

Introducción a MATLAB

1 Introducción

MATLAB es un software orientado a realizar cálculos numéricos. Es una herramienta multifuncional que posee muchas características para resolver diversos problemas, facilidades para programar, herramientas de graficación, entre otras. Utiliza un interprete para compilar y ejecutar secuencias de código dadas por el usuario.

MATLAB no es un sistema de álgebra computacional (cálculo simbólico) como lo es Mathematica o Maxima, a pesar de que algunas herramientas si podrían ser similares a estos últimos programas.

Una de las principales características de MATLAB es que es sencillo de utilizar lo que representa una característica importante si el lector no posee experiencia con lenguajes de programación. Además, MATLAB hace sencillas las operaciones con vectores/matrices y en muchos casos las funciones involucradas para estas operaciones están optimizadas. Como el lector podrá notar, los vectores/matrices son muy útiles para implementar los algoritmos del curso de métodos numéricos. Muchos de los códigos de hechos en MATLAB que se estudiarán en aquí pueden ser escritos de la misma forma en Octave (una versión libre de MATLAB), en una menor cantidad de casos, se debe realizar pequeños ajustes.

MATLAB está disponible para sistemas operativos Linux, Windows y MacOS.

2 Interfaz

Una vez instalado, la interfaz de MATLAB contiene varias ventanas que pueden ser útiles a la hora de trabajar con códigos. En la figura 1 se muestra la interfaz luego de la instalación donde se pueden observar las ventanas: *Explorador de archivos*, *Editor* y *Ventana de Comandos*. Si el lector dispone de un monitor grande quizás prefiera utilizar todas estas ventanas, sin embargo, puede también cerrar varias ventanas. En este capítulo utilizaremos esencialmente la *Ventana de Comandos* y el *Editor*, este último se puede abrir haciendo uso del botón *nuevo guión (script)* disponible en la barra de herramientas.

Se pueden *arrastrar* las ventanas para ordenarlas a conveniencia. En la figura 2 se muestra la interfaz únicamente con las ventanas *Editor* y *Ventana de Comandos*. Aquí se hace una breve descripción de como utilizar estas ventanas.

Ventana de Comandos (Command Window)

En la ventana de comandos (en inglés *Command Window*) situada a la derecha en la figura 2 puede ser utilizada de muchas maneras. En esta sección se exploran varias que son de mucha utilidad para el curso de métodos numéricos. Para iniciar, digite en la ventana de comandos la siguiente la operación $1 + 2$ como se muestra en la siguiente línea

```
>> 1+2
```

luego presione enter para obtener el resultado. Como el lector puede imaginar, en esta ventana es posible realizar muchas operaciones aritméticas con la ventaja que se cuenta con más poder de cómputo que en

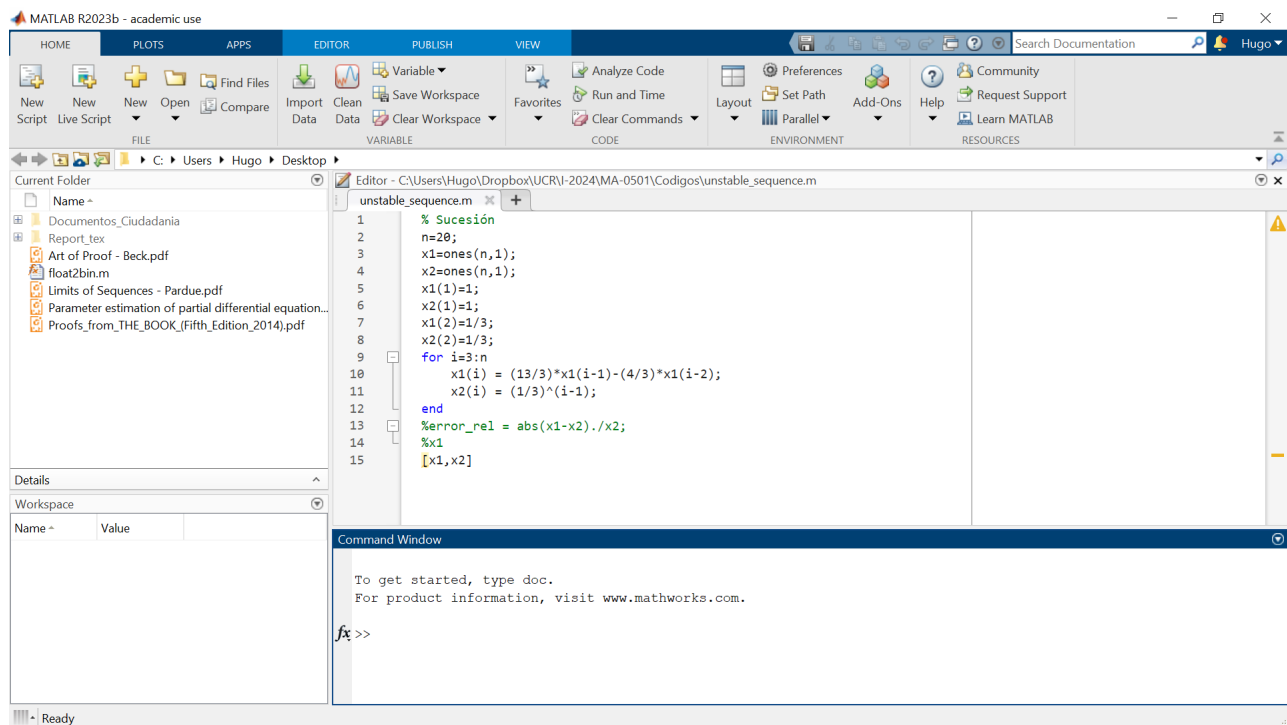


Figure 1: Interfaz de MATLAB luego de la instalación.

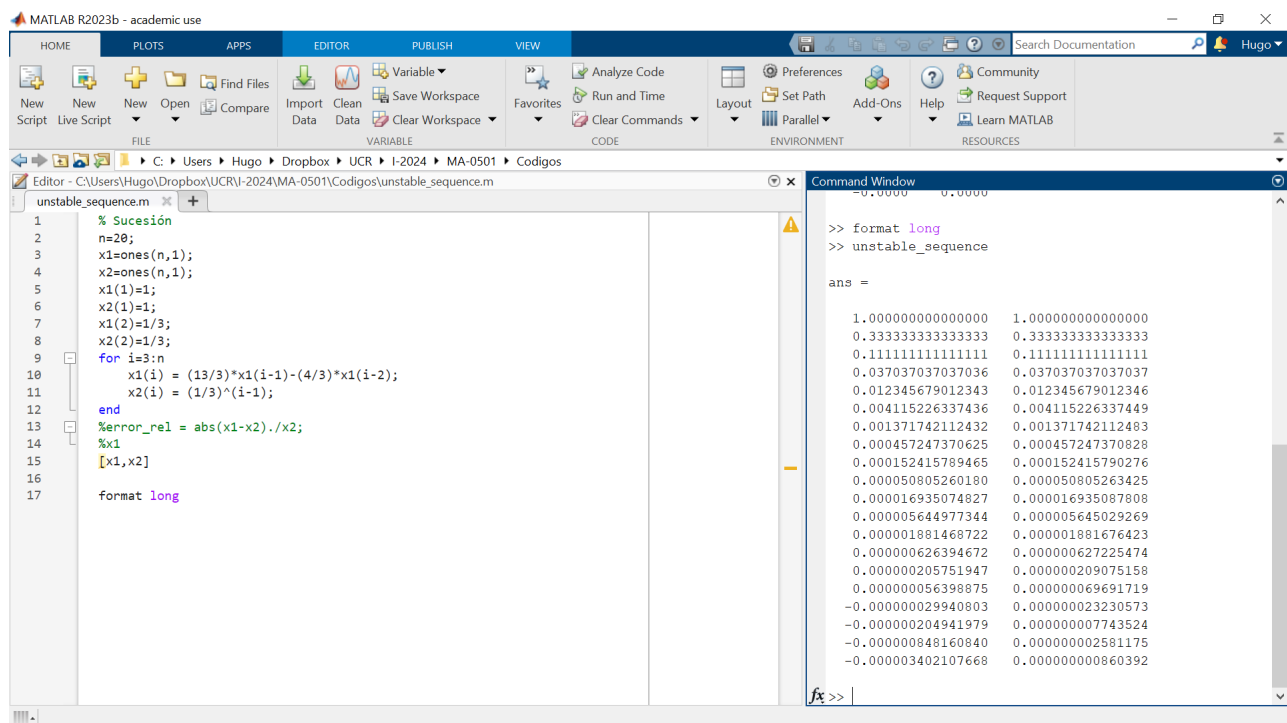


Figure 2: Interfaz simplificada de MATLAB

una calculadora sencilla. El cuadro (2) muestra los símbolos para realizar operaciones aritméticas en MATLAB.

Operación	Símbolo
Suma	+
Resta	-
Multipliación	*
División	/
Potenciación	^

Table 1: Símbolos de operaciones básicas en MATLAB.

Ejemplo 1 Verifique que el valor $x_0 = 1.2$ es cero del polinomio $p(x) = x^3 - 5.2x^2 + 7.8x - 3.6$, $p : \mathbb{R} \rightarrow \mathbb{R}$.

Solución: para verificar que x_0 es un cero del polinomio basta calcular su imagen. Esto se puede realizar fácilmente en la ventana de comandos como se puede ver en la siguientes líneas

```
>> (1.2)^3 - 5.2*(1.2)^2 + 7.8*(1.2) - 3.6
```

al presionar enter se obtiene como resultado 0. Note en particular la necesidad del símbolo del producto * para calcular $p(1.2)$.

Nota 1 Orden de operaciones: es fundamental tener en cuenta que en algunas situaciones se debe utilizar paréntesis adecuadamente para respetar el orden de las operaciones. Por ejemplo, si se quiere calcular el valor $\frac{1}{3+2}$, entonces, se debe digitar $1/(3+2)$ en la Ventana de Comandos.

Variables

El concepto de variable es de mucha importancia en la programación. Una variable está compuesta por un nombre, un valor asociado y una clase (o tipo). Una variable puede ser un número, un vector, una matriz o una cadena de texto. Una vez que se le indica a MATLAB el nombre y el valor de la variable, esta se almacena en la memoria y puede ser utilizada de diversas formas. De ser necesario se puede también cambiar el valor de la variable.

Por ejemplo, para declarar una variable¹ x que guarde el valor 1 basta digitar la instrucción $x = 1$ en la ventana de comandos y presionar enter, como se muestra a continuación

```
>> x = 1
```

Una vez que se presiona enter, se puede ver como MATLAB imprime nuevamente en pantalla el valor de la variable

```
>> x = 1
x=1
```

En este caso el nombre de la variable es x , el valor asignado es 1 y por defecto es de clase *double* (valor numérico con precisión doble).

En la mayoría de las situaciones cuando queremos asignar una variable nos interesa simplemente que se guarde en memoria y no queremos que MATLAB imprima nuevamente su valor. Para evitar que se imprima nuevamente se utiliza el comando $x = 1;$ (note el punto y coma), entonces al digitar esto y presionar enter, la variable se guarda y se evita que se despliegue nuevamente su valor en pantalla.

Los siguientes ejemplos muestran como declaran variables en MATLAB.

¹Usualmente se utiliza una letra minúscula para declarar variables que guardan valores numéricos. MATLAB hace diferencia entre mayúsculas y minúsculas por lo que también se puede declarar una variable diferente con nombre X .

Ejemplo 2 Es posible reasignar valores a las variables. Si se guarda el valor $x = 1$ en la ventana de comandos y luego se necesita utilizar otro valor, por ejemplo $x = 1.5$ basta digitar nuevamente en la ventana de comandos

```
>> x = 1.5;
```

esto actualizará el valor de la variable en la memoria de MATLAB.

Ejemplo 3 Guardar el valor $a = 5$ en MATLAB y calcular el resultado de la operación:

$$b = \frac{\sqrt{a^7} + (a - 3)^5}{a + 2025}$$

Solución:

```
>> format long
>> a=5;
>> b=(a^(7/2) + (a-3)^5)/(a+2025);
>> b
>> b=0.153452461668706
```

Nota 2 Explicaremos aquí cada una de las líneas de la imagen anterior. El comando **format long** le indica a MATLAB que se quiere el resultado en formato de punto fijo con 16 dígitos significativos.

Luego de realizar la asignación $a = 5$ se digita un punto y coma, esto es para que MATLAB no despliegue en pantalla el valor de a . Análogamente cuando se calcula el valor de b . Por último, en la penúltima línea se digitó b y luego se presionó enter, esto para desplegar (imprimir) en pantalla el valor de b .

Ejemplo 4 (Consejo importante) Suponga que se quiere usar el valor $a = 6$ y calcular una operación muy similar al ejemplo anterior pero con una leve diferencia

$$b = \frac{\sqrt{a^7} + (a - 3)^5}{2a + 2025}$$

No es necesario digitar todo nuevamente. Una vez que el cursor (mouse) esté en la ventana de comandos basta presionar en el teclado de su computadora la flecha hacia arriba (\uparrow), podrá ver que en la línea de comandos aparece la última instrucción dada a MATLAB.

Si se presiona varias veces la flecha hacia arriba podrá ver todas las instrucciones que se digitaron desde que se abrió MATLAB. Una vez ubicada la instrucción $a = 5$; se utiliza la flecha izquierda (\leftarrow) del teclado para cambiar esta instrucción a $a = 6$; y se presiona enter para actualizar el valor de la variable. Similarmente modifique la expresión para calcular b y despliegue el valor de b .

El nuevo valor de la variable b debe ser 0.379032785685403.

Nota 3 (Sobre la asignación de un valor) Note que en los ejemplos anteriores se utilizó el símbolo ' $=$ ' para asignar valores a las variables. Es importante que el lector sepa que en MATLAB la instrucción $x = 1.5$ crea en memoria una variable de nombre x y le asigna el valor de 1.5.

Si se digita la instrucción $y = x$, esto nuevamente crea una variable y a la que se le asigna el valor de 1.5. Sin embargo, note que si luego se modifica el valor de x a 2 con la instrucción $x = 2$; el valor de y seguirá siendo 1.5 a menos de que se modifique manualmente el valor de la variable y .

Editor

La ventana **Editor** se muestra cuando se abre un nuevo archivo, esto haciendo uso del botón *nuevo guión (script)* disponible en la barra de herramientas. Esto despliega la ventana que se señala con color azul en la figura (2). Esta ventana se utiliza para crear, editar y guardar códigos por parte del usuario.

Para ejecutar un código creado en esta ventana (lo que comúnmente se conoce como correr el código) se debe guardar el archivo y luego seleccionar el botón *Ejecutar (Run)* de la barra de *Editor*. Los archivos creados con MATLAB son archivos de texto que **no pueden contener en el nombre caracteres especiales ni espacios**.

Una vez que se corra el código, todas las variables creadas quedan en la memoria de MATLAB y se pueden consultar desde en la Ventana de Comandos.

Ejercicio 1 En la ventana **editor** digite las siguientes líneas y corra el código.

```
format long % Este le dice a MATLAB que muestre las salidas con 15 dígitos
a=5;
b=(a^(7/2) + (a-3)^5)/(a+2025);
b
```

MATLAB le va a solicitar que guarde el archivo para poder compilar el código. Como nombre del archivo digite *mi_primer_codigo* y seleccione el directorio (carpeta) donde desea guardarlo. Recuerde que nombre de un archivo no puede contener caracteres especiales ni espacios. Una vez guardado el archivo, MATLAB le mostrará como resultado en la **Ventana de Comandos** el nombre del archivo que se compiló así como los resultados que se solicitaron en dicho archivo. En el caso de este ejercicio solamente se solicitó el valor de *b*.

```
>> mi_primer_codigo
b = 0.153452461668706
```

3 Vectores y matrices

Hay diferentes formas de definir un vector en MATLAB. En la ecuación (1) se define un vector fila de nombre *v* indicando explícitamente cada entrada del vector.

$$v = [10.25 \ 11.3 \ 10.50 \ 13 \ 12.5]; \quad (1)$$

Note el uso de paréntesis cuadrados, además, los números están separados con espacios para indicar las diferentes entradas del vector.

Es posible acceder a la entrada *i* del vector digitando $v(i)$, para $1 \leq i \leq 5$. MATLAB enumera la primera entrada de un vector con el índice 1, así el vector *v* definido anteriormente tiene 5 entradas, la primera tiene el valor de $v(1) = 10.25$, la segunda el valor de $v(2) = 11.3$ y así sucesivamente.

Los vectores son muy útiles para resolver diversos tipo de problemas. El siguiente ejemplo muestra una aplicación muy sencilla.

Ejemplo 5 Los tiempos, en segundos, obtenidos por cinco velocistas en los 100m planos son los siguientes:

10.25 11.3 10.50 13 12.5

Utilizando el **Command Window** de MATLAB realice lo siguiente:

a) Defina un vector *v* en MATLAB que guarde la información anterior.

Solución:

```
>> v=[10.25 11.3 10.50 13 12.5];
```

- b) El primer tiempo de la lista anterior fue una marca en uno de los entrenamientos de Nery Brenes antes de los juegos Olímpicos Río 2016. Calcule la velocidad de Nery en esa prueba de entrenamiento y guarde el dato en una variable llamada ***vnergy***.

Solución:

```
>> vnergy=100/v(1);
>> vnergy
vnergy = 9.7561
```

- c) Calcule la velocidad promedio de todos los velocistas durante dicha prueba y guardela en una variable llamada ***promTotal***.

Solución:

```
>> promTotal = ( v(1)+v(2)+v(3)+v(4)+v(5) )/5;
>> promTotal
promTotal = 11.510
```

Ejemplo 6 Considere el sistema de ecuaciones

$$\begin{cases} x - \frac{2}{5}y - \frac{z}{2} = 8 \\ y - \frac{2}{3}z = -9 \\ z = 6 \end{cases}$$

La matriz asociada al sistema es la matriz triangular superior

$$\begin{pmatrix} 1 & -2/5 & -1/2 \\ 0 & 1 & -2/3 \\ 0 & 0 & 1 \end{pmatrix}.$$

Defina $b = (8, -9, 6)^t$.

- Utilice el comando `help` en la ventana de comandos para ver la descripción del operador `\`, i.e, digite `help \`
- En un script, defina la matriz A y el vector b de la siguiente manera $A = [1, -2/5, -1/2; 0, 1, -2/3; 0, 0, 1]$; $b = [8; -9; 6]$; y resuelva el sistema $Ax = b$ utilizando el operador `\`.
- En la ventana de comandos digite cada uno de los siguientes comandos y precione enter. Describa brevemente que hace cada comando.
 - A
 - $A(:)$
 - $A(1 : 3, 1)$
 - $A(1 : 2, 2 : 3)$
- ¿Qué comando puede utilizarse para extraer la primera fila de A ?

Ejemplo 7 Introduzca en matlab la siguiente matriz:

$$A = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \end{pmatrix},$$

y realice lo siguiente:

- Asigne a un vector v la fila 2 de la matriz A .
- Reemplace la fila 2 por el vector $(-1, -2, 0, 1)$.
- Reemplace el bloque B definido por $2 \leq i \leq 3, 3 \leq j \leq 4$ de A por el bloque $C = \begin{pmatrix} -10 & -20 \\ -30 & -40 \end{pmatrix}$.

Nota 4 Otros comandos útiles para trabajar con matrices son los siguientes

- $A = \text{zeros}(n, n)$: Crea una matriz de tamaño $n \times n$ donde todas las entradas con ceros.
- $A = \text{zeros}(m, n)$: Crea una matriz de tamaño $m \times n$ donde todas las entradas con ceros.
- $A = \text{eye}(m, n)$: Crea una matriz $m \times n$ con 1's en la diagonal principal.
- $B = \text{tril}(A)$: extrae la parte triangular inferior de la matriz A y la asigna a la matriz B .
- $B = \text{triu}(A)$: extrae la parte triangular superior de la matriz A y la asigna a la matriz B .
- $v = \text{diag}(A, k)$: extrae como vector columna la diagonal k -ésima de la matriz A .
- $n = \text{length}(v)$: asigna a la variable n el número de entradas del vector v .

4 Funciones predeterminadas

MATLAB tiene un gran número de funciones predeterminadas que pueden ser de mucha utilidad. En la siguiente tabla se encuentra algunas:

Función	Código para MATLAB	Comentarios
Valor absoluto	<code>abs(x)</code>	
Raíz cuadrada	<code>sqrt(x)</code>	La función <code>nthroot</code> se puede utilizar para calcular raíces con otros índices diferentes de 2.
Función exponencial	<code>exp(x)</code>	En particular note que <code>exp(1)</code> devuelve una aproximación para el valor de e .
Logaritmo natural	<code>log(x)</code>	
Logaritmo en base 2	<code>log2(x)</code>	
Logaritmo en base 10	<code>log10(x)</code>	
Función signo	<code>sign(x)</code>	
Función techo	<code>ceil(x)</code>	
Función parte entera	<code>floor(x)</code>	
seno	<code>sin(x)</code>	
coseno	<code>cos(x)</code>	
tangente	<code>tan(x)</code>	

Esta es solamente una muestra de la cantidad de funciones que están implementadas en MATLAB. Hay muchas más que puede consultar en la siguiente dirección la página de **MathWorks**.

5 Otros Comandos

MATLAB tiene un gran número de comandos. En la siguiente tabla se encuentra algunos:

- `clc`: Clear command window.
- `clearvars`: borra todas las variables de la memoria. También se puede especificar la variable(s) que se desea borrar.
- `whos x`: despliega información sobre la variable `x`.
- `help clc`: despliega la documentación del comando introducido, en este caso, `clc`.
- `diary on/off`: si se activa Matlab creará un documento con todo el texto desplegado en la ventana de comandos. Si se desactiva, Matlab guarda el documento y deja de escribir el diario.