

Matlab / Octave

Instructor

M.C. Miguelangel Fraga Aguilar

Origenes de Matlab

- Creado a finales de los años 1970 por Cleve Moler en la Universidad de Nuevo México.
- Se diseñó con el objeto de permitir a los estudiantes acceso a las librerías LINPACK y EISPACK (antecesoras de LAPACK) sin que tuvieran que aprender Fortran
- Es un lenguaje interpretado y multiparadigma, aunque incluye un compilador justo a tiempo en las últimas versiones
- Los operadores vectoriales llaman a las rutinas altamente optimizadas de BLAS y LAPACK (Intel Math Kernel Library, openBlas, Atlas)

MatLab / Octave

- MatLab – Programa para procesamiento numérico – Comercial desde 1984 – Matworks
- Octave – 95% compatible con Matlab a nivel de lenguaje de programación U. Texas – Open Source
- Otras alternativas Open Source: Scilab y Freemath
- (Casi) todo en MatLab es una matriz
- Se pueden usar matrices de números de punto flotante, de números complejos, de enteros, valores lógicos (verdadero 1, falso 0) y de caracteres
- No hay necesidad de declarar las variables y pueden cambiar de tipo dinámicamente

Operadores

- Se puede usar como una calculadora avanzada
- Operadores matriciales
 - = (asignación)
 - + - * (suma, resta, multiplicación)
 - / (Multiplicación por el inverso por la derecha)
 - \ (Multiplicación por el inverso por la izquierda o resolver un sistema de ecuaciones)
 - ^ (elevar a una potencia)

Creación de vectores y matrices

- Operador de concatenación:
 - $[1\ 2\ 3]$ concatenación horizontal (renglón)
 - $[1;2;3]$ concatenación vertical (columna)
 - $[1\ 2; 3\ 4]$ Matriz
- No es necesario declarar a las variables, se crean en su primer uso.
 $A=[1\ -5;-3\ 2]$
- Solución de un sistema de ecuaciones $Ax=b$, $x=A\backslash b$.
Ejemplo: $[1\ -5;-3\ 2]\backslash[4; -1]$

Creación de vectores (2)

- Vector renglón con enteros consecutivos: inicio:fin Ejemplos: 1:10
9900:10000
- Vector renglón con números reales con incremento arbitrario:
inicio:incremento:fin
Ejemplos: -1:0.1:1 0:0.5:100
- La función `linspace(inicio,fin,n)` con n elementos linealmente espaciados.
Ejemplo: `linspace(0,1,5)`
- Si no se desea ver el resultado de un comando se puede poner ; al final

Operadores Matriciales vs elemento a elemento

- La mayoría de los operadores de Matlab manejan las variables como matrices
 - + - * / \
- Operadores elemento a elemento
 - Requieren que los operandos sean vectores o matrices con las mismas dimensiones
 - Realizan la operación sobre los elementos correspondientes de los dos operandos para obtener un elemento del resultado
 - El resultado tiene el mismo tamaño que los operandos
 - .* ./ .^

Almacenamiento y carga de variables

- `save archivo` – almacena todas las variables actuales en el `archivo.mat`
- `save archivo v1 v2` – almacena las variables `v1` y `v2` en el `archivo.mat`
- `load archivo` – carga del `archivo.mat` todas las variables que contiene
- `var=cvsread('archivo.csv');` - lee el contenido del archivo de valores separados por comas como una matriz y lo almacena en la variable `var`
- `cvswrite('archivo.csv',var);` - escribe la variable `var` en el archivo de valores separados por comas

Funciones matemáticas

- La mayor parte de ellas puede trabajar tanto con escalares como con vectores y matrices
- Trigonométricas: `sin()` `cos()` `tan()` `asin()` `acos()` `atan()` `atan2()`
- Exponenciales: `exp()`, `log()`, `log10()`, `log2()`, `pow()`, `sqrt()`, `cbrt()`, `nthroot()`
- Complejos: `abs()`, `angle()`, `imag()`, `real()`
- Sumas y productos: `sum()`, `cumsum()`, `prod()`, `cumprod()`
- Redondeo: `ceil()`, `fix()`, `floor()`, `round()`

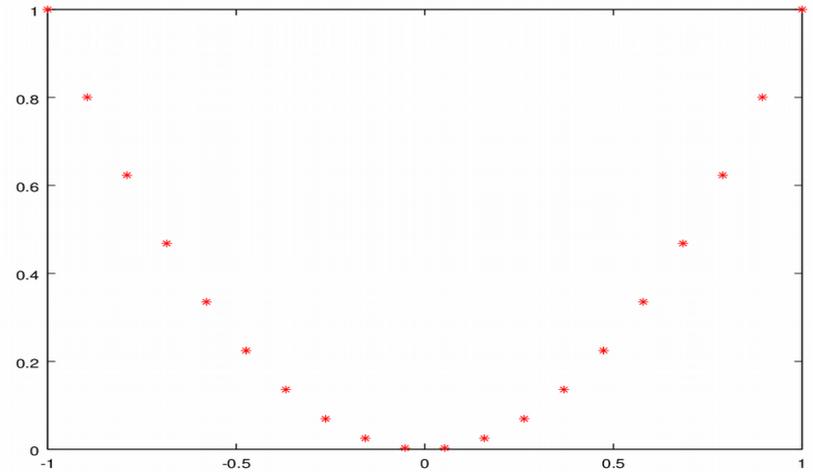
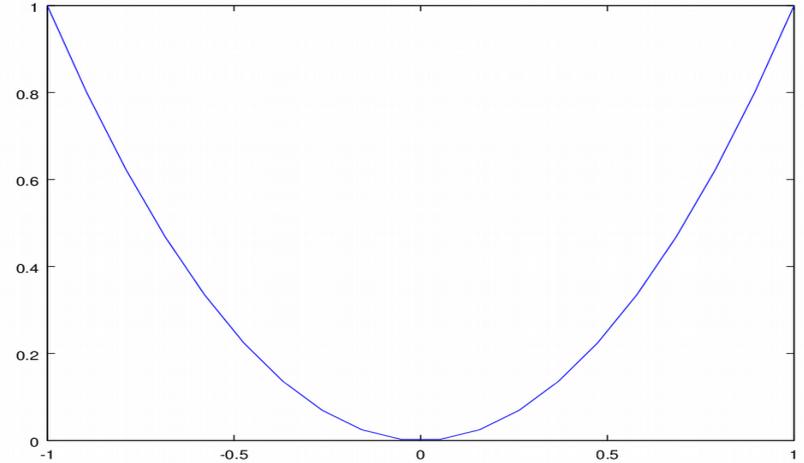
Gráficas de funciones (1 variable)

- Función `ezplot(cadena,[inicio,fin])`
 - `ezplot('x^2+x+10',[-2,2])`
- Función `plot(x,y,'formato')`
- Formato:
 - `.` solo poner los puntos, no unirlos con líneas
 - `+ * x` usar estos signos como marcadores de puntos
 - `--` usar línea discontinua para unir los puntos
 - `r g b c m y k` colores para los marcadores o la línea (rojo verde azul cian magenta amarillo negro)

Gráficas de funciones (2)

- `x=linspace(-1,1,20);`
- `plot(x,x.^2)`

- `plot(x,x.^2,'*r')`



Indexación de matrices

- vector(índice) – índice debe ser un entero
 - $b=v(5)$
 - $v(3)=0$
- matriz(r,c) - r es el índice del renglón y c el de la columna.
 - $b=A(3,5)$
 - $A(2,4)=0$
- Se puede usar un vector de índices para escoger varios elementos a la vez.
 - $V(2:5)$
 - $A(1:3,1)$

Indexación de matrices (2)

- Se pueden escoger todos los elementos de un renglón o columna con :
 - $A(11,:)$ Selecciona el renglón 11 de la matriz
 - $A(:,2)$ Selecciona la columna 2 de la matriz
- Un operador de comparación o lógico regresa un vector o matriz del mismo tamaño con 1 representando verdadero y 0 representando falso
 - $v==5$
 - $A \geq 0$

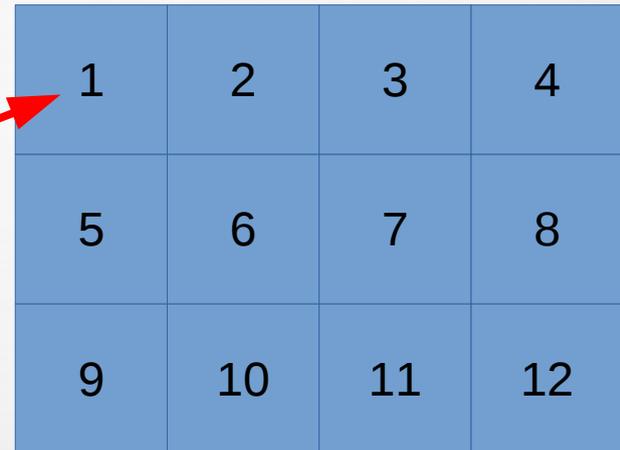
Indexación de matrices (3)

- Un vector de valores booleanos se puede usar como índice
 - $p=v(v \geq 0)$
 - $v(v > 5)=5$
- Se puede convertir una matriz en un vector con : - se concatenan las columnas verticalmente y se obtiene un vector columna de $n_{ren} \times n_{col}$
 - $v=A(:)$
- La función $A=\text{reshape}(v,n_{ren},n_{col})$ convierte de vector a matriz

Múltiples ejes en una figura

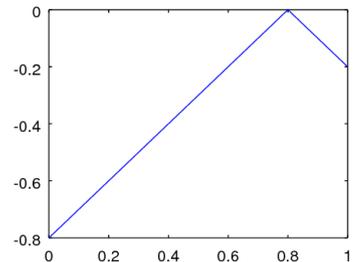
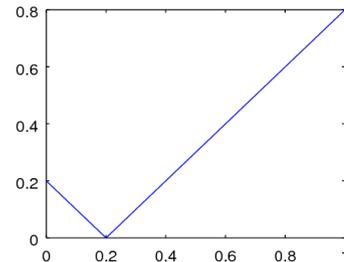
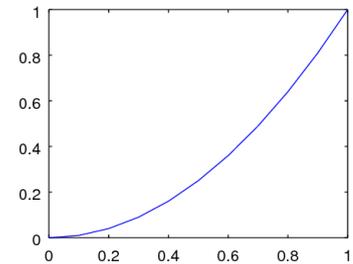
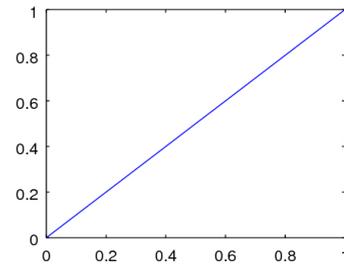
- `subplot(nren, ncol, indice)`
- Divide el espacio de la figura actual en `nren` renglones y `ncol` columnas
- `Indice` permite seleccionar en que posición se dibujara el resultado del siguiente comando gráfico

Indice



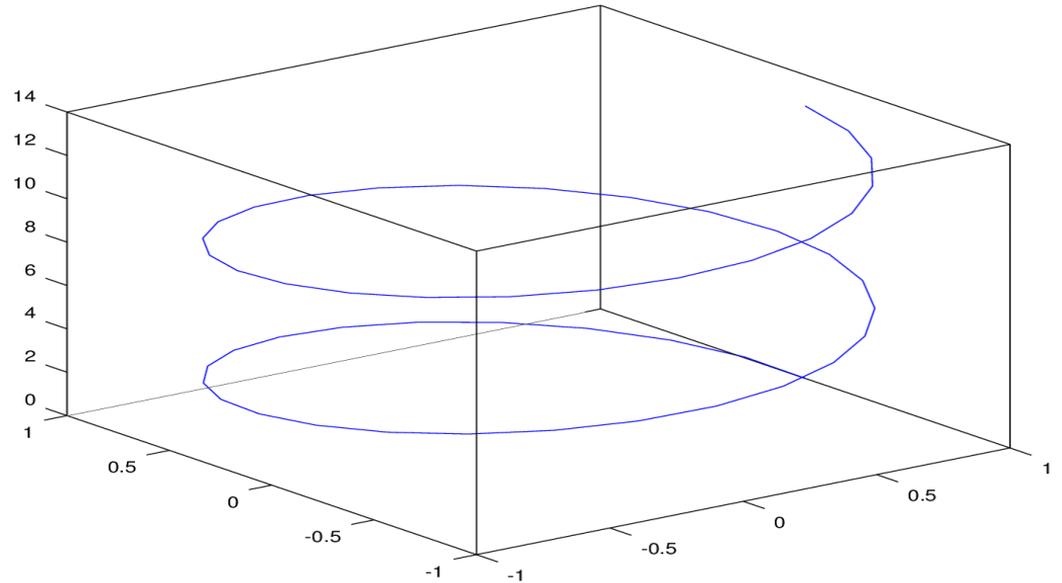
1	2	3	4
5	6	7	8
9	10	11	12

```
x=0:0.1:1;  
subplot(2,2,1);  
plot(x,x);  
subplot(2,2,2);  
plot(x,x.^2);  
subplot(2,2,3);  
plot(x,abs(x-0.2));  
subplot(2,2,4);  
plot(x,-abs(x-0.8));
```



Gráficas de curvas 3d

- `plot3(x,y,z,'formato')`
 - `t=linspace(0,8*pi,50);`
 - `plot3(cos(t),sin(t),t)`

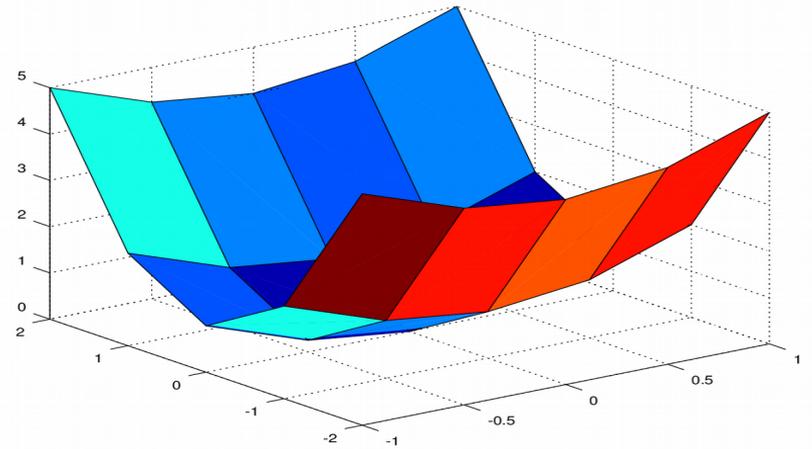
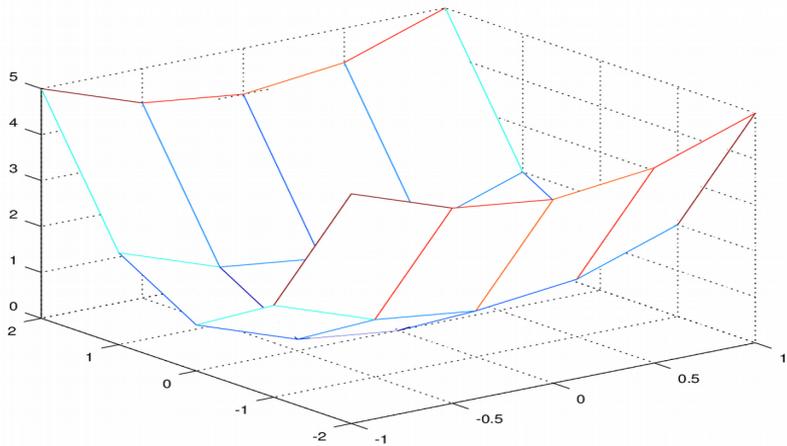


Gráfica de funciones de 2 variables

- $[X,Y]=\text{meshgrid}(v_x,v_y)$ – genera las matrices de rejilla con valores de x y y para graficar una función de 2 variables. v_x,v_y – vectores con los valores de x y y que se usaran en la rejilla
 - $[X,Y]=\text{meshgrid}(-1:0.5:1,-2:2)$

Gráficas de funciones de dos variables

- `mesh(X,Y,X.^2+Y.^2)`
- `surf(X,Y,X.^2+Y.^2)`



Algunas funciones útiles

- Leer la ayuda de una función: `help funcion`
- `repmat(var,n,m)` – replica una variable en un arreglo de n renglones por m columnas
- `reshape(var,n,m)` – cambia la forma de una variable a que sea una matriz de n renglones por m columnas
- Funciones estadísticas y descriptivas
 - `mean`, `var`, `std`, `min`, `max`

Archivos .m

- Son una serie de comandos de MatLab/Octave escritos en un archivo de texto y que se ejecutan secuencialmente
- Se pueden llamar como una sola orden (guiones o scripts) o como funciones.
- Deben encontrarse en el directorio actual o en el camino de búsqueda (path). Cambiar de directorio con “file browser” o el comando cd

Funciones en archivos .m

```
function [salidas]=nombre(entradas)
```

```
end
```

- El archivo .m debe llamarse igual que la función
- Se debe asignar el valor de retorno a los parámetros de salida antes de terminar la función

Estructuras de control

- if condición
 expresión
end
- while condición
 expresión
end
- for var=renglon
 expresión
end

Manejo de archivos

- Abrir el archivo fopen
- Entrada fscanf, fread
- Salida fprintf, fwrite
- Cerrar el archivo fclose