600
107700	IC = IR	00107700
107800	LL 22 L = 1,II	00107800
107900	HLLI = HOLD + A(III) * A(IC)	00107900
108000	IC = IC -1	00108000
108100	22 III. = III. *M	00108100
108200	23 A(III) = A(III) - HOLD	00108200
108300	24 CONTINUE	00108300
108400	25 II (N-IRANK)3,3,26	00108400
108500	26 II. = LL	00108500
108600	II. = LL + K	00108600

```

108700      DL 30 J = 1, INAMF                                00108700
108800      DL 29 I = KK, NM, H                                00108800
108900      JJ = IR                                           00108900
109000      LL = I                                           00109000
109100      HLLD = 0.0                                         00109100
109200      II = J                                           00109200
109300      27 II = II - 1                                     00109300
109400      IF (II) 29, 29, 28                                00109400
109500      28 HLLD = HLLD - A(JJ) * A(LL)                    00109500
109600      JJ = JJ - M                                        00109600
109700      LL = LL - 1                                       00109700
109800      GO TO 27                                          00109800
109900      29 A(LL) = (HLLD - A(LL)) / A(JJ)                 00109900
110000      30 II = II - 1                                     00110000
110100      RETURN                                           00110100
110200      END                                             00110200
110300      SUBROUTINE GCRD(A, M, L, P, ID1, TOL, ILL)          00110300
110400      *                                                  00110400
110500      * *****                                             00110500
110600      * *                                                  * 00110600
110700      * * THE SUBROUTINE ZGCRDZ FINDS THE GREATEST COMMON RIGHT DIVISION * 00110700
110800      * * IN UPPER TRIANGULAR FORM OF ALL THE ROWS OF THE INPUT MATRIX. * 00110800
110900      * * THIS IS DONE BY PERFORMING ELEMENTARY ROW OPERATIONS ON THE * 00110900
111000      * * INPUT MATRIX. CONSEQUENTLY THE INPUT MATRIX IS ALTERED BY * 00111000
111100      * * THE SUBROUTINE. IN ADDITION A SECOND MATRIX IS PASSED TO THIS * 00111100
111200      * * ROUTINE WHICH IS CONVERTED TO THE IDENTITY MATRIX OF THE ROW * 00111200
111300      * * DIMENSION OF THE INPUT MATRIX. THE ELEMENTARY ROW OPERATIONS * 00111300
111400      * * ARE ALSO PERFORMED ON THIS MATRIX, THE RESULT BEING THE UNIT * 00111400
111500      * * REGULAR PRE-MULTIPLICATION MATRIX NECESSARY TO CREATE THE GCRD * 00111500
111600      * * DETERMINED. IF THE INPUT MATRIX IS UNIMODULAR THEN THIS MATRIX * 00111600
111700      * * WILL ALSO BE THE INVERSE OF THE INPUT MATRIX. * 00111700

```

111800	C	*		*	00111800
111900	C	*	VARIABLES PASSED:	*	00111900
112000	C	*	A(1,2,3) = 3-D DOUBLE PRECISION ARRAY STORING POLY	*	00112000
112100	C	*	MATRIX WHOSE GCD IS TO BE DETERMINED	*	00112100
112200	C	*	INDICES REP COLS, ROWS, AND COEFS RESPECTIVELY. GCD IS RETURNED IN THIS ARRAY	*	00112200
112300	C	*		*	00112300
112400	C	*	A1(1,2,3) = ARRAY WHICH RETURNS PRINCIPAL PRODUCT OF INVERSE OF INPUT	SS	00112400
112500	C	*	MATRIX	*	00112500
112600	C	*		*	00112600
112700	C	*	L,R = ROW AND COLUMN DIMENSIONS OF INPUT A MATRIX	*	00112700
112800	C	*	UL = UPPER BOUND POLYNOMIAL SIZE	*	00112800
112900	C	*	TOL = ROUND-OFF TOLERANCE (DOUBLE PREC)	*	00112900
113000	C	*	ILR = RETURNS A 1 IF INPUT MATRIX HAS RANK LESS THAN ITS MINIMUM DIMENSION	*	00113000
113100	C	*		*	00113100
113200	C	*		*	00113200
113300	C	*	OTHER ROUTINES CALLED:	*	00113300
113400	C	*	ROWOP = DETERMINES A PROPER PIVOT ROW FOR ROW OPERATIONS	*	00113400
113500	C	*		*	00113500
113600	C	*	ROWOP = PERFORMS ACTUAL ROW OPERATIONS ON 3-D A, AND A1 ARRAYS (POLY MATRICES)	*	00113600
113700	C	*		*	00113700
113800	C	*	RSWPT = EFFECTIVELY SWITCHES TWO ROWS OF A POLY MATRIX (ACTUALLY 3-D EQUIVALENT)	*	00113800
113900	C	*		*	00113900
114000	C	*	INZEL = EXPLAINED WITH SUBROUTINE	*	00114000
114100	C	*		*	00114100
114200	C	*	*****	*	00114200
114300	C		IMPLICIT REAL*8 (A,T,X)		00114300
114400	C		DOUBLE PRECISION PARAMS		00114400
114500	C		DIMENSION A(10,10,20), A1(10,10,20)		00114500
114600	C		INTEGER POSK, POSJ		00114600
114700	C	*	* CREATE IDENTITY MATRIX		00114700
114800	C		DO 2 I=1,L		00114800



(	114900	DL 2 J=1,L	00114900
(	115000	DL 2 K = 1, ID 1	00115000
(	115100	2 A1(I,J,K)=0.0	00115100
(	115200	DL 3 I=1,L	00115200
(	115300	3 A1(I,I,I)=1.0	00115300
(	115400	C	00115400
(	115500	C * MAIN LOOP FOR MATRIX MANIPULATION.	00115500
(	115600	C	00115600
(	115700	ML=M	00115700
(	115800	IF (L.LT.M) ML=L	00115800
(	115900	DL 100 I=1,ML	00115900
(	116000	C	00116000
(	116100	C * FIND NEW MAIN ROW SUCH THAT LOWEST DEGREE	00116100
(	116200	C * POLYNOMIAL IS IN DIAGONAL POSITION	00116200
(	116300	C	00116300
(	116400	7 CALL REDUC(A,I,L>ID1,MFOR,IPAX,TOL)	00116400
(	116500	IF (IPAX.EQ.1) GO TO 500	00116500
(	116600	IF (I.LQ.L) GO TO 56	00116600
(	116700	IF (I.LE.MROW) CALL RSIFT(A,I,L,M,II,1,I,MROW)	00116700
(	116800	II,DI=I+1	00116800
(	116900	PUSH=MZLL(A,I,I,ID1,TOL)	00116900
(	117000	C	00117000
(	117100	C * ELIMINATE ROW OPERATIONS ON ROWS BELOW MAIN ROW	00117100
(	117200	C	00117200
(	117300	DL 50 J=MROW,L	00117300
(	117400	FLSJ=MZLL(A,I,J,II,TOL)	00117400
(	117500	IF (FSJ.EQ.0) GO TO 50	00117500
(	117600	CALL REDOP(A,A1,I,J,FSJ,PSH,L,II,1,TOL)	00117600
(	117700	50 CONTINUE	00117700
(	117800	C	00117800
(	117900	C * SEE IF SUBMATRIX IS REDUCED BELOW MAIN ROW	00117900

118000	DL 55 J=MROWB*L	00118000
118100	K=NZLL(A,I,J,I01,TOL)	00118100
118200	IF (K.LE.0) GO TO 7	00118200
118300	55 CONTINUE	00118300
118400	C	00118400
118500	C * REDUCE ORDER OF POLYNOMIAL ABOVE MAIN ROW IF POSSIBLE	00118500
118600	C	00118600
118700	56 IF (I.LQ.1) GO TO 100	00118700
118800	POSJ=NZLL(A,I,I01,TOL)	00118800
118900	MIDNA=I-1	00118900
119000	LL 95 J=1,MROWA	00119000
119100	57 POSJ=NZLL(A,I,I01,TOL)	00119100
119200	IF (POSJ.LI.POSH) GO TO 95	00119200
119300	CALL ROWOP(A,A1,I,J,POSJ,POSH,L,M,I01,TOL)	00119300
119400	GL TO 57	00119400
119500	95 CONTINUE	00119500
119600	100 CONTINUE	00119600
119700	C	00119700
119800	C * NORMALIZE ROWS SO THAT LEADING COEFFICIENT OF DIAGONAL	00119800
119900	C * ELEMENTS EQUALS ONE	00119900
120000	C	00120000
120100	LL 25 J=1,ML	00120100
120200	K3=NZLL(A,J,J,I01,TOL)	00120200
120300	IF (K3.LQ.0) GO TO 25	00120300
120400	X=A(J,J,K3)	00120400
120500	LL 20 I=J,M	00120500
120600	I2=NZLL(A,I,J,I01,TOL)	00120600
120700	IF (I2.LQ.0) GO TO 20	00120700
120800	LL 15 I=1,K2	00120800
120900	IF (DAES(A(I,J,K)).GE.TOL) GO TO 13	00120900
121000	A(I,J,I)=0.0	00121000

	121100	GL TO 15	00121100
(	121200	13 A(I,J,K)=A(I,J,K)/X	00121200
	121300	15 CONTINUE	00121300
(	121400	20 CONTINUE	00121400
	121500	LL 24 I=1,L	00121500
(	121600	K2=1,ZEL(A1,I,J,ID1,TOL)	00121600
	121700	IF (K2.EQ.0) GO TO 24	00121700
(	121800	LL 23 K=1,K2	00121800
	121900	IF (ABS(A1(I,J,K)).GE.TOL) GO TO 22	00121900
(	122000	A1(I,J,K)=0.0	00122000
	122100	GL TO 23	00122100
(	122200	22 A1(I,J,K)=A1(I,J,K)/X	00122200
	122300	23 CONTINUE	00122300
(	122400	24 CONTINUE	00122400
	122500	25 CONTINUE	00122500
(	122600	n	00122600
	122700	ILR=0	00122700
(	122800	RETURN	00122800
	122900	50n ILR=1	00122900
(	123000	RETURN	00123000
	123100	98n FORMAT(4I3)	00123100
(	123200	END	00123200
	123300	SUBROUTINE MNRUN(A,I,L,ID1,MRDH,IMAX,TOL)	00123300
(	123400	IMPLICIT REAL*8 (A,T,X)	00123400
	123500	DOUBLE PRECISION PARAMS	00123500
(	123600	DIMENSION A(10,10,20)	00123600
	123700	IMAX=-1	00123700
(	123800	LL 10 J=1,L	00123800
	123900	D=0	00123900
(	124000	LL 5 K=1,101	00124000
	124100	K1=IL1+1-K	00124100

	124200	X=ABS(A(I,J,K1))	00124200
(	124300	IF (X.GT.TOL) GO TO 6	00124300
	124400	A(I,J,K1)=0.0	00124400
(	124500	K=K+1	00124500
	124600	5 CONTINUE	00124600
(	124700	GO TO 10	00124700
	124800	6 IF (N.LE.IMAX) GO TO 10	00124800
(	124900	IMAX=N	00124900
	125000	IFON=J	00125000
(	125100	10 CONTINUE	00125100
	125200	RETURN	00125200
(	125300	END	00125300
	125400	SUBROUTINE RSHFT(A,A1,I,M,IP1,IRDW1,IFON2)	00125400
(	125500	IMPLICIT REAL*8 (A,T,X)	00125500
	125600	DOUBLE PRECISION PARAMS	00125600
(	125700	DIMENSION A(10,10,20),A1(10,10,20),TMP1(20),TMP2(20)	00125700
	125800	DO 5 I=1,M	00125800
(	125900	DO 5 K=1,101	00125900
	126000	TMP1(K)=A(I,IRDW1,K)	00126000
(	126100	A(I,IRDW1,K)=A(I,IRDW2,K)	00126100
	126200	A(I,IRDW2,K)=TMP1(K)	00126200
(	126300	5 CONTINUE	00126300
	126400	DO 10 I=1,L	00126400
(	126500	DO 10 K=1,101	00126500
	126600	TMP2(K)=A1(I,IRDW1,K)	00126600
(	126700	A1(I,IRDW1,K)=A1(I,IRDW2,K)	00126700
	126800	A1(I,IRDW2,K)=TMP2(K)	00126800
(	126900	10 CONTINUE	00126900
	127000	RETURN	00127000
(	127100	END	00127100
(	127200	SUBROUTINE ROKOP(A,A1,I,J,PC5J,POS1,L,M,IP1,TOL)	00127200

127300	II. LICIT REAL*0 (A,T,X)	00127300
127400	DOUBLE PRECISION DARS	00127400
127500	DIMENSION A(10,10,20),A1(10,10,20),TMP1(20),TMP2(20)	00127500
127600	INTEGER POSJ,PUSH	00127600
127700	NSHFT=POSJ+PUSH	00127700
127800	X=A(I,J,POSJ)	00127800
127900	DL 55 I1=1,N	00127900
128000	IF (LSHFT.EQ.0) GO TO 18	00128000
128100	DL 15 K=1,NSHFT	00128100
128200	15 TMP1(K)=0.0	00128200
128300	18 K1=I1-NSHFT	00128300
128400	DL 20 K=1,K1	00128400
128500	20 TMP1(K+NSHFT)=A(I1,I,K)	00128500
128600	45 DL 50 K=1,I1	00128600
128700	IF (LABS(TMP1(N))*GE.TPL) GO TO 48	00128700
128800	TMP1(K)=0.0	00128800
128900	GL TL 50	00128900
129000	48 TMP1(K)=TMP1(K)*X/A(I,I,POSJ)	00129000
129100	50 A(I1,J,K)=A(I1,J,K)-TMP1(K)	00129100
129200	55 CONTINUE	00129200
129300	DL 75 I1=1,L	00129300
129400	IF (LSHFT.EQ.0) GO TO 60	00129400
129500	DL 50 K=1,NSHFT	00129500
129600	58 TMP2(K)=0.0	00129600
129700	60 K1=I1-NSHFT	00129700
129800	DL 65 K=1,K1	00129800
129900	65 TMP2(K+NSHFT)=A1(I1,I,K)	00129900
130000	DL 70 K=1,I1	00130000
130100	IF (LABS(TMP2(N))*GE.TPL) GO TO 68	00130100
130200	TMP2(K)=0.0	00130200
130300	GL TL 70	00130300

130400	6P	TIP2(K)=TMP2(K)*X/A(I,I,POSK)	00130400
130500	70	A1(I1,J,K)=A1(I1,J,K)-TIP2(K)	00130500
130600	75	CONTINUE	00130600
130700		RETURN	00130700
130800		END	00130800
130900		SUBROUTINE FACT (R,A,X,N>ID1,TOL)	00130900
131000	C	*	00131000
131100	C	*****	00131100
131200	C	*	00131200
131300	C	* THE SUBROUTINE ZFACT? DETERMINES THE POLYNOMIAL MATRIX ZXZ IN *	00131300
131400	C	* THE EQUATION: R = (X) (A)	00131400
131500	C	* WHERE ZRZ IS A POLYNOMIAL MATRIX OF FULL COLUMN RANK? AND ZAZ *	00131500
131600	C	* IS A GREATEST COMMON RIGHT DIVISOR OF THE ROWS OF ZRZ IN UPPER *	00131600
131700	C	* TRIANGULAR FORM. IT COMPUTES ZXZ BY TAKING ADVANTAGE OF THE *	00131700
131800	C	* UPPER TRIANGULAR FORM OF ZAZ. THE UPPER TRIANGULAR FORM ALLOWS *	00131800
131900	C	* YOU TO UNIQUELY DETERMINE THE COLUMNS OF ZXZ STARTING WITH COL. *	00131900
132000	C	* COL. 1 AND GOING FROM LEFT TO RIGHT. *	00132000
132100	C	*	00132100
132200	C	* VARIABLES PASSED: *	00132200
132300	C	* I(1,2,3) = N(1,2,3) *	00132300
132400	C	* X(1,2,3) = 3-D DOUBLE PRECISION ARRAYS CON- *	00132400
132500	C	TAINING POLY. MATRICES AS EX- *	00132500
132600	C	* PLAINED ABOVE. R AND A ARE PASSED *	00132600
132700	C	* AND REMAIN UNCHANGED. X IS RE- *	00132700
132800	C	* TURNED WITH THE RESULT *	00132800
132900	C	* N?N = NO. OF ROWS AND COLS. OF R *	00132900
133000	C	* ID1 = UPPER BOUND ON POLYNOMIAL SIZE *	00133000
133100	C	* TOL = ROUNDOFF TOLERANCE (DOUB PREC) *	00133100
133200	C	*	00133200
133300	C	* OTHER ROUTINES USED: *	00133300
133400	C	* SUBPR?DIV = REDUCE ALGEBRAIC MANIPULATIONS ON *	00133400

133500	C	THE POLYNOMIAL ELEMENTS AS RE*	*	00133500
133600	C	QUIRED BY THE ALGORITHM*	*	00133600
133700	C	*	*	00133700
133800	C	*****		00133800
133900	C	*		00133900
134000		IMPLICIT REAL*8 (A,P,T,X)		00134000
134100		DIMENSION A(10,10,20),P(10,10,20),X(10,10,20),TEMP(2,20)		00134100
134200		TITLE: ERR		00134200
134300	C			00134300
134400	C			00134400
134500		DO 5 I=1,N		00134500
134600		DO 5 J=1,M		00134600
134700		DO 5 I=1,101		00134700
134800		X(I,J,I)=0.0		00134800
134900	C			00134900
135000	C	* DETERMINE COLUMN 1 OF X(S)		00135000
135100	C			00135100
135200		I=1		00135200
135300		DO 20 J=1,M		00135300
135400		DO 15 K=1,101		00135400
135500		TEMP(1,K)=R(I,J,K)		00135500
135600	15	TEMP(2,K)=A(I,J,K)		00135600
135700		CALL DIV(A,X,TEMP,I,J,101,TOL,ERR)		00135700
135800		IF (LIE.E0.1) RETURN		00135800
135900	20	CONTINUE		00135900
136000	C			00136000
136100	C	* DETERMINE REMAINING COLUMNS OF X(S)		00136100
136200	C			00136200
136300		DO 25 I=2,N		00136300
136400		DO 25 J=1,M		00136400
136500		CALL SUBPR(A,X,TEMP,I,J,101,TOL)		00136500

136600	DO 22 K=1,101	00136600
136700	TEMP(1,K)=R(1,J,K)-TEMP(1,K)	00136700
136800	TEMP(2,K)=A(1,1,K)	00136800
136900	22 CONTINUE	00136900
137000	CALL DIV(A,X,TEMP,I,J,IDI,TOL,ERR)	00137000
137100	IF (LINE.EQ.1) RETURN	00137100
137200	29 CONTINUE	00137200
137300	RETURN	00137300
137400	END	00137400
137500	SUBROUTINE DETERM (P, A, H, IDI, TOL)	00137500
137600	IMPLICIT REAL*8 (A, P, T, X, D)	00137600
137700	COMMON /POLGDS/DRDCTR,DRDCTI	00137700
137800	DIMENSION A(10,10,20),R(10,10,20),IND(10),DET(20),X(20),	00137800
137900	* DLT(20),DRDCTR(20),DRDCTI(20)	00137900
138000	C	00138000
138100	C TRANSFORM MATRIX TO UPPER TRIANGULAR EQUIVALENT	00138100
138200	C	00138200
138300	CALL LIMPIA (DRDCTR,20,1,1)	00138300
138400	CALL LIMPIA (DRDCTI,20,1,1)	00138400
138500	CALL GCRD (R, A, H, H, IDI, TOL, IER)	00138500
138600	IF (IER.EQ.1) GO TO 60	00138600
138700	ICL = 0	00138700
138800	C	00138800
138900	C DETERMINE IF MATRIX IS UNIMODULAR	00138900
139000	C	00139000
139100	DO 10 I = 1,H	00139100
139200	IDR(I) = NZEL (R, 1, I, IDI, TOL)	00139200
139300	10 ICD = ICD + IDR(I)	00139300
139400	IF (ICD.GT.H) GO TO 20	00139400
139500	WRITE (6,900)	00139500
139600	RETURN	00139600



139700	C		00139700
139800	C	MULTIPLY DIAGONAL ELEMENTS OF UPPER TRIANGULAR	00139800
139900	C	EQUIVALENT TO FORM NORMALIZED DETERMINANT	00139900
140000	C		00140000
140100	20	I = 1	00140100
140200		IX = IDR(I)	00140200
140300		CALL VLOAD (R, X, I, I, KX, ID1, TOL)	00140300
140400		IF (M:EQ:1) GO TO 45	00140400
140500		PL 40 I = 2:M	00140500
140600		IX = IDR(I)	00140600
140700		CALL MMULT (R, X, I, I, KX, IF1, TOL)	00140700
140800	40	CONTINUE	00140800
140900	45	WRITE (6,905)	00140900
141000		CALL PRN13 (X, ID1, TOL)	00141000
141100		GO TO 1	00141100
141200	C	WRITE (6,910)	00141200
141300	C		00141300
141400	C	FIND AND PRINT ROOTS OF DIAGONAL ELEMENTS INDIVIDUALLY	00141400
141500	C		00141500
141600	C	PL 55 I=1:M	00141600
141700		IF (IDR(I):EQ:1) GO TO 55	00141700
141800		ILLG = IDR(I)	00141800
141900		PL 50 K = 1:IDEG	00141900
142000	50	DET(K) = R(1, I, K)	00142000
142100		IDEG = IDEG - 1	00142100
142200	1	CALL PLRT (X, DET1, M, DEGREE, DEGREE, IEP)	00142200
142300		GO TO (200,300,400,500),IEP	00142300
142400	C	PL 100 K = 1:IDEG	00142400
142500	C 100	WRITE (6,920) UROOTS(K),DROOTS(K)	00142500
142600		GO TO 55	00142600
142700	200	WRITE (6,950)	00142700

142800		GO TO 55	00142800
142900	300	WRITE (6,960)	00142900
143000		GO TO 55	00143000
143100	400	WRITE (6,970)	00143100
143200		GO TO 55	00143200
143300	500	WRITE (6,980)	00143300
143400		GO TO 55	00143400
143500	55	CONTINUE	00143500
143600		RETURN	00143600
143700	60	WRITE (6,907)	00143700
143800		RETURN	00143800
143900	900	FORMAT (1X//1X,"THIS MATRIX IS UNILQUPLAR")	00143900
144000	905	FORMAT (1X//1X,"THE NORMALIZED DETERMINANT POLYNOMIAL"	00144000
144100	910	* " OF THIS MATRIX IS: ")	00144100
144200	905	FORMAT (/1X," POLINOMIO CARACTERISTICO DEL OBSERVADOR :"/)	00144200
144300	907	FORMAT (1X//1X,"THIS MATRIX IS SINGULAR")	00144300
144400	910	FORMAT (1X//1X,"THE ROOTS OF THIS POLYNOMIAL MATRIX ARE: ")	00144400
144500	920	FORMAT (1X//15X,"S = "F13.6," + "F13.6," J ")	00144500
144600	950	FORMAT (1X//1X,"THE INPUT DEGREE IS LESS THAN ONE")	00144600
144700	960	FORMAT (1X//1X,"THE INPUT DEGREE IS GREATER THAN 36")	00144700
144800	970	FORMAT (1X//1X,"UNABLE TO DETERMINE ROOT WITHIN 500"	00144800
144900		* " ITERATIONS ON 5 STARTING VALUES")	00144900
145000	980	FORMAT (1X//1X,"THE HIGHEST DEG. COEFFICIENT INPUT IS ZERO")	00145000
145100		END	00145100
145200		SUBROUTINE DIV(A,X,TEMP,I,J,IP1,TOL,ERR)	00145200
145300		IMPLICIT REAL*8 (A,E,T,X)	00145300
145400		DOUBLE PRECISION DAPS	00145400
145500		DIMENSION A(10,10,20),X(10,10,20),TEMP(2,20),IPDS(2)	00145500
145600		INTEGER ERR	00145600
145700		DO 10 I=1,2	00145700
145800		DO 5 L=1,IP1	00145800

145900	IF DS(I1)=ID1+1-K	00145900
146000	IF (ABS(TEMP(1,IPDS(I1))).GT.TOL) GO TO 10	00146000
146100	5 CONTINUE	00146100
146200	IF (I1-1) 25,25,30	00146200
146300	10 CONTINUE	00146300
146400	NSHFT=IPDS(1)-IPDS(2)+1	00146400
146500	I12=IPDS(2)	00146500
146600	IF (NSHFT.LE.0) GO TO 30	00146600
146700	LL 20 I1=1+NSHFT	00146700
146800	I1=NSHFT+1-I1	00146800
146900	IF 1=IPDS(1)+1-I1	00146900
147000	X(I,J+1,1)=TEMP(1,IP1)/TEMP(2,IP2)	00147000
147100	LL 15 I2=1+IP2	00147100
147200	15 TEMP(1,I2+I1-1)=TEMP(1,I2+I1-1)*X(I,J,I1)+TEMP(2,I2)	00147200
147300	20 CONTINUE	00147300
147400	IF 1=IPDS(1)	00147400
147500	LL 22 I=1+IP1	00147500
147600	IF (ABS(TEMP(1,K)).LE.TOL) GO TO 22	00147600
147700	WRITE (6,901) I,J	00147700
147800	LFR=0	00147800
147900	RETURN	00147900
148000	22 CONTINUE	00148000
148100	25 LFR=0	00148100
148200	RETURN	00148200
148300	30 WRITE (6,903) I,J	00148300
148400	LFR=1	00148400
148500	RETURN	00148500
148600	C	00148600
148700	C	00148700
148800	901 FORMAT (1X/1X/27)WARNING REMAINDER ABOVE TOL(2I3)	00148800
148900	903 FORMAT (1X/1X/27)ZERO DIVIDE OR DEGREE ERROR(2I3)	00148900

149000		LID		00149000
149100		SUBROUTINE PREAD (A, N, M)		00149100
149200	C	*		00149200
149300	C	*****		00149300
149400	C	*		00149400
149500	C	* THE SUBROUTINE "PREAD" READS IN A POLYNOMIAL MATRIX INTO A	*	00149500
149600	C	* 3-D DOUBLE PREC. ARRAY. THE FIRST LINE OF INPUT SHOULD BE A	*	00149600
149700	C	* LINE OF (13) INTEGERS WHICH ARE THE ROW COLUMN MATRIX DIMEN-	*	00149700
149800	C	* SIONS, AND THE DEGREES OF THE HIGHEST DEGREE POLYNOMIAL IN	*	00149800
149900	C	* EACH COLUMN OF THE MATRIX TO BE INPUT. THE MATRIX POLYNOMIAL	*	00149900
150000	C	* COEFFICIENTS ARE THEN READ IN COLUMNWISE ASSUMING EACH POLY-	*	00150000
150100	C	* NOMIAL IS OF THE SAME DEGREE OF THE MAXIMUM DEGREE POLYNOMIAL	*	00150100
150200	C	* OF THAT COLUMN. CONSEQUENTLY ZERO COEFFICIENTS SHOULD BE	*	00150200
150300	C	* ADDED APPROPRIATELY. IN ADDITION COEFFICIENTS FOR EACH COLUMN	*	00150300
150400	C	* START ON A NEW LINE WITH FORMAT (10.2). FOR EACH POLYNOMIAL	*	00150400
150500	C	* COEFFICIENTS ARE ORDERED FROM THE SCALAR TERM TO THE HIGHER	*	00150500
150600	C	* ORDERED TERMS. NOTE: THE FORTRAN READ STATEMENTS USE DATA SET	*	00150600
150700	C	* REFERENCE NUMBER 1. THIS MEANS THAT FOR TERMINAL USE INPUT	*	00150700
150800	C	* MUST BE PLACED IN THE FILE "FILE ITC1F001". FOR BATCH USE THE	*	00150800
150900	C	* REFERENCE NUMBER MUST BE CHANGED FROM 1 TO 5.	*	00150900
151000	C	* A(1,2,3) = 3-D DOUBLE PREC ARRAY TO BE FILLED	*	00151000
151100	C	* VARIABLES PASSED:	*	00151100
151200	C	*	*	00151200
151300	C	* WITH POLY. MATRIX	*	00151300
151400	C	*	*	00151400
151500	C	* N, M = ROW COLUMN DIMENSIONS OF MATRIX	*	00151500
151600	C	*****		00151600
151700	C	*		00151700
151800		IMPLICIT REAL*8 (A)		00151800
151900		DIMENSION A(10,10,20), IDA(10)		00151900
152000	C			00152000

152100	C	REAL (5,800) N, M, (IDA(I), I = 1, N)	00152100
152200		REAL (5,7 ) N, M, (IDA(I), I = 1, N)	00152200
152300	C		00152300
152400		DO 5 I = 1, n	00152400
152500		IDA1 = IDA(I) + 1	00152500
152600	C 5	READ (5,810) ((A(I,J,K), K = 1, IDA1), J = 1, N)	00152600
152700	5	READ (5,7 ) ((A(I,J,K), K = 1, IDA1), J = 1, N)	00152700
152800		RETURN	00152800
152900	600	FORMAT (24I3,24I3)	00152900
153000	610	FORMAT (9F8,2)	00153000
153100		END	00153100
153200		SUBROUTINE SUMPR(A,X,TEMP,I,J>ID1,TOL)	00153200
153300		IMPLICIT REAL*8 (A-P,T,X)	00153300
153400		LINEAR A(10,10,20), Y(10,10,20), TEMP(2,20), TEMP1(10,20)	00153400
153500		I1=I-1	00153500
153600		DO 20 K=1, I1	00153600
153700		IX=IZEL(X,K,J>ID1,TOL)	00153700
153800		IA=IZEL(A,I,K,J>ID1,TOL)	00153800
153900		DO 5 K1=1, ID1	00153900
154000	5	TEMP1(K,K1)=0.0	00154000
154100		IF (IX.EQ.0) GO TO 20	00154100
154200		IF (IA.EQ.0) GO TO 20	00154200
154300		DO 10 I1=1, IA	00154300
154400		DO 10 K1=1, IX	00154400
154500		TEMP1(K,K1+I1-1)=TEMP1(K,K1+I1-1)+X(K,J,K1)+A(I,K,I1)	00154500
154600	10	CONTINUE	00154600
154700	20	CONTINUE	00154700
154800		DO 30 K1=1, ID1	00154800
154900		TEMP1(1,K1)=0.0	00154900
155000		DO 25 K=1, I1	00155000
155100	25	TEMP(1,K1)=TEMP(1,K1)+TEMP1(K,K1)	00155100

155200	30	CONTINUE		00155200
155300		RETURN		00155300
155400		END		00155400
155500		SUBROUTINE INVERT(P>PI>T1>T2>X>H>K>IP1>TOL>IEP)		00155500
155600	C	*****		00155600
155700	C	*	*	00155700
155800	C	* THE SUBROUTINE ZINVERTZ TAKES THE INVERSE OF ANY NONSINGULAR	*	00155800
155900	C	* SQUARE POLYNOMIAL MATRIX P. THE RESULTS IS RETURNED IN THE	*	00155900
156000	C	* FORM OF A POLYNOMIAL MATRIX PI AND A DIVISOR POLYNOMIAL X	*	00156000
156100	C	* SUCH THAT:	*	00156100
156200	C	* $P^{-1}$	*	00156200
156300	C	* $P^{-1} = PI/X$		00156300
156400	C	* NORMALLY X WOULD BE THE DETERMINANT OF P AND PI WOULD BE THE	*	00156400
156500	C	* ADJUGATE OF P, BUT IN THIS CASE THE POLYNOMIAL X IS NORMALIZED	*	00156500
156600	C	SO THAT ITS LARGEST COEFFICIENT IS 1. THEREFORE X DIFFERS	*	00156600
156700	C	* FROM DET P AND PI FROM ADJ P BY SCALAR FACTORS	*	00156700
156800	C	*	*	00156800
156900	C	* VARIABLE PASSED:	*	00156900
157000	C	* P(1,2,3) - 3-D ARRAY (DOUBLE PREC) CONTAINING	*	00157000
157100	C	MATRIX TO BE INVERTED. PROGRAM RETURNS	*	00157100
157200	C	AN UPPER TRIANGULAR COEFFICIENT MATRIX	*	00157200
157300	C	* EQUIVALENT TO THE MATRIX PASSED.	*	00157300
157400	C	* PI(1,2,3) - CONTAINS RESULTANT POLY MATRIX AS	*	00157400
157500	C	* EXPLAINED ABOVE (3-D DOUBLE PREC)	*	00157500
157600	C	* T1(1,2,3) - 3-D (DOUBLE PREC) ARRAYS USED FOR TEMP	*	00157600
157700	C	* T2(1,2,3) - ARRAY MANIPULATION WITHIN ROUTINE	*	00157700
157800	C	* X(1) - VECTOR CONTAINING RESULTANT DIVISOR	*	00157800
157900	C	* POLYNOMIAL AS EXPLAINED ABOVE (DOUBLE	*	00157900
158000	C	* PRECISION)		00158000
158100	C	* H - SQUARE MATRIX DIMENSION OF P	*	00158100
158200	C	* EX - (DEGREE+1) OF RESULTANT POLY X	*	00158200

158300	C	ID1	=	UPPER BOUND OF POLYNOMIAL SIZE	*	00158300
158400	C	* TOL	=	ROUND-OFF TOLERANCE (DOUB PREC)	*	00158400
158500	C	* ILR	=	RETURNS 1 IF INPUT MATRIX IS SINGULAR	*	00158500
158600	C	*			*	00158600
158700	C	* OTHER ROUTINES CALLED:			*	00158700
158800	C	* P, RMULT, FACT, HVRMUT, VLOAD, NZEL			*	00158800
158900	C	* (REFER TO ROUTINE FOR DESCRIPTION)			*	00158900
159000	C	*			*	00159000
159100	C	*****				00159100
159200	C	*				00159200
159300		IMPLICIT REAL*8 (P,T,X)				00159300
159400		1) DIMENSION P(10,10,20), F1(10,10,20), T1(10,10,20), T2(10,10,20),				00159400
159500		1 X(20), IDIM(10)				00159500
159600	C					00159600
159700	C					00159700
159800	C	* CASE WHERE MATRIX IS 1X1				00159800
159900	C					00159900
160000		IF (I=IDIM) GO TO 10				00160000
160100		LL 5 I=1, IDIM				00160100
160200		F1(I,I,K)=0.0				00160200
160300		= X(I)=P(I,I,K)				00160300
160400		F1(I,I,I)=1.0				00160400
160500		IX=NZEL(P,M,M, IDIM, TOL)				00160500
160600		IF IX=0				00160600
160700		RETURN				00160700
160800	C					00160800
160900	C	* FILE G(S) AN UPPER TRIANGULAR EQUIVALENT TO P(S)				00160900
161000	C					00161000
161100		10 CALL GCRD(P, F1, R, P, IDIM, TOL, IEP)				00161100
161200		IF (IEP=EQ.1) RETURN				00161200
161300		I=0				00161300

161400	DL 15 I=1,M	00161400
161500	ILIN(I)=NZEL(P,I,I>ID1,TOL)	00161500
161600	15 K=K+ILIN(I)	00161600
161700	IF (I.ECT.M) GO TO 25	00161700
161800	C	00161800
161900	C * CASE WHERE INPUT MATRIX IS UNIMODULAR (G(S)=I)	00161900
162000	C	00162000
162100	DL 20 K=2,ID1	00162100
162200	20 X(K)=0.0	00162200
162300	X(1)=1.0	00162300
162400	KX=1	00162400
162500	KC=1	00162500
162600	CALL SPLIT (I1,P1,K0,I1,P0,I1,M,K>ID1)	00162600
162700	RETURN	00162700
162800	C	00162800
162900	C * CASE WHERE INPUT MATRIX HAS NO SPECIAL FORM	00162900
163000	C	00163000
163100	25 I1=ILIN(1)	00163100
163200	I=1	00163200
163300	CALL VLOAD(P,X,I,I,K1,ID1,TOL)	00163300
163400	DL 30 I=2,M	00163400
163500	I2=ILIN(I)	00163500
163600	CALL MVMULT(P,X,I,I,K2,K1,ID1,TOL)	00163600
163700	30 CONTINUE	00163700
163800	KX=K1	00163800
163900	C	00163900
164000	C * CREATE DIAGONAL MATRIX OF G(S) DETERMINANTS	00164000
164100	C	00164100
164200	DL 35 I=1,M	00164200
164300	DL 35 J=1,M	00164300
164400	DL 35 K=1,ID1	00164400



```

164500      35 P1(I,J,K)=0.0                                00164500
164600      LL 40 I=1,K                                     00164600
164700      LL 40 K=1,KX                                    00164700
164800      40 P1(I,I,K)=X(K)                              00164800
164900      C                                             00164900
165000      C      * DETERMINE ADJOINT(G(S)) AND THEN I(S) 00165000
165100      C                                             00165100
165200      CALL FACT (I1,I2,H1,H2,ID1,TOL)                00165200
165300      CALL PMMULT(I2,T1,P1,H1,H2,K1,K2,ID1,TOL)      00165300
165400      RETURN                                          00165400
165500      END                                            00165500
165600      FUNCTION NZEL(I,J,ID1,TOL)                     00165600
165700      C      *                                       00165700
165800      C      * *****                                       00165800
165900      C      *                                       00165900
166000      C      * THE FUNCTION NZEL RETURNS AN INTEGER ONE GREATER THAN THE * 00166000
166100      C      * DEGREE OF THE POLYNOMIAL MATRIX ELEMENT PASSED, IF THIS EL- * 00166100
166200      C      * EMENT HAS THE VALUE ZERO, THEN ZERO IS RETURNED ELSE ONE OF * 00166200
166300      C      * GREATR IS RETURNED. * 00166300
166400      C      *                                       * 00166400
166500      C      *   VARIABLES PASSED: * 00166500
166600      C      *   A(1,2,3)   -   3-D DOUBLE PRECISION ARRAY WHICH STORES * 00166600
166700      C      *   POLYNOMIAL MATRIX WHOSE ELEMENT IS OF * 00166700
166800      C      *   INTEREST * 00166800
166900      C      *   I,J       -   COLUMN AND ROW INDICES OF DESIRED ELEMENT * 00166900
167000      C      *   IL1      -   UPPER BOUND OF POLYNOMIAL SIZE * 00167000
167100      C      *   TOL      -   ROUNDOFF TOLERANCE (DOUBLE PREC) * 00167100
167200      C      * * 00167200
167300      C      * *****                                       00167300
167400      C      *                                       00167400
167500      C      *                                       00167500

```

167000	C			00167000
167100		IMPLICIT REAL*8 (A-D,X)		00167100
167200		DOUBLE PRECISION DABS		00167200
167300		DIMENSION A(10,10,20)		00167300
167400		DO 5 K=1,101		00167400
167500		K1=IL1+1-K		00167500
167600		X=DABS(A(I,J,K1))		00167600
167700		IF (X.GT.TOL) GO TO 7		00167700
167800		5 CONTINUE		00167800
167900		6 K1=0		00167900
168000		7 NZEL=K1		00168000
168100		RETURN		00168100
168200		END		00168200
168300		SUBROUTINE PMMULT(A1,A2,A3,L1,M1,L2,M2,L3,M3,LD1,TOL)		00168300
168400	C			00168400
168500	C	*****		00168500
168600	C	*		00168600
168700	C	* THE SUBROUTINE ZPMULT PERFORMS MATRIX MULTIPLICATION ON POLY-		00168700
168800	C	* NOMIAL MATRICES. IN TERMS OF THE ARRAYS PASSED IT COMPUTES:		00168800
168900	C	* A3 = (A1) (A2)		00168900
169000	C	*		00169000
169100	C	* VARIABLES PASSED:		00169100
169200	C	* A1(1,2,3),A2(1,2,3)		00169200
169300	C	* A3(1,2,3) - 3-D DOUBLE PREC. ARRAYS STORING		00169300
169400	C	* POLY. MATRICES. A3 CONTAINS		00169400
169500	C	* RESULT. A1,A2 REMAIN UNCHANGED		00169500
169600	C	* L1,L2,L3 - NO. OF ROWS IN A1,A2,A3 RESPECT.		00169600
169700	C	* L3 IS RETURNED, L1 AND L2 RE-		00169700
169800	C	* MAIN UNCHANGED		00169800
169900	C	* M1,M2,M3 - NO. OF COLS. IN A1,A2,A3 RESP.		00169900
170000	C	* L3 IS RETURNED, M1 AND M2 RE-		00170000
170100	C			00170100
170200	C			00170200
170300	C			00170300
170400	C			00170400
170500	C			00170500
170600	C			00170600

170700	C		PAIN UNCHANGED	*	00170700		
170800	C	*	ID1	-	UPPER BOUND ON POLYNOMIAL SIZE	*	00170800
170900	C	*	TOL	-	BOUNDING TOLERANCE (DOUBLE PREC)	*	00170900
171000	C					*	00171000
171100	C		*****				00171100
171200	C		*				00171200
171300			IMPLICIT REAL*8 (A-T)				00171300
171400			DIMENSION A1(10,10,20),A2(10,10,20),A3(10,10,20),TEMP(10,20)				00171400
171500			IF (N1.NE.L2) GO TO 100				00171500
171600			L3=L1				00171600
171700			K3=K2				00171700
171800			DO 50 I=1,K3				00171800
171900			LL 50 J=1,L3				00171900
172000			LL 20 L=1,M1				00172000
172100			LA1=LZLL(A1,N,J,I,1,TOL)				00172100
172200			LA2=LZLL(A2,N,I,1,TOL)				00172200
172300			LL 5 L1=1,TOL				00172300
172400			5 TEMP(I,K1)=0.0				00172400
172500			IF (LA1.EQ.0) GO TO 20				00172500
172600			IF (LA2.EQ.0) GO TO 20				00172600
172700			LL 10 K2=1,KA2				00172700
172800			LL 10 K1=1,KA1				00172800
172900			10 TEMP(I,K1+K2-1)=TEMP(I,K1+K2-1)+A1(I,J,K1)+A2(I,K,K2)				00172900
173000			20 CONTINUE				00173000
173100			LL 30 K1=1,101				00173100
173200			A3(I,J,K1)=0.0				00173200
173300			LL 25 L=1,M1				00173300
173400			25 A3(I,J,K1)=A3(I,J,K1)+TEMP(L,I)				00173400
173500			30 CONTINUE				00173500
173600			50 CONTINUE				00173600
173700			LLTUEE				00173700

```

173800      100 WRITE(6,900)                                00173800
173900          LLTLEN                                       00173900
174000      500 FLERRAT(1X//1X//ILLEGAL MATRIX MULTIPLICATION?) 00174000
174100          ELD                                          00174100
174200          SUBROUTINE SPLIT (A1,A2,LO,L1,N0,M1,L2,M2,LD1) 00174200
174300      C      *                                          00174300
174400      C      * *****                                          00174400
174500      C      *                                          00174500
174600      C      * THE SUBROUTINE ZSPLITZ TAKES A SECTION OF ROWS AND COLUMNS OUT * 00174600
174700      C      * OF A 3-D REP OF A POLYNOMIAL MATRIX AND PLACES IT INTO AN * 00174700
174800      C      * OTHER ARRAY *                               * 00174800
174900      C      *                                          * 00174900
175000      C      *   VARIABLES PASSED:                          * 00175000
175100      C      *   A1(1,2,3),A2(1,2,3) = 3-D DOUBLE PREC REP OF POLY MAT * 00175100
175200      C      *   RICES. A1 CONTAINS CONTAINS OR *         * 00175200
175300      C      *   ORIGINAL MATRIX AND REMAINS UN *         * 00175300
175400      C      *   CHANGED. SUBSECTION OF A1 IS *           * 00175400
175500      C      *   PLACED INTO A2 *                           * 00175500
175600      C      *   LO,L1      - BEGINNING AND FINAL ROWS TO BE * 00175600
175700      C      *   TAKEN OUT OF A1. 1 = LO = L1. *           * 00175700
175800      C      *   N0,M1      - BEGINNING AND FINAL COLS. TO BE * 00175800
175900      C      *   TAKEN OUT OF A1. 1 = N0 = M1. *           * 00175900
176000      C      *   L2,M2      - RESULTING ROW AND COLUMN DIM * 00176000
176100      C      *  ENSIONS OF A2 *                             * 00176100
176200      C      *   IL1      - UPPER BOUND OF POLYNOMIAL SIZE * 00176200
176300      C      *                                          * 00176300
176400      C      * *****                                          00176400
176500      C      *                                          00176500
176600      C      *                                          00176600
176700          IMPLICIT REAL*8 (A)                          00176700
176800          DIMENSION A1(10,10,20),A2(10,10,20)          00176800

```

```

176900      DL 10 I=M0,M1                                00176900
177000      DL 10 J=L0,L1                                00177000
177100      DL 10 K=1,TOL1                              00177100
177200      10 A2(I+1=L0,J+1=L0,K)=A1(I,J,K)           00177200
177300      L2=L1-L0+1                                   00177300
177400      K2=K1-K0+1                                   00177400
177500      ELTOL=                                       00177500
177600      END                                          00177600
177700      SUBROUTINE ZVLOAD(A,X,I,J,KX,IP1,TOL)        00177700
177800      *                                             00177800
177900      * *****                                       00177900
178000      * *                                             00178000
178100      * * THE SUBROUTINE ZVLOAD COPIES A POLYNOMIAL MATRIX ELEMENT FROM * 00178100
178200      * * A DOUBLE PRECISION ARRAY INTO A DOUBLE PRECISION VECTOR * 00178200
178300      * * *                                             00178300
178400      * * VARIABLES PASSED: * 00178400
178500      * * A(1,2,3) - 3-D DOUBLE PREC ARRAY WHOSE ELEMENT * 00178500
178600      * * IS DESIRED. * 00178600
178700      * * X(1) - DOUBLE PREC VECTOR WHICH RECEIVES * 00178700
178800      * * POLYNOMIAL. * 00178800
178900      * * I,J - COLUMN INDICES OF DESIRED MATRIX * 00178900
179000      * * ELEMENT * 00179000
179100      * * IX - (DEGREE+1) OF POLYNOMIAL RETURNED * 00179100
179200      * * IN X * 00179200
179300      * * IP1 - UPPER BOUND OF POLYNOMIAL SIZE * 00179300
179400      * * TOL - ROUND OFF TOLERANCE (DOUBLE PREC) * 00179400
179500      * * *                                             00179500
179600      * * *****                                       00179600
179700      * * *                                             00179700
179800      * * IMPLICIT REAL*8(A,X,T) 00179800
179900      * * DIMENSION A(10,10,20),Y(20) 00179900

```

```

160000      C      00180000
160100      DL 5 K=1,KX      00180100
160200      5 X(K)=A(I,J,K)      00180200
160300      IF (IX.EQ.101) RETURN      00180300
160400      K1=KX+1      00180400
160500      DL 10 K=K1,101      00180500
160600      10 X(K)=0.0      00180600
160700      FLT01:      00180700
160800      END      00180800
160900      SUBROUTINE MVMULT(A,X,I,J,KA,IX,IL1,TCL)      00180900
161000      C      *      00181000
161100      C      *****      00181100
161200      C      *      *      00181200
161300      C      * THE SUBROUTINE MVMULT MULTIPLIES A POLYNOMIAL MATRIX ELEMENT *      00181300
161400      C      * BY ANOTHER POLYNOMIAL IN A VECTOR X AND RETURNS THE PRODUCT *      00181400
161500      C      * POLYNOMIAL IN THE VECTOR X. *      00181500
161600      C      *      *      00181600
161700      C      * VARIABLES PASSED: *      00181700
161800      C      * A(1,2,3) - 3-D DOUB PREC ARRAY CONTAINING POLY *      00181800
161900      C      * MATRIX WHOSE ELEMENT IS BEING MULTIPLIED *      00181900
162000      C      * X(1) - VECTOR INITIALLY CONTAINING POLYNOMIAL *      00182000
162100      C      * TO BE MULTIPLIED AND RETURNING RESULTANT *      00182100
162200      C      * POLYNOMIAL *      00182200
162300      C      * KA - (DEGREE+1) OF MATRIX ELEMENT TO BE MULT *      00182300
162400      C      * IX - (DEGREE+1) OF OTHER POLY TO BE MULT. *      00182400
162500      C      * AND RETURNING SAME INFO ABOUT RESULTANT *      00182500
162600      C      * PRODUCT POLY. *      00182600
162700      C      * I,J - COL,ROW INDICIES OF MATRIX ELEMENT *      00182700
162800      C      * IL1 - UPPER BOUND OF POLYNOMIAL SIZE *      00182800
162900      C      * TCL - ROUNDOFF TOL (LEUP PREC) *      00182900
163000      C      *      *      00183000

```

183100	C	*****	00183100
183200	C	*	00183200
183300		IMPLICIT REAL*8 (A,B,T,X)	00183300
183400		DECLARATION A(10,10,20),Y(20),R(20)	00183400
183500	C		00183500
183600		IF (I,X.EQ.0) RETURN	00183600
183700		IF (I,A.EQ.0) KA=1	00183700
183800		DO 5 K1=1,101	00183800
183900		5 R(K1)=0.0	00183900
184000		DO 10 K1=1,KA	00184000
184100		DO 10 K2=1,KX	00184100
184200		10 R(K1+K2-1)=R(K1+K2-1)+A(I,J,K1)*X(K2)	00184200
184300		DO 15 K1=1,101	00184300
184400		15 X(K1)=R(K1)	00184400
184500		I,X=K1+1,X=1	00184500
184600		RETURN	00184600
184700		END	00184700
184800		SUBROUTINE PRN13(A,IP1,TOL)	00184800
184900	C	*	00184900
185000	C	*****	00185000
185100	C	*	00185100
185200	C	* ZPL132 IS A MODIFIED VERSION OF THE ZPRINT2 SUBROUTINE. IT	00185200
185300	C	* TAKES IN A SINGLE POLYNOMIAL AND PRINTS IT OUT. THIS ROUTINE	00185300
185400	C	* IS DOUBLE PRECISION, AND CAN PRINT UP TO A 19TH DEGREE POLY.	00185400
185500	C	*	00185500
185600	C	* VARIABLES PASSED:	00185600
185700	C	A(I)    -  A VECTOR OF POLYNOMIAL COEFFICIENTS	00185700
185800	C	ORDERED FROM THE SCALE TO THE HIGHEST	00185800
185900	C	DEGREE TERMS	00185900
186000	C	*          IP1    -  UPPER BOUND ON POLYNOMIAL SIZE	00186000
186100	C	*          TOL    -  ROUND-OFF TOLERANCE (DOUBLE PREC)	00186100

186200	C	*	*	00186200
186300	C	*****	*****	00186300
186400	C	*		00186400
186500	C			00186500
186600		IMPLICIT REAL*8 (A-T,X)		00186600
186700		DIMENSION A(20),TEMP(20),ITEMP(20)		00186700
186800		DO 2 I=1,101		00186800
186900		K1=I+1-K		00186900
187000		X=DABS(A(K1))		00187000
187100		IF (X*LT*TOL) GO TO 4		00187100
187200		2 CONTINUE		00187200
187300	C			00187300
187400	C	* BRANCH FOR PRINTING ACCORDING TO POLYNOMIAL		00187400
187500	C	* (1,DEGREE+1)		00187500
187600		4 DO 5 I=1,K1		00187600
187700		K2=I+1-K		00187700
187800		ITEMP(I)=K2-1		00187800
187900		TEMP(I)=A(K2)		00187900
188000		5 CONTINUE		00188000
188100	C			00188100
188200	C			00188200
188300	C	BRANCH FOR PRINTING ACCORDING TO POLYNOMIAL (DEGREE+1)		00188300
188400	C			00188400
188500		IF (I1.LE.1) GO TO 7		00188500
188600		IF (I1.LE.6) GO TO 9		00188600
188700		IF (I1.LE.13) GO TO 11		00188700
188800		WRITE (6,910) (ITEMP(K),K=1,6)		00188800
188900		WRITE (6,911) (ITEMP(K),K=7,6)		00188900
189000		WRITE (6,912) (ITEMP(K),K=7,13)		00189000
189100		WRITE (6,913) (ITEMP(K),K=7,13)		00189100
189200		WRITE (6,912) (ITEMP(K),K=14,1)		00189200



```

189300      WRITE (6,915) (TEMP(K),K=14,K1)
189400      RETURN
189500      7 WRITE (6,915) TEMP(1)
189600      RETURN
189700      9 WRITE (6,916) (TEMP(K),K=1,K1)
189800      WRITE (6,917)(TEMP(K),K=1,K1)
189900      RETURN
190000      11 WRITE (6,916) (TEMP(K),K=1,6)
190100      WRITE (6,917)(TEMP(K),K=1,6)
190200      WRITE (6,912)(TEMP(K),K=7,K1)
190300      WRITE (6,913)(TEMP(K),K=7,1:1)
190400      RETURN
190500      C *
190600      912 FORMAT(1X,7(15A,I2))
190700      913 FORMAT(1X,1X,7(F13.7,AH S +))
190800      915 FORMAT (1X//1X,15X,F13.7)
190900      916 FORMAT (1X/1X,14X,6(15Y,I2))
191000      917 FORMAT (1X,15X,6(F13.7,AH S +))
191100      C
191200      END
191300      SUBROUTINE PRINT (A, L, H, ID1, TOL)
191400      C *
191500      C +*****
191600      C *
191700      C * THE SUBROUTINE "PRINT" PRINTS OUT THE ELEMENTS OF A POLYNOMIAL *
191800      C * MATRIX IN A 3-D DOUBLE PRECISION ARRAY REPRESENTATION. *
191900      C * EACH POLYNOMIAL CAN BE UP TO 19TH DEGREE. *
192000      C *
192100      C *          VARIABLES PASSED: *
192200      C *          A(1,2,3) - 3-D DOUBLE PREC. ARRAY CONTAING POLY. *
192300      C *          MATRIX TO BE PRINTED.REMAINS UNCHANGED*

```

192400	C	*	LDH	= ROW AND COLUMN DIMENSIONS OF A	*	00192400	
192500	C	*	LD1	= UPPER BOUND ON POLYNOMIAL SIZE	*	00192500	
192600	C	*	TOL	= ROUND OFF TOLERANCE (DOUBLE PREC)	*	00192600	
192700	C	*			*	00192700	
192800	C	*****					00192800
192900	C	*				00192900	
193000			IMPLICIT REAL*8 (A-Z)			00193000	
193100			DOUBLE PRECISION DAPS			00193100	
193200			DIMENSION A(10,10,20), TEMP(20), ITEMP(20)			00193200	
193300			LU 20 I = 1,M			00193300	
193400			LU 20 J = 1,L			00193400	
193500	C					00193500	
193600	C	*	LET LKLINE (DEGREE + 1) OF A MATRIX ELEMENT			00193600	
193700	C					00193700	
193800			DO 2 K = 1,LD1			00193800	
193900			K1 = ID1 + 1 - K			00193900	
194000			X = DAPS(A(I,J,K1))			00194000	
194100			IF (X.GT.TOL) GO TO 4			00194100	
194200		2	CONTINUE			00194200	
194300	C					00194300	
194400	C	*	LEADER COEFFICIENTS FOR PRINTING			00194400	
194500	C					00194500	
194600		4	DO 5 K = 1,K1			00194600	
194700			K2 = K1 + 1 - K			00194700	
194800			ITEMP(K) = K2 - 1			00194800	
194900			TEMP(K) = A(I,J,K2)			00194900	
195000		5	CONTINUE			00195000	
195100	C					00195100	
195200	C	*	LEADER FOR PRINTING ACCORDING TO POLYNOMIAL			00195200	
195300	C	*	(DEGREE + 1)			00195300	
195400	C					00195400	

195500		IF (K1.LE.9.1) GO TO 7	00195500
195600		IF (K1.LE.6) GO TO 9	00195600
195700		IF (K1.LE.13) GO TO 11	00195700
195800		WRITE (6,916) (ITEMP(K),K = 1,6)	00195800
195900		WRITE (6,917) J, I, (TEMP(I),K = 1,6)	00195900
196000		WRITE (6,912) (ITEMP(K),K = 7,13)	00196000
196100		WRITE (6,913) (TEMP(K),K = 7,13)	00196100
196200		WRITE (6,912) (ITEMP(K),K = 14,K1)	00196200
196300		WRITE (6,913) (TEMP(K),K = 14,K1)	00196300
196400		GO TO 20	00196400
196500	7	WRITE (6,915) J, I, TEMP(1)	00196500
196600		GO TO 20	00196600
196700	0	WRITE (6,916) (ITEMP(K),K = 1,K1)	00196700
196800		WRITE (6,917) J, I, (TEMP(I),K = 1,K1)	00196800
196900		GO TO 20	00196900
197000	11	WRITE (6,916) (ITEMP(K),K = 1,6)	00197000
197100		WRITE (6,917) J, I, (TEMP(I),K = 1,6)	00197100
197200		WRITE (6,912) (ITEMP(K),K = 7,K1)	00197200
197300		WRITE (6,913) (TEMP(K),K = 7,K1)	00197300
197400	20	CONTINUE	00197400
197500		RETURN	00197500
197600	c	I:	00197600
197700		912 FORMAT (1X,7(15X,I2))	00197700
197800		913 FORMAT (1X,1X,7(F13.7,4H S +))	00197800
197900		915 FORMAT (1X,7(1X,7H ELEM (I2,1I,3I)) : 7(F13.7))	00197900
198000		916 FORMAT (1X,1X,14X,6(15X,I2))	00198000
198100		917 FORMAT (1X,7H ELEM (I2,1I,3I) : 6(F13.7,4H S +))	00198100
198200	c		00198200
198300		END	00198300
198400		SUBROUTINE POLRT(XCOF,COF,IPROTE,ECOTI,IER)	00198400
198500	c	SUBROUTINE POLRT	00198500

	198600	c	PURPOSE	00198600
(	198700	c	COMPUTES THE REAL AND COMPLEX ROOTS OF A REAL POLYNOMIAL	00198700
(	198800	c	USAGE	00198800
(	198900	c	CALL QLRT(XCOF,COF,H,ROOTR,ROOTI,IER)	00198900
(	199000	c	DESCRIPTION OF PARAMETERS	00199000
(	199100	c	XCOF -VECTOR OF P+1 COEFFICIENTS OF THE POLYNOMIAL	00199100
(	199200	c	ORDERED FROM SMALLEST TO LARGEST POWER	00199200
(	199300	c	COF -WORKING VECTOR OF LENGTH P+1	00199300
(	199400	c	H -ORDER OF POLYNOMIAL	00199400
(	199500	c	ROOTR-RESULTANT VECTOR OF LENGTH H CONTAINING REAL ROOTS	00199500
(	199600	c	OF THE POLYNOMIAL	00199600
(	199700	c	ROOTI-RESULTANT VECTOR OF LENGTH H CONTAINING THE	00199700
(	199800	c	CORRESPONDING IMAGINARY ROOTS OF THE POLYNOMIAL.	00199800
(	199900	c	IER -ERROR CODE WHERE	00199900
(	200000	c	IER=0 NO ERROR	00200000
(	200100	c	IER=1 P LESS THAN ONE	00200100
(	200200	c	IER=2 H GREATER THAN 36	00200200
(	200300	c	IER=3 UNABLE TO DETERMINE ROOT WITH 500 ITERATIONS	00200300
(	200400	c	ON 5 STARTING VALUES	00200400
(	200500	c	IER=4 HIGH ORDER COEFFICIENT IS ZERO	00200500
(	200600	c	REMARKS	00200600
(	200700	c	LIMITED TO 36TH ORDER POLYNOMIAL OR LESS.	00200700
(	200800	c	FLOATING POINT OVERFLOW MAY OCCUR FOR HIGH ORDER	00200800
(	200900	c	POLYNOMIALS BUT WILL NOT AFFECT THE ACCURACY OF THE RESULTS.	00200900
(	201000	c	SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED	00201000
(	201100	c	NONE	00201100
(	201200	c	METHOD	00201200
(	201300	c	NEWTON-RAPHSON ITERATIVE TECHNIQUE. THE FINAL ITERATIONS	00201300
(	201400	c	ON EACH ROOT ARE PERFORMED USING THE ORIGINAL POLYNOMIAL	00201400
(	201500	c	RATHER THAN THE REDUCED POLYNOMIAL TO AVOID ACCUMULATED	00201500
(	201600	c	ERRORS IN THE REDUCED POLYNOMIAL.	00201600

201700		DIMENSION XCOF(1),COF(1),RODTP(1),RODTI(1)	00201700
201800		DOUBLE PRECISION XD,YD,X,Y,XPP,YPR,UX,UY,V,YT,XT,U,XTZ,YTZ,SUMSQ,	00201800
201900		IX,DX,TEMP,ALPHA	00201900
202000	C	IF A DOUBLE PRECISION VERSION OF THIS ROUTINE IS DESIRED, THE	00202000
202100	C	C IN COLUMN 1 SHOULD BE REMOVED FROM THE DOUBLE PRECISION	00202100
202200	C	STATEMENT WHICH FOLLOWS.	00202200
202300		DOUBLE PRECISION XCOF,COF,RODTR,RODTI	00202300
202400	C	THE C MUST ALSO BE REMOVED FROM DOUBLE PRECISION STATEMENTS	00202400
202500	C	APPEARING IN OTHER ROUTINES USED IN CONJUNCTION WITH THIS	00202500
202600	C	ROUTINE.	00202600
202700	C	THE DOUBLE PRECISION VERSION MAY BE MODIFIED BY CHANGING THE	00202700
202800	C	CONSTANT IN STATEMENT 78 TO 1.00 <sup>-12</sup> AND IN STATEMENT 122 TO	00202800
202900	C	1.00 <sup>-10</sup> . THIS WILL PROVIDE HIGHER PRECISION RESULTS AT THE	00202900
203000	C	COST OF EXECUTION TIME	00203000
203100	C	SET ERROR CODE TO 1	00203100
203200	C	SET ERROR CODE TO 4	00203200
203300	C	SET ERROR CODE TO 2	00203300
203400	C	SET INITIAL VALUES	00203400
203500	C	ZLEU INITIAL VALUE COUNTER	00203500
203600	C	SET X AND Y TO CURRENT VALUE	00203600
203700	C	EVALUATE POLYNOMIAL AND DERIVATIVES	00203700
203800	C	STEP ITERATION COUNTER	00203800
203900	C	SET ERROR CODE TO 3	00203900
204000		IF IT=0	00204000
204100		I:=1	00204100
204200		ILR=0	00204200
204300		IF (XCOF (N+1) .EQ. 0.0) GO TO 25	00204300
204400	10	IF (N) 10,15,32	00204400
204500	15	ILR=1	00204500
204600	20	RTURN	00204600
204700	25	ILR=4	00204700

204800	GL TU 20	00204800
204900	30 ILR=2	00204900
205000	GL TU 20	00205000
205100	32 IF (I=36) 35, 35, 30	00205100
205200	35 I: X=I:	00205200
205300	I: XX=I: +1	00205300
205400	I: 2=1	00205400
205500	I: J1= I: +1	00205500
205600	DL 40 L=1, KU1	00205600
205700	I: T=I: J1-L+1	00205700
205800	40 CLF (I: T)=XCOR (L)	00205800
205900	45 X0=.00500101	00205900
206000	Y0=0.01000101	00206000
206100	I: I.=0	00206100
206200	50 X=X0	00206200
206300	X0=-10.0*Y0	00206300
206400	Y0=-10.0*X	00206400
206500	X=X0	00206500
206600	Y=Y0	00206600
206700	I: I.=I: I.+1	00206700
206800	GL TU 59	00206800
206900	55 IF IT=1	00206900
207000	XI R=X	00207000
207100	YI R=Y	00207100
207200	50 I: CT=0	00207200
207300	60 UX=0.0	00207300
207400	UY=0.0	00207400
207500	V =0.0	00207500
207600	YT=0.0	00207600
207700	XT=1.0	00207700
207800	U=CEI (I: +1)	00207800

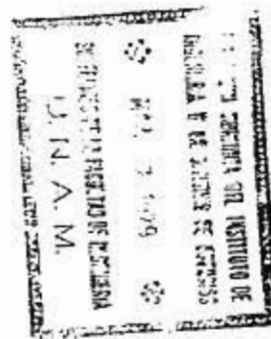
207500	IF (U) 65,130,05	00207900
208000	65 DL 70 I=1,N	00208000
208100	L =N-I+1	00208100
208200	TEHP=COF(L)	00208200
208300	XT2=X*XT-Y*YI	00208300
208400	YT2=X*YT+Y*XI	00208400
208500	U=U+TEHP*XT2	00208500
208600	V=V+TEHP*YT2	00208600
208700	I=I	00208700
208800	UX = UX+FI*XI+IENP	00208800
208900	UY=UY-FI*YI+IENP	00208900
209000	XT=XT2	00209000
209100	70 YT=YT2	00209100
209200	SUMSQ=UX*UX+UY*UY	00209200
209300	IF (SUMSQ) 75,110,75	00209300
209400	75 LX=(V*UY-U*UX)/SUMSQ	00209400
209500	X=X+LX	00209500
209600	LY=(U*UY+V*UX)/SUMSQ	00209600
209700	Y=Y+LY	00209700
209800	78 IF (DABS(DY)+DABS(DX)=1.0D-05) 100,20,80	00209800
209900	80 ICT=ICT+1	00209900
210000	IF (ICT=500) 60,85,85	00210000
210100	85 II (II IT) 100,90,100	00210100
210200	90 IF (II=5) 50,95,95	00210200
210300	95 ILR=3	00210300
210400	GL TO 20	00210400
210500	100 DL 105 L=1,N,XX	00210500
210600	HT=KJ1=L+1	00210600
210700	TEHP=XCOF(M1)	00210700
210800	XCOF(HT)=COF(L)	00210800
210900	105 CLF(L)=TEHP	00210900

211000	ITEMP=I:	00211000
211100	I=IX	00211100
211200	I,X=ITEMP	00211200
211300	IF (IF IT) 120,50,120	00211300
211400	110 IF (II IT) 115,50,115	00211400
211500	115 X=XPI:	00211500
211600	Y=YPI:	00211600
211700	120 IF IT=0	00211700
211800	122 IF (LAES (Y)-1.00-4*DAES (X))135,125,125	00211800
211900	125 ALPHA=X+X	00211900
212000	SUMSQ=X*X+Y*Y	00212000
212100	I=I+2	00212100
212200	GL TL 140	00212200
212300	130 X=0.0	00212300
212400	IX=IX-1	00212400
212500	I:XX=I:XX-1	00212500
212600	135 Y=0.0	00212600
212700	SUMSQ=0.0	00212700
212800	ALPHA=X	00212800
212900	I=I-1	00212900
213000	140 CLF (2)=CDF (2)+ALPHA*CDF (1)	00213000
213100	145 LL 150 L=2+N	00213100
213200	150 CLF (L+1)=CDF (L+1)+ALPHA*CDF (L)-SUMSQ*CDF (L-1)	00213200
213300	155 FLOTT1 (1,2)=Y	00213300
213400	FLOTT1 (1,2)=X	00213400
213500	I:2=I:2+1	00213500
213600	IF (SUMSQ) 160,165,160	00213600
213700	160 Y=-Y	00213700
213800	SUMSQ=0.0	00213800
213900	GL TL 155	00213900
214000	165 IF (I,) 20,20,40	00214000



	214100		END	00214100
	214200		SUBROUTINE MINV (A,N,D,L,M)	00214200
	214300	C	SUBROUTINE MINV	00214300
	214400	C	PURPOSE	00214400
	214500	C	INVERT A MATRIX	00214500
	214600	C	USAGE	00214600
	214700	C	CALL MINV(A,N,D,L,M)	00214700
	214800	C	DESCRIPTION OF PARAMETERS	00214800
	214900	C	A *INPUT MATRIX* DESTROYED IN COMPUTATION AND REPLACED BY	00214900
	215000	C	RESULTANT INVERSE.	00215000
	215100	C	N = ORDER OF MATRIX A	00215100
	215200	C	D = RESULTANT DETERMINANT	00215200
	215300	C	L = WORK VECTOR OF LENGTH L	00215300
	215400	C	M = WORK VECTOR OF LENGTH M	00215400
	215500	C	REMARKS	00215500
	215600	C	MATRIX A MUST BE A GENERAL MATRIX	00215600
	215700	C	SUBROUTINES AND FUNCTIONS SUBPROGRAMS REQUIRED	00215700
	215800	C	NONE	00215800
	215900	C	METHOD	00215900
	216000	C	THE STANDARD GAUSS-JORDAN METHOD IS USED. THE DETERMINANT	00216000
	216100	C	IS ALSO CALCULATED. A DETERMINANT OF ZERO INDICATES THAT	00216100
	216200	C	THE MATRIX IS SINGULAR.	00216200
	216300	C	IF A DOUBLE PRECISION VERSION OF THE ROUTINE IS DESIRED,THE	00216300
	216400	C	C IN COLUMN 1 SHOULD BE REMOVED FROM THE DOUBLE PRECISION	00216400
	216500		DOUBLE PRECISION AND *BIG* HOLD	00216500
	216600	C	THE C MUST ALSO BE REMOVED FROM DOUBLE PRECISION STATEMENTS	00216600
	216700	C	APPEARING IN OTHER ROUTINES USED IN CONJUNCTION WITH THIS	00216700
	216800	C	ROUTINE.	00216800
	216900	C	THE DOUBLE PRECISION VERSION OF THIS SUBROUTINE MUST ALSO	00216900
	217000	C	CONTAIN DOUBLE PRECISION FORTRAN FUNCTIONS, ABS STATEMENT	00217000
	217100	C	MUST BE CHANGED TO DAPS.	00217100

217200	r	SEARCH FOR LARGEST ELEMENT	00217200
217300		DIHENSION L(1),M(1),N(1)	00217300
217400		D=1.0	00217400
217500		NI="I.	00217500
217600		DL 80 I=1,N	00217600
217700		NI=NI+I.	00217700
217800		L(K)=I.	00217800
217900		H(K)=K.	00217900
218000		KI = NI + K	00218000
218100		BIGA = A(KK)	00218100
218200		DL 20 J = K,N	00218200
218300		IJ = I+(J-1)	00218300
218400		DL 20 I=K,N	00218400
218500		IJ = IJ + 1	00218500
218600		10 IF (ABS(BIGA)-ABS(A(IJ))) 15,20,20	00218600
218700		15 BIGA=A(IJ)	00218700
218800		L(K)=I	00218800
218900		H(K)=J	00218900
219000		20 CONTINUE	00219000
219100	r	INTERCHANGE ROWS	00219100
219200		J=L(K)	00219200
219300		I(J-1) 35,35,25	00219300
219400		25 KI=K+I	00219400
219500		DL 30 I=1,N	00219500
219600		KI=KI+I	00219600
219700		H(KI)=A(KI)	00219700
219800		JI = KI - K + J	00219800
219900		A(KI) = A(JI)	00219900
220000		30 A(JI) =HOLD	00220000
220100	r	INTERCHANGE COLUMNS	00220100
220200		35 I=H(L)	00220200



220300		IF (I=1.) 45,45,38	00220300
220400	3P	J1=I1*(I-1)	00220400
220500		LU 40 J=1,N	00220500
220600		J1=IK+J	00220600
220700		J1=J1+J	00220700
220800		HLL1=-A(JK)	00220800
220900		A(JK)=A(J1)	00220900
221000	4N	A(J1)=HDL0	00221000
221100	C	DIVIDE COLUMN BY HIGHS PIVOT (VALUE OF PIVOT ELEMENT IS	00221100
221200	C	CONTAINED IN BIGA)	00221200
221300	45	IF (BIGA) 48,46,48	00221300
221400	4A	L=0.0	00221400
221500		RETURN	00221500
221600	4P	LU 55 I=1,N	00221600
221700		IF (I=1.) 50,55,50	00221700
221800	5N	II=II+I	00221800
221900		A(II)=A(IK)/(=BIGA)	00221900
222000	55	CONTINUE	00222000
222100	C	REDUCE MATRIX	00222100
222200		LU 65 I=1,N	00222200
222300		II=II+I	00222300
222400		HLL1=A(IK)	00222400
222500		IJ=I-II	00222500
222600		LU 65 J=1,N	00222600
222700		IJ=IJ+I	00222700
222800		IF (I=1.) 60,65,60	00222800
222900	6N	IF (J=1.) 62,65,62	00222900
223000	6P	IJ=IJ-I+K	00223000
223100		A(IJ)=HLL1*A(KJ)+A(IJ)	00223100
223200	65	CONTINUE	00223200
223300	C	DIVIDE ROW BY PIVOT	00223300

223400		KJ=K+1	00223400
223500		DL 75 J=1,N	00223500
223600		I,J=I,J+1	00223600
223700		II (J=I) 70,75,70	00223700
223800	70	A(KJ)=A(KJ)/B10A	00223800
223900	75	CONTINUE	00223900
224000	C	PRODUCT OF PIVOTS	00224000
224100		C=D*L10A	00224100
224200	C	REPLACE PIVOT BY RECIPROCAL	00224200
224300		A(KK)=1.0/H10A	00224300
224400	80	CONTINUE	00224400
224500	C	FINAL ROW AND COLUMN INTERCHANGE	00224500
224600		K=I	00224600
224700	100	K=(I-1)	00224700
224800		II (K) 150,150,105	00224800
224900	105	I=L(K)	00224900
225000		II (I=I) 120,120,108	00225000
225100	108	JL=I*(I-1)	00225100
225200		JL=I*(I-1)	00225200
225300		DL 110 J=1,N	00225300
225400		JL=J0+J	00225400
225500		HLLI=A(JK)	00225500
225600		J1=J1+J	00225600
225700		A(JK)=-A(J1)	00225700
225800	110	A(J1) =HOLD	00225800
225900	120	J=I(I)	00225900
226000		II (J=I) 100,100,125	00226000
226100	125	K1=K-1	00226100
226200		DL 130 I=1,N	00226200
226300		K1=K1+1	00226300
226400		HLLI=A(K1)	00226400

