

ÍNDICE

PREFACIO	IX
CAPÍTULO 1. FUNDAMENTOS BÁSICOS DE MATLAB®	1
1.1 Introducción.....	1
1.2 Entorno MATLAB®	1
1.3 Lenguaje MATLAB®	4
1.4 Relación de MATLAB® con Matrices y Vectores.....	9
1.5 Estructuras de control de flujo en MATLAB®	11
1.6 Declaración de funciones en MATLAB® y archivos .m.....	16
CAPÍTULO 2. GRAFICACIÓN CON MATLAB®	19
2.1 Introducción.....	19
2.2 Graficación Simple	20
2.3 Contorno de funciones bidimensionales.....	33
2.4 Malla triangular y contornos.....	35
2.5 Graficación de Superficies y Mallas.....	38
2.6 MATLAB® como herramienta de graficación implementando Mapping Toolbox.....	42
CAPÍTULO 3. ÁLGEBRA LINEAL	55
3.1 Introducción.....	55
3.2 Matrices y Vectores	55
3.3 Operaciones de Matrices y Vectores.....	58
3.4 Sistemas de Ecuaciones Lineales.....	60
3.5 Determinante de una Matriz Cuadrada	62
3.6 Problemas mal acondicionados	63
3.7 Eliminación de Gauss	66
3.8 Eliminación de Gauss-Jordan	69
3.9 Descomposición LU	72

3.10 Valores propios de matrices.....	73
CAPÍTULO 4. INTERPOLACIÓN Y POLINOMIOS	77
4.1 Introducción	77
4.2 Expresiones polinomiales en MATLAB®	77
4.3 Interpolación Lineal.....	82
4.4 Interpolación Polinómica con Series de Potencias	85
4.5 Interpolación de Lagrange.....	88
4.6 Error en Interpolación Polinomial	93
4.7 Diferenciación e Integración del Polinomio de Interpolación de Lagrange	94
4.8 Interpolación de Chebyshev y Legendre	97
4.9 Interpolación de Hermite cúbica	100
4.10 Interpolación Bidimensional	104
CAPÍTULO 5. MÉTODOS NUMÉRICOS DE DIFERENCIACIÓN	107
5.1 Introducción	107
5.2 Diferenciación de Polinomios de Interpolación.....	107
5.3 Aproximación de Diferencias	110
5.4 Diferenciación por Método de Expansión de Taylor	111
5.5 Aproximación de Diferencias en Derivadas Parciales.....	113
5.6 Cálculo Numérico de Derivadas de Orden Superior	114
CAPÍTULO 6. MÉTODOS NUMÉRICOS DE INTEGRACIÓN.....	119
6.1 Introducción	119
6.2 Regla Trapezoidal	120
6.3 Regla de Integración de Simpson	123
6.4 Integración por Fórmula Cerrada de Newton-Cotes	130
6.5 Integración Numérica en Dominios Bidimensionales.....	134
6.6 Comandos para Integración Numérica Propios de MATLAB®	137
CAPÍTULO 7. RAÍCES DE ECUACIONES NO LINEALES	139
7.1 Introducción	139
7.2 Método gráfico.....	139
7.3 Método de la Bisectriz.....	143
7.4 Iteración de Newton	146
7.5 Método de la secante.....	150
7.6 Método de sustituciones sucesivas.....	151
7.7 Ecuaciones simultáneas no lineales	154

CAPÍTULO 8. INTERPOLACIÓN POR PARTES (SPLINES)..... 159

8.1 Introduccion..... 159

8.2 Splines Lineales 164

8.3 Splines Cuadráticos 169

8.4 Splines Cúbicos 173

 8.4.1 Splines cúbicos con polinomios de forma estándar 174

 8.4.2 Splines cúbicos basados en polinomios de forma de Lagrange 176

8.5 Funciones en MATLAB® 184

CAPÍTULO 9. ECUACIONES DIFERENCIALES..... 191

9.1 Introducción..... 191

9.2 Solución Simbólica de Ecuaciones..... 194

9.3 Método de Euler 195

9.4 Método Trapezoidal..... 199

9.5 Método de Runge-Kutta..... 201

9.6 Método Predictor-Corrector 205

 9.6.1 Funciones para la implementación genérica 206

9.7 Sistema de Ecuaciones Diferenciales 209

9.8 Ecuaciones Diferenciales de Segundo Grado 213

9.9 Ecuaciones Diferenciales con Valor de Frontera 219

9.10 Métodos Gráficos..... 225

ÍNDICE ANALÍTICO 229

PREFACIO

El objetivo de este libro es presentar una visión amplia y consistente de los métodos numéricos para estudiantes universitarios y profesionales en el área de ingeniería. El libro puede también servir como referencia para profesionales en la ingeniería que tienen la necesidad de usar MATLAB® en sus aplicaciones. Esto se debe a que muchos de los códigos de MATLAB®, presentados después de introducir las ideas básicas de cada algoritmo, se pueden modificar fácilmente para resolver problemas similares, incluso por aquellos que no saben lo que sucede dentro de las rutinas de MATLAB® y los algoritmos que utilizan.

Este libro puede ser utilizado con éxito por dos tipos de lectores; en primer lugar, por aquellos que requieran una explicación detallada de cada algoritmo y su potencial, porque están interesados en conocer cada método de tal forma que puedan modificar la estructura y realizar adaptaciones para resolver sus problemas particulares. En segundo lugar, por aquellos lectores que quieren usar genéricamente los métodos numéricos como subrutinas. Así como la mayoría de los usuarios de un electrodoméstico únicamente tienen que saber cómo usarlo para desempeñar una tarea, estos usuarios deben saber solo cómo formular sus problemas que quieren resolver utilizando MATLAB® y cómo usar las rutinas correspondientes para solucionarlos. Es importante aclarar que el conocimiento detallado sobre el método numérico es útil para encontrar una solución a un problema particular de ingeniería. Sin embargo, solo implica que los usuarios de una sola vez de cualquier método numérico pueden utilizar este libro, así como los lectores que quieren entender el principio/ecuaciones subyacentes de cada algoritmo.

Su contenido se enfoca principalmente en ayudar a los lectores a comprender los conceptos matemáticos fundamentales de los métodos numéricos y practicar habilidades de resolución de problemas usando MATLAB®. La metodología del libro es primero enseñar conceptos básicos para que los lectores puedan formular correctamente los problemas matemáticamente, saltándose algunas comprobaciones tediosas e innecesarias; y después, implementar directamente en MATLAB® los códigos para resolver problemas prácticos. Todos los algoritmos presentados en este libro van seguidos de un ejemplo de código MATLAB® para que los estudiantes puedan modificarlo fácilmente para resolver sus propios problemas. Esta metodología utilizada es por el hecho de considerar que ayuda a la mayoría de los estudiantes y profesionales –particularmente aquellos que no son de matemáticas–, a comprender cómo usar herramientas numéricas correctamente en la solución de sus problemas de interés es más importante que hacer largas demostraciones y comprobaciones.

El libro usa mucho más código que matemáticas formales. Esto es porque estamos convencidos de que incluso lectores con una excelente formación matemática tienen problemas en entender un enfoque hasta que ellos observan el algoritmo implementado en código. Este hecho es porque el código implementado resuelve todas las ambigüedades.

Durante más de dos años hemos ensayado múltiples maneras de exponer este material a auditorios disímiles. En el camino se ha contado con la invaluable tolerancia de nuestros alumnos, principalmente del CUCEI en la Universidad de Guadalajara en México. Se agradece de manera especial a nuestros compañeros profesores del Centro Universitario de Ciencias Exactas e Ingenierías. Tantas colaboraciones, ayudas y discusiones con colegas merecerían un capítulo adicional. A todos, los mencionados y en especial a los omitidos, nuestro testimonio de gratitud.

Erik Cuevas
Alberto Luque
Héctor Becerra
Héctor Escobar
Daniel Zaldívar

Guadalajara, México

Fundamentos Básicos de MATLAB[®]

1.1 INTRODUCCIÓN

MATLAB[®] es un software originalmente escrito por Cleve Moler, con el objetivo de proporcionar un software matricial a finales de los 70. Con el paso de los años la plataforma se ha desarrollado hasta el punto de ser hoy en día un estándar en ingeniería y en ciencias computacionales.

Siendo MATLAB[®] un lenguaje interpretado, sencillo y a la vez rápido y potente, aunado a su capacidad para generar gráficos, incorporación de animaciones, y su capacidad para el desarrollo de diversas aplicaciones complejas, se ha vuelto una herramienta clave en el avance de la investigación científica.

Durante este capítulo se estudiarán los fundamentos básicos del entorno y lenguaje mismo que nos permitirán comprender, analizar y crear códigos que se presentarán en los capítulos posteriores del texto.

1.2 ENTORNO MATLAB[®]

El entorno es el conjunto de herramientas que permiten trabajar como usuario o programador [1-3]. Permite: importar y procesar datos, crear y modificar ficheros, generar gráficos y animaciones y desarrollar aplicaciones de usuario.

La interfaz de MATLAB® se compone de distintas ventanas en las que se puede ejecutar instrucciones o bien obtener información necesaria del espacio de trabajo, en la figura 1.1 se muestra la pantalla inicial al arrancar el programa:

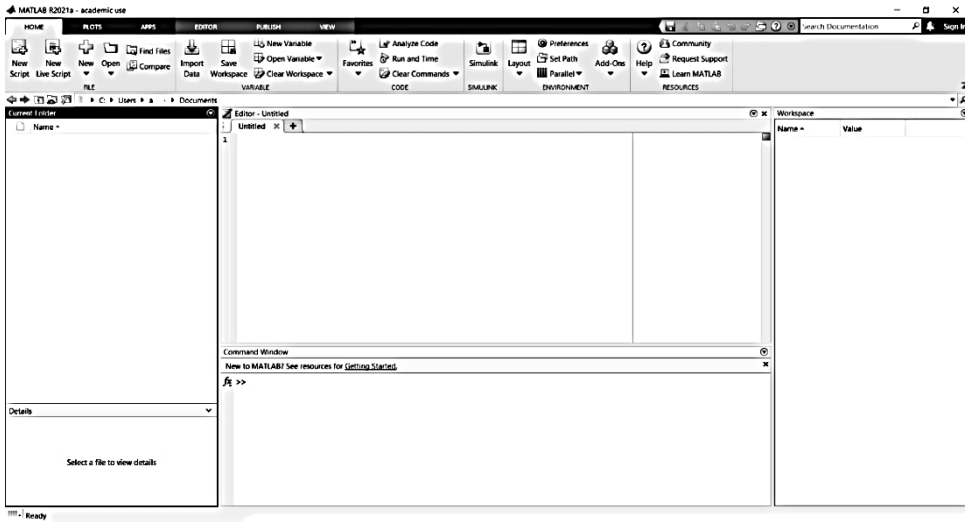


Figura 1.1 Espacio de trabajo Principal de MATLAB®

Dependiendo de la necesidad que se tenga, se pueden realizar instrucciones en tiempo real en la ventana “Command Window”(ventana de comandos) que es la que se presenta en la figura 1.2:

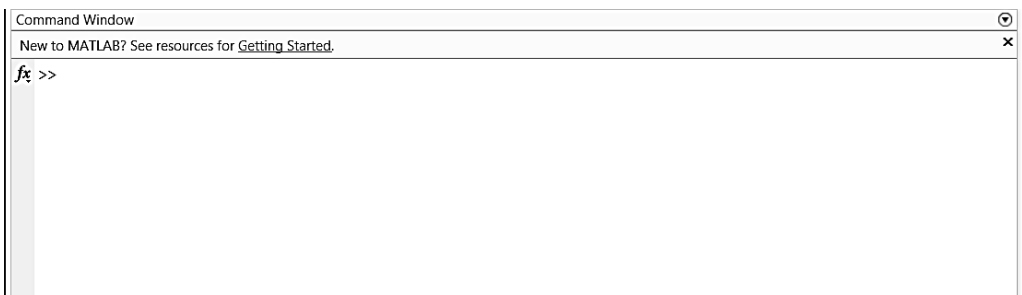


Figura 1.2 Ventana de Comandos de MATLAB®

Para obtener Información de las variables con las que se está trabajando, solo es necesario consultar la ventana de “workspace” presentada en la figura 1.3:



Figura 1.3 Workspace

Si se desea conocer el directorio en el que estamos trabajando solo es cuestión de verificar en la ventana “current folder” mostrada en la figura 1.4:

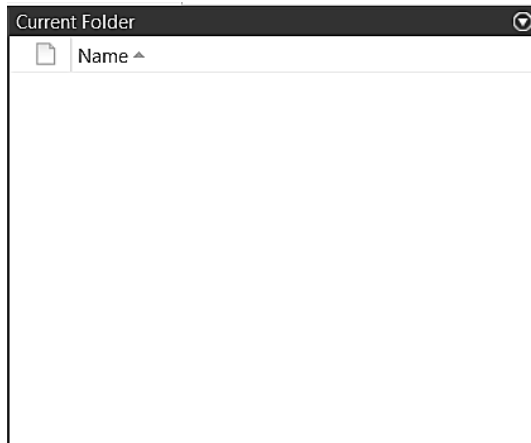


Figura 1.4 Ventana de directorio actual “current folder”

Para el caso de crear un script en un archivo nuevo, será necesario dar un clic en el botón “+” que se muestra en la figura 1.5 y seleccionar la opción script, esto nos abrirá la ventana del editor que se muestra en la figura 1.6 que será la zona donde podremos editar y crear los programas que se deseen llevar a cabo.

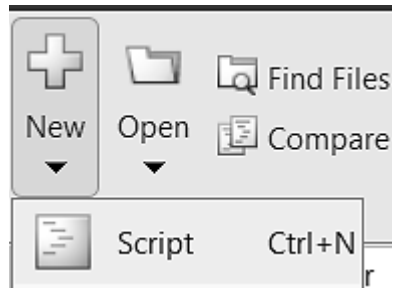


Figura 1.5 Botón para crear un script nuevo

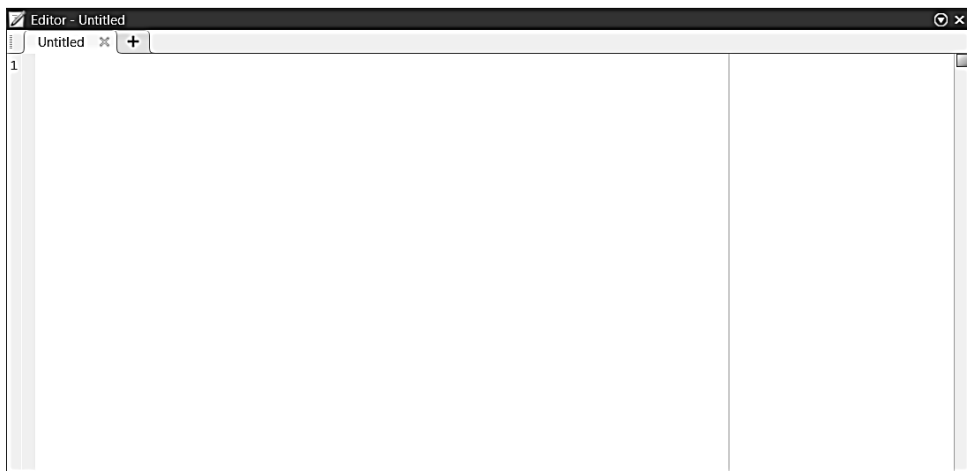


Figura 1.6 Editor de funciones

1.3 LENGUAJE MATLAB®

1.3.1 Asignación de Variables

MATLAB® es un lenguaje científico de programación que, a diferencia de muchos otros, no necesita una declaración de variables para utilizarlas [4]. La única condición que pone para ello es el uso correcto de la sintaxis siguiente:

```
Nombre_variable = Valor_Variable
```

Por ejemplo:

```
A = 32
```

Al ser MATLAB® un lenguaje con orientación científica, es necesario también tener en cuenta ciertas palabras reservadas y variables predefinidas en el lenguaje tales como:

Nombre	Significado
eps	Precisión de la máquina (número más pequeño que se puede expresar en la computadora)
flops	Contador de operaciones de punto Flotante
inf	Infinito
Pi	Número pi π
nargin	Número de argumentos de entrada a función
nargout	Número de argumentos de salida en una función
NaN	No es un número (se produce en indefiniciones)
date	Fecha de la máquina

Tabla 1.1 Variables predefinidas en MATLAB®

1.3.2 Operadores en MATLAB®

Como se verá más adelante, la unidad fundamental de MATLAB® son las matrices, a continuación se muestran algunos operadores que se pueden aplicar tanto a matrices como a magnitudes escalares en el lenguaje:

Aritméticos:

Operador	Símbolo / Comando
Suma	+
Resta	-
Multiplicación	*
División	/
Potenciación	^
Raíz cuadrada	sqrt ()
Residuo de una división a/b	mod (a , b)

Tabla 1.2 Operadores Aritméticos en MATLAB®(R)

En el caso de operar con matrices y desear realizar la operación elemento a elemento de estas, basta con agregar un "." antes del operador a utilizar, por ejemplo, la suma elemento a elemento de matrices puede realizarse con:

$$C = a .+ b$$

Operadores trigonométricos:

Operador	Comando
seno	<code>sin()</code>
coseno	<code>cos()</code>
tangente	<code>tan()</code>
cotangente	<code>cot()</code>
secante	<code>sec()</code>
cosecante	<code>csc()</code>

Tabla 1.3 Operadores Trigonométricos MATLAB®

Operadores Especiales números complejos:

Operador	Comando
Magnitud de un número complejo	<code>abs(x)</code>
Angulo en radianes del número complejo	<code>angle(x)</code>
Generación de un número complejo $x + yi$	<code>complex(x,y)</code>
Conjugado de un número complejo	<code>conj(x)</code>
Parte real de un número complejo	<code>real(x)</code>
Parte imaginaria de un número complejo	<code>imag(x)</code>
Devuelve 1 si el número es real o 0 si el número es complejo	<code>isreal()</code>

Tabla 1.4 Operadores Especiales para números complejos en MATLAB®

Operadores Relacionales:

Son operadores que se utilizan generalmente para evaluar una condición para los cuales su salida puede ser un 1 en caso de ser verdadera o un 0 en caso de ser falsa la afirmación:

Operador	Símbolo
Mayor que	<code>></code>
Menor que	<code><</code>
Igual que	<code>==</code>
Mayor o igual que	<code>>=</code>
Menor o igual que	<code><=</code>
Diferente que	<code>~=</code>

Tabla 1.5 Operadores Relacionales en MATLAB®

Operadores Lógicos:

Estos operadores suelen ser utilizados para evaluar más de una condición en una sola condición y son:

Operador	Símbolo
Y lógico da un 1 siempre que las dos condiciones sean verdaderas	&
O lógico da 1 siempre que alguna de las dos condiciones sean verdaderas	

Tabla 1.6 Operadores Lógicos en MATLAB®

Operadores con Arreglos Matriciales:

Dado que la estructura básica de MATLAB® son las matrices, se muestran a continuación algunos operadores especiales para arreglos matriciales:

Operador	Comando
Dimensión de un arreglo	size()
Longitud de un vector, o número de líneas en una matriz	length()
Valor máximo de un arreglo	max()
Valor mínimo de un arreglo	min()
Mediana de los elementos de un arreglo	median()
Promedio de los elementos de un arreglo	mean()
Desviación estándar de los elementos de un arreglo	std()

Tabla 1.7 Operadores Propios de Arreglos en MATLAB®

Operadores de Entrada y Salida

En diversas ocasiones es necesario realizar una interacción con el usuario del código que se está ejecutando, por lo que es necesario tener instrucciones de adquisición de datos de entrada o de despliegue de mensajes. En MATLAB® estas instrucciones son: