

INFORMÁTICA CIENTÍFICA MATLAB

**Matemática Superior Aplicada
3er. Nivel de la Carrera en Ingeniería Química
Universidad Tecnológica Nacional – Facultad Regional Rosario**

**Prof.: Dr. Alejandro S. M. Santa Cruz
Auxiliar Docente: Ing. Javier A. Francesconi**



ÍNDICE

- **Matlab Básico**
 - ✓ **Introducción**
 - ✓ **Variables, ctes., operadores y expresiones**
 - ✓ **Vectores y matrices**
 - ✓ **Funciones**

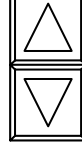
- **Matlab Avanzado**
 - ✓ **Análisis de datos**
 - ✓ **Polinomios**
 - ✓ **Gráficos**
 - ✓ **Uso de Matlab con Windows**
 - ✓ **Programación**

15/05/2007



Descripción de **MATLAB**

Alejandro S. M. Santa Cruz – Javier A. Francesconi



2

INTRODUCCIÓN

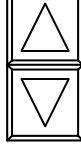
- **Lenguaje interpretado**
- **Ampliación: Toolboxes + Programas de usuarios**
- **Arranque desde Windows**
- **Ventana de comandos**
- **Ventana gráfica**
- **Demos**
- **Ayuda en línea**

15/05/2007



Descripción de MATLAB

Alejandro S. M. Santa Cruz – Javier A. Francesconi



3

FUNCIONES Y SÍMBOLOS DEL ENTORNO

- **Variables válidas hasta que se acaba la sesión.**
- **who: muestra las variables que hay.**
- **Grabarlas: save fichero o save fichero variables.**
- **Se recuperan con load fichero.**
- **Borrado: clear variables.**
- **! Cualquier comando del sistema operativo.**
- **ctrl-c: detener ejecución de comando.**
- **Salir: quit o exit.**

15/05/2007



Descripción de MATLAB

Alejandro S. M. Santa Cruz – Javier A. Francesconi



VARIABLES

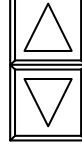
- Preparado para matrices (MATrix LABoratory).
- Variable: cualquier nombre: x, X, rad12, temp ...
- Asignación: a = 3; matriz = [1 2 3; 4 5 6]
 - ✓ Elementos: blancos o comas.
 - ✓ Filas: punto y coma o return.
 - ✓ Matriz: []
- Elementos reales, complejos o expresiones de otras variables y constantes.

15/05/2007



Descripción de MATLAB

Alejandro S. M. Santa Cruz – Javier A. Francesconi



5

VARIABLES DEL ENTORNO

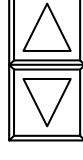
- **ans**: resultado del último comando cuando no se asigna.
- **pi**, **eps**, **inf**, **NaN**.
- **i**, **j** = $\sqrt{-1}$
- **flops** = número de operaciones.
- **clock**, **etime**: mide tiempos de cálculo.
- **Ejemplo**: ángulo = $180 \cdot \text{alfa} / \text{pi}$
- **format**: cambia formato de salida de datos
- **A** = $[1 + 5i, 2 - 4.5i]$

15/05/2007



Descripción de MATLAB

Alejandro S. M. Santa Cruz – Javier A. Francesconi



OPERADORES Y EXPRESIONES ARITMÉTICAS

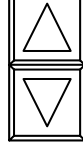
- **Constantes:** cualquier valor real, entero o cadena de caracteres: 3, -1.23e-3, 3i, 'Hola', ...
- **Precisión:** eps (epsilon): 16 cifras significativas
- **Operadores aritméticos:** +, -, *, /, ^
- **Funciones matemáticas:** sin, cos, log, sqrt, ...
- **Funciones creadas por el usuario:** archivos *.m
- **Expresión:** cualquier combinación válida de variables, constantes, operadores y funciones:
 - ✓ Alfa = 3 * sin(-0.5/x)
 - ✓ Cadena = 3 + 'AAA'. Resultado: cadena = [68 68 68]

15/05/2007



Descripción de MATLAB

Alejandro S. M. Santa Cruz – Javier A. Francesconi



VECTORES Y MATRICES

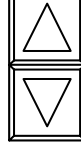
- Cualquier variable es una matriz.
- $A = [1 \ 2 \ 3; 4 \ 5 \ 6]$
- Transpuesta: $B = A'$. Resultado: $B = [1 \ 4; 2 \ 5; 3 \ 6]$
- Si es compleja: transpuesta conjugada.
- Suma y resta: siempre que tengan igual dimensión. Si una es un escalar: a todos los elementos de la otra se le suma dicho valor.
- Multiplicación: $*$. Dimensiones congruentes.
- Producto escalar: $x'*y$
- Vector por matriz o producto por escalar.

15/05/2007



Descripción de MATLAB

Alejandro S. M. Santa Cruz – Javier A. Francesconi



8

OPERACIONES CON MATRICES

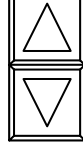
- División. Se hace sin invertir la matriz.
 - ✓ División derecha: $x = b/A$ es solución de $x*A = b$
 - ✓ División izquierda: $x = A\b b$ es solución de $A*x = b$
- Inversión: $B = \text{inv}(A)$ (si A es no singular)
- Potencias: A^p . A cuadrada y p entero
- `det`, `eig`, `rank`, `trace`, `real`, `imag`, `abs`
- Cualquier operador con un `.` es elemento a elemento:
 - ✓ $A.*B$ ó $A./B$
 - ✓ $z = x.^y$ siendo x y y vectores

15/05/2007



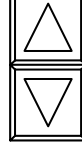
Descripción de MATLAB

Alejandro S. M. Santa Cruz – Javier A. Francesconi



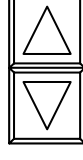
OPERADORES RELACIONALES

- Actúan sobre todos los elementos de la matriz.
- $<$, $<=$, $>$, $>=$, $==$, $~=$
- Se comparan los elementos y el resultado es otra matriz con ceros (0) si falso y unos (1) si cierto.
- $2 + 2 > 4$. El resultado es 0 (falso).
- Si $x = [1.3 \ 2.1]$ e $y = [1.3 \ 2.1]$, el resultado de:
 $z = x == y$ es $z = [1 \ 1]$
- Útil para expresiones condicionadas.



OPERADORES LÓGICOS

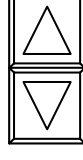
- Producto lógico (and): &
- Suma lógica (or): |
- Negación (not): ~
- $C = A \& B$ tiene elementos 1 donde A y B los tengan.
- `any(x)` devuelve 1 si alguno de los elementos es distinto de cero.
- `all(x)` devuelve 1 si todos los elementos son distintos de 0.
- `exist(x)` comprueba si la variable existe.
- `isinf(x)` detecta valores infinitos.
- `isempty(x)` detecta matrices vacías.



MANEJO DE VECTORES Y DE MATRICES (I)

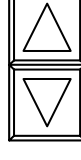
■ Generación de datos

- $x = 1:5$ genera el vector $x = [1\ 2\ 3\ 4\ 5]$
- $y = 0:0.1:0.5$ genera $y = [0\ 0.1\ 0.2\ 0.3\ 0.4\ 0.5]$
- En general vector = $v_inicial:incremento:v_final$
- Si se quiere un determinado número de puntos:
`linspace (v_inicial, v_final, puntos)`
- Para datos espaciados logarítmicamente:
`logspace (d1, d2, N)` genera N puntos entre 10^{d1} y 10^{d2}



MANEJO DE VECTORES Y DE MATRICES (II)

- **Acceso a los elementos**
 - **Elementos individuales $A(i, j)$. Ej. $A(1, 2) = -7.5$**
 - **El primer índice es el 1 (no el 0)**
 - **El índice puede ser a su vez un vector**
 - **Ejemplo $A = [1 \ 2 \ 3; 4 \ 5 \ 6]$**
 - ✓ $A(1:2,1)$ corresponde a la primera columna
 - ✓ $A(1, :)$ sería la primera fila
 - ✓ $A(1, [1 \ 3]) = 5$ pone un 5 en los elementos 1ro. y 3ero. de la 1era. fila



MANEJO DE VECTORES Y DE MATRICES (III)

- **Generación de Matrices a Partir de Otras**
 - Se pueden formar matrices mayores con elementos de otras matrices o de vectores.
 - Si $x = [1\ 2\ 3]$ y $P = [4\ 5; 6\ 7]$, se puede crear.
 $A = [x(1:2); P]$. Cuidado con las dimensiones!
 - **Manipulación:**
 - ✓ Rotación $\text{rot90}(A)$
 - ✓ Cambio de dimensiones $B = \text{reshape}(A, 2, 6)$ (linealizadas por columnas).
 - **Ver funciones en hoja resumen.**
 - **Matriz vacía $A = []$ (existe pero está vacía).**

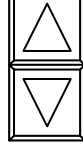


FUNCIONES

- **Existen subrutinas con argumentos:**
 - **Predefinidas**
 - **Toolboxes**
 - **Usuario**
 - **[s1, s2, ..., sm] = funcion(e1, e2, ..., en)**
 - **Ejemplos:**
 - ✓ **p = [1 2 1]; raices = roots(p)**
 - ✓ **datos = 1:10; media = mean(datos)**
- **Se irán viendo funciones concretas.**



15/05/2007



Descripción de MATLAB

Alejandro S. M. Santa Cruz – Javier A. Francesconi

15

MATLAB AVANZADO

- **Análisis de datos**
- **Polinomios**
- **Gráficos**
- **Programación**
- **Ficheros**

15/05/2007



Descripción de MATLAB

Alejandro S. M. Santa Cruz – Javier A. Francesconi



16

ANÁLISIS DE DATOS

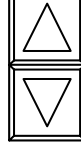
- Conjunto de datos en un archivo datos.dat
- Creados por cualquier programa. Por columnas.
- Para cargarlos:
 - ✓ Load datos.dat
 - ✓ Se crea una variable datos.
- Funciones: max, min, mean, median, std, sort, sum, prod, corrcoef, cov, ...
- Operan sobre matrices (varias columnas).
- Ejemplo: fichero con datos de temperaturas de un año.
 - ✓ Cálculo media, máxima y mínima. (Atención al path).

15/05/2007



Descripción de MATLAB

Alejandro S. M. Santa Cruz – Javier A. Francesconi



17

POLINOMIOS (I)

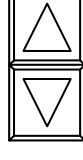
- Representación: como vector
- $x^3-6x^2-72x-27$: [1 -6 -72 -27]
- Potencias descendentes. 0 si falta alguno
- Raíces: roots(). Polinomio: poly()
- $p = [1 \ -6 \ -72 \ -27]$; $r = \text{roots}(p)$ (reales o complejas)
 - ✓ $r = [12.13 \ -5.73 \ -0.3884]$
 - ✓ poly(r) es p
- Producto: conv(p1, p2)
- Suma: suma de vectores p1 + p2

15/05/2007



Descripción de MATLAB

Alejandro S. M. Santa Cruz – Javier A. Francesconi



18

POLINOMIOS (II)

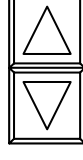
- Evaluación: `polyval(p, x)`
- Ajuste polinómico: `polyfit(x, y, n)`
 - ✓ Ajuste de $y = f(x)$ a un polinomio de grado n por mínimos cuadrados.
- Ejemplo: aproximar x^2+x+1 por una recta y calcular el error de aproximación en el intervalo $[0, 1]$
 - ✓ `plot(x, [y' yaprox'])`
- También ajuste por *splines*

15/05/2007



Descripción de MATLAB

Alejandro S. M. Santa Cruz – Javier A. Francesconi



19

GRÁFICOS

- **Elevadas prestaciones**
- **2 dimensiones**
 - ✓ **Lineal**
 - ✓ **Logarítmico**
 - ✓ **Polar, etc.**
- **3 dimensiones**
 - ✓ **Superficies**
 - ✓ **Diferentes vistas**
 - ✓ **Iluminación**

15/05/2007



Descripción de MATLAB

Alejandro S. M. Santa Cruz – Javier A. Francesconi



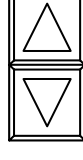
20

GRÁFICOS 2-D (I)

- **Ver demo**
 - ✓ `plot()`
 - ✓ `title`
 - ✓ `xlabel, ylabel`
 - ✓ `txt`
 - ✓ `gtext`
 - ✓ `grid`
- **Distintos colores, símbolos y tipos de línea.**
 - ✓ `plot(x, y, 'r-', x, z, 'g--')`
 - ✓ Varios dibujos con `hold`
- **Dibujar seno y coseno entre 0 y 180 grados en dos colores y ponerle etiquetas a los ejes.**



15/05/2007



21

GRÁFICOS 2-D (II)

- Si es una matriz, se dibuja una curva por cada fila.
- `plot(peaks)`, `plot(peaks, rot90(peaks))`
- Cambio de escalas axis()
- Tipos: `bar`, `hist`, `polar`, `stairs`, ...
- Variables complejas:
 - ✓ `plot(z)`. Equivale a: `plot(real(z), imag(z))`

15/05/2007



Descripción de MATLAB

Alejandro S. M. Santa Cruz – Javier A. Francesconi



22

GRÁFICOS 3-D

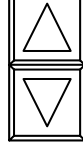
- **Muy potentes**
- **Líneas: `plot3(x, y, z)`**
 - ✓ `t = 0:pi50:10*pi`
 - ✓ `plot3(sin(t), cos(t), t)`. Hélice circular
- **Superficies: `mesh(z)`, donde `z` es una matriz**
- **`mesh(peaks)`, `contour(peaks, 20)` (20 curvas de nivel)**
- **`contour3(peaks, 20)`. Curvas de nivel en 3-D**
- **`view(az, el)`. Vista (azimut y elevación).**

15/05/2007



Descripción de MATLAB

Alejandro S. M. Santa Cruz – Javier A. Francesconi



23

PROGRAMACIÓN (I)

- Se pueden hacer programas y funciones
- Sentencias de control de flujo:
 - ✓ Bucles
 - ✓ Sentencias condicionales
- Bucle for. Iteraciones
 - ✓ for v = expresión sentencias end
 - ✓ for i = 1:n x(i) = 0 end
 - ✓ Anidados. Indentación

15/05/2007



Descripción de MATLAB

Alejandro S. M. Santa Cruz – Javier A. Francesconi



24

PROGRAMACIÓN (II)

- **Bucle while**

while expresion sentencias **end**

n = 1;

while prod(1:n) < 1e100, **n = n +1;**

end

n

- **Sentencia condicional if**

if expresión, **sentencia1**

else **sentencia2**

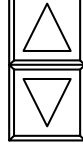
end

15/05/2007



Descripción de MATLAB

Alejandro S. M. Santa Cruz – Javier A. Francesconi



FICHEROS

- **Datos**
 - ✓ Formato MATLAB: save
 - ✓ Formato ASCII (por columnas). Para crearlo así desde MATLAB: save temp.dat A -ascii
 - ✓ Formateado con otro lenguaje
- **Comandos**
- **Funciones**
- **Importante el path**

15/05/2007



Descripción de MATLAB

Alejandro S. M. Santa Cruz – Javier A. Francesconi



26

FICHEROS *.m

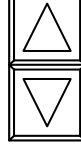
- Ficheros de:
 - ✓ Comando MATLAB
 - ✓ Funciones de usuario
- Como si fueran una función interna
- Function $y = \text{media}(x)$
 $[m, n] = \text{size}(x)$
If $m == 1$ $m = n$; end
 $y = \text{sum}(x)/m$;
- Uso: $p = \text{media}(z)$

15/05/2007



Descripción de MATLAB

Alejandro S. M. Santa Cruz – Javier A. Francesconi



27