

Universidad Autónoma del Estado de México

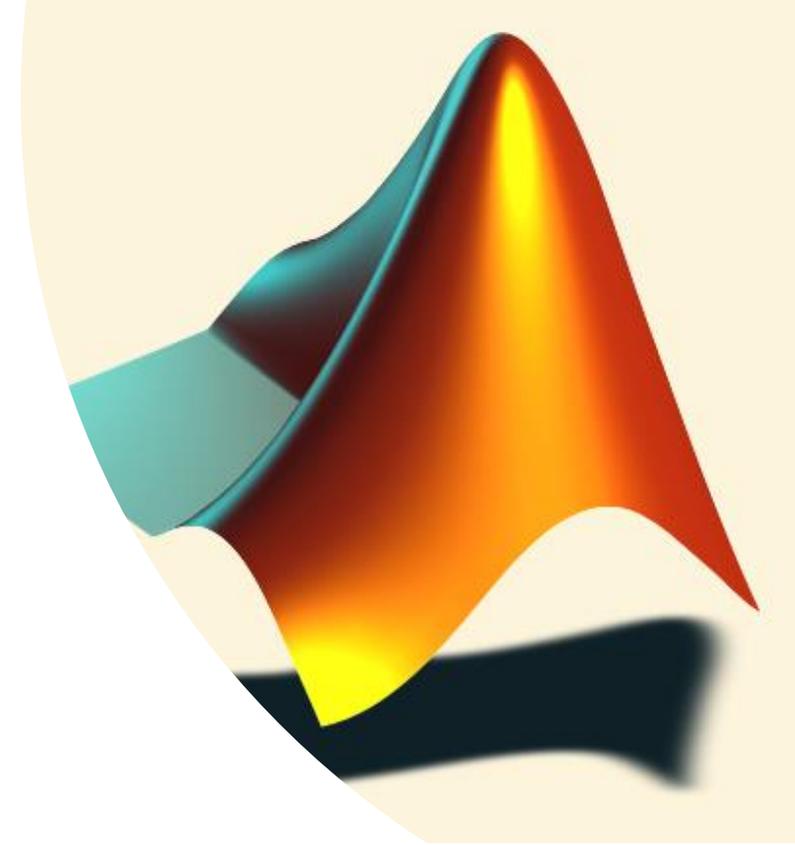
Facultad de Medicina

Licenciatura: *Bioingeniería Médica*

Unidad de Aprendizaje: *Métodos Numéricos*

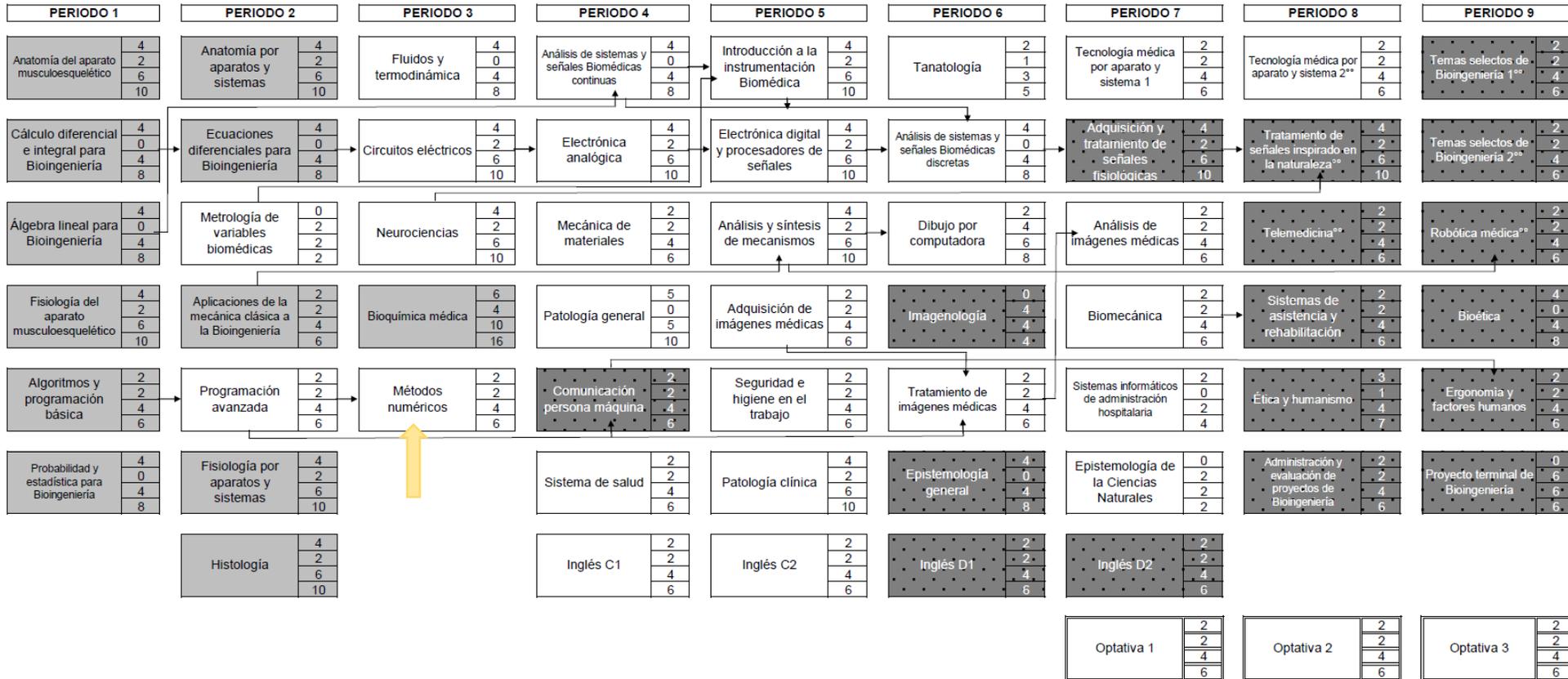
Unidad de competencia 1: *Introducción a Matlab*

Elaboró: M. en I. Ma. de Lourdes Najera López

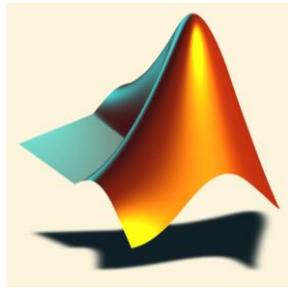


Septiembre, 2019.

MAPA CURRICULAR DE LA LICENCIATURA EN BIOINGENIERÍA MÉDICA 2010



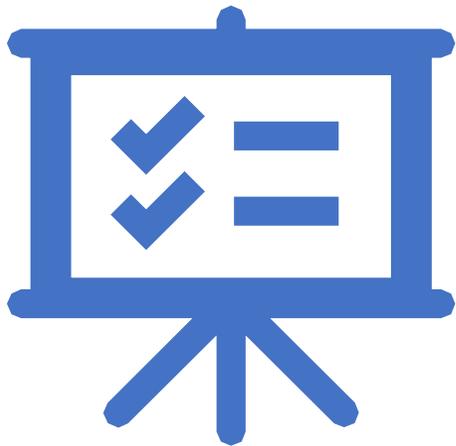
Ubicación de la unidad de aprendizaje de **Métodos Numéricos** dentro del programa de **Bioingeniería Médica**



1. INTRODUCCIÓN A MATLAB

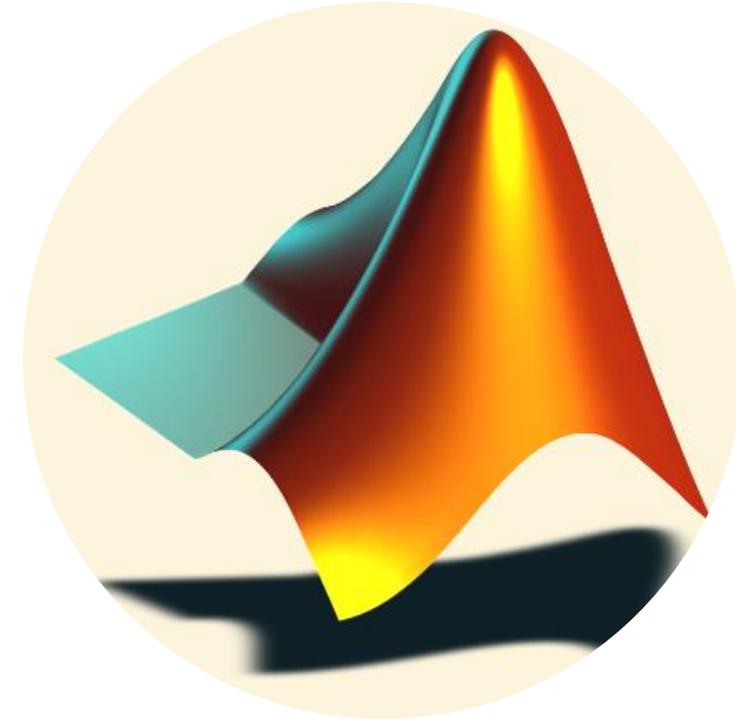
OBJETIVO

Identificar los fundamentos del software Matlab a través de la revisión documental de los manuales de referencia para ser capaces de codificar los algoritmos de posteriores unidades.



- 1.1 Características básicas
- 1.2 Variables
- 1.3 Operadores
- 1.4 Entrada y salida de datos
- 1.5 Estructuras de selección
- 1.6 Estructuras de iteración
- 1.7 Manejo de vectores y Matrices
- 1.8 Uso de funciones predefinidas
- 1.9 Representación gráfica de datos

CONTENIDO



PRESENTACIÓN

En el campo profesional de la medicina se requiere utilizar modelos matemáticos para la predicción y explicación de ciertos fenómenos relacionados con el cuidado de la salud y la mejora de la calidad de vida del ser humano; en particular en el ámbito de la tecnología aplicada al servicio médico directo al paciente, por tal motivo, la unidad de aprendizaje de **Métodos Numéricos** es fundamental para la formación de profesionistas en la licenciatura de **Bioingeniería Médica**, ya que son técnicas mediante las cuales es posible plantear soluciones a los problemas, a través de modelos matemáticos en conjunto con algoritmos ejecutados en Software especializado.

PRESENTACIÓN

Para que los alumnos logren lo anterior, se utilizan las siguientes estrategias de aprendizaje:

1. Enseñanza directa: El profesor describe los conceptos fundamentales de las características básicas de Matlab.
2. El alumno junto con el profesor interactúa con dicho software para identificar los operadores más útiles de acuerdo a la unidad de aprendizaje.
3. El alumno junto con el profesor desarrolla algoritmos para operaciones con matrices.
4. El profesor propone series de ejercicios extra clase al alumno para su solución por medio de algoritmos en Matlab.

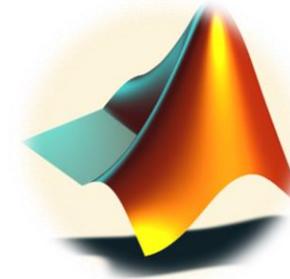
Introducción

El análisis numérico trata de diseñar métodos para aproximar, de una manera eficiente, las soluciones a problemas expresados matemáticamente. La eficiencia del método depende tanto de la precisión que se requiera como de la facilidad con la que pueda implementarse. En situaciones prácticas, el problema matemático se deriva de un fenómeno físico sobre el cual se han hecho algunas suposiciones para simplificarlo y para poderlo representar matemáticamente. Por lo anterior, se debe estudiar **métodos numéricos** para conocer las técnicas de solución a ciertos problemas. El futuro bioingeniero médico debe saber pronosticar un fenómeno mediante un modelo que se ejecuta en software de alta precisión y rapidez.

¿Qué es Matlab?

MATriX LABoratory

MATLAB es un programa para realizar cálculos numéricos con *vectores* y *matrices*. Como caso particular puede también trabajar con números escalares, tanto reales como complejos. Una de las capacidades más atractivas es la de realizar una amplia variedad de *gráficos* en dos y tres dimensiones. MATLAB.



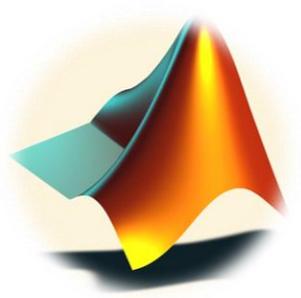
The image displays the MATLAB R2017b software interface. The top menu bar includes HOME, PLOTS, APPS, EDITOR, PUBLISH, and VIEW. The Editor window shows a function definition for 'untitled' with the following code:

```
1 function [outputArg1,outputArg2] = untitled(inputArg1,inputArg2)
2 %UNTITLED Summary of this function goes here
3 % Detailed explanation goes here
4 outputArg1 = inputArg1;
5 outputArg2 = inputArg2;
6 end
7
8
```

The Command Window at the bottom shows the prompt `>>` and the text: "Ventana de comandos, que a través de esta nos comunicamos con MATLAB". Below this, it states: "`>>` comando se denomina *prompt* e indica que MATLAB está desocupado, disponible para ejecutar órdenes".

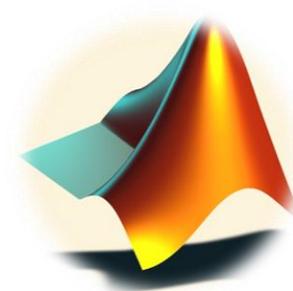
The Workspace window on the right is empty, showing columns for Name and Value.

The Command Window text is:
Ventana de comandos, que a través de esta nos comunicamos con MATLAB
`>>` comando se denomina *prompt* e indica que MATLAB está desocupado, disponible para ejecutar órdenes



1.1 CARACTERÍSTICAS BÁSICAS

- **MATLAB** es un software que posibilita la ejecución del cálculo numérico y simbólico de forma rápida y precisa, acompañado de gráficas para el uso científico de ciencias e ingeniería.
- El espacio de trabajo es interactivo y tiene más de 500 funciones.
- Cuenta con un lenguaje de programación basado en vectores y matrices.
- Cuenta con apartados específicos para álgebra lineal, análisis matemático y análisis numérico, estadística, series de tiempo, finanzas, etc.
- Permite diseñar sistemas dinámicos a través de Simulink
- Admite sistemas de tiempo continuo, sistemas de control y control inteligente, procesamiento de señal digital y comunicaciones.



1.1 CARACTERÍSTICAS BÁSICAS

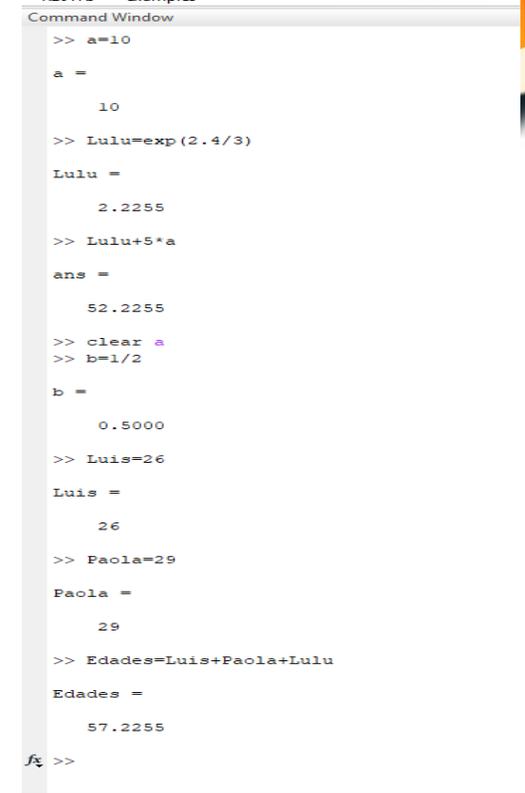
- Los toolboxes de Matlab son paquetes de aplicación básica, como:
 - **Symbolic Math**: integra el lenguaje matemático
 - **Database Toolbox**: permite intercambio de datos con las bases de datos más populares
 - **Excel Link Toolbox**: se interrelaciona con Microsoft Excel y permite la transferencia de datos en ambos sentidos
 - **Statistics Toolbox**: contiene 20 tipos de distribuciones de probabilidad, estadística descriptiva, estadística de procesos, diseño de experimentos, etc.
 - **Optimization Toolbox**: proporciona algoritmos para solución de problemas
 - **Spline Toolbox**: a través de un interface gráfico da potentes funciones para el ajuste de datos, visualización, interpolación y extrapolación.
 - **Partial Differential Equation Toolbox**: soluciona problemas como transferencia de calor, flujo en medios porosos, medios conductores, estructuras, magnetismo, etc.
 - **Neural Network Toolbox**: diseña y simula redes neuronales, aplicación de control y simulación de sistemas.

1.2 VARIABLES

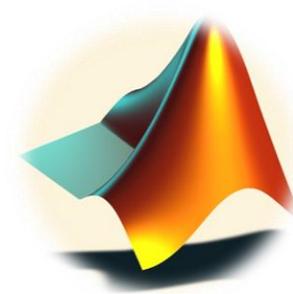
- **Variables:** se crea una variable asignando un valor directo.
- **Expresiones numéricas:** son un conjunto de números, funciones y variables previamente definidas, relacionados todos ellos por operadores aritméticos.
 - Nombre de una variable: **LETRA+NUMERO=VALOR**

Variables vectoriales: $V = [v_1, v_2, \dots, v_n]$

Variables matriciales: $A = [a_{11} a_{12} a_{13}; a_{21} a_{22} a_{23}; a_{31} a_{32} a_{33}]$

A screenshot of the MATLAB Command Window showing a series of commands and their outputs. The window title is 'Command Window'. The commands and outputs are: '>> a=10' followed by 'a = 10'; '>> Lulu=exp(2.4/3)' followed by 'Lulu = 2.2255'; '>> Lulu+5*a' followed by 'ans = 52.2255'; '>> clear a' followed by '>> b=1/2' followed by 'b = 0.5000'; '>> Luis=26' followed by 'Luis = 26'; '>> Paola=29' followed by 'Paola = 29'; and '>> Edades=Luis+Paola+Lulu' followed by 'Edades = 57.2255'. The prompt 'fx >>' is visible at the bottom.

```
Command Window
>> a=10
a =
    10
>> Lulu=exp(2.4/3)
Lulu =
    2.2255
>> Lulu+5*a
ans =
    52.2255
>> clear a
>> b=1/2
b =
    0.5000
>> Luis=26
Luis =
    26
>> Paola=29
Paola =
    29
>> Edades=Luis+Paola+Lulu
Edades =
    57.2255
fx >>
```



1.3 OPERADORES

Concepto	Comando en Matlab	Ejemplo
Unidad imaginaria	i, j	$5 + 7i; \quad -1 - 4j$
Número π	pi	
Infinito, el mayor número que puede almacenar.	Inf	Se produce cuando $\frac{x}{0}$ con $x \neq 0$
Not a Number, magnitud no numérica, resultado de cálculos indefinidos	NaN	Es cuando $\frac{0}{0}$ o ∞/∞
Otras constantes	eps, intmax, intmin, realmax, realmin	Consultar ayuda on-line

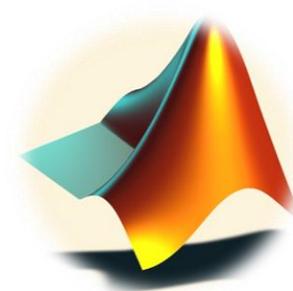
Algunas constantes pre-definidas



1.3 OPERADORES

Concepto	Ejemplo
Números Reales	$-3, 2, -1, 0, -0,00658, 1,5e + 5, 6,21e - 25$
Números complejos	$5 + 7i; -1 - 4j$
Booleanos	true, false
Caracteres	'carácter' (entre apóstrofes)
Operadores aritméticos	$+, -, *, /, ^$
Operadores de comparación	$==, \cong$ (ó $<>$), $>, <, \geq, \leq$
Operadores lógicos	$\&, , \sim$
Operadores lógicos para escalares	$\&\&, $

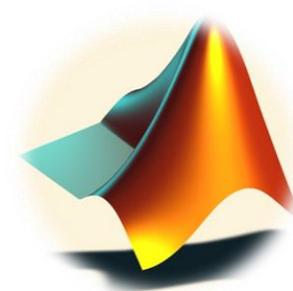
Constantes y operadores diversos



1.3 OPERADORES

Concepto	Matlab	Concepto	Matlab
Raíz cuadrada	<code>sqrt(x)</code>	Seno (radianes)	<code>sin(x)</code>
Módulo o magnitud	<code>abs(x)</code>	Coseno (radianes)	<code>cos(x)</code>
Complejo conjugado	<code>conj(z)</code>	Tangente (radianes)	<code>tan(x)</code>
Parte real	<code>real(z)</code>	Cotangente (radianes)	<code>cotg(x)</code>
Parte imaginaria	<code>imag(z)</code>	Arcoseno	<code>asin(x)</code>
Exponencial	<code>exp(x)</code>	Arcocoseno	<code>acos(x)</code>
Logaritmo natural	<code>log(x)</code>	Arcotangente	<code>atan(x)</code>
Logaritmo decimal	<code>log10(x)</code>	Coseno hiperbólico	<code>cosh(x)</code>
Aprox. Racional	<code>rat(aprox. Racional)</code>	Seno hiperbólico	<code>sinh(x)</code>
Resto de dividir x por y iguales si $x, y > 0$.	<code>mod(x,y)</code> <code>rem(x,y)</code>	Tangente hiperbólico	<code>tanh(x)</code>
Redondeo hacia 0	<code>fix(x)</code>	Arcoseno hiperbólico	<code>acosh(x)</code>
Redondeo hacia $+\infty$ y $-\infty$	<code>ceil(x), floor(x)</code>	Arcocoseno hiperbólico	<code>asin(x)</code>
Redondeo al entero más próximo	<code>round(x)</code>	Arcotangente hiperbólico	<code>atanh(x)</code>

Algunas funciones matemáticas elementales



1.3 OPERADORES

Concepto	Matlab
Ver lista de variables actuales	<code>who</code>
Lo anterior más detallado	<code>whos</code>
Eliminar todas las variables	<code>clear</code>
Eliminar las variables a, b, c, etc (sin comas)	<code>clear a b c</code>
Cambia el formato de salida a su valor por defecto, short	<code>format</code>
El formato por defecto	<code>format short</code>
Muestra 15 dígitos	<code>format long</code>
Formato short, en coma flotante	<code>format short e</code>
Formato long, en coma flotante	<code>format long e</code>
Muestra de números como cociente de enteros	<code>format rat</code>

Por defecto MATLAB muestra los números en formato de punto fijo con 5 dígitos



1.3 OPERADORES

Concepto	Matlab
Lista de ficheros del directoria de trabajo	ls o dir
Devuelve el nombre y ruta (path) del directorio de trabajo	pwd
Para cambiar de directorio	cd
Limpia la ventana de Command Window	clc
Fecha actual	date

Documentación y ayuda *on-line*

>>*help* nombre_de_comando

>>*helpwin* Ayuda *on-line* con ventana de navegador

Ejercicios



Ejercicio 1 Calcular el valor de la expresión

$$J = \frac{42.1768 + 234}{2^{10} - 10247}$$

Ejercicio 2 Calcular el valor de la expresión

$$H = \frac{9.8 * 10^{14} + 5.876 * 10^{-5}}{9.987 * 10^5 - 10^6}$$

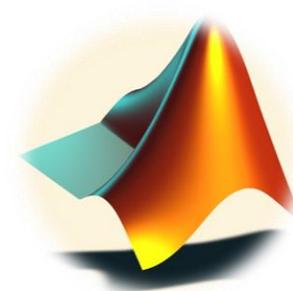
Y escribir el resultado en al menos 2 formatos

Ejercicio 3 Calcular

$$I = \sqrt[7]{\frac{3\text{sen}(32^\circ)}{42.1^3}}$$

Ejercicio 4 Según Hill y Lounasmaa, la ecuación de la curva de inversión del helio es $P = -21.0 + 5.44T - 0.132T^2$ Donde la presión viene dada en atmósfera y la temperatura T en grados Kelvin. Calcular el valor de la presión a una temperatura de 293 °K. Calcular el valor de la temperatura para una presión de 1N/m².

Nota: 1N/m²=9.265*10⁻⁶atm



1.4 Entrada y salida de datos

Todos los ficheros de comandos matlab deben de llevar la extensión .m hay 2 tipos

Ficheros de función la primera línea es ejecutable y comienza con la palabra

Function `arg_salida=nombre_funcion(arg_entrada)`

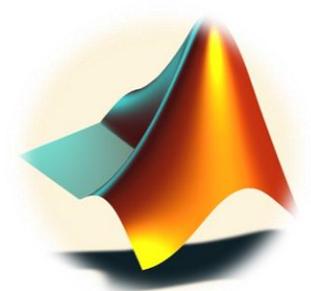
Después cuantos comandos sean necesarios incluidos comentarios como si se tratase de un fichero programa

El fichero se debe guardar con **nombre_funcion.m**

Ficheros de programa no constituyen funciones y se construye mediante una secuencia de comandos se ejecuta tecleando el nombre sin extensión

Ejemplo: Construir una función que convierta de °F a °C

```
MATLAB R2017b
HOME PLOTS APPS EDITOR PUBLISH VIEW
New Open Save Find Files Compare Go To Insert Comment Indent Breakpoints Run Run and Advance Run and Time
FILE NAVIGATE EDIT BREAKPOINTS RUN
C:\Users\DELL\Documents\Metodos_Numéricos\Matlab
Current Folder
Name
mincuadrados1.m
mincuadrados.m
LU.m
grafica11.fig
grafica5.m
grafica4.m
grafica3.m
grafica2.m
grafica1.m
grafica1.fig
grafica.m
convert.m
areatriang.m
areaequi.m
APUNTESMATLAB.pdf
Editor - C:\Users\DELL\Documents\Metodos_Numéricos\Matlab\convert.m
LU.m mincuadrados1.m convert.m
1 function temp_c=convert(temp_f)
2 %CREAR UNA FUNCION
3 % PASA DE GRADOS °F A °C
4 temp_f=54;
5 temp_c=5/9*temp_f-5/9*32;
6 end
7
Command Window
>> convert
ans =
12.2222
fx >>
```



1.4 Entrada y salida de datos

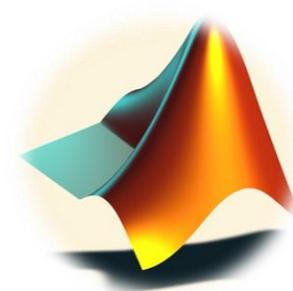
Comandos de entrada salida

input: permite introducir datos
variable=input(*'mensaje a pantalla'*);

disp: muestra un texto por pantalla
disp(*'El algoritmo no ha convergido'*)

menu: genera un menú que permite al usuario elegir entre distintas opciones
opcion=menu(*'titulo del mensaje','opcion1',...,'opcionp'*)

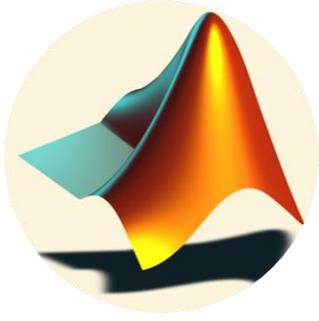
error: informa de la existencia de un error y detiene la ejecución del programa devolviendo el control al teclado.



1.5 Estructuras de selección

Desde Matlab 5 se ha incorporado la estructura `switch` que permite realizar bifurcaciones en un programa atendiendo al resultado de una expresión:

```
switch expresion
  case expresion_0
    comandos_0 que deben ejecutarse
  case expresion_1
    comandos_1 que deben ejecutarse
  case expresion_2
    comandos_2 que deben ejecutarse
  otherwise
    comandos escoba que deben ejecutarse
end
```



1.5 Estructuras de selección

Lectura y escritura en ficheros externos

La lectura y escritura de información en ficheros externos se lleva a cabo esencialmente con los comandos

`fread` y `fscanf` para lectura

`fprintf` y `fwrite` para escritura

El procedimiento general en todos los casos es:

- Abrir el fichero del que se desea leer o en el cual deseamos escribir.
- Colocar el puntero de lectura o escritura en la posición deseada.
- Leer o escribir las variables.
- Cerrar el fichero.



1.6 Estructuras de iteración

Bucles. Bucles anidados

```
for k=n1:incr:n2  
end
```

```
for k=vector_columna  
end
```

Con `break` se rompe la ejecución

Estructuras de control condicionadas

if	if	if	if	
	else	elseif	elseif	while
			else	
end	end	end	end	end



Current Folder: C:\Users\DELL\Documents\Metodos_Numéricos\Matlab

Editor - C:\Users\DELL\Documents\Metodos_Numéricos\Matlab\solucion_ecuacion_2do.m

```

1 function raices=solucion_ecuacion_2do(a,b,c)
2   %raiz=sole2(a,b,c)
3   %solucion de la ecuacion de segundo grado
4   %ax^2+bx+c=0, a~=0
5
6   a=input('valor de a: ');
7   b=input('valor de b: ');
8   c=input('valor de c: ');
9   d=b^2-4*a*c;
10  if d>0
11      x1=(-b+sqrt(d))/(2*a);
12      x2=(-b-sqrt(d))/(2*a);
13      disp('valor positivo de la raiz')
14  elseif d==0
15      x1=-b/(2*a);
16      x2=-b/(2*a);
17      disp('la raiz vale 0')
18  else
19      x1=(-b+i*sqrt(-d))/(2*a);
20      x2=(-b-i*sqrt(-d))/(2*a);
21      disp('la raiz es negativa')
22  end
23  disp('valores de la ecuación de segundo grado:')
24  x1, x2
25  end
26

```

Workspace

Name	Value
ans	12.2222

Command Window

```

valor de a: 2
valor de b: 4
valor de c: 1
valor positivo de la raiz
valores de la ecuación de segundo grado:

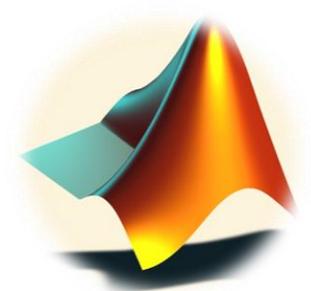
x1 =
-0.2929

x2 =
-1.7071

solucion_ecuacion_2do
valor de a:

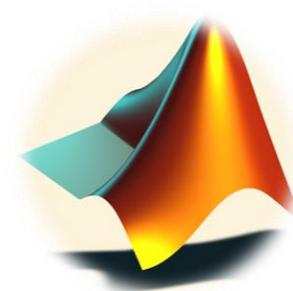
```

Ejercicio 5:
Utilizando estructuras de control construir una función que nos calcule las raíces de una ecuación de segundo grado $ax^2 + bx + c = 0$



1.7 Vectores y matrices

- Como ya se ha dicho, las matrices bidimensionales de números reales o complejos son los objetos básicos con los que trabaja MATLAB. Los vectores y escalares son casos particulares de matrices.
- La forma más elemental de introducir matrices en MATLAB es describir sus elementos de forma exhaustiva (por filas y entre corchetes rectos `[]`): elementos de una fila se separan unos de otros por comas y una fila de la siguiente por punto y coma.



1.7 Vectores y matrices

- Construir los siguientes vectores y matrices
- `v1=a:h:b` %%crea un vector fila
- `v2=a:b` %% crea un vector fila como el anterior con `h=1`
- `v3=v2'` %% matriz transpuesta (conjugada si es complejo)
- `v4=v2.'` %% matriz transpuesta sin conjugar

MATLAB posee, además, decenas de funciones útiles para generar distintos tipos de matrices. Para ver una lista exhaustiva consultar:

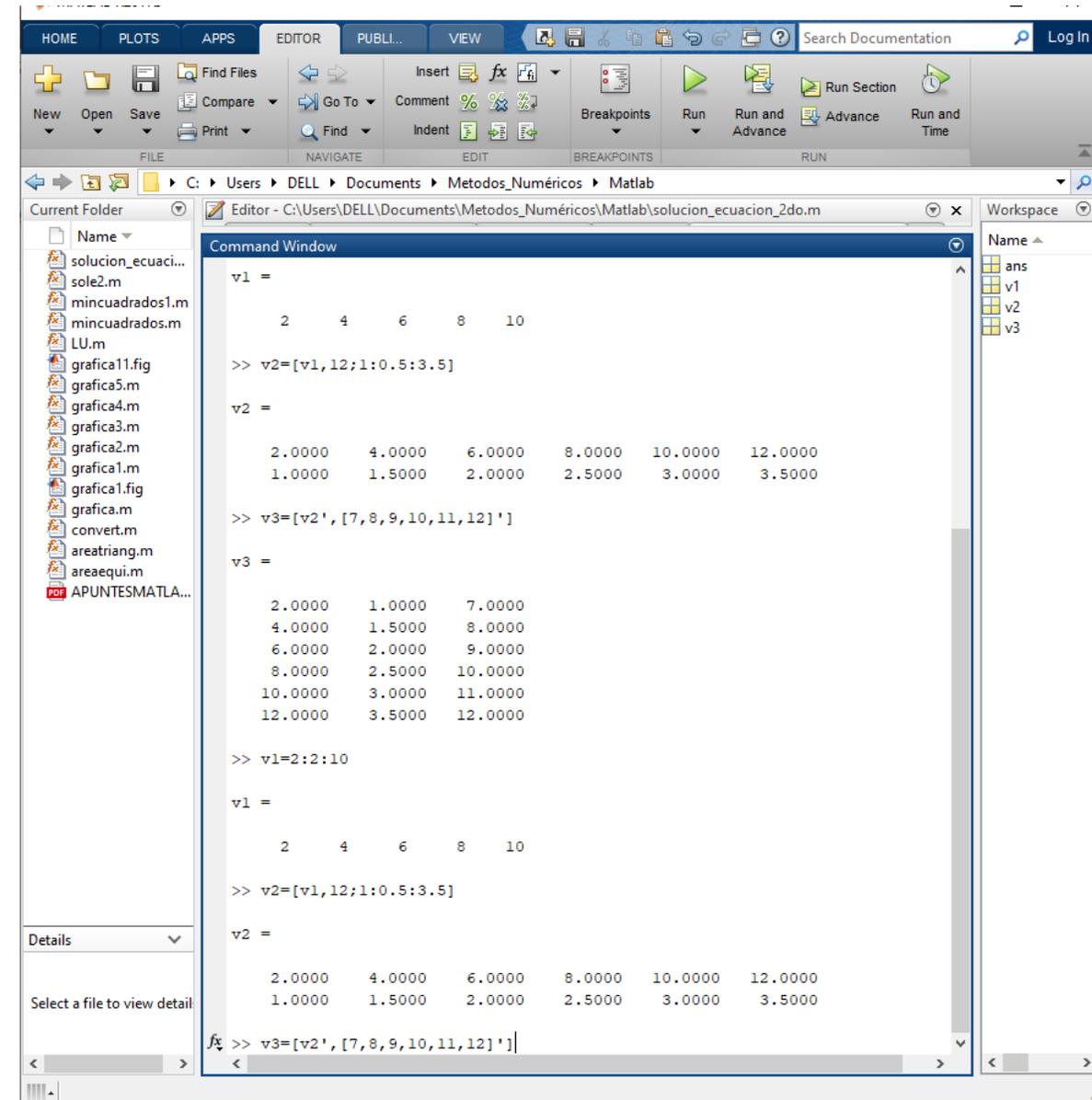
[Help](#) → [MATLAB](#) → [Functions: By Category](#) → [Mathematics](#) → [Arrays and Matrices](#)

Concepto	Operador Matlab
Matriz $n \times m$ con todas sus elementos igual a 0	<code>zeros(n,m)</code>
Matriz $n \times m$ con todos sus elementos iguales a 1	<code>ones(n,m)</code>
Matriz unidad $n \times m$ con diagonal principal igual a 1 y el resto de los elementos iguales a 0	<code>eye(n,m)</code>
Si v es un vector, entonces una matriz cuadrada de ceros con diagonal principal= v	<code>diag(v)</code>
Si A es una matriz, es su diagonal principal	<code>diag(A)</code>

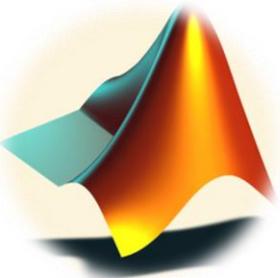
1.7 Vectores y matrices

Ejercicio: Introducir a Matlab los siguientes vectores

- $v1 = 2:2:10;$
- $v2 = [v1,12; 1:0.5:3.5]$
- $v3 = [v2', [7,8,9,10,11,12]']$



```
v1 =  
     2     4     6     8    10  
  
>> v2=[v1,12;1:0.5:3.5]  
  
v2 =  
     2.0000     4.0000     6.0000     8.0000    10.0000    12.0000  
     1.0000     1.5000     2.0000     2.5000     3.0000     3.5000  
  
>> v3=[v2', [7,8,9,10,11,12]']  
  
v3 =  
     2.0000     1.0000     7.0000  
     4.0000     1.5000     8.0000  
     6.0000     2.0000     9.0000  
     8.0000     2.5000    10.0000  
    10.0000     3.0000    11.0000  
    12.0000     3.5000    12.0000  
  
>> v1=2:2:10  
  
v1 =  
     2     4     6     8    10  
  
>> v2=[v1,12;1:0.5:3.5]  
  
v2 =  
     2.0000     4.0000     6.0000     8.0000    10.0000    12.0000  
     1.0000     1.5000     2.0000     2.5000     3.0000     3.5000  
  
fx >> v3=[v2', [7,8,9,10,11,12]']
```



1.7 Vectores y matrices

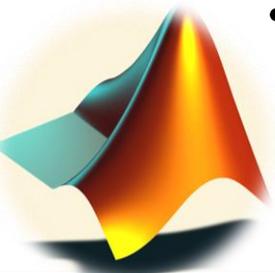
- **Acceso a elementos de vectores y matrices**
- El acceso a elementos individuales de una matriz se hace indicando entre paréntesis sus índices de fila y columna. Se puede así utilizar o modificar su valor

Dada la matriz $A = [2,4,6; 3,5,7; 1,2,3]$

- $A(3,2)$ selecciona el elemento del renglón 3 columna 2
- $A(3,2)=-8$ cambia el valor del elemento

Sea la matriz $w = [5,3,1,7,3,0, -2]$

- $w(\text{end})$
- $w(\text{end}-1)=8$

A screenshot of the MATLAB R2017b software interface. The Command Window shows the following code and output:

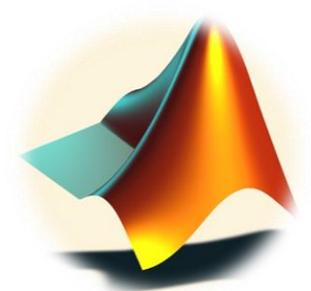
```
>> A=[2, 4, 6; 3, 5, 7; 1, 2, 3]
A =
     2     4     6
     3     5     7
     1     2     3

>> A(3,2)
ans =
     2

>> A(3,2)=-8
A =
     2     4     6
     3     5     7
     1    -8     3

fx >> |
```

The interface also shows a file explorer on the left with a list of files in the current folder, including 'solucion_ecuacion_2do.m', 'sole2.m', 'mincuadrados1.m', 'mincuadrados.m', 'LU.m', and several 'grafica' files.



1.7 Vectores y matrices

- **Acceso a elementos de vectores y matrices**

- Se puede acceder a una fila o columna completa sustituyendo el índice adecuado por : (dos puntos)

```
>>a=[1,2,3;4,5,6;7,8,9]
```

```
>>a(:,2)
```

```
>> aa(1,:)
```

- **Acceso a submatrices**

```
>> v=[4,6,1,7,6,9,-4,]
```

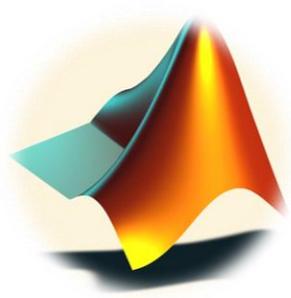
```
>>v([1,2,3])
```

```
>>a=[1,2,3;5,4,8;-1,0,2;4,5,9]
```

```
>>a(1:2,:)
```

```
>> B=a(3:4,[1,3])
```

1.7 Vectores y matrices

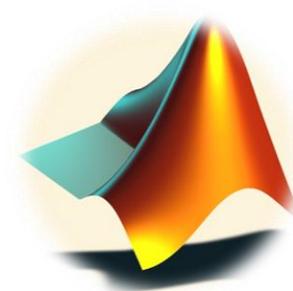


Sea A y B dos matrices de elementos a_{ij} y b_{ij} , y k un escalar

Sea v un vector y A una matriz

$A \pm B$	Si las dimensiones son iguales $a_{ij} \pm b_{ij}$	sum(v)	Suma de los elementos del vector v
$A \pm K$	Matrices de elementos respectivos $a_{ij} \pm k$	sum(A) o sum(A,1)	Suma de los elementos de la matriz A por columnas
$K * A$, A/K	Matrices de elementos respectivos ka_{ij} , $\frac{1}{k}a_{ij}$	sum(A,2)	Suma de los elementos de la matriz A por filas
A^n	Si n es entero positivo	prod(v), prod(A), prod(A,1), prod(A,2)	Como la suma, pero para el producto
$A.* B$	Producto de matrices con las dimensiones adecuadas $a_{ij} \times b_{ij}$	max(v), min(v)	Máximo/mínimo elemento del vector v
$A./B$	ídem de elementos a_{ij}/b_{ij}	max(A), min(A)	Máximo/mínimo elementos de la matriz A , por columnas
$A.^B$	ídem de elementos $a_{ij}^{b_{ij}}$	mean(v)	Media de los elementos del vector v
$k./A$	Matriz de la misma dimensión que A , con elementos k/a_{ij}	mean(A)	Media de los elementos de la matriz A por columnas
$A.^k$	ídem de elementos a_{ij}^k		

MATLAB dispone de ciertos operadores aritméticos que operan elemento a elemento. Son los **operadores** `.*` `./` y `.^`, muy útiles para aprovechar las funcionalidades vectoriales de MATLAB.



1.8 Uso de funciones predefinidas

Existen dos tipos de archivos de programa:

- **Scripts**, que no aceptan argumentos de entrada ni devuelven argumentos de salida. Operan sobre datos en el área de trabajo.
- **Funciones**, que aceptan argumentos de entrada y devuelven argumentos de salida. Las variables internas son locales para la función.

Scripts

- Cuando invoca un *script*, MATLAB simplemente ejecuta los comandos que se encuentran en el archivo. Los scripts pueden operar en datos existentes en el área de trabajo, o pueden crear datos nuevos en los cuales operar.

Funciones

- Una *función anónima* es una forma simple de la función de MATLAB que se define dentro de una sola instrucción de MATLAB.
- Cada archivo de función contiene una *función principal* obligatoria que aparece primero, y cualquier número de *funciones locales* que pueden seguir a la principal.
- Una *función privada* es un tipo de función principal. La característica que la distingue es que es visible solo para un grupo limitado de otras funciones.
- Una *función animada*, se dice que éstas se encuentran *anidadas* dentro de la función exterior

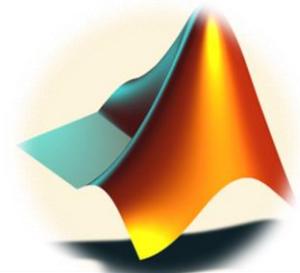
1.8 Uso de funciones predefinidas

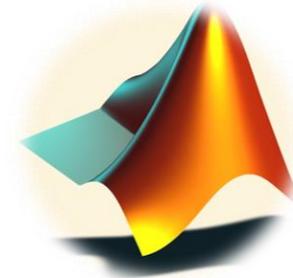
- **Scripts** (guión o archivo por lotes)
 - Es un conjunto de instrucciones (programa)
 - Debe guardarse en un fichero con sufijo **.m** para ser reconocido
- **M-Funciones**
 - Es un programa con una interfaz de comunicaciones con el exterior mediante argumentos de entrada y salida.

```
function [argumentos de salida] = nombre(argumentos de entrada)
%
% comentarios
%
....
instrucciones (normalmente terminadas por ; para evitar eco en pantalla)
....
```

- Algunas funciones sencillas, que devuelven el resultado de una expresión, se pueden definir mediante una sola instrucción, en mitad de un programa (script o función) o en la línea de comandos. Se llaman **funciones anónimas**. La sintaxis para definir las es:

nombre_funcion = @(argumentos) expresion_funcion





1.9 Representación gráfica de datos

- La representación gráfica de una curva en un ordenador es una línea poligonal construida uniendo mediante segmentos rectos un conjunto discreto y ordenado de puntos: $\{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$.

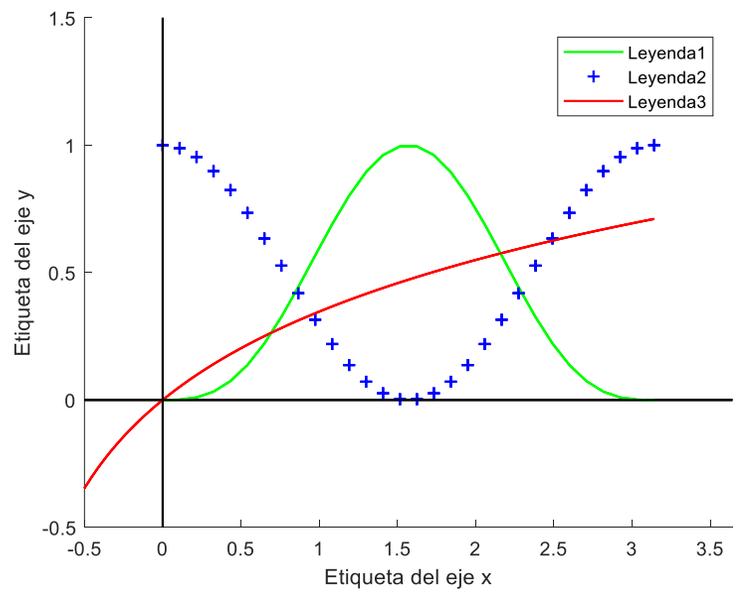
Colores: Blue ('b*'), Green, Red ('r*'), Cyan, Magenta, Yellow, Black ('k+')

Para establecer el **grosor LineWidth** (atención a mayúsculas)

Para establecer el **rango x=linspace(-1.0,1.0);**

```
function [y5] = grafica5()
1
2
3 x=linspace(0,pi,30);
4 axis([-0.5,pi+0.5,-0.5,1.5])
5 hold on
6 plot(x,sin(x).^3,'g',x,cos(x).^2,'b+', 'LineWidth',1.1)
7
8 x=linspace(-0.95,pi);
9 plot(x,log(x+1)/2,'r', 'LineWidth',1.1)
10 plot([-5,5],[0,0],'k', 'LineWidth',1.0)
11 plot([0,0],[-5,5],'k', 'LineWidth',1.0)
12 legend('Leyenda1','Leyenda2','Leyenda3')
13 xlabel('Etiqueta del eje x')
14 ylabel('Etiqueta del eje y')
15 hold off
16 shg
17
```

plot(x,y)



BIBLIOGRAFÍA

Burden RL y Faires JD. Análisis numérico. 8a ed. México: Cengage Learning Editores; 2001.

King RK y Mody NA. Numerical and statistical methods for bioengineering. 1a ed. Reino Unido: Cambridge University Press; 2011.

Larson R y Falvo DC. Elementary linear algebra. 7a ed. Estados Unidos: Brooks/Cole CENGAGE learning; 2012.

Nieves A y Domínguez F. Métodos numéricos aplicados a la ingeniería. 3ra ed. México: Grupo editorial Patria; 2007.

Press WH, Teukolsky SA, Vetterling WT y Flannery BP. Numerical recipes: The Art of Scientific Computing. 3a ed. Reino Unido: Cambridge University Press; 2007.

Peter Katta, MATLAB for Beginners: A Gentle Approach:Smashwords Edition, 2016.

Hahn, Brian, Essential MATLAB for Scientists and Engineers 2ed, Elsevier 2013.