MANUAL DE USUARIO PARA PROGRAMACION DE MATLAB FOR WINDOWS CON APLICACION A ALGEBRA DE MATRICES.

JOSE TRINIDAD RODRIGUEZ ESTRADA

T E S I S

PRESENTADA COMO REQUISITO PARCIAL
PARA OBTENER EL GRADO DE

MAESTRO EN CIENCIAS
EN ESTADISTICA EXPERIMENTAL

Universidad Autonoma Agraria
Antonio Narro
PROGRAMA DE GRADUADOS
Buenavista, Saltillo, Coah.
FEBRERO 1995

MANUAL DE USUARIO PARA PROGRAMACION DE MATLAB FOR WINDOWS CON APLICACION A ALGEBRA DE MATRICES

JOSE TRINIDAD RODRIGUEZ ESTRADA

T E S I S

PRESENTADA COMO REQUISITO PARCIAL
PARA OBTENER EL GRADO DE
MAESTRO EN CIENCIAS
EN
ESTADISTICA EXPERIMENTAL

Universidad Autónoma
Antonio Narro

PROGRAMA DE GRADUADOS
Bunavista Saltillo, Coah.
FEBRERO DE 1995

Tesis elaborada bajo la supervisión del comité particular de asesoría y aprobada como requisito parcial , para optar al grado de

MAESTRO EN CIENCIAS

EN ESTADISTICA EXPERIMENTAL

COMITE PARTICULAR

Asesor Principal: _____

M.C. Mario Cantú Sifuentes.

Asesor: _____

M.C. Félix de Jesús Sánchez.

Asesor: _____

M.C. Héctor Franco López.

Asesor: _____

M.C. Cesar González Rivera.

Asesor: _____

M.C. Victor Cantú Hernández

_____

Dr. Jesús Fuentes Rodríguez

Subdirector de Postgrado

Buenavista, Saltillo, Coahuila. Febrero, 1995.

DEDICATORIA

A MIS PADRES :

Leonor Estrada

José Rodríguez

Por todo su apoyo.

A MIS HERMANOS

Victoria

María Elena

Francisco

Petra

Juan

Margarita

Cony

Angeles

Por ser como son.

AL AMOR DE MI VIDA

LAURA ALICIA TORRES FLORES

Por que siempre estuvo con migo.

A MI NIÑA

Mi hija Lesli

Por quien todo tiene sentido.

AGRADECIMIENTO

El autor de este trabajo expresa su agradecimiento a las siguientes personas e instituciones.

A la Universidad Autónoma Agraria "Antonio Narro", por darme la oportunidad de estudiar una maestría.

Al M.C. Mario Cantu Sifuentes, por su gran apoyo, colaboración y solidaridad.

A mis Amigos y Compañeros de cubiculo Sergio Sánchez y Alberto Rodríguez , por sus Sugerencias.

A los maestros del Departamento de Estadística y Calculo, por sus consejos.

COMPENDIO

Manual de usuario para programación de Matlab for Windows con aplicación a Algebra de Matrices

POR

JOSE TRINIDAD RODRIGUEZ ESTRADA

MAESTRIA

ESTADISTICA EXPERIMENTAL

UNIVERSIDAD AUTONOMA AGRARIA ANTONIO NARRO

BUENAVISTA, SALTILLO, COAHUILA. SEPTIEMBRE 1994

M.C. Mario Cantú Sifuentes - Asesor-

Palabras claves: Matlab, Matrices, Programación, Windows

En Este trabajo se presenta un manual de usuario para programación en Matlab for Windows para solución de matrices, así como un programa donde se presentan algunas aplicaciones de las funciones predefinidas de Matlab.

El manual esta diseñado para dar una introducción tanto a programación para usuarios novatos así como para la aplicación directa en Algebra de Matrices, el programa a la vez se presenta de una forma demostrativa para utilizarse como ejemplo para diseñar nuevos programas y dar una idea precisa del alcance de matlab tanto como lenguaje de programación así como herramienta potencial para solucionar problemas que involucren matrices y algunos otros problemas matemático.

ABSTRACT

Manual for programation of Matlab for Windows with algebraic application

BY

JOSE TRINIDAD RODRIGUEZ ESTRADA

MASTER OF SINCE

EXPERIMENTAL STATISTICS

UNIVERSIDAD AUTONOMA AGRARIA ANTONIO NARRO

BUENAVISTA, SALTILLO, COAHUILA. SEPTEMBER 1994

M.C. Mario Cantú Sifuentes -Advisor-

Key words: Matlab, Matrix, Programation, Windows

In this presentation a handbook of user for programation in matlab for windows to solve matrix, and a software where is presented some applications of the functions predefine of matlab.

The handbook is designed for an introduction for programation for new users and for the application in algebra of matrix.

The software is presented in a demostrative for as an exemple to design new, programs and to give a precise idea of the lenguaje of programation and tool to solve problems with matrix and some mathematics problems.

INDICE DEL CONTENIDO

CAPITULO I

INTRODUCCION

Los sistemas de cómputo han evolucionado durante los últimos veinte años desde su invención como experimento informático hasta llegar a convertirse en uno de los entornos más populares debido a su aplicabilidad a gran diversidad de disciplinas científicas y de ingeniería en todo el mundo. Hoy día mas de un millón de usuarios de computadoras cuyos equipos se extienden desde las micro computadoras con recursos limitados hasta las mayores maxi y supercomputadoras utilizan software diseñado específicamente para sus actividades principales. Este crecimiento se está acelerando cada vez más conforme más y más usuarios sucumben a la sorprendente flexibilidad, potencia y elegancia que proporcionan los sistemas computacionales en general.

A pesar de que las primeras investigaciones de cómputo se realizaron en laboratorios de instituciones investigadoras del extranjero, la historia de los sistemas de cómputo es casi única, debido a que sus avances son en gran medida, aportaciones de personas con ideas creativas singulares. Para obtener un nuevo sistema de cómputo, se encierra bajo llave a un puñado de programadores a sueldo en una habitación cerrada (con una computadora), y cada año o así, se recoge el software que más de la mitad de ellos estén utilizando, y esa es la nueva versión de software que se use en el mercado computacional. La implicación de lo anterior es que los avances importantes no han venido de decisiones burocráticas, sino más bien de las necesidades y la creatividad de los usuarios. esto sigue siendo cierto hoy, lo que hace de los sistemas de cómputo uno de los jardines más fértiles para la creación de nuevos conceptos en aplicaciones computacionales.

El propósito original al diseñar esta Tesis es la de proporcionar un Software que sirva de apoyo para los Cursos de Postgrado, tanto de Programación, como para Algebra de Matrices, que se imparten en la Universidad Autónoma Agraria "Antonio Narro", el Material está especificado para la Maestría de Estadística Experimental, sin embargo, a otras Maestrías del postgrado les será útil si se interesan por los cursos de Programación y Algebra de Matrices.

El Principal Propósito al Analizar el Software para Algebra Lineal es el de Incrementar la Variedad de Herramientas, así como seleccionar el Lenguaje más apropiado para el tratamiento de Operaciones Matriciales y Simplificar lo más posible la Implementacion de Algoritmos en Rutinas Computacionales que logren conducir con éxito el desarrollo del Algebra de Matrices.

Por un lado se ha seleccionado el software Matlab for Windows, para facilitar los cálculos científicos y de ingeniería, por su versatilidad como sistema interactivo basado en rutinas de Cómputo diseñadas en Fortran y Lenguaje C, para aplicaciones de Operaciones con Matrices, y por otro lado con todas las facilidades de acceso que proporciona un ambiente Multitarea de Windows 3.1.

Por otra parte el manejo de opción múltiple en el sistema diseñado para la selección de operaciones con Matrices constituye un apoyo rápido y eficiente en la solución de problemas de Algebra de Matrices, y por último constituye un Lenguaje Sencillo de Programación fácil de aprender y de aplicar en futuros cursos de diseño de Modelos Lineales.

Considero que la atención particular se debe procurar a la comprensión de aspectos Algebraicos básicos, por esta razón se ha evitado introducir demasiados conceptos en la programación que hagan complicado y difícil de entender el software diseñado.

A lo largo de la descripción tanto de los conceptos fundamentales así como su Programación se incluyen algunos ejemplos, que hacen de este trabajo un manual práctico de usuario tanto de introducción a programación Matlab, así como de Introducción de operaciones básicas de Algebra de Matrices, se agregan ademas algunos tópicos con comandos avanzados y funciones definidas proporcionados por Matlab, que se trataran de forma independiente.

Este material podrá usarse opcionalmente en forma conjunta en los cursos de Programación y de Algebra de Matrices, en Programación podrá iniciarse en el diseño de software tomando las herramientas que proporciona Matlab y en Algebra de Matrices podrá percibir la teoría del curso en su clase y tendrá acceso a un apoyo práctico con el uso de este software.

Objetivos

Crear un manual adecuado a las necesidades del curso de programación y además introducir un programa para realizar operaciones básicas de álgebra de matrices, en cuanto a conceptos, terminología y diseño de presentación de resultados. Este trabajo lograra la continuación del proceso del avance de las ciencias de la computación a la Universidad y el aprovechamiento del cómputo como un valioso instrumento de desarrollo en las materias de cada especialidad.

CAPITULO II

REVISION DE LITERATURA

Para el diseño de este software así como para el desarrollo de manual adjunto, se obtuvo en base a creatividad y la experiencia vivida en los cursos de programación y de álgebra de matrices. De tal forma en el presente capítulo se dará una descripción de las técnicas utilizadas en los conceptos que construyeron tanto el software como su manual.

La estructura principal de las técnicas empleadas, son las proporcionadas por la naturaleza del lenguaje computacional matlab for windows y su interpretación práctica relacionada con álgebra de matrices, (principalmente con Hoffman (1973)) los conceptos complementarios y sus referencias se concentran en Sigmon, k., (1992). Así como la teoría matemática que respalda el análisis de programación de los algoritmos tratados computacionalmente (Moler, C., et al . , 1987).

Sin duda la notación utilizada será fácil de entender por ser congruente con la notación establecida en el curso de álgebra de matrices y por otro lado la simplicidad que proporciona el manual para programación de estructuras tanto de álgebra de matrices, como para manejo de datos estadísticos en general serán de utilidad para introducir a usuarios de nivel principiante en análisis y diseño de sistemas computacionales (Coffin, S., 1992), enfocados a problemáticas matemático - estadísticas, así como los usuarios intermedios que tienen experiencia en otros lenguajes de alto nivel , les resultara una plataforma interesante y una opción mas para programar , podrán comparar las ventajas que proporciona la ejecución de programas en matlab for windows en aplicaciones de análisis numérico y podrán notar la conveniencia de utilizar este lenguaje en los cursos antes mencionados.

El manual de Matlab presentará las características de un manual de consulta y referencia básico su fácil acceso a conceptos no rebuscados permitirá la utilización de comandos potentes en aplicaciones prácticas del manual como si se tratara de un tutorial tanto para aprender sintaxis y lógica computacional, así como si fuera un libro para consultas rápidas de detalles para implementacion de algoritmos estadísticos que involucren tratamiento de datos en forma de sistemas de ecuaciones lineales, también presentará la ventaja de no ser un manual tan extenso que se pierda en la generalidad de los conceptos de programación y por otro lado tampoco resultará en un manual ambiguo que pierda profundidad en sus conceptos , más bien se pretenderá proporcionar un manual de matlab para usuarios estadísticos que aprovechen las facilidades que proporciona matlab para el tratamiento de datos.

CAPITULO III

MATERIALES Y METODOS

Los materiales que intervinieron en la realización de esta obra son los que a continuación se listan:

Micro computadora PC con sus respectivos componentes tanto de hardware como elementos de software:

Hardware:

1. Marca: Acer
2. Modelo: AcerPower 386SX
3. Memoria en RAM: 2 M
4. Unidades de disco : Flexible, doble lado, alta densidad, de 3.5 pulgadas.
   Flexible, doble lado, alta densidad, de 5.25 pulgadas.
5. Mause: Acer M/N : M-SG14
6. Impresora: Láser jet HP plus series II

Software :

6. Paquete Compilador : Matlab for windows.
7. Ambiente multitarea : Windows 3.1.
8. Procesador de palabras : Microsoft Word .
9. Sistema Operativo : DOS Microsoft 5.

Métodos

Se comprendieron los siguientes aspectos:

Técnica de Programación:

El lenguaje proporcionado por matlab for windows fue el seleccionado debido a que contiene características únicas que lo hacen el más indicado para el desarrollo de software que involucre operaciones con matrices y por otro lado, por su sencillez sintáctica resulta un lenguaje fácil de entender y accesible para aprender, por otro lado debido al carácter de sus librerías de rutinas predefinidas, hacen del software diseñado un programa fácil de ejecutar y dar mantenimiento de así requerir.

Técnica Estadística.

La principal técnica utilizada en el desarrollo del software está influenciada por la técnica actual que se imparte en el curso de Algebra de Matrices, así como los conceptos seleccionados para constituir el manual están relacionados tanto con el contenido analítico de Programación como de Algebra de matrices, por otro lado no menos importante se considera el respaldo bibliográfico mencionado en el capítulo de Literatura Citada.

CAPITULO IV

RESULTADOS

Manual de Usuario para Programación de MATLAB for Windows

con Aplicación a Algebra de Matrices

José T. Rdz. Estrada

Departamento de Estadística y Cálculo, Universidad Autónoma Agraria Antonio Narro,

Buenavista Saltillo, Coah. México 1995 .

INTRODUCCION

Matlab es un lenguaje computacional diseñado para trabajar de manera interactiva con matrices para efectuar operaciones que involucren cálculos científicos y de ingeniería. Debido a su gran potencia es posible implementar rutinas para resolver problemas de análisis numérico de relativa complejidad sin elaborar códigos fuente demasiado extensos, como sucedería en otros lenguajes de orientación científica. El nombre de Matlab resulta ser la abreviatura para MATrix LABoratory.

Un uso común es el de efectuar cálculos numéricos, sin embargo debido a su versatilidad es posible implementar prototipos de algoritmos y resolver problemas especiales relacionados con fórmulas que utilicen matrices, dichas fórmulas son principalmente estadísticas.

MATLAB trabaja esencialmente con objetos numéricos rectangulares denominados matrices donde es posible tener elementos complejos. para algunas situaciones especiales se trabajara con matrices de una fila (renglón) y una columna, es decir con un escalar, y por otro lado se trabajara con solamente una fila o una columna, en tal caso a la estructura en cuestión se le denominara vector.

Introducción de Matrices

Si se tiene un sistema de ecuaciones lineales, cuyas incógnitas se denominan como x1,..,xn, y sus coeficientes se representan como los elementos Aij y los escalares de la parte derecha de las ecuaciones son los yi, entonces a tal sistema podremos representarlo de la siguiente manera :

$$AX = Y$$

Donde

$$A = \begin{bmatrix} A11 \ldots A1n \\ \vdots \quad \vdots \\ Am1 \ldots Amn \end{bmatrix}$$

$$X = \begin{bmatrix} x1 \\ \vdots \\ xn \end{bmatrix} \quad y \quad Y = \begin{bmatrix} y1 \\ \vdots \\ ym \end{bmatrix}$$

A se llama matriz de coeficientes del sistema. Estrictamente hablando, la disposición rectangular expuesta no es una matriz, sino una representación de una matriz, con las siguientes características:

* una matriz m*n.

* contiene a los pares de enteros (i, j), 1<= i < = m, 1<= j <= n.

* Los elementos de la matriz A son escalares A (i, j) = Aij.

Ahora definiremos las diferentes formas para introducir una matriz en Matlab:

- Introduciendo un lista explícita de elementos.

- Generándola mediante funciones predefinidas de Matlab.

- Escribiendo un archivo .m.

- Leyéndola de un archivo de datos externo.

Introduciendo una lista explícita de elementos

Es la entrada mas común utilizada frecuentemente especialmente cuando se trata de matrices relativamente pequeñas donde es posible manejar sin ningún problema sus elementos, de esta forma se listan los elementos de la matriz separados por blancos encerrados entre corchetes ([ ]) y usando punto y coma (;) para indicar fin de línea, a excepción del último fin de linea.

ejemplo

A = [1 2 3; 4 5 6; 7 8 9]

El resultado que se obtendrá después de teclear intro.

A =

1 2 3

4 5 6

7 8 9

En forma opcional los datos pudieron introducirse también de la siguiente manera

A = [1 2 3

4 5 6

7 8 9]

Generándola mediante funciones predefinidas de Matlab

Existen algunas funciones de matlab predefinidas que proporcionan algunas matrices especiales que se encuentran disponibles solo con llamarlas.

ejemplo

rand

magic

hilb

Las cuales se describirán en una sección de funciones definidas mas delante.

Escribiendo un archivo .m

Este método generalmente se utiliza cuando se tiene una matriz m*n demasiado grande y donde es factible cometer errores en el momento de su edición por tal razón resulta mas conveniente introducir los datos en un archivo .m donde es posible realizar correcciones en los elementos de la matriz, la manera de generar archivos m. se tratara por separado.

Leyéndola de un archivo de datos externo

Esta opción se aplica para matrices que se exportan de otros archivos diferentes de los de matlab (.m) , tales como lotus o dbase.

Cabe mencionar que cuando algunos de los números se escriben en forma exponencial , deben evitarse espacios en blanco.

ejemplo

A= [5.87e-9 12.23e-9 23.65e-9

1.11e-9 22.22e-9 33.33e-9

4.44e-9 55.55e-9 66.66e-9]

Operaciones con Matrices

Para trabajar operaciones entre matrices es necesario trabajar con operadores, los que matlab soporta son los siguientes:

+ adición

- sustracción

* multiplicación

^ potenciacion

' traspuesta

\ matiz inversa por la izquierda

/ matriz inversa por la derecha

De igual manera se utilizan los anteriores operadores para trabajar con elementos individuales (escalares), para los casos en que las reglas tanto de multiplicación de matrices, adición , resta o inversión no se satisfagan el compilador matlab envía un mensaje de error.

adición y sustracción

Para sumar (+) y para restar (-) matrices es necesario que las matrices tengan la misma dimensión, es decir que tengan el mismo numero (finito) de elementos.

ejemplo

si tenemos dos matrices A y B ambas de 3*3 :

A = [1 2 3 B = [1 4 7

4 5 6 2 5 8

7 8 9] 3 6 0]

y sumamos (instrucción matlab)

C = A + B

obtenemos

C =

2 6 10

6 10 14

10 14 0

si restamos (instrucción matlab)

C = A-B

obtenemos

C =

0 -2 -4

2 0 -2

4 2 9

multiplicación

Es factible la multiplicación de un escalar por una matriz , sin embargo la multiplicación entre matrices debe tener ciertas características.

Si A es una matriz m * n y si B es una matriz n * p, entonces el producto C = A*B es la matriz m * p, es decir para que una multiplicación sea probable es necesario que el número de columnas de la primer matriz sea igual al numero de filas de la segunda. Si multiplicamos (instrucción matlab)

C = A*B

potenciacion

Se utiliza principalmente para elevar al cuadrado una matriz , la expresión que se utiliza es

A^ p

Donde A es una matriz y p es un escalar, generalmente se utiliza un p entero y positivo para evitar errores en la operación aún cuando en algunos casos es posible elevar a fracciones la matriz usando valores propios (eigenvalues).

traspuesta

La traspuesta de una matriz consiste en intercambiar las filas por los renglones de la matriz y su expresión en matlab es:

A'

A

matiz inversa por la izquierda

matriz inversa por la derecha

Uno de los conceptos más importantes en el álgebra de matrices es la operación de inversión, tanto por la izquierda como por la derecha, ya que por medio de ellas se obtienen las soluciones en los sistemas de ecuaciones, además de que representan la base para realizar cálculos en teoremas más elaborados de modelos lineales.

En la inversa por la izquierda su interpretación matemática es la siguiente:

X = A\ B es la solución para A*X = B

En la inversa por la derecha su interpretación matemática es la siguiente:

X = B/A es la solución para X*A = B

Para que las divisiones (inversión por la izquierda) A\ B y B/A puedan estar definidas es necesario que ambas matrices sean cuadradas y que B*A = I y A*B = I donde I es la matriz identidad , el cálculo es realizado a través de eliminación Gaussiana.

Sintaxis de Matlab

Para realizar expresiones en matlab se utiliza la siguiente regla de sintaxis.

variable = expresión

o simplemente

expresión.

Las expresiones se componen de operadores, variables, y funciones. cuando se evalúa una expresión por el compilador de matlab se produce una matriz la cual se despliega en pantalla y se asigna a la variable declarada a la izquierda del signo igual en la regla de sintaxis antes mencionada, el contenido de la variable puede utilizarse posteriormente, si simplemente se utiliza una expresión el resultado se guarda en la variable ans de "answer".

ejemplo

instrucción matlab

```
18/6
```

produce

```
ans =

     6
```

Las instrucciones matlab terminan cuando tecleamos intro, cuando una instrucción es demasiado larga es posible escribirla en dos o más lineas introduciendo al final de cada

línea a excepción de la última los siguientes caracteres, " ... ", los cuales le indican al compilador que la instrucción continúa en la siguiente linea

ejemplo

suma = 1 - 1/2 + 1/3 - 1/4 + 1/5 - 1/6 + 1/7 ...

- 1/8 + 1/9 - 1/10 + 1/11 - 1/12

Si se quieren introducir varias instrucciones por línea se pueden introducir separándolas por comas o puntos y comas, si el último caracter de una instrucción es un punto y coma el resultado no se mostrará en pantalla, aunque sí se realizará la asignación, esto se usará para evitar pérdidas de tiempo al mostrar los resultados intermedios.

Los nombres de variables y funciones se construyen con letras y dígitos (hasta 19 caracteres), para el compilador de matlab no es lo mismo, a, que, A, por lo que se tomará la siguiente regla para evitar confusiones: todos los nombres de funciones serán escritos por letras minúsculas y las matrices serán escritas con letras mayúsculas.

ejemplo

inv (A)

la función se denomina inv

y la matriz es A

la operación de función es sobre la matriz dentro del paréntesis,

note que

inv (A) no es igual a INV (A)

puesto que INV se tomaría como una indefinicion

Cabe decir que existe un comando que hace al compilador leer en forma indistinta las letras mayúsculas y minúsculas este comando es : casesen.

ejemplo

casesen

INV(a)

será una instrucción válida

por lo tanto

inv (a) será equivalente a INV (A) e INV(a).

Variables permanentes (constantes): hasta ahora conocemos la variable permanente answ proporcionada por el compilador, de este tipo existen algunas otras tales como:

eps precisión de la computadora eps = 2 ^ -52

pi pi = 3.1416

inf se utiliza para indeterminaciones como divisiones por cero, donde se envía un mensaje de warning, la variable inf, representa la bandera que marca que ha sucedido una indeterminación, sin ciclar o abortar la ejecución de las instrucciones.

NaN se relaciona con inf para los casos en que se realizan operaciones indebidas como 0/0, bandera de " no es un número"

Tales constantes son llamadas a pantalla con la instrucción who, para eliminar el contenido de una variable no permanente de la memoria se utiliza la instrucción clear nombre - de - variable, si requiere borrar todas las variables no permanentes se escribe solo clear.

Números y expresiones aritméticas:

Los números que pueden utilizarse sin problemas de longitud van en el rango de 10 ^-308 a 10^308, salvo restricciones del computador en particular es decir que pueden introducirse dígitos como los siguientes:

3 99 0.0001

9.6397238 1.6021E-20 6.02522e23

Para manejar números complejos es necesario generar una unidad compleja:

i = sqr (-1)

ahora puede utilizarse en instrucciones como:

a = 4+2*i

Generalizando para matrices

ejemplo

D = [1 2 ; 3 4] + i * [8 3 ; 3 2]

ejemplo

B = [1+5*i 2+6*i; 3+7*i 4+8*i]

Instrucción Save, Load, Quit, !, Hardcopy

Para salir del editor de matlab, es necesario escribir "quit" o "exit", lo cual indica que termina la sesión con matlab un cuyo caso las variables utilizadas en el espacio de trabajo se pierden, por tal razón antes de salir es posible salvar dichas variables con la instrucción "save", esta instrucción salva todas las variables y sus contenidos en un archivo en disco llamado por default "matlab.m",dependiendo del drive que se esté manejando (A, B , C,D, etc.) , en la siguiente sesión para abrir el archivo salvado como "matlab.mat" se utilizará la instrucción "load", en forma general "save" y "load" serán usados para salvar y llamar otros archivos, para salvar variables seleccionadas se utilizará el archivo "tem.mat", la instrucción se manejara así:

save temp X

salva solo la variable X, mientras que

save temp A B C

guarda A B C.

El signo "!" se utiliza para ejecutar operaciones del sistema operativo sin salir de matlab, la instrucción seguida de "!" se efectuará y al finalizar trasladará el control del programa a la siguiente instrucción de matlab.

ejemplo

```
>> ! edit archivo.m
```

si edit es un editor de texto del sistema operativo, la instrucción anterior editará el archivo.m y regresara el control del programa a matlab.

Hardcopy: es una copia por impresora o en un archivo de disco, para obtener un hardcopy se utiliza la instrucción "diary" nombre-de-archivo, hace que todo lo escrito a continuación en pantalla sea guardado en el nombre - de - archivo (si se omite el nombre de-archivo, se toma por defecto "diary") esto se realizara hasta que se encuentre el compilador con la instrucción "diary off", si se requiere seguir guardando (código) después de haber tecleado "diary off " puede reanudarse la opción de copia utilizando "diary on". al final de la sesión puede editarse el archivo, imprimirse etc.

Instrucciones de ciclo (Loops) For, While, If

Matlab contiene instrucciones de control de flujo, que se pueden encontrar en otros lenguajes, tales como for, while e if .

Instrucción for: se utiliza para generar ciclos repetitivos con control de número de veces a realizar el ciclo.

sintaxis:

```
for v = condición
    instrucciones
end
```

ejemplo:

```
for i = 1 : n
x ( i ) = 0
end
```

inicializa a cero los n elementos de x

Instrucción While : la instrucción while se utiliza para ciclos que se repetirán mientras la expresión de control sea verdadera.

sintaxis:

```
while condición
instrucciones
end
```

ejemplo:

cual es el primer entero n para el cual su factorial es un numero de 100 dígitos.

```
n=1
while  prod(1:1) < 1.e 100, n=n+1; end
n
```

Instrucción if: la instrucción if se utiliza para ciclos de opción.

sintaxis:

```
if  condición
instrucciones
end
```

ejemplo:

```
if  n = = 1
a = input ( ' introduce un numero positivo ' )
end
```

los operadores relacionales que se utilizan para condiciones son:

<	menor que
>	mayor que
< =	menor igual que
> =	mayor o igual que
= =	igual
- =	no igual

los operadores lógicos

&	y	(and)
|	o	(or)
-	no	(not)

Cuando se aplican a escalares, una condición es realmente el escalar 1 o 0 dependiendo de si la condición es verdadera o falsa.

Cuando se aplica a matrices del mismo orden, una condición entre ellas da lugar a una matriz de ceros y unos, dando el valor de la condición entre las correspondientes entradas.

Cuando se utiliza una condición entre matrices dentro de un ciclo while o if, la condición se entiende verdadera si cada una de las entradas de la matriz de la condición es no nula. Entonces, si se quiere efectuar alguna expresión cuando las matrices A y B son iguales, se escribirá

```
if A = = B
expresión
end
```

Pero si se desea ejecutar la misma instrucción si A y B son distintas, hay que recurrir a:

```
if any ( any ( A-= B ))
```

expresión

end.

Donde any es una función de operación vectorial por tal razón se utiliza dos veces, para reducir relaciones entre matrices a relaciones entre vectores escalares, es decir en la condición if pregunta si algún elemento de A no es igual a algún elemento de B entonces expresión

o

if A = = B else

expresión

end

note que

if A - = B , expresión, end

no se ejecutará como se desea ya que la instrucción sólo se ejecutará si todas las entradas de A son distintas de las de B.

Funciones: Escalares, Vectoriales, Matriciales

La razón más importante que da un poder matemático con matrices a matlab es sin duda sus funciones. las funciones de matlab pueden estar guardadas en librerías o en archivos.m o bien como funciones de tipo instrucción comando, ademas matlab permite la creación de funciones diseñadas por el usuario según sus necesidades. las siguientes funciones están disponibles en matlab:

Funciones Matriciales:

eye	matriz identidad
zeros	matriz de ceros
ones	matriz de unos
diag	matriz diagonal

triu	parte triangular superior de una matriz
tril	parte triangular inferior de una matriz
rand	matriz generada aleatoriamente
hilb	matriz de hilbert
magic	matriz mágica
inv	matriz inversa
det	el determinante de la matriz
eig	valores propios de la matriz (eigenvalues)
rank	rango de la matriz
chol	factorizacion de Cholesky
svd	descomposición en valores singulares
lu	factorizacion triangular superior e inferior
qr	factorizacion ortogonal
hess	forma de henssenberg
schur	descomposición de schur
rref	forma escalonada reducida por filas
expm	matriz exponencial
sqrtm	matriz raíz cuadrada
poly	polinomio característico
size	tamaño
norm	norma 1, norma 2, norma de Frobenius, norma infinita
cond	numero de condición en la norma 2

ejemplo

zeros (m, n)

produce una matriz nula m*n

zeros(n)

produce una matriz cuadrada de orden n

si A es una matriz, entonces

zeros (A)

produce una matriz de ceros del mismo orden que A.

ejemplo

si x es un vector

diag (x)

es la matriz diagonal con x en su diagonal;

si A es una matriz cuadrada

diag (A)

es un vector formado por la diagonal de A

Funciones Escalares:

Algunas funciones de matlab operan esencialmente sobre escalares, aunque lo hacen también sobre matrices (elemento a elemento). Las funciones mas comunes entre estas son:

sin	seno
exp	exponencial regresa e ^ x
cos	coseno
tan	tangente
asin	arcseno
acos	arcoseno
atan	arcotangente

log	logaritmo (natural)
rem	remanente
abs	valor absoluto
sqrt	raíz cuadrada
sign	signo, para cada elemento de la matriz, regresa 1 si el elemento es mas grande que cero, 0 si es igual acero y -1 si el elementos menor que cero.
round	redondea
floor	redondea a menos infinito
ceil	redondea a mas infinito
real	la parte real
imag	parte imaginaria
conj	conjugados complejos
fix	redondeando a cero
atan	cuarto cuadrante arcotangente
sinh	seno hiperbólico
cosh	coseno hiperbólico
tanh	tangente hiperbólica
bessel	funciones bessel
gamma	función gamma
rat	aproximación racional

Funciones Vectoriales: otras funciones de matlab operan fundamentalmente sobre vectores (fila o columna) aunque también pueden operar sobre matrices m * n (m > = 2) haciéndolo en este caso columna a columna, produciendo, por tanto, un vector fila que contiene el resultado de su aplicación a cada columna para que actúen por filas basta usar su traspuesta;

ejemplo

mean (A ') '

max	elemento mas grande
min	elemento mas pequeño
sort	ordena cada columna en orden ascendente
sum	suma de los elementos del vector
prod	para vectores el producto de sus elementos
median	calcula el valor mediana del vector
mean	calcula el valor medio del vector
std	calcula la desviación standard del vector
any	para vectores regresa un 1 si alguno de los elementos del vector no es cero,en otro caso regresa cero
all	para vectores regresa 1 si todos los elementos del vector no son cero, n otro caso regresa 0

Edición: De Línea, Archivos .M

De Linea

Resulta de fácil uso la edición de línea de matlab. El cursor se sitúa en la posición deseada utilizando las teclas de flechas izquierda y derecha mientras que para borrar caracteres pueden usarse las teclas de retroceso (backspace) o suprime (supr).

Se recomienda usar las teclas de flechas arriba y abajo para recuperar los comandos previos. Se puede, por tanto recuperar una linea de comandos previa,

editarla,ejecutarla revisada. Para pequeñas rutinas, esto es más conveniente que usar un archivo .m

ejemplo

Si se quieren comparar las gráficas de las funciones y = sen mx e y = cos nx en el intervalo [0, 2pi] para varios m y n, se podrá hacer desde la línea de comandos:

m=2; n=3 ; x=0 :.01: * pi ; y = sin (m*x) ; z = cos (n * x) ; plot (x, y, x, z)

Archivos .m

Matlab puede ejecutar una sucesión de instrucciones almacenadas en archivos de disco. Estos archivos se denominan "archivos .m ", debido a que su extensión que lo identifica como archivo matlab es " .m ", en matlab se recomienda trabajar con archivos .m.

Los archivos .m se clasifican en dos tipos : archivos de instrucciones y archivos de funciones.

Archivos de instrucciones.

Un archivo de instrucciones consiste en una sucesión de instrucciones matlab. los archivos de instrucciones se ejecutan al escribir su nombre en la línea de comandos, sus variables son globales y, por lo tanto, cambiaran los valores del espacio de trabajo.

Los archivos de instrucciones se utilizan a menudo para introducir datos en una matriz grande. la ventaja de utilizar archivos es que resulta más sencillo corregir errores sin tener que elaborar nuevamente todo el trabajo.

ejemplo

se escribe en el archivo " datos.m "

```
A = [
      1 2 3 4
      5 6 7 8 ];
```

entonces se escribe la instrucción matlab " datos " (nombre del archivo .m) se efectuará la asignación especificada en datos.m

Un archivo .m puede hacer llamadas a otros, incluyendo a el mismo, es decir puede ser recursivo.

Archivos de funciones.

Los archivos de funciones de Matlab hacen posible aumentar la capacidad de crecimiento al programar funciones especificas para un problema determinado, y , a partir de su introducción, tendrán el mismo rango que las demás funciones del sistema. Las variables en las funciones se consideran como locales.

Sintaxis

function argumentos de entrada y argumentos de salida

instrucciones

ejemplo

```
function [ media, desv] = estad ( x )
% función estadística ESTAD que calcula la media y desviación típica. para un
vector,
% estad (x) calcula la media y la desviación típica de x.
% Para una matriz x, estad ( x ) proporciona dos vectores fila conteniendo,
% resp l a media y la desviación típica de cada columna.
[ m,n ] = size ( x )
if m = = 1
        m = n ; % caso de un vector fila
end
media = sum (x ) / m;
desv = sqrt ( sum (x.^2) / m - media. ^2 )
```

llamada de la función

[xm, xd] = estad (x)

matlab asignara la media y la desviación típica de x a las variables xm y xd, respectivamente.

Cuando se dispone de una función con argumento de salida múltiple,se hacen asignaciones simples.

ejemplo

xm = esad (x)

se asignara la media de x a xm

no son necesarios los corchetes alrededor de xm .

El símbolo % indica que el resto de la linea es un comentario, que documenta el archivo el compilador matlab ignora el resto de la línea, existe un comando help a través del cual es posible editar en pantalla los comentarios que documentan una archivo,

ejemplo

help estad.

Es recomendable utilizar documentación en cualquier archivo .m diseñado ya sea función o archivo de instrucciones, dicha documentación debe especificar que hace el archivo, como utilizarlo y algunos otros detalles importantes a considerar durante la ejecución del archivo (programa).

En la función anterior se puede apreciar que

x.^2	es la matriz de los cuadrados de las entradas de x
sum	es una función vectorial
sqrt	es una función escalar
sum (x) / m	la división opera una matriz con un escalar

ejemplo

Función para obtener el máximo común divisor de dos enteros usando el método de Euclides, donde es posible distinguir la programación de un mensaje de error para una rutina matlab:

```
función a = mcd (a , b )
% MCD Máximo común divisor
% mcd (a,b) es el máximo común divisor de a y b no nulos a la vez.

a = round ( abs ( a )); b = round ( abs ( b ) ) ;
if a = = 0 & b = = 0
error ( ' El mcd no está definido cuando ambos números son
nulos ' )
        else
                while b - = 0
                        r = rem ( a, b );
                        a = b ;
                        b = r ;
                end
end
```

Notación de Dos Puntos (:)

La notación de dos puntos (:) que introduce matlab para programar es con el fin de hacer un mejor manejo de ciclos con incrementos variables para generación de vectores y matrices, de esta forma matlab reduce considerablemente el uso de loops y hace que las instrucciones sean más simples y legibles.

sintaxis

numero base : incremento : numero limite

ejemplo

0.2 : 0.2 : 2

dará como resultado

[0.2 0.4 0.6 0.8 1.0 1.2]

los números a trabajarse con notación de dos puntos no deben ser necesariamente enteros ni los incrementos tampoco.

ejemplo

5 : -1 : 1

se genera el vector

[5 4 3 2 1]

La notación de dos puntos permite :

Acceder a submatrices

ejemplo

A (1 : 4 , 3)

es el vector columna con los cuatro primeros datos de la tercera columna de A.

Índices vectores arbitrarios

ejemplo

A(: ,[2 4])

vectores de la segunda y cuarta columna de la matriz A.

Asignación

ejemplo

A (: , [2 4 5]) = B (: , 1: 3)

reemplaza las columnas 2,4 y 5 de A por las tres primeras de B, se despliega y se asigna en pantalla a la matriz A actualizada.

Multiplicación por la derecha

ejemplo

A (: , [2, 4]) = A (: , [2, 4] * [1 2 ; 3 4]

las columnas 2 y 4 se multiplican por la derecha por una matriz m*n = 2*2, se asigna y se despliega en pantalla la matriz A actualizada.

Strings: Texto, Mensajes de Error, Input

Los strings o cadenas de caracteres en matlab son tratados como filas sencillas, los caracteres que componen la cadena son guardados en vectores caracter por caracter por tal razón dichos caracteres son almacenados en su respectivo código ASCII.

setstr	para ver su representación en código ASCII
disp	despliega el texto en pantalla
sprintf	convierte números a su representación en matlab
num2str	convierte un numero x en un representación t con cuatro dígitos de precisión
int2str	convierte un entero a una cadena con formato entero

Las cadenas de texto se introducen en matlab entre comillas simples

ejemplo

s = ' cadena de caracteres '

asigna la cadena de texto a la variable s

Opcionalmente se puede utilizar "disp"

ejemplo

disp (' cadena de caracteres ')

aquí no hay asignación solo se despliega en pantalla el letrero.

Para mensajes de error "error" despliega los mensaje en la ejecución de un archivo .m.

ejemplo

error (' dato no numérico ')

Por otra parte es posible utilizar "break " para terminar la ejecución de un ciclo while, para evitar errores de ciclos infinitos, en estructuras fuera de control y además se puede utilizar " return " para trabajar en funciones m., donde causará un regreso normal, al archivo de trabajo actual el cual haya llamado a dicha función al terminar de ejecutar las instrucciones contenidas en la función

Para trabajar de una forma interactiva con archivos .m se utiliza la función input

ejemplo

num = input (' introduce un número ')

el string entre comillas se despliega en pantalla, y la ejecución del archivo m. que contiene la instrucción se detiene, hasta que el usuario introduce datos, y pulsa intro, es cuando el compilador matlab asigna el dato introducido desde el teclado a la variable num, y continua la ejecución del archivo .m

Normalmente al trabajar archivos m. al momento de su ejecución, los comandos en el archivo no se despliegan en pantalla, por tal razón no es posible ver que está pasando con el archivo, sino hasta el final cuando proporciona el resultado de la salida de la ejecución, sin embargo es posible usar el comando llamado " echo " para ver la ejecución de archivos m., este comando puede ser usado como un debug, o para generar archivos demostrativos (demos).

En forma similar a input, pero con más potencia es posible utilizar keyboard, esta función llama a introducir datos desde el teclado, este comando se utiliza para modificar el valor de variables durante la ejecución de los archivos .m

Un comando llamado " pause " produce un procedimiento para parar la ejecución de un archivo y esperar hasta que el usuario pulse una tecla para continuar.

ejemplo

Pause (n)

detiene la ejecución del programa para n segundos antes de continuar.

Salida Format

Hasta ahora sólo hemos trabajado con entrada de datos y procesamientos de datos, sin embargo la salida de datos es muy importante para análisis e interpretación de resultados, es conveniente que la salida de un proceso sea lo más clara posible para poder interpretarla correctamente, es por eso que resulta conveniente tratar la salida formateada.

Los resultados de las asignaciones de algunas instrucciones matlab son desplegadas en la pantalla, como una asignación especificada a una variable, el formato numérico desplegado puede ser controlado usando los comandos de format, los efectos de format son solamente considerados para la impresión de matrices. (usualmente matlab realiza todos los cálculos en doble precisión).

Si todos los elementos de una matriz son enteros (como hasta ahora), la matriz se imprime con un formato que no cosidera un punto decimal

ejemplo

x = [-1 0 1]

su resultado es

x =

-1 0 1

Ahora si uno de los elementos de la matriz no es exactamente entero, (lo cual es probable) entonces existe una serie de posibles formatos de salida que pueden ser utilizados. el formato por default es el denominado formato short, el cual contiene 5 dígitos decimales significativos, los restantes formatos contienen mas dígitos significativos o su notación científica.

ejemplo

x = [4/3 1.2345e-6]

Los posibles formatos de salida para el vector x son:

format short	1.3333	0.0000
format short e	1.3333E+000	1.2345E-006
format long	1.333333333333338	0.000001234500000
format long e	1.333333333333338E+000	1.234500000000003E-006
format hex	3FF5555555555555	3EB4B6231ABFD271
format +	+	+

Para los formatos cortos el elemento más grande en una matriz es 1000 y el más pequeño es 0.001.

Para el formato + despliega en forma compacta los caracteres +, - y blanco se imprimen para valores en la matriz positivos, negativos, y ceros.

Una vez ordenado un formato, este permanece hasta que se ordena un cambio, la orden:

format compact

evitará la mayor parte de las líneas en blanco, con lo que se puede mostrar más información en pantalla y ademas actúa en forma independiente a los formatos anteriores.

Gráficos: Plot, Print, Meta, Malla, Mesh

Otra herramienta importante que proporciona matlab es la producción de gráficos a partir de instrucciones sencillas, así como una variedad de opciones gráficas.

Matlab puede producir gráficos planos y gráficos de malla de superficies tridimensionales.

Gráficos planos:

La instrucción plot crea gráficos en el plano XY: si x e y son vectores de la misma longitud, la orden

plot (x, y)

accede a la pantalla gráfica y realiza un gráfico plano de elementos de x contra los elementos de y.

El área de trabajo de matlab se divide en dos partes principales, la alfanumérica, que es en la que hemos trabajado hasta ahora las operaciones de vectores, matrices, funciones y archivos m., y la pantalla gráfica, que es la ventana donde se proyectan los gráficos diseñados.

Generalmente las gráficas serán llamadas desde la pantalla alfanumérica por medio de órdenes para hacer gráficos en la línea de comandos o bien desde funciones o archivos .m, entonces se produce un cambio de la ventana alfanumérica a la ventana gráfica, para regresar a la ventana numérica basta con pulsar cualquier tecla, en contra parte se usa la orden shg (show graph) , para acceder a la ventana gráfica.

La instrucción Grid efectúa un cuadriculado en el gráfico.

Al trabajar con gráficos puede incluirse:

title titulo del gráfico

xlabel comentario en el eje x

ylabel comentario en el eje y

gtext texto posicionado interactivamente

text texto posicionado mediante coordenadas

ejemplo

title (' primer gráfico ')

proporciona un titulo gráfico

ejemplo

gtext (' posición ')

permite posicionar una cruz en el gráfico con el mouse (ratón) o con las flechas de dirección del teclado, donde se situara el texto cuando se pulse cualquier tecla.

Por default los ejes para gráficos se auto escalan, para evitar que esto suceda se usa el comando axis.

ejemplo

si

c = [xmin, xmax, ymin, ymax]

es un vector con cuatro elementos variables entonces

axis (c)

establece el escalado de ejes a los límites deseados en las variables xmin, xmax, ymin, ymax.

Introduciendo axis de nuevo volvemos a auto escalado, para asegurar que se use la misma escala en ambos ejes se usa:

axis ('square')

La opción de sobreponer una gráfica sobre otra , se obtiene con el comando hold el cual congela la ventana gráfica actual de forma que los gráficos posteriores se sobreponen en ella, al escribir hold de desactiva el congelamiento.

Existen opciones para tipos de linea y de puntos en los diseños de gráficas:

Tipos de línea

sólido	(-)
a trazos	(--)
puntos	(:)
punto y trazo	(.-)

Tipos de puntos

punto	(.)
mas	(+)
asterisco	(*)
círculo	(°)
equis	(x)

Gráficas hardcopy

Para obtener un hardcopy de la ventana gráfica se usa el comando print, este enviará una copia de alta resolución de la ventana gráfica a la impresora, situando el gráfico en la mitad superior de la pagina impresa.

Para imprimir dos gráficos por página es necesario utilizar el comando meta el cual genera un archivo donde se van guardando los gráficos seleccionados en el comando meta

sintaxis

meta nombre-de-archivo.met

posteriores meta (sin nombre)

El comando prtscr envia la ventana gráfica actual a la impresora , pero en general este resultado arroja una gráfica de baja resolución, debido a que los pixeles de la pantalla son transformados a pixeles de la impresora

Agregarán las nuevas gráficas al archivo meta anterior

Los gráficos de malla de superficies tridimensionales se obtienen al utilizar el comando mesh.

sintaxis

mesh (A)

donde A es una matriz,

la instrucción mesh creara un gráfico tridimensional en perspectiva de la matriz A, la superficie de malla está definida por las coordenadas de A de los puntos sobre un cuadriculado rectangular en el plano xy .

ejemplo

graficar z = e ^ (-x^2 - y^2), sobre el área [-2, 2] * [-2, 2]

```
xx = -2:.1:2;
yy =  xx;
[x, y] = meshdom( xx, yy );
a = exp ( -x.^2 -y.^2 );
mesh ( a )
```

donde meshdom es una función que crea una matriz x, en la que cada fila es igual a xx, y de igual forma una matriz y, con todas sus columnas iguales a yy, de esta manera genera el dominio en los ejes coordenados para posicionar la gráfica.

Ayuda (Help)

Matlab presenta un comando help (ayuda) el cual facilita considerablemente la obtención de información en linea de comandos para los diferentes tópicos de matlab.

sintaxis

help nombre-del-topico

El comando help especifica el tópico señalado a continuación de help.

comandos que complementan a help

who	lista las variables actuales en memoria
what	muestra un directorio listando los archivos .m en disco
matlabpath	es una operación del medio ambiente del sistema, la cual ha sido predefinida para usuarios matlab, pero esta se puede modificar si así se desea, al cambiar el directorio especificado. ejemplo el matlabpath para ms-dos esta especificado de la siguiente manera set matlabpath = c: \ wmatlab;c:\matlab\control

Por otro lado existe una gran cantidad de funciones (tópicos) integrados a matlab, en particular, varias cajas de herramientas (tool box) de funciones para aplicaciones especificas, como proceso de señales

teoría de control

control robusto

identificación de sistemas

optimizacion

splines

quimiometria

m-analisis y síntesis

identificación

redes neurales

a continuación se listan las funciones más importantes, así como operadores y demás caracteres que constituyen el lenguaje matlab y a los cuales se puede tener acceso mediante el comando help para mayor información, los cuales se pueden clasificar de la siguiente manera:

generales

operadores matriciales

operadores puntuales

operadores lógicos y relaciónales

caracteres especiales

valores especiales

archivos de disco

matrices especiales

manipulación de matrices

funciones lógicas y relaciónales

control de flujo

programación y archivos .m

texto y cadenas

ventana alfanumérica

gráficos

anotación gráfica

control de la ventana gráfica

impresión de gráficos

funciones elementales

funciones trigonométricas

funciones especiales

descomposición y factrorización

condicionamiento de matrices

funciones matriciales elementales

polinomios

análisis de datos por columnas

proceso de señales

integración numérica

solución de ecuaciones diferenciales

ecuaciones no lineales y optimización

interpolación

GENERAL

help	ayuda
demo	demostraciones
who	muestra variables en memoria
what	muestra archivos .m en disco
size	numero de filas y columnas
lenght	longitud de un vector
clear	limpia el espacio de trabajo
computer	tipo de computadora
^C	interrupción local
exit	salida de MATLAB
quit	lo mismo que exit

OPERADORES MATRICIALES

+	suma
-	resta
*	multiplicación
/	división por la derecha
\	división por la izquierda
^	potenciación
'	conjugada traspuesta

OPERADORES PUNTUALES

+	suma
-	resta
.*	multiplicación
./	división por la derecha
.\	división por la izquierda
.^	potenciación
.'	traspuesta

OPERADORES LOGICOS Y RELACIONALES

<	menor que
<=	menor o igual que
>	mayor que
>=	mayor o igual que
==	igual
~=	no igual
&	y

|	o
-	no

CARACTERES ESPECIALES

=	instrucción de asignación
[	usado para formar vectores y matrices
]	usado para cerrar la formación de vectores y matrices
(	precedencia aritmética
)	usado para cerrar la precedencia aritmética
.	punto decimal
...	instrucción actual continúa en la siguiente linea
,	separa índices y argumentos de función
;	fin de linea y suprime el eco
%	inicia cadenas de comentarios
:	indexación, generación de vectores (ciclos con incremento
!	ejecuta instrucción del sistema operativo (interrupciones)

VALORES ESPECIALES

ans	respuesta cuando no hay asignación de variable en operaciones
eps	precisión
pi	constante pi = 3.1416
i, j	-1 ^ (1/2)
inf	infinito
NaN	No Número (Not-a-Number)
clock	reloj
date	fecha
flops	número de operaciones en la sesión matlab
nargin	número de argumentos de entrada de una función
nargout	número de argumentos de salida de una función

ARCHIVOS DE DISCO

chdir	cambiar de directorio
delete	borrar archivo
diary	diario de la sesión
dir	directorio de archivos en disco
load	cargar variables de un archivo
save	guardar variables en un archivo
type	mostrar función o archivo

what	mostrar archivos .m en el disco
fprintf	escribir en un archivo
pack	compactar memoria vía save
translate	traslado de datos

MATRICES ESPECIALES

compan	compañera
diag	diagonal
eye	identidad
gallery	esotérica
hadamard	hadamard
hankel	hankel
hilb	Hilbert
invhilb	inversa de Hilbert
linspace	vectores igualmente espaciados
logspace	vectores logarítmicamente espaciados
magic	mágica cuadrada
meshdom	dominio para puntos de malla
ones	constante
pascal	Pascal
rand	elementos aleatorios
teoplitz	Teoplitz
vander	Vandermonde
zeros	cero

MANIPULACION DE MATRICES

rot90	rotación
fliplr	invierte el orden de las columnas
flipud	invierte el orden de las filas
diag	diagonal
tril	parte triangular superior
triu	parte triangular inferior
reshape	reordena una matriz en otra
	traspuesta
.'	traspuesta
:	convierte una matriz en una columna simple ; A (:)

FUNCIONES LOGICAS Y RELACIONALES

any	condiciones lógicas, regresa 1 si algunos de los elementos de un vector x son > 0
all	condiciones lógicas, regresa 1 si todos los elementos de un vector son > 0
find	encuentra índices de valores lógicos
isnan	detectaNaNs
finite	detecta infinitos
isempty	detecta matrices vacías
isstr	detecta variables de cadena

strcomp	compara variables de cadena

CONTROL DE CICLOS (LOOPS)

if	ejecuta instrucciones condicionalmente
elseif	opción del if en caso de que la condición no sea verdadera
end	termina un if, for, while
for	repite instrucciones mientras una sentencia lógica sea verdadera
while	repite instrucciones mientras una sentencia lógica sea verdadera
break	sale de los ciclos for y while
return	salida desde funciones
pause	pause hasta que se pulse una tecla

PROGRAMACION Y ARCHIVOS .M

input	obtiene números desde el teclado
keyboard	llamada al teclado como si fuera un archivo .m
error	muestra un mensaje de error
function	define función
eval	evalúa un texto en variables
feval	evalúa función dada por una cadena

echo	permite mostrar las instrucciones en pantalla
exist	comprueba si las variables existen
casesen	sensibilidad a las mayúsculas
global	define variables globales
startup	archivo de inicialización
getenv	accede a una variable de entorno
menu	genera un menú
etime	tiempo gastado

TEXTO Y CADENAS

abs	convierte cadena en valores ASCII
eval	evalúa texto como instrucciones
num2str	convierte números en cadenas
int2str	convierte enteros en cadenas
setstr	indicador de cadenas
sprintf	convierte números en cadenas
isstr	detecta variables de cadena
strcomp	compara variables de cadena
hex2num	convierte cadenas hexadecimales en números

VENTANA ALFANUMERICA

clc	limpia pantalla
home	mueve cursor al comienzo
format	establece el formato de salida
disp	muestra matriz o texto
fprintf	imprime número formateado
echo	permite se desplieguen las instrucciones en pantalla

GRAFICOS

plot	gráfico lineal en el plano XY
loglog	gráfico logarítmico en el plano XY
semilogx	gráfico semilogarítmico
semilogy	gráfico semilogarítmico
polar	gráfico polar
mesh	superficie de malla tridimensional
contour	plano de contornos
meshdom	dominio para gráficos de superficie
bar	gráficos de barras
stairs	gráficos de escaleras
errobar	añade barras de errores

ANOTACION GRAFICA

title	título
xlabel	anotación en el eje x
ylabel	anotación en el eje y
grid	cuadrícula el gráfico
text	posiciona un texto arbitrariamente
gtext	posiciona un texto con el mouse (ratón)
ginput	input gráfico

CONTROL DE LA VENTANA GRAFICA

axis	escalado manual de ejes
hold	mantiene gráfico en pantalla
shg	muestra la pantalla gráfica
clg	limpia la pantalla gráfica
subplot	divide la pantalla gráfica

IMPRESION DE GRAFICOS

print	envía gráfico a impresora
prtsc	impresión desde el teclado
meta	archivo de gráficos

FUNCIONES ELEMENTALES

abs	modulo complejo
angle	argumento complejo

sqrt	raíz cuadrada
real	parte real
imag	parte imaginaria
conj	conjugado complejo
round	redondeo al entero más cercano
fix	redondeo hacia cero
floor	redondeo hacia - infinito
ceil	redondeo hacia infinito
sign	función signo
rem	resto
exp	exponencial base e
log	logaritmo natural
log10	logaritmo base 10

FUNCIONES TRIGONOMETRICAS

sin	seno
cos	coseno
tan	tangente
asin	arco seno
acos	arco coseno
atan	arco tangente
atan2	arco tangente de x/y
sinh	seno hiperbólico
cosh	coseno hiperbólico
tanh	tangente hiperbólica
asinh	arco seno hiperbólico

acosh	arco coseno hiperbólico
atanh	arco tangente hiperbólica

FUNCIONES ESPECIALES

bessel	función de Bessel
gamma	función Gamma
rat	aproximación racional
erf	función de error
inverf	inversa de la función error
ellipk	integral completa elíptica de primera especie
ellipj	integral elíptica de jacobi

DESCOMPOSICIONES Y FACTORIZACIONES

balance	forma equilibrada
backsub	sustitución regresiva
cdf2rdf	convierte diagonales complejas en diagonales reales
chol	factorizacion de cholesky
eig	auto valores y auto vectores (valores propios)
hess	forma de Hessenberg
inv	inversa
lu	factores de eliminación gaussiana

nnls	mínimos cuadrados con restricciones
null	base orto normal del núcleo
orth	base orto normal de la imagen
pinv	pseudo inversa (inversa generalizada)
qr	factorizacion QR
qz	algoritmo QZ
rref	forma escalonada reducida por filas
schur	descomposición de schur
svd	descomposición en valores singulares

CONDICIONAMIENTO DE MATRICES

cond	número de condición en la norma 2
norm	norma 1, norma 2, norma de Frobenius, norma infinita
rank	rango
rcond	estimación de la condición (inverso)

FUNCIONES MATRICIALES ELEMENTALES

expm	matriz exponencial
logm	matriz logaritmo

sqrtm	matriz raíz cuadrada
funm	función arbitraria de matriz
poly	polinomio característico
det	determinante
trace	traza
kron	producto tensorial de kronecker

POLINOMIOS

poly	polinomio característico
roots	raíces de polinomios - método de raiz compañera
roots1	raíces de polinomios - método de Laguerre
polyval	evaluación de polinomios
polyvalm	evaluación polinomio matricial
conv	multiplicación
deconv	división
residue	desarrollo en fracciones parciales
polyfit	ajuste por un polinomio

ANALISIS DE DATOS POR COLUMNAS

max	valor máximo
min	valor mínimo
mean	valor medio
median	mediana
std	desviación típica

sort	ordenación
sum	suma de elementos
prod	producto de elementos
cumsum	suma acumulativa de elementos
cumprod	producto acumulativo de elementos
diff	derivadas aproximadas
hist	histogramas
corrcoef	coeficientes de correlación
cov	matriz de covarianza
cplxpair	reordena en pares complejos
corr	matriz de correlación

PROCESO DE SEÑALES

abs	módulo complejo
angle	argumento complejo
conv	convolución
corrcoef	coeficientes de correlación
cov	covarianza
deconv	deconvolución
fft	transformada rápida de Fourier
fft2	FFT 2 - dimensiones
ifft	FFT inversa
ifft2	FFT inversa 2 - dimensiones

fftshift	cambia las dos mitades de un vector
dft	transformada discreta de Fourier
filter	implementacion directa de filtro
freqz	respuesta digital de frecuencia
freqs	frecuencia de respuesta análoga
xcorr	función de cruce de correlación
xcorr2	correlación de cruse en 2 dimensiones

INTEGRACION NUMERICA

quad	función de interacción numérica
quad8	función de integración numérica

SOLUCION DE ECUACIONES DIFERENCIALES

ode23	método de Runge - Kutta de orden 2/3
ode45	método de Runge - Kutta - Fehlberg de orden 4/5

ECUACIONES NO LINEALES Y OPTIMIZACION

fmin	mínimo de una función de una variable
fmins	mínimo de una función de varias variables

fsolve	solución de un sistema de ecuaciones no lineales
fzero	cero de una función de una variable

INTERPOLACION

spline	spline cúbico
table1	genera tablas 1- dimensión
table2	genera tablas 2 - dimensiones

MANUAL TECNICO

DE MATLAB FOR WINDOWS

UNIVERSIDAD AUTONOMA AGRARIA ANTONIO NARRO

MANUAL TECNICO DE MATLAB

FOR WINDOWS

MATLAB FOR WINDOWS

José Trinidad Rodríguez Estrada

INTRODUCCION

El análisis estadístico de matrices requiere de complicados cálculos numéricos.

Para facilitar el trabajo de cálculos numéricos, se hace imprescindible el uso de los recursos de cómputo electrónico, aprovechando la abundante disposición de software que se tiene hoy en día.

En el mercado de software se encuentran sistemas de análisis estadístico que corren diferentes equipos de cómputo: mainframes, minicomputadoras y micro computadora, y que pueden usarse a través de menús o comandos.

Uno de estos lenguajes que facilita la operación de matrices es Matlab, que al interactuar con Windows, aumenta en forma considerable la ayuda tanto para programar rutinas específicas, como para tratar directamente matrices.

El presente Manual Técnico de Matlab for Windows, además de ser un complemento ilustrativo del Manual de Usuario, tiene como objetivo presentar una guía sencilla para introducir al usuario en el Matlab for Windows, para despertar el interés en aquellos usuarios que apenas se inician en el uso del software para análisis de matrices.

El PRINCIPIO

MATLAB for WINDOWS es un sistema que funciona a través de ventanas (windows) y trabaja en línea de comandos o bien con archivos . m, cuando trabajamos en un ambiente interactivo como windows es posible elegir opciones mediante el uso del mouse, o usando las teclas para el movimiento del cursor y presionando la tecla < enter >

Una vez dentro de windows se busca en el administrador de archivos el grupo en el cual se encuentra Matlab, (generalmente se encuentra en el grupo de aplicaciones) , después de abrir la ventana de aplicaciones se selecciona el icono correspondiente a

matlab y el control de ejecución de programa se traslada de windows a matlab, dentro del ambiente matlab el modo comando opera a partir de la línea de comandos

EJECUCIÓN DEL PROGRAMA

Si MATLAB for WINDOWS ha sido instalado correctamente en el disco duro de la computadora, se deben seguir los siguientes pasos para empezar a ejecutar el softwer, a partir del indicador C> del sistema operativo.

Teclear win < Enter >	ejecuta el programa WINDOWS
Seleccionar con el mouse (o con las teclas de movimiento cursor) en el administrador de archivos de windows el grupo de aplicaciones.	carga en memoria del aplicaciones
Seleccionar el icono de matlab	ejecuta el programa MATLAB

Después del despliegue del logotipo y otros mensajes se desplegará la línea de comandos Matlab:

>>

y listo ya puede comenzar a introducir vectores y matrices (ver el manual de usuario adjunto).

DOCUMENTACION DEL SISTEMA

OPEMAT

UNIVERSIDAD AUTONOMA AGRARIA ANTONIO NARRO

EN QUE CONSISTE OPEMAT

MATLAB for WINDOWS

José Trinidad Rodríguez Estrada

Objetivo General

Aplicar los conceptos fundamentales de matlab para el diseño y construcción de un sistema que resuelva problemas de operaciones matriciales.

Conocimientos del área de especialidad:

Lógica computacional

Lenguaje computacional Matlab básico

Lógica Aplicada al lenguaje matlab

Agrupación de datos en vectores y matrices

Obtención de reportes

Introducción al Análisis y Diseño de Sistemas, con comandos avanzados

Características terminales:

Diseño de el diagrama de flujo del sistema opemat

Diseño de el Algoritmo del sistema opemat

Diseño estructurado del codigo fuente del sistema computacional opemat

Almacenamiento y recuperacion de informacion grabada en unidades de

Almacenamiento secundario

Obtención de reportes

Creación de procedimientos a nivel sistema operativo

Manejo de funciones predefinida en matlab

Cálculo de operaciones matriciales

ALGORITMO

[A1] Inicio

[A2] Creación de presentación

" opermat "

" álgebra de matrices"

[A3] Manejo de archivos

Nuevo

Abrir

Guardar

Imprimir

salir

[A4] Creación de Menú de opciones de operación con matrices

Transpuesta

Adición

Multiplicación

División

Elevación a Potencia

Inversión

[A5] Creación de Menu de Funciones Trascendentales con matrices

Exponenciación

Raiz cuadrada

Logaritmos

[A6] Creación de Menú de Funciones Matemáticas Elementales

módulo complejo

raiz cuadrada

parte real

parte imaginaria

conjugada compleja

redondeo al entero más cercano

redondeo a cero

redondeo a -infinito

redondeo a +infinito

función signo

resto

seno

coseno

tangente

arco seno

arco coseno

arco tangente

arcotangente de x/y

seno hiperbólico

coseno hiperbólico

tangente hiperbólica

exponencial base e

logaritmo natural

logaritmo base 10

funciones Bessel

funciones Gamma

aproximación racional

[A7] Creación del Menu de Analisis de Datos

valor máximo

valor mínimo

valor medio

valor de la mediana

ordenación

suma de elementos

producto de elementos

suma acumulativa de elementos

producto acumulativo de elementos

derivadas aproximadas

histogramas

generación de tablas de 1 dimensión

matriz de correlación

matriz de covarianza

condiciones lógicas

encontrar arreglos de índices de valores lógicos

[A8] Creación de Menu de Funciones de matrices

factorización triangular

factorización ortogonal

descomposición de valores propios

descomposición de valores singulares

[A9] Creación de Menu de Polinomios

[A10] Creación de Menu de Gaficos

tipos

colores

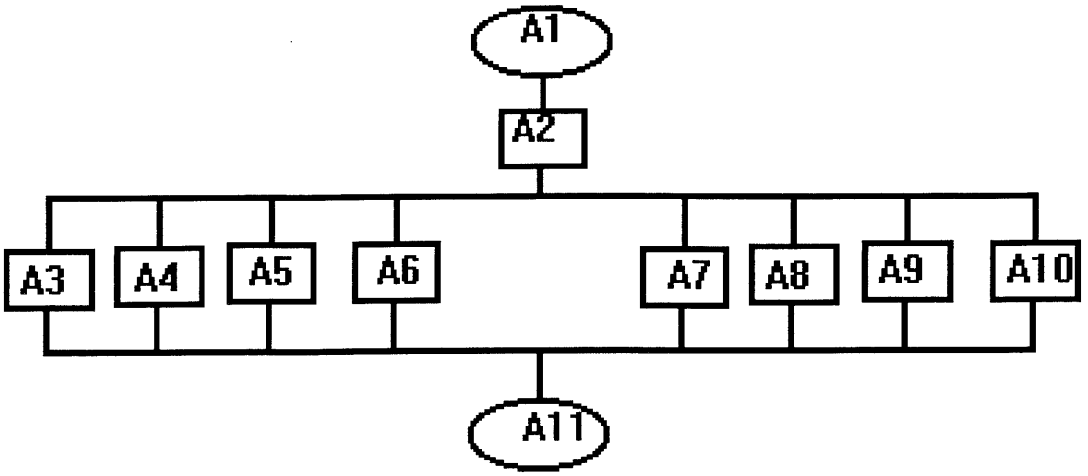
barras

3 - dimensiones

impresión

[A11] Fin

Diagrama de Flujo



Listado del codigo fuente

```
function ope
labels = str2mat(...
  'Transpuesta',...
  'Adicion',...
  'Multiplicacion',...
  'Division',...
  'Elevacion a Potencia',...
  'Inversion');

end
callbacks = [ ...
  'tra          '
  'adi          '
  'mul           '
  'div          '
  'ele          '
  'inverc       '];
choices('OPE', 'OPERACIONES CON MATRICES', labels, callbacks);

disp('matriz introducida')
home
A
disp('matriz transpuesta')
B=A'
plot(B)
disp('presione cualquier tecla para continuar...')
pause

disp('matriz introducida 1')
home
A
disp('matriz introducida 2')
B
disp('suma de las matrices introducidas ')
C=A+B
plot(C)
disp('presione cualquier tecla para continuar...')
pause
```

```
disp('matriz introducida 1')
home
A
disp('matriz introducida 2')
B
disp('multiplicacion de las matrices introducidas ')
C=A*B
plot(C)
Disp('presione cualquier tecla para continuar...')
pause

disp('matriz introducida 1')
home
A
disp('matriz introducida 2')
B
disp('division de las matriz 1 / la matriz 2 ')
C=A/B
plot(C)
disp('presione cualquier tecla para continuar...')
pause

disp('matriz introducida 1')
home
A
disp('potencia introducida')
B
disp('suma de las matrices introducidas ')
C=A^B
plot(C)
disp('presione cualquier tecla para continuar...')
pause
disp('matriz introducida 1')
home
A
disp('inversion')
B=inv(A)
plot(B)
disp('presione cualquier tecla para continuar...')
pause


 function funtra
 labels = str2mat(...
```

```
  'Exponencicion',...
  'Raiz Cuadrada',...
  'Logaritmo',...
  'Logaritmo base 10');
c = computer;
if c(1:2) ~= 'PC' & c(1:2) ~= 'MA'
  label = str2mat(labels, ...
    'Hurricane Andrew');
end
callbacks = [ ...
  'exp          '
  'raicua       '
  'log          '
  'logbas10     '];
choices('FUNTRA', 'FUNCIONES TRASCENDENTALES', labels, callbacks);
disp('matriz introducida')
home
A
disp('matriz Exponencial')
B=exp(A)
plot(B)
disp('presione cualquier tecla para continuar...')
pause
disp('matriz introducida')
home
A
disp('matriz Raiz Cuadrada')
B=sqrt(A)
plot(B)
disp('presione cualquier tecla para continuar...')
pause

disp('matriz introducida')
home
A
disp('matriz Raiz Cuadrada')
B=sqrt(A)
plot(B)
disp('presione cualquier tecla para continuar...')
pause

disp('matriz introducida')
home
A
```

```
disp('matriz Logaritmo')
B=log(A)
plot(B)
disp('presione cualquier tecla para continuar...')
pause

disp('matriz introducida')
home
A
disp('matriz Logaritmo base 10')
B=log10(A)
plot(B)
disp('presione cualquier tecla para continuar...')
pause

 function funmat
labels = str2mat(...
  'Modulo Complejo',...
  'Parte Real',...
  'Parte Imaginaria',...
  'Conjugada Compleja',...
  'Funcion Signo',...
  'Resto',...
  'Funciones Bessel',...
  'Aproximacion Racional');
c = computer;
if c(1:2) ~= 'PC' & c(1:2) ~= 'MA'
  label = str2mat(labels, ...
    'Hurricane Andrew');
end
callbacks = [ ...
  'modcom       '
  'parrea       '
  'parima       '
  'concom       '
  'funsig       '
  'res          '
  'funbes       '
  'aprrac       '];
choices('FUNMAT', 'FUNCIONES MATEMATICAS ELEMENTALES', labels,
callbacks);


disp('matriz introducida')
```

```
home
A
disp('matriz Modulo Complejo')
B=abs(A)
plot(B)
disp('presione cualquier tecla para continuar...')
pause

disp('matriz introducida')
home
A
disp('matriz Parte Real')
B=real(A)
plot(B)
disp('presione cualquier tecla para continuar...')
pause

disp('matriz introducida')
home
A
disp('matriz Parte Imaginaria')
B=imag(A)
plot(B)
disp('presione cualquier tecla para continuar...')
pause

disp('matriz introducida')
home
A
disp('matriz Conjugada Compleja')
B=conj(A)
plot(B)
disp('presione cualquier tecla para continuar...')
pause
disp('matriz introducida')
home
A
disp('matriz Resto')
B=rem(A)
plot(B)
disp('presione cualquier tecla para continuar...')
pause

disp('matriz introducida')
```

```
home
A
disp('matriz Funcion Bessel')
B=bessel(A)
plot(B)
disp('presione cualquier tecla para continuar...')
pause

disp('matriz introducida')
home
A
disp('matriz Aproximacion Racional')
B=rat(A)
plot(B)
disp('presione cualquier tecla para continuar...')
pause

 function funred
 labels = str2mat(...
   'Redondeo al Entero mas Cercano',...
   'Redondeo a Cero',...
   'Redondeo -Infinito',...
   'Redondeo +Infinito');
c = computer;
if c(1:2) ~= 'PC' & c(1:2) ~= 'MA'
   label = str2mat(labels, ...
      'Hurricane Andrew');
end
callbacks = [ ...
   'redent         '
   'redcer         '
   'redmeinf       '
   'redmainf       '];
choices('FUNRED', 'FUNCIONES MATEMATICAS DE REDONDEO', labels,
callbacks);

disp('matriz introducida')
home
A
disp('matriz Redondeo a cero')
B=fix(A)
plot(B)
disp('presione cualquier tecla para continuar...')
pause
```

```
disp('matriz introducida')
home
A
disp('matriz Redondeo -Infinito')
B=floor(A)
disp('presione cualquier tecla para continuar...')
pause
disp('matriz introducida')
home
A
disp('matriz Redondeo -Infinito')
B=floor(A)
disp('presione cualquier tecla para continuar...')
pause

 function funtri
labels = str2mat(...
  'Seno',...
  'Coseno',...
  'Tangente',...
  'Arco Seno',...
  'Arco Coseno',...
  'Arco Tangente',...
  'Arcotangente de x/y',...
  'Seno Hiperbolico',...
  'Coseno Hiperbolico',...
  'Tangente Hiperbolica',...
  'Funciones Gamma');
c = computer;
if c(1:2) ~= 'PC' & c(1:2) ~= 'MA'
  label = str2mat(labels, ...
    'Hurricane Andrew');
end
callbacks = [ ...
  'sen          '
  'cose         '
  'tane         '
  'arcsen       '
  'arccos       '
  'arctan       '
  'arctanxy     '
  'senhip       '
  'coship       '
  'tanhip       '
```

```
  'fungam        '];
choices('FUNTRI', 'FUNCIONES MATEMATICAS ELEMENTALES
TRIGONOMETRICAS', labels, callbacks);


disp('matriz introducida')
home
A
disp('matriz Seno')

B=sin(A)
plot(B)
disp('presione cualquier tecla para continuar...')
pause

disp('matriz introducida')
home
A
disp('matriz Coseno')
B=cos(A)
plot(B)
disp('presione cualquier tecla para continuar...')
pause

disp('matriz introducida')
home
A
disp('matriz Tangente')
B=tan(A)
plot(B)
disp('presione cualquier tecla para continuar...')
pause

disp('matriz introducida')
home
A
disp('matriz Arcoseno')
B=asin(A)
plot(B)
disp('presione cualquier tecla para continuar...')
pause

disp('matriz introducida')
home
```

```
A
disp('matriz Arcocoseno')
B=acos(A)
plot(B)
disp('presione cualquier tecla para continuar...')
pause

disp('matriz introducida')
home
A
disp('matriz Arcotangente')
B=atan(A)
plot(B)
disp('presione cualquier tecla para continuar...')
pause

disp('matriz introducida')
home
A
disp('matriz Arcotangente de x/y')
B=atan2(A)
plot(B)
disp('presione cualquier tecla para continuar...')
pause

disp('matriz introducida')
home
A
disp('matriz Arcotangente de x/y')
B=atan2(A)
plot(B)
disp('presione cualquier tecla para continuar...')
pause

disp('matriz introducida')
home
A
disp('matriz Seno Hiperbolico')
B=sinh(A)
plot(B)
disp('presione cualquier tecla para continuar...')
pause

disp('matriz introducida')
```

```
home
A
disp('matriz Coseno Hiperbolico')
B=cosh(A)
plot(B)
disp('presione cualquier tecla para continuar...')
pause

disp('matriz introducida')
home
A
disp('matriz Tangente Hiperbolico')
B=tanh(A)
plot(B)
disp('presione cualquier tecla para continuar...')
pause

disp('matriz introducida')
home
A
disp('matriz Funciones Gamma')
B=gamma(A)
plot(B)
disp('presione cualquier tecla para continuar...')
pause

 function anadati
labels = str2mat(...
  'Valor Maximo',...
  'Valor Minimo',...
  'Ordenacion',...
  'Suma de Elementos',...
  'Producto de Elementos',...
  'Derivadas Aproximadas',...
  'Histogramas');
c = computer;
if c(1:2) ~= 'PC' & c(1:2) ~= 'MA'
  label = str2mat(labels, ...
    'Hurricane Andrew');
end
callbacks = [ ...
  'valmax        '
  'valmin        '
  'ord           '
```

```
  'sumele        '
  'proele        '
  'derapr        '
  'his          ' ];
choices('ANADATI', 'ANALISIS DE DATOS I', labels, callbacks);


disp('matriz introducida')
home
A
disp('Valor Maximo')
B=max(A)
plot(B)
disp('presione cualquier tecla para continuar...')
pause

disp('matriz introducida')
home
A
disp('Valor Minimo')
B=min(A)
plot(B)
disp('presione cualquier tecla para continuar...')
pause

disp('matriz introducida')
home
A
disp('Ordenacion')
B=sort(A)
plot(B)
disp('presione cualquier tecla para continuar...')
pause

disp('matriz introducida')
home
A
disp('Suma de Elementos')
B=sum(A)
plot(B)
disp('presione cualquier tecla para continuar...')
pause

disp('matriz introducida')
```

```
home
A
disp('Producto de Elementos')
B=prod(A)
disp('presione cualquier tecla para continuar...')
pause


disp('matriz introducida')
home
A
disp('Derivadas Aproximadas')
B=diff(A)
plot(B)
disp('presione cualquier tecla para continuar...')
pause

disp('matriz introducida')
home
A
disp('Histogramas')
B=hist(A)
plot(B)
disp('presione cualquier tecla para continuar...')
pause

 function anadatii
labels = str2mat(...
  'Valor Medio',...
  'Valor de la Mediana',...
  'Generacion de Tablas de 1 dimension',...
  'Matriz de Correlacion',...
  'Matriz de Covarianza');
c = computer;
if c(1:2) ~= 'PC' & c(1:2) ~= 'MA'
  label = str2mat(labels, ...
    'Hurricane Andrew');
end
callbacks = [ ...
  'valmed      '
  'valmedi     '
  'gentab      '
  'matcor      '
  'matcov      '];
```

```
choices('ANADATII', 'ANALISIS DE DATOS II', labels, callbacks);

disp('matriz introducida')
home
A
disp('Valor Medio')
B=mean(A)
plot(B)
disp('presione cualquier tecla para continuar...')
pause

disp('matriz introducida')
home
A
disp('Valor de la Mediana')
B=median(A)
plot(B)
disp('presione cualquier tecla para continuar...')
pause

disp('matriz introducida')
home
A
disp('Generacion de Tablas de 1 Dimension')
B=table1(A)
plot(B)
disp('presione cualquier tecla para continuar...')
pause

disp('matriz introducida')
home
A
disp('Matriz de Correlacion')
B=corr(A)
plot(B)
disp('presione cualquier tecla para continuar...')
pause

disp('matriz introducida')
home
A
disp('Matriz de Covarianza')
B=cov(A)
plot(B)
```

```
disp('presione cualquier tecla para continuar...')
pause

 function funmat
labels = str2mat(...
  'Modulo Complejo',...
  'Parte Real',...
  'Parte Imaginaria',...
  'Conjugada Compleja',...
  'Funcion Signo',...
  'Resto',...
  'Funciones Bessel',...
  'Aproximacion Racional');
c = computer;
if c(1:2) ~= 'PC' & c(1:2) ~= 'MA'
  label = str2mat(labels, ...
    'Hurricane Andrew');
end
callbacks = [ ...
  'modcom       '
  'parrea       '
  'parima       '
  'concom       '
  'funsig       '
  'res          '
  'funbes       '
  'aprrac       '];
choices('FUNMAT', 'FUNCIONES MATEMATICAS ELEMENTALES', labels,
callbacks);


disp('matriz introducida')
home
A
disp('Factorizacion Triangular')
B=qr(A)
plot(B)
disp('presione cualquier tecla para continuar...')
pause

disp('matriz introducida')
home
A
disp('Factorizacion Ortogonal')
```

```
B=orth(A)
plot(B)
disp('presione cualquier tecla para continuar...')
pause

disp('matriz introducida')
home
A
disp('Descomposicion de Valores Propios')
B=eig(A)
plot(B)
disp('presione cualquier tecla para continuar...')
pause

disp('matriz introducida')
home
A
disp('Descomposicion de Valores Singulares')
B=svd(A)
plot(B)
disp('presione cualquier tecla para continuar...')
Pause

 function pol
labels = str2mat(...
  'Polinomio Caracteristico',...
  'Raices de Polinomios',...
  'Raices de Polinomios Laguerre',...
  'Evaluacion de polinomios',...
  'Evalucion de Polinomio Matricial',...
  'Multiplicacion',...
  'Division',...
  'Desarrollo en Fracciones Parciales',...
  'Ajuste por un Polinomio');
c = computer;
if c(1:2) ~= 'PC' & c(1:2) ~= 'MA'
  label = str2mat(labels, ...
    'Hurricane Andrew');
end
callbacks = [ ...
  'polcar       '
  'raipol       '
  'railag       '
  'evapol       '
```

```
  'evamat        '
  'multi         '
  'divi          '
  'desfra        '
  'ajupol        '];
choices('POL', 'POLINOMIOS', labels, callbacks);


disp('matriz introducida')
home
A
disp('Polinomio Caracteristico')
B=poly(A)
plot(B)
disp('presione cualquier tecla para continuar...')
pause

disp('matriz introducida')
home
A
disp('Raices de Poloinomios')
B=roots(A)
plot(B)
disp('presione cualquier tecla para continuar...')
pause

disp('matriz introducida')
home
A
disp('Raices de Poloinomios Laguerre')
B=roots1(A)
plot(B)
disp('presione cualquier tecla para continuar...')
pause

disp('matriz introducida')
home
A
disp('Evaluacion Poloinomio Matricial')
B=polyvalm(A)
plot(B)
disp('presione cualquier tecla para continuar...')
pause
```

```
disp('matriz introducida')
home
A
disp('Evaluacion de Polinomio Matricial')
B=polyvalm(A)
plot(B)
disp('presione cualquier tecla para continuar...')
pause

disp('matriz introducida')
home
A
disp('Multiplicacion')
B=conv(A)
plot(B)
disp('presione cualquier tecla para continuar...')
pause

disp('matriz introducida')
home
A
disp('Division')
B=deconv(A)
plot(B)
disp('presione cualquier tecla para continuar...')
pause

disp('matriz introducida')
home
A
disp('Desarrollo en Fracciones Parciales')
B=residue(A)
plot(B)
disp('presione cualquier tecla para continuar...')
pause


disp('matriz introducida')
home
A
disp('Ajuste por un Polinomio')
B=polyfit(A)
plot(B)
disp('presione cualquier tecla para continuar...')
```

```
pause

 function gra
labels = str2mat(...
  'Grafico Lineal',...
  'Grafico Logaritmico',...
  'Grafico Semilogaritmico x',...
  'Grafico Semilogaritmico y',...
  'Grafico Polar',...
  'Superficie de malla Tridimensional',...
  'Plano de Contornos',...
  'Dominio para Graficos de Superficie',...
  'Grafico de Barras',...
  'Graficos de Escalares',...
  'Añade Barras de errores');
c = computer;
if c(1:2) ~= 'PC' & c(1:2) ~= 'MA'
  label = str2mat(labels, ...
    'Hurricane Andrew');
end
callbacks = [ ...
  'gralin      '
  'gralog      '
  'grasemx     '
  'grasemy     '
  'grapol      '
  'supmal      '
  'placon      '
  'domgra      '
  'grabar      '
  'graesc      '
  'barerr      '];
choices('GRA', 'GRAFICOS', labels, callbacks);


disp('matriz introducida')
home
A
disp('Grafico Lineal')
B=plot(A)
disp('presione cualquier tecla para continuar...')
pause

disp('matriz introducida')
```

```
home
A
disp('Grafico Logaritmico')
B=loglog(A)
disp('presione cualquier tecla para continuar...')
pause

disp('matriz introducida')
home
A
disp('Grafico Semilogaritmico x')
B=semilogx(A)
disp('presione cualquier tecla para continuar...')
pause

disp('matriz introducida')
home
A
disp('Grafico Semilogaritmico y')
B=semilogy(A)
disp('presione cualquier tecla para continuar...')
pause

disp('matriz introducida')
home
A
disp('Grafico Polar')
B=polar(A)
disp('presione cualquier tecla para continuar...')
pause

disp('matriz introducida')
home
A
disp('Superficie de Malla Tridimensional')
B=mesh(A)
disp('presione cualquier tecla para continuar...')
pause

disp('matriz introducida')
home
A
disp('Plano de Contornos')
B=contour(A)
```

```
disp('presione cualquier tecla para continuar...')
pause

disp('matriz introducida')
home
A
disp('Dominio para Graficos de Superficie')
B=meshdom(A)
disp('presione cualquier tecla para continuar...')
pause

disp('matriz introducida')
home
A
disp('Graficos de Barras')
B=bar(A)
disp('presione cualquier tecla para continuar...')
pause

disp('matriz introducida')
home
A
disp('Graficos de Escalares')
B=stairs(A)
disp('presione cualquier tecla para continuar...')
pause

disp('matriz introducida')
home
A
disp('Añadir Barras de Errores')
B=errorbar(A)
disp('presione cualquier tecla para continuar...')
pause
```

CAPITULO V

DISCUSION

Hasta no hace mucho tiempo atrás los cursos en las diferentes Universidades se llevaban de una forma Tradicional, enfocados principalmente a generar una destreza mecánica de las técnicas, principalmente de aquellas que involucraban análisis numérico, sin embargo hoy en día con el desarrollo tecnológico de las computadoras, el estudio de las diferentes disciplinas se ha revolucionado poniendo al alcance de todo tipo de usuario diferentes herramientas computacionales que le ayuden a mejor comprender la inferencia teórica de sus cursos.

Las ventajas de usar equipos de cómputo se reflejan principalmente en eficiencia, rapidez y precisión, por otro lado es innegable que todo tipo de investigador tarde o temprano tendrá que integrarse al número de personas que utilizan computadoras para poder ser competitivo en la presentación de su trabajo.

CAPITULO VI

CONCLUSIONES

En conclusión, en particular se recomienda reconsiderar el lenguaje de programación utilizado hasta ahora en el curso de programación, no porque el lenguaje que se utiliza actualmente no sea apropiado sino porque el que se propone en este trabajo representa una buena oportunidad, dadas sus características particulares, de aprovechar de forma óptima los recursos de computación que matlab comprende, asociado a la orientación matemático - estadística, por sus funciones predefinidas para tales usos.

Como justificación a lo anterior, en la segunda parte del trabajo se presenta la implementación computacional de operaciones básicas de matrices, donde podrá observarse a manera de ejemplo didáctico, las ventajas y herramientas computacionales de las que se ha hablado en el transcurso de este escrito.

CAPITULO VII

RESUMEN

En resumen, el contenido de este trabajo puede dividirse en dos partes principales, la primera la que corresponde a un manual de introducción a la programación por medio de un lenguaje de computación denominado matlab, y la segunda que comprende un caso particular de programación de matlab en lo que son operaciones básicas de álgebra de matrices, orientado al programa analítico del curso de álgebra de matrices.

En la primera parte se explica como realizar la introducción de datos en forma matricial, se especifican las operaciones básicas con matrices, así como la sintaxis de programación de matlab, ademas se explican los comandos save, load, quit , !, hardcopy, así como instrucciones de ciclos como if for y while, con respecto a funciones se expresan escalares, vectoriales y matriciales, para la edición de se explican las opciones de archivos.m y edición de linea, se especifica por otro lado la notación de programación de dos puntos (:), se trabaja con strings en sus modalidades de texto, input y mensajes de error, se explica también la salida format, se da una explicación sobre los gráficos en sus opciones de plot print meta malla y mesh se incluye ademas una sección de help .

En la segunda parte se da un tratamiento especial a una aplicación de programación específicamente a la programación de operaciones básicas de matrices, donde se incluye un manual técnico que contiene tanto el diagrama de flujo del software diseñado así como su algoritmo y los detalles de interés que se consideraron para su programación.

CONTENIDO DEL PROGRAMA

En el programa se trabaja en base a ventanas se maneja una barra de títulos, una barra de menús, y por último los comandos se alojan en los menús, de esta manera se obtiene un programa de buena presentación y fácil acceso a su funcionamiento tanto para usuarios novatos como para usuarios intermedio, por otro lado se trata de introducir además todo el contenido del manual relacionado con operaciones básicas de matrices, dejando además oportunidad de seguir desarrollando rutinas para depurar y maximizar este programa o bien que este sea la base para generar más y mejores versiones de aplicaciones ya sea de operaciones básicas de matrices o de otras disciplinas.

CAPITULO VIII

LITERATURA CITADA

Hoffman K., Kunze R. 1973. Algebra Lineal. Editorial Prentice - Hall Hispanoamericana, S.A. Engliwood Cliffs. México. p. 1-395.

Sigmon K. 1992. Introducción a Matlab. Universidad de Florida. USA. p. 1-27.

Moler C., Little J., Bangert S. 1987. PC - Matlab for MS-DOS Personal Computers. The Math Works, Inc. Sherborn, MA 01770. USA. p. 1 - 271.

Coffin S. 1992. Introducción a los Sistemas Computacionales. Mc Graw Hill / Interamericana de España, S. A. España. p. 1 - 745