

Laboratorio 0

1. Introducción a Matlab

Introducción al entorno de trabajo de Matlab

Matlab es un sistema interactivo cuyo elemento básico para el manejo de datos son arreglos que no requieren ser dimensionados. El nombre de Matlab proviene de la contracción de Matrix y Laboratory y fue originalmente escrito para facilitar el acceso a *software* matricial de otros proyectos. Actualmente es utilizado en diferentes áreas por su versatilidad y facilidades que ha ido incorporando a través de los llamados “toolboxes”. La versión de Matlab que estaremos utilizando en el laboratorio, corre bajo el sistema operativo Windows y el proceso a seguir para iniciar Matlab es como sigue:

- Pulse sobre el botón “Inicio” de la barra de tareas de Windows, seleccione “Programas” y seguidamente “Matlab”.
- En la ventana de comandos aparece un texto seguido de la señal de espera de entrada (“prompt”) de Matlab en forma de `>>` .
- Para finalizar la sesión bastará con teclear `exit` o `quit`, o simplemente cerrar la ventana de Matlab.

La ventana por defecto de Matlab (Desktop → Desktop Layout → Default) puede verse en la Figura 1, en donde podemos distinguir cuatro mini ventanas:

1. **Current Directory** (izquierda): es el directorio o ruta actual de Matlab y se puede navegar como el si fuera el explorador de Windows. Es importante verificar que la ruta actual sea la deseada. Mensajes de error como `??? Undefined function or method...` son muy comunes si no se está en el directorio correcto.
2. **Command Window** (centro): donde se ejecutan todas las instrucciones en Matlab.
3. **Workspace** (derecha y arriba): variable existentes en la sesión actual.
4. **Command History** (derecha y abajo): instrucciones recientemente utilizadas.

Es conveniente hacer uso intensivo de la ayuda de Matlab accediendo a ella desde el menú en el tope de la hoja de Matlab o tecleando en la ventana de comandos `help <palabra clave>` para la ayuda sobre el comando especificado de `<palabra clave>`. Por ejemplo, `>> help for` invoca la ayuda sobre el comando `for`.

Algunas puntos a tomar en cuenta al usar Matlab:

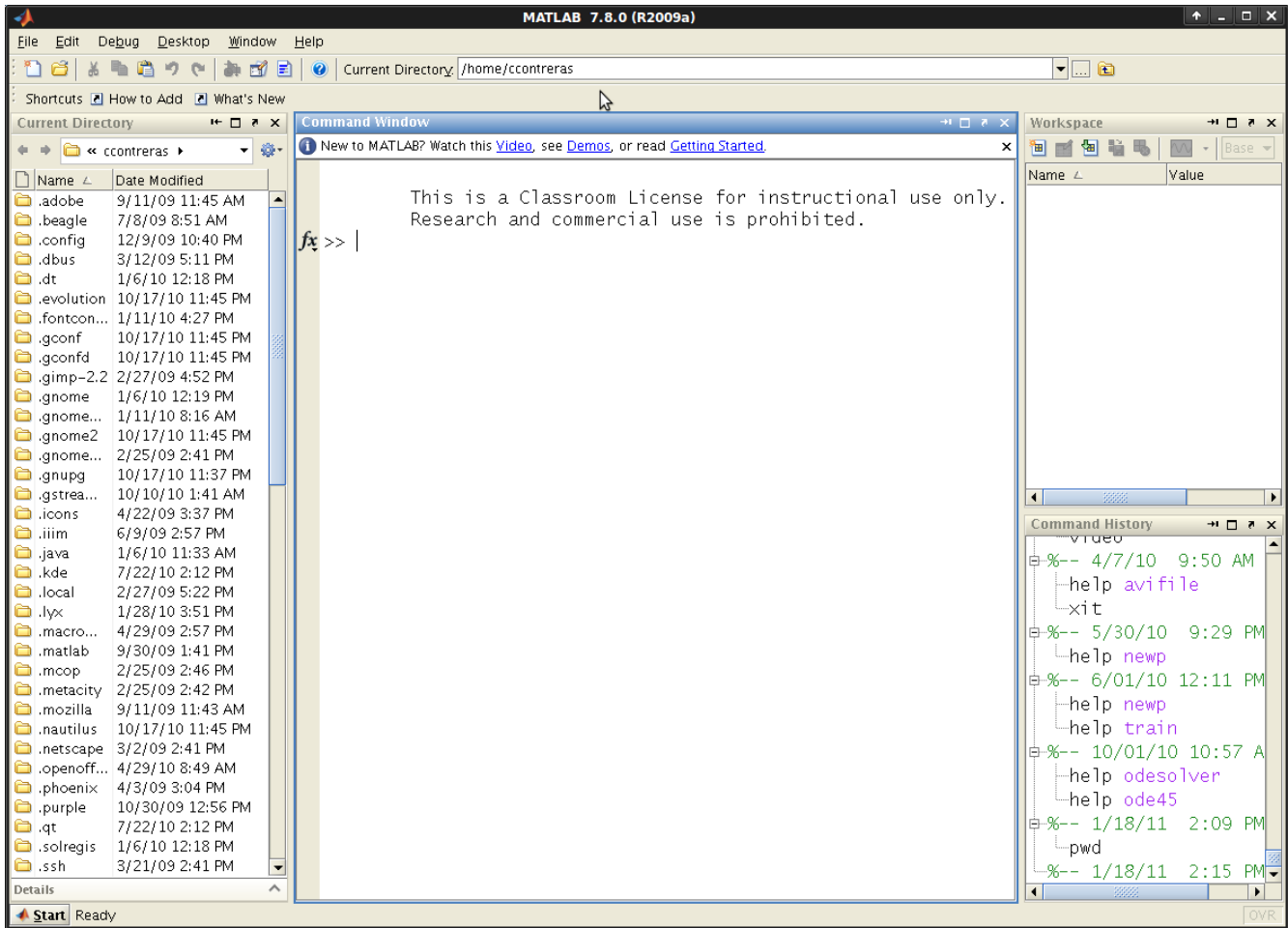


Figura 1: Ventana de Matlab por defecto

- Se pueden utilizar las flechas ↓ y ↑ para evocar los comandos introducidos anteriormente.
- Las letras mayúsculas y las minúsculas para Matlab no son equivalentes: por ejemplo, **valor** y **Valor** serían dos variables diferentes.
- Un punto y coma (;) al final de cualquier expresión suprime la salida del resultado por la pantalla (pero no la evaluación de dicha expresión).
- En Matlab se puede usar tanto paréntesis () como corchetes [], pero éstos tienen significados diferentes: los paréntesis sirven para agrupar expresiones e indicar los argumentos de funciones, mientras que los corchetes especifican vectores y matrices.

Operaciones básicas en Matlab

La forma más sencilla de utilizar Matlab es hacerlo funcionar como una calculadora. Las operaciones básicas disponibles son: + (suma), - (resta), * (producto), / (división), ^ (potenciación). Ejemplo, calcule: a) $156 - 123$, b) $456/56$, c) $152^{1.5}$, d) $(135 - 25/3) * 10^3$.

El resultado de las operaciones es asignado por defecto a la variable **ans** (de “answer”), pero puede asignarse a otras variables. Éstas pueden tener cualquier nombre que cumpla con las reglas

habituales (caracteres alfanuméricos, empezando por una letra). Por ejemplo, teclee: `x = 12 / 2;`. Esta instrucción realiza la operación indicada y asigna el resultado a la variable `x` sin mostrar el resultado. Para imprimir en pantalla el valor de la variable `x` basta teclear: `x`.

La variable conserva su valor a lo largo de toda la sesión, salvo que se le asigne otro o se le prive de valor por medio de la instrucción: `clear x`. Varios comandos de Matlab pueden ir en una misma línea, separados por una coma o un punto y coma (el punto y coma suprime la salida del resultado por pantalla; así, si únicamente desea que se asigne el valor, pero no quiere ver el resultado, basta con añadir `;` al final de la instrucción). Casi todas las funciones elementales (y muchas de las funciones especiales) se encuentran incluidas en Matlab: funciones trigonométricas (`sin`, `cos`, `tan`, `acos`, `asin`, `atan`), logaritmos (`log`, `log10`), función exponencial (`exp`), raíz cuadrada (`sqrt`), funciones hiperbólicas (`sinh`, `cosh`), etc. Para utilizar cualquiera de estas funciones se debe colocar el argumento entre paréntesis. Ejemplo, teclee: `y = log(18*x)`

Revise el “help” de Matlab en la parte de funciones para conocer más del tema.

Matlab conoce también varias constantes universales, como `pi` (π), `e` y la unidad imaginaria `i` o `j`. Pruébelas... Ejemplo: Realice las siguientes operaciones: a) $(\sin(\pi/8))^{\log 5}$, b) $\sqrt{1 + \arctan 1^2}$.

Aritmética de la máquina

Matlab realiza todos los cálculos siguiendo el estándar IEEE en punto flotante, doble precisión. El valor del ϵ de la máquina se puede comprobar con el comando: `eps` El valor NaN (“Not-A-Number”) de Matlab está en correspondencia con el estándar mencionado, y es producido por cualquier “indeterminación”, por ejemplo: `0/0`. Una vez generado, es propagado a cualquier cálculo: `0*NaN`.

El mayor número real positivo con representación normalizada en doble precisión, es almacenado por Matlab en la variable `realmax` (respectivamente, el menor número real positivo es almacenado en la variable `realmin`). a) Compruebe estos valores. b) ¿Qué valor espera obtener usted como resultado de cada una de las operaciones siguientes:

- `1.1 * realmax`
- `-2 * realmin`
- `eps * realmin`
- `eps * realmin / 2`
- `1023`
- `1024`
- `-10742`
- `-1075`

Verifíquelo. Calcule 2^{1023} , 2^{1024} , 2^{-1074} , 2^{-1075} . ¿qué sucede?

¿Qué resultado esperaría encontrar para `w` y `v` tras la ejecución del código MATLAB que sigue?
`w = 5; w = 2*w; v = w2; v = v/w;` Intente teclear `who`. ¿Qué observa? ¿Cuál es la diferencia de teclear `whos`? Si ejecuta los comandos `clear w; whos` ¿qué ocurre?

El editor de Matlab y la ejecución de programas y el manejo de errores

Con el Mouse valla a la opción new dentro del menú de file y marque la casilla **m-file**, lo que hará que abra el editor para un archivo de nombre *Untitled*. Ahora escriba los comandos dados en la sección anterior (una por línea y eliminando el punto y coma final) y verá como el editor las irá numerando a medida que apriete la tecla Intro.

Una vez completadas las instrucciones podrá salvar su programa (cuya extensión será **.m** por defecto) utilizando la opción save en el menú file. Note que el programa será salvado con el nombre que usted disponga en la carpeta donde se esté trabajando en ese momento, a menos que indique lo contrario; dicha carpeta y el path hasta ella, aparecen en la parte superior de la consola de Matlab, señalados como Current directory.

Una vez salvado el programa, bastará con escribir el nombre dado al mismo en la ventana de comandos de Matlab para que se ejecute. Es importante que cuando lo haga se asegure que el indicador de Current directory apunte hacia el directorio donde está el programa a ejecutar pues de lo contrario Matlab presentará un mensaje diciendo que no pudo encontrar el archivo en cuestión. Edite nuevamente el programa creado a través de la opción open del menú file y cambie la variable **w** por la variable **W** en la instrucción **v = v/w**, salve nuevamente el programa y ciérrelo antes de volver a correrlo (si el programa no es salvado antes de correrlo, Matlab trabajará con la copia del programa que tiene almacenada sin el cambio).

Note como Matlab envía un mensaje avisando que el programa tiene un error, mostrándole además la instrucción que contiene el error. Note también que aparece el nombre del programa subrayado seguido de un número; el número se refiere a la línea del programa donde se encuentra el error y haciendo clic con el mouse sobre la línea subrayada, el programa es automáticamente abierto en el editor y el cursor es enviado a la instrucción que contiene el error. Dependiendo del tipo de error, Matlab puede indicar también la posición del error dentro de la línea (columna donde aparece el error).

Variabes Lógicas

Cuando se utilizan operadores lógicos o relacionales, el resultado es una variable lógica que si bien puede ser utilizada como variable aritmética, tiene algunas características que permiten abreviar algunas operaciones matriciales.

Cree un vector **x** que contenga 10 números aleatorios con distribución normal.

Cree un vector **y** que contenga el resultado lógico de **x<0**.

Cree un vector **z = y * 1**.

Vea las características de cada variable en el workspace. ¿Qué obtiene si escribe la expresión **x(y)**? ¿Qué obtiene si escribe la expresión **x(z)**? ¿Son diferentes? ¿porqué?. Revise si existen comandos que conviertan variables lógicas en aritméticas y viceversa.