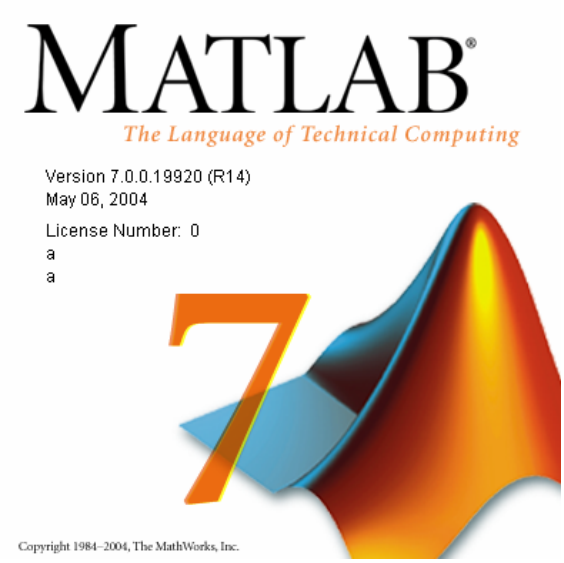
UNIVERSIDAD DE
CASTILLA-LA MANCHA
Escuela Superior de Informática

MATLAB

1.Capacidades Básicas de Matlab



- Matemáticas básicas
- Salvar y recuperar datos
- Formatos de numeración
- Variables
- Funciones matemáticas
- Ficheros de comandos
- Manejo de ficheros
- Arranque de Matlab

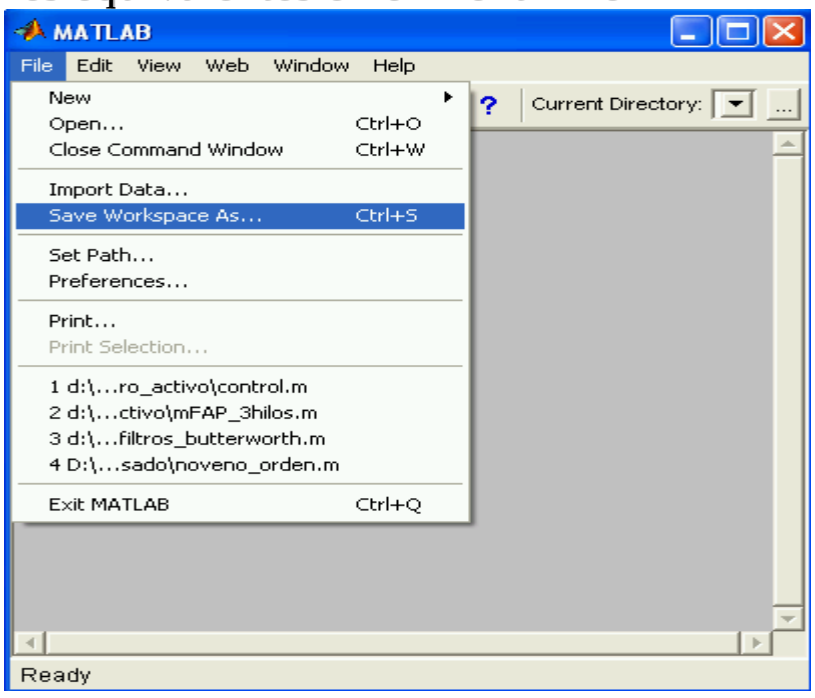
Matemáticas básicas

Operación	Símbolo	Ejemplo
adición	+	5+3
substracción	-	23-12
multiplicación	*	3.14*0.85
división	/ ó \	56/8 = 8\56
potencias	^	5^2

Las expresiones se expresan de izquierda a derecha con el mayor orden de precedencia en la potencia, seguida de la multiplicación y división (con la misma precedencia) y por último la suma y la resta (con la misma precedencia).

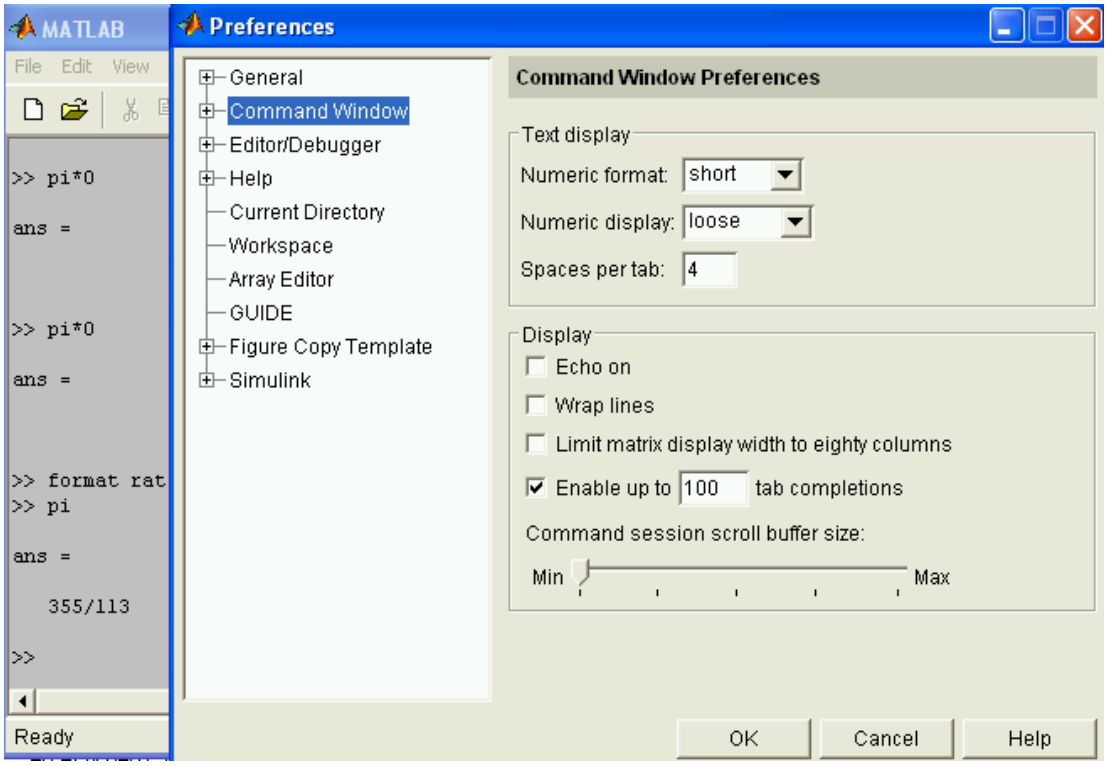
Salvar y recuperar datos

- save
  guarda todas las variables en el fichero “matlab.mat”
- load
  recupera la información de ese fichero
- save datos / load datos
  guarda/recupera todas las variables en el fichero “datos.mat”
- save data manzanas naranjas coste
  guarda las variables “manzanas”, “naranjas” y “coste” en el fichero “datos.mat”.
- Opciones equivalentes en el menú File



Formatos de numeración

Comando	ejemplo	características
format long	35.833333…4	16 dígitos
format short e	35.833e+01	5 dígitos+exponente
format long e	35.833…4e+01	16 dígitos+exponente
format hex	4041eaaaa…b	hexadecimal
format bank	35.83	2 dígitos decimales
format +	+	+, - ó 0
format rat	215/6	aproximación racional
format short	35.8333	por defecto



Variables

Variables especiales

- ans resultados por defecto
- pi relación entre la circunferencia y el diámetro de un círculo
- eps El menor número tal que añadido a uno da un número mayor que uno como resultado
- inf Infinito (1/0)
- NaN Not a Number ej: 0/0
- i, j número imaginario
- realmin El menor número positivo utilizable
- realmax El mayor número positivo utilizable

- clear manzanas borra esa variable
- clear borra todas las variables **sin esperar confirmación**
- hay memoria suficiente => no usar “clear”

Comentarios y puntuación

- Todo lo que aparece después del signo % es un comentario y, por lo tanto, Matlab lo ignora

```
>>manzanas=4  %es el número de manzanas
manzanas =
     4
```

- Se pueden poner múltiples comandos en la misma línea si se separan por comas o punto y coma:

```
>>manzanas=3, naranjas=2; fresas=5
manzanas=
     3
fresas=
     5
```

- Las "," muestran el resultado, los ";" no
- Los "…" le dicen a Matlab que la operación sigue en la siguiente línea (no se puede cortar una palabra):

```
>>1/…
4
ans =
   0.25
```

Funciones matemáticas

- Trigonometric.
  - sin - Sine.
  - sinh - Hyperbolic sine.
  - asin - Inverse sine.
  - cos - Cosine.
  - cosh - Hyperbolic cosine.
  - acos - Inverse cosine.
  - tan - Tangent.
  - tanh - Hyperbolic tangent.
  - atan - Inverse tangent.
  - atan2 - Four quadrant inverse tangent.
  - ……
- Exponential.
  - exp - Exponential.
  - log - Natural logarithm.
  - log10 - Common logarithm.
  - sqrt - Square root.

Funciones matemáticas III

- Complex.
  - abs - Absolute value.
  - angle - Phase angle.
  - conj - Complex conjugate.
  - imag - Complex imaginary part.
  - real - Complex real part.

- Numeric.
  - fix - Round towards zero.
  - floor - Round towards minus infinity.
  - ceil - Round towards plus infinity.
  - round - Round towards nearest integer.
  - rem - Remainder after division.
  - sign - Signum function.

Ficheros de comandos

» Funciones de Matlab útiles cuando se utilizan ficheros son:

Funciones útiles para los ficheros de comandos

disp	muestra los resultados
echo	eco en la ventana de comandos
input	entrada de valores por el usuario
keyboard	da el control al teclado temporalmente
pause	para hasta que se pulse una tecla
pause(n)	pausa de n segundos

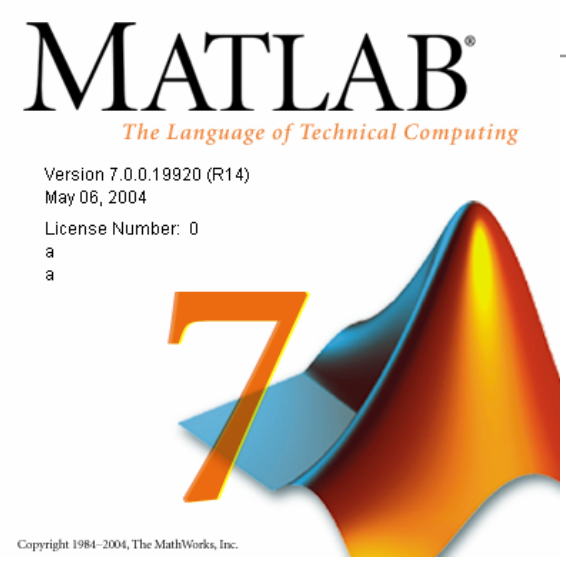
Arranque de Matlab

- Se ejecutan dos ficheros de comandos al arrancar:

  matlabrc.m startup.m
- matlabrc.m viene con Matlab y ***no debe ser modificado*** pues es donde se encuentran definidas características como las de las ventanas de figuras.
- startup.m añade características propias del usuario como el “path”.
- como es un fichero de comandos normal vale cualquier instrucción
- tener cuidado con no escribir en startup.m instrucciones demasiado específicas que se deben escribir en otros ficheros .m . Ejemplo:

  si se escribe quit en startup.m nunca arrancará Matlab

2.Matrices en Matlab



- Construcción de matrices
- Orientación de las matrices
- Operaciones entre escalares y matrices
- Operaciones entre matrices
- Manipulación de matrices
- Búsqueda en submatrices
- Tamaño de las matrices
- Funciones con matrices
- Operadores relacionales y lógicos

Construcción de Matrices II

Comandos útiles en la construcción de matrices

x=[2 2*pi sqrt(2) 2-3j]	creación de un vector fila que contiene los elementos especificados
x=primero:ultimo	desde el elemento primero hasta el último con incrementos de 1
x=prim:increm:ultim	desde prim hasta ultim con incrementos de increm
x=linspace(prim,ultim,n)	desde prim a ultim con n elementos en el vector
x=logspace(prim,ultim,n)	desde 10^prim hasta 10^ultim con n elementos (por si se desea utilizar una escala logarítmica)

Orientación de las matrices

- Vector columna:

```
» c=[1;2;3;4;5]
c =
     1
     2
     3
     4
     5
```

- y para hacerlo utilizando las formas rápidas ya vistas:

```
» a=1:5
a =
     1     2     3     4     5
» b=a'
b =
     1
     2
     3
     4
     5
```

Orientación de las matrices III

- También se pueden hacer matrices con varias filas y columnas, basta con separar las filas con ; o con retornos de línea y con tener cuidado de que todas las filas tengan igual número de elementos:

```
» g=[1 2 3 4;5 6 7 8]
g =
     1     2     3     4
     5     6     7     8
» h=[1 2 3 4
5 6 7 8
9 10 11 12]
h =
     1     2     3     4
     5     6     7     8
     9    10    11    12
» k=[1 2 3;4 5 6 7]
??? All rows in the bracketed expression must have the
    same number of columns.
```

Operaciones entre escalares y matrices

- Se puede operar como si las matrices fueran también escalares:

```
» h
h =
     1     2     3     4
     5     6     7     8
     9    10    11    12
» 2*h-2
ans =
     0     2     4     6
     8    10    12    14
    16    18    20    22
```

Operaciones entre matrices

```
» k=[1 1 1 1;2 2 2 2;3 3 3 3]  % nueva matriz
k =
     1     1     1     1
     2     2     2     2
     3     3     3     3
» h+k            % sumando elemento a elemento
ans =
     2     3     4     5
     7     8     9    10
    12    13    14    15
» h.*k           % multiplica elemento a elemento
ans =
     1     2     3     4
    10    12    14    16
    27    30    33    36
```

- Para multiplicar matrices basta con aplicar *

```
h*k
??? Error using ==> *
Inner matrix dimensions must agree.
```

- Pero hay que tener cuidado de que los rangos de las matrices permitan la operación.

Operaciones entre matrices II

Operaciones con matrices elemento a elemento

datos:	a=[a_1 a_2 … a_n], b=[b_1 b_2 … b_n], c=escalar
suma de escalar	a+b = [a_1+c a_2+c … a_n+c]
multiplicación	a*c = [a_1*c a_2*c … a_n*c]
suma de matrices	a+b = [a_1+b_1 a_2+b_2 … a_n+b_n]
multip. de matrices	a.*b = [a_1*b_1 a_2*b_2 … a_n*b_n]
división de matrices	a./b = [a_1/b_1 a_2/b_2 … a_n/b_n]
división inversa de matrices	a.\b = [a_1\$b_1$ a_2\$b_2$ … a_n\$b_n$]
potencias	a.^c = [a_1^c a_2^c … a_n^c] c.^a = [c^a_1 c^a_2 … c^a_n] a.^b = [a_1^b_1 a_2^b_2 … a_n^b_n]

Manipulación de matrices I

```
» A=[1 2 3;4 5 6;7 8 9]   % Crea la matriz A
A =
     1     2     3
     4     5     6
     7     8     9
» B=A(3:-1:1,:) % Esto también vale
B =
     7     8     9
     4     5     6
     1     2     3
» C=[A B(:,[1 3])] % C añade dos columnas de B a A
C =
     1     2     3     7     9
     4     5     6     4     6
     7     8     9     1     3
```

Manipulación de matrices III

```
» B=A(1:2,2:3)
B =
     2     3
     5     6
» C=[1 3]
C =
     1     3
» B=A(C,C)
B =
     1     3
     7     9
» B=A(:)
B =
     1
     4
     7
     2
     5
     8
     3
     6
     9
```

```
» B=B.'
B =
     1     4     7     2     5     8     3     6     9
» B=A
B =
     1     2     3
     4     5     6
     7     8     9
» B(:,2)=[]
B =
     1     3
     4     6
     7     9
»
```

Manipulación de matrices III

Direccionamiento de matrices

A(r , c)	Direcciona una submatriz de A definida por el vector índice de las filas deseadas r y el vector índice de las columnas deseadas c.
A(r, :)	Direcciona una submatriz de A definida por el vector índice de las filas deseadas r y todas las columnas.
A(: , c)	Direcciona una submatriz de A definida por todas las filas y por el vector índice de las columnas deseadas c.
A(:)	Direcciona todos los elementos de A tomados columna a columna.
A(i)	Direcciona una submatriz de A definida por un único vector índice como si A fuese el vector columna A(:).
A(x)	Direcciona una submatriz de A definida por la matriz lógica x. Donde x debe contener solo valores 0 y 1, y debe tener el mismo tamaño que A.

Búsqueda en submatrices

Comandos de búsqueda en matrices

i=find(x)	Devuelve los índices de la matriz x donde sus elementos son distintos de cero.
[r, c]=find(x)	Devuelve las filas y columnas de los elementos de la matriz x que son distintos de cero.

```
» x=-3:3
x =
   -3 -2 -1  0  1  2  3
» k=find(abs(x)>1)
k =
    1    2    6    7
» y=x(k)
y =
   -3   -2    2    3
» A=[1 2 3;4 5 6;7 8 9]
A =
    1    2    3
    4    5    6
    7    8    9
```

```
» [i,j]=find(A>5)
i =
    3
    3
    2
    3
j =
    1
    2
    3
    3
```

Tamaño de las matrices

Comandos relacionados con el tamaño de las matrices

whos	Muestra las variables y su tamaño
s=size(A)	Devuelve un vector de dos elementos donde el primero es el número de filas y el segundo el número de columnas.
[r,c]=size(A)	Devuelve dos escalares r y c conteniendo el número de filas y columnas de A.
r=size(A,1)	Devuelve el número de filas de A.
c=size(A,2)	Devuelve el número de columnas de A.
n=length(A)	Devuelve max(size(A)).

Funciones de manipulación de matrices

Comandos relacionados con la manipulación de matrices

flipud(A)	Intercambia lo de arriba abajo
fliplr(A)	Intercambia de izquierda a derecha
rot90(A)	Rota una matriz en el sentido contrario a las agujas del reloj 90º
diag(A)	Extrae la diagonal de la matriz A como un vector columna
diag(v)	Crea una matriz diagonal con el vector v en su diagonal
tril(A)	Extrae la parte triangular inferior de la matriz A
triu(A)	Extrae la parte triangular superior de la matriz A

Funciones con matrices

MATLAB Command Window

File Edit Options Windows Help

```
» help matfun

 Matrix functions - numerical linear algebra.

 Matrix analysis.
   cond         - Matrix condition number.
   norm         - Matrix or vector norm.
   rcond        - LINPACK reciprocal condition estimator.
   rank         - Number of linearly independent rows or columns.
   det          - Determinant.
   trace        - Sum of diagonal elements.
   null         - Null space.
   orth         - Orthogonalization.
   rref         - Reduced row echelon form.

 Linear equations.
   \ and /      - Linear equation solution; use "help slash".
   chol         - Cholesky factorization.
   lu           - Factors from Gaussian elimination.
   inv          - Matrix inverse.
   qr           - Orthogonal-triangular decomposition.
   qrdelete     - Delete a column from the QR factorization.
   qrinsert     - Insert a column in the QR factorization.
   nnls         - Non-negative least-squares.
   pinv         - Pseudoinverse.
   lscov        - Least squares in the presence of known covariance.
```

MATLAB Command Window

File Edit Options Windows Help

```
 Eigenvalues and singular values.
   eig          - Eigenvalues and eigenvectors.
   poly         - Characteristic polynomial.
   hess         - Hessenberg form.
   qz           - Generalized eigenvalues.
   rsf2csf      - Real block diagonal form to complex diagonal form.
   cdf2rdf      - Complex diagonal form to real block diagonal form.
   schur        - Schur decomposition.
   balance      - Diagonal scaling to improve eigenvalue accuracy.
   svd          - Singular value decomposition.

 Matrix functions.
   expm         - Matrix exponential.
   expm1        - M-file implementation of expm.
   expm2        - Matrix exponential via Taylor series.
   expm3        - Matrix exponential via eigenvalues and eigenvectors.
   logm         - Matrix logarithm.
   sqrtm        - Matrix square root.
   funm         - Evaluate general matrix function.

»
```

Operadores relacionales

Operador relacional	Descripción
<	menor
<=	menor o igual
>	mayor
>=	mayor o igual
==	igual
~=	no igual

Nota: el símbolo ~ se obtiene manteniendo pulsada la tecla 'Alt' mientras se escribe el número 126.

- Ejemplo:

```
» A=1:9
A =
    1    2    3    4    5    6    7    8    9
» C=A>4
C =
    0    0    0    0    1    1    1    1    1
```

Operadores lógicos

Operador lógico	Descripción
&	AND
|	OR
~	NOT

- Ejemplo: C vale 1 para los valores de A que no son mayores que 4.

```
» A=1:9
A =
     1     2     3     4     5     6     7     8     9
» C=~(A>4)
C =
     1     1     1     1     0     0     0     0     0
```

Resumen de operadores y caracteres especiales

MATLAB Help

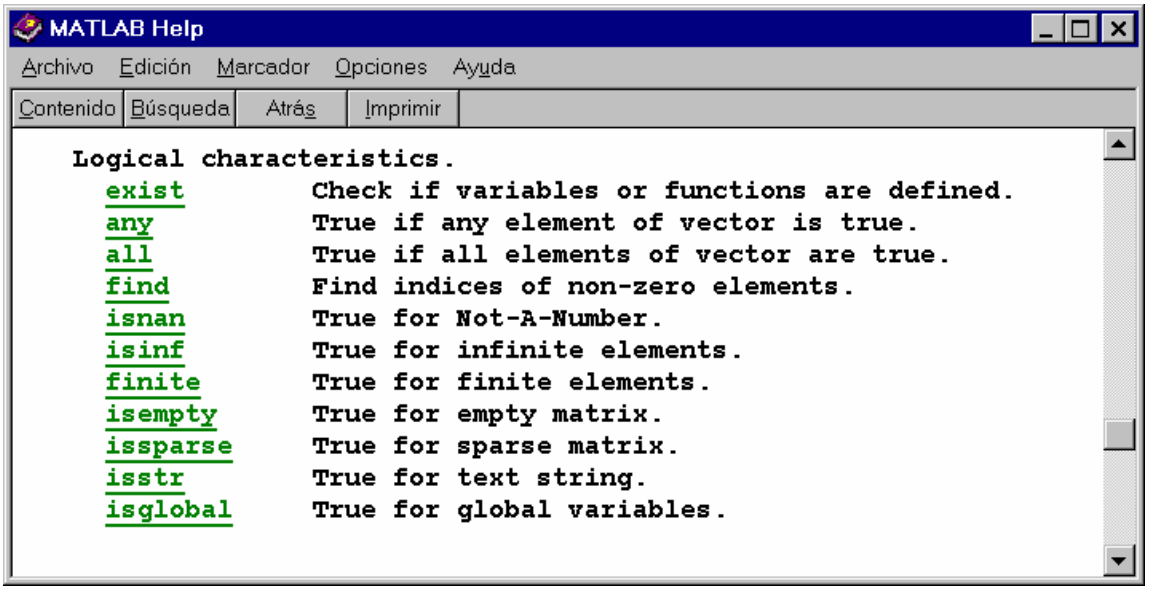
Archivo Edición Marcador Opciones Ayuda

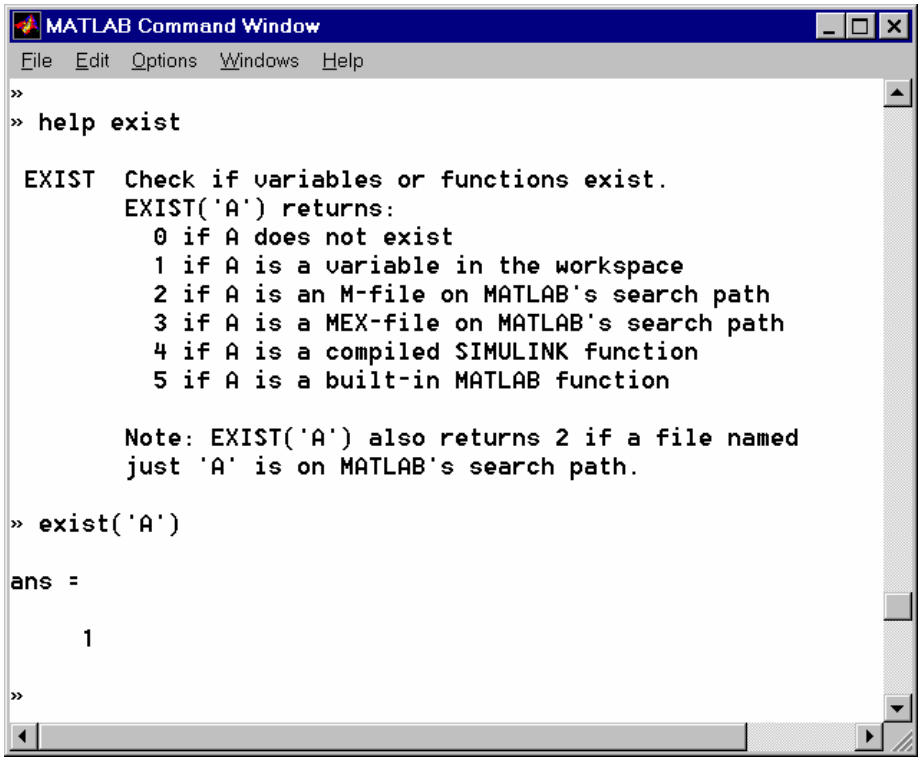
Contenido Búsqueda Atrás Imprimir

Operators and special characters.

Char	Name	HELP topic
+	Plus	arith
-	Minus	arith
*	Matrix multiplication	arith
.*	Array multiplication	arith
^	Matrix power	arith
.^	Array power	arith
\	Backslash or left division	slash
/	Slash or right division	slash
./	Array division	slash
kron	Kronecker tensor product	kron
:	Colon	colon
( )	Parentheses	paren
[ ]	Brackets	paren
.	Decimal point	punct
..	Parent directory	punct
...	Continuation	punct
,	Comma	punct
;	Semicolon	punct
%	Comment	punct
!	Exclamation point	punct
'	Transpose and quote	punct
=	Assignment	punct
==	Equality	relop
< >	Relational operators	relop
&	Logical AND	relop
|	Logical OR	relop
~	Logical NOT	relop
xor	Logical EXCLUSIVE OR	xor

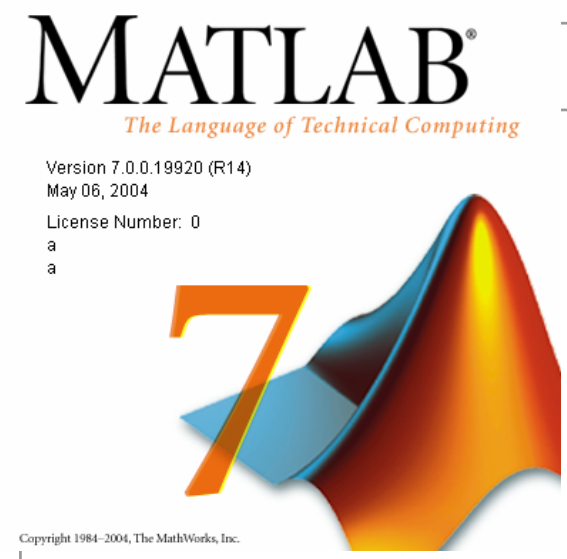
Funciones relacionales y lógicas





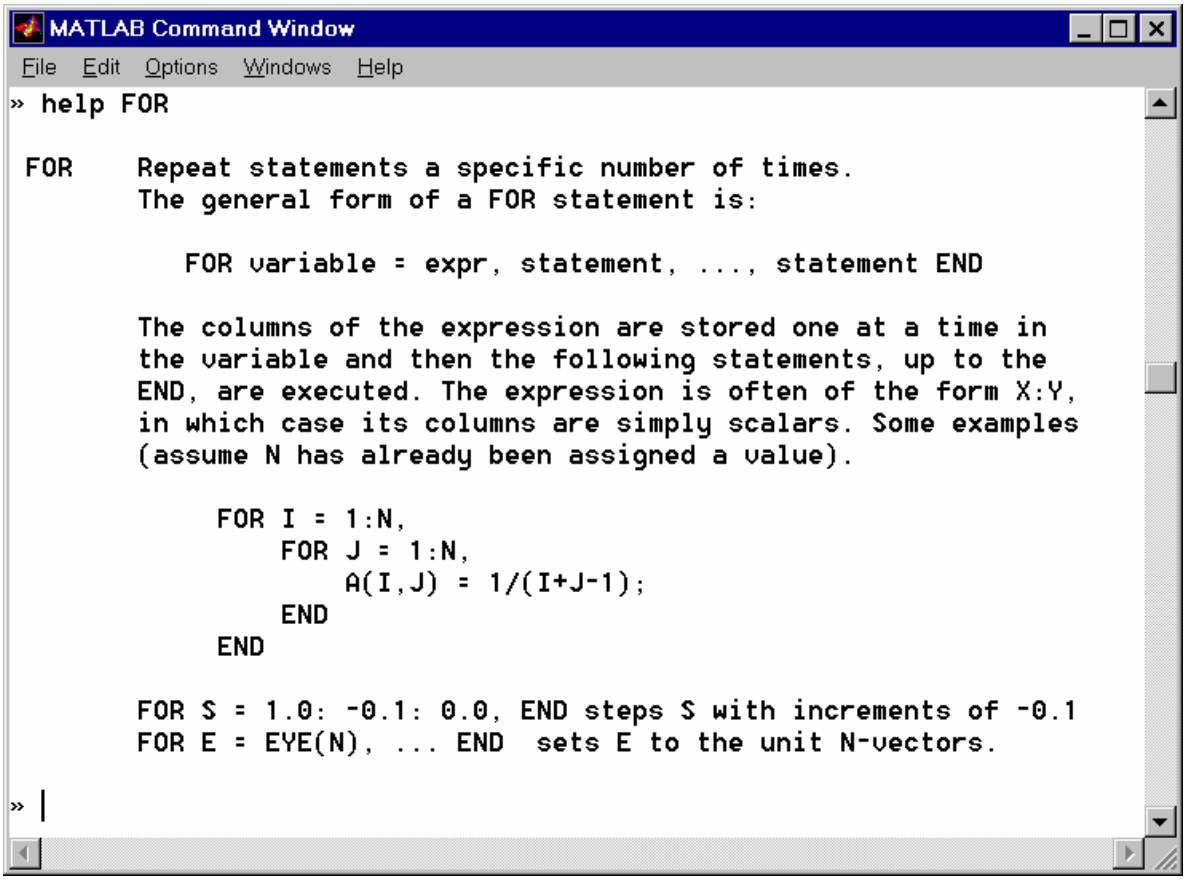
UNIVERSIDAD DE CASTILLA-LA MANCHA
Escuela Superior de Informática

3. Control de flujo y M-files



- Bucles FOR
- Bucles WHILE
- Estructuras IF-ELSE-END
- Sumario
- Funciones en ficheros de comandos M-files

Bucles FOR


MATLAB Command Window
File Edit Options Windows Help
» help FOR
FOR Repeat statements a specific number of times.
The general form of a FOR statement is:
FOR variable = expr, statement, ..., statement END
The columns of the expression are stored one at a time in
the variable and then the following statements, up to the
END, are executed. The expression is often of the form X:Y,
in which case its columns are simply scalars. Some examples
(assume N has already been assigned a value).
FOR I = 1:N,
FOR J = 1:N,
A(I,J) = 1/(I+J-1);
END
END
FOR S = 1.0: -0.1: 0.0, END steps S with increments of -0.1
FOR E = EYE(N), ... END sets E to the unit N-vectors.
»

Bucles FOR II

- Cualquier matriz de matlab es aceptada por un bucle FOR

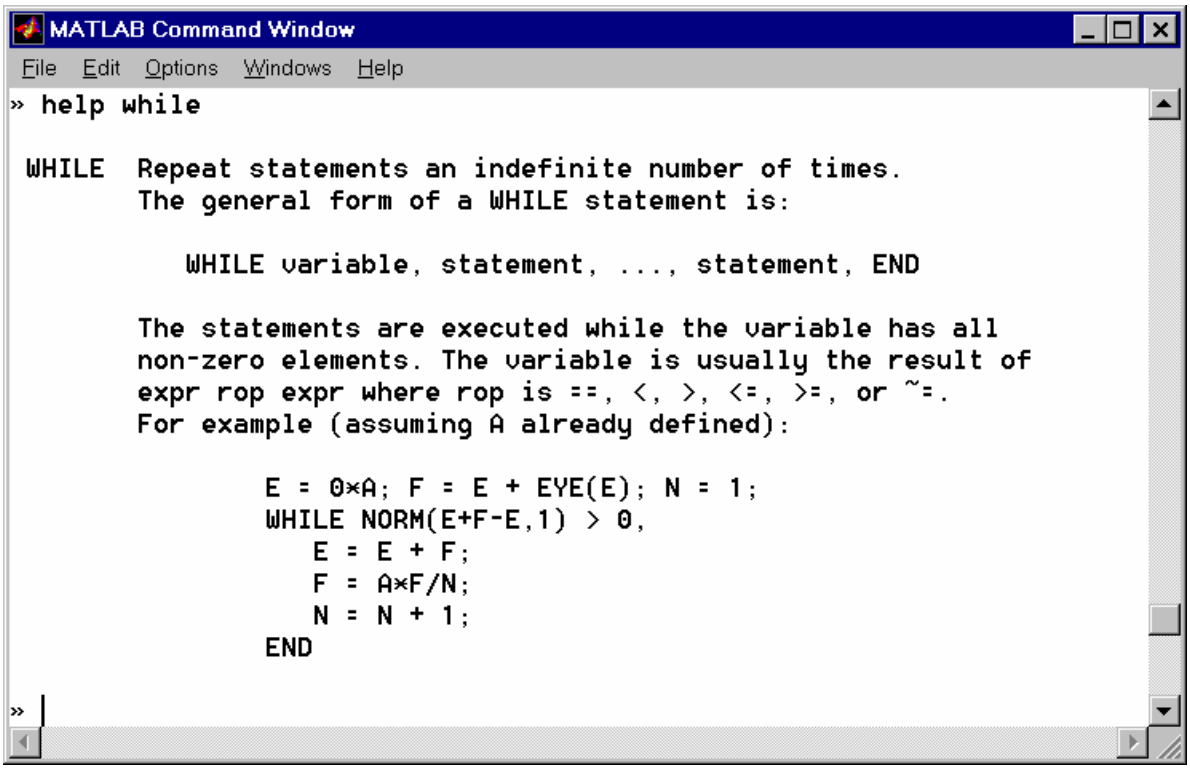
```
» datos=[3 9 45 6;7 16 -1 5]
datos =
     3     9    45     6
     7    16    -1     5
» for n=datos
    x=n(1)-n(2)
  end
x =
    -4
x =
    -7
x =
    46
x =
     1
»
```

Bucles FOR III

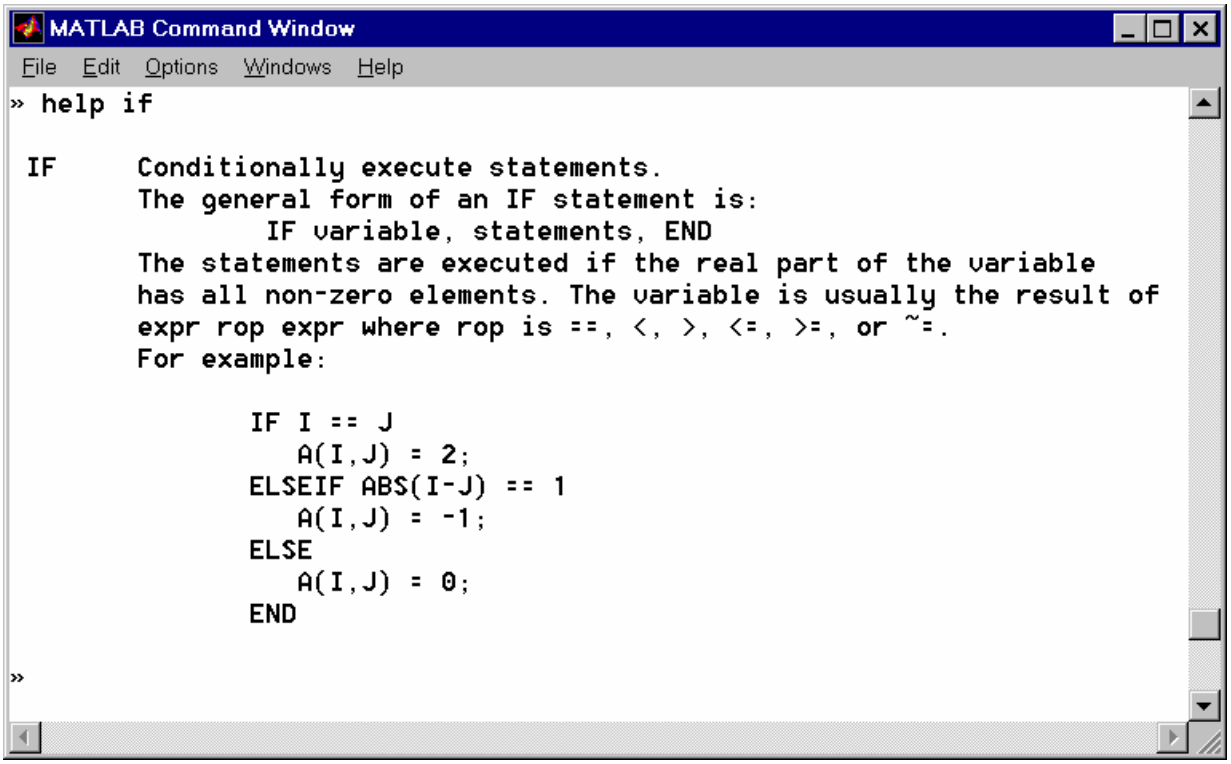
- Los bucles FOR pueden ser anidados a voluntad

```
» for n=1:5
    for m=5:-1:1
      A(n,m)=n^2+m^2;
    end
    disp(n)
  end
   1
   2
   3
   4
   5
» A
A =
    2    5   10   17   26
    5    8   13   20   29
   10   13   18   25   34
   17   20   25   32   41
   26   29   34   41   50
»
```

Bucles WHILE



Estructuras IF-ELSE-END



Sumario

Estructuras de control de flujo

for x=matriz comandos end	Un bucle FOR, que para cada iteración, asigna x y ejecuta los comandos
while expresión comandos end	Un bucle WHILE que ejecuta comandos mientras se cumpla la exprexión
if expresión comandos end	Se ejecutan los comandos si se cumple la expresión (todos sus elementos son verdaderos o <>0)

if expresión
  comandos1
else
  comandos2
end
	Si se cumple la expresión se ejecutan los comandos1 y si no se cumple se ejecutan los comandos2

if expresión1
  comandos1
elseif expresión2
  comandos2
elseif expresión3
…
else
  comandosE
end
	Si se cumple la expresión1 se ejecutan los comandos1,
	si no y se cumple la expresión2 se ejecutan los comandos2,
	si no y se cumple la expresión3
	…
	si no se cumple ninguna de las expresiones anteriores se ejecutan los comandosE

break	Termina la ejecución de un bucle

Funciones .m

- Cuando se utiliza una función de Matlab, como:
  inv, abs, angle, sqrt …
  esas funciones reciben los datos a la entrada, realizan una serie de operaciones que quedan ocultas al usuario y después devuelven los resultados.
- Los ficheros de comandos M-files vistos hasta ahora realizaban operaciones como si se estuviesen haciendo en la propia línea de comandos y, por tanto interaccionan directamnete con las variables del entorno.
- El nombre de la función y el del fichero que la contiene deben ser idénticos. Ej: la función fliplr está almacenada en el fichero fliplr.m.

Reglas y propiedades I

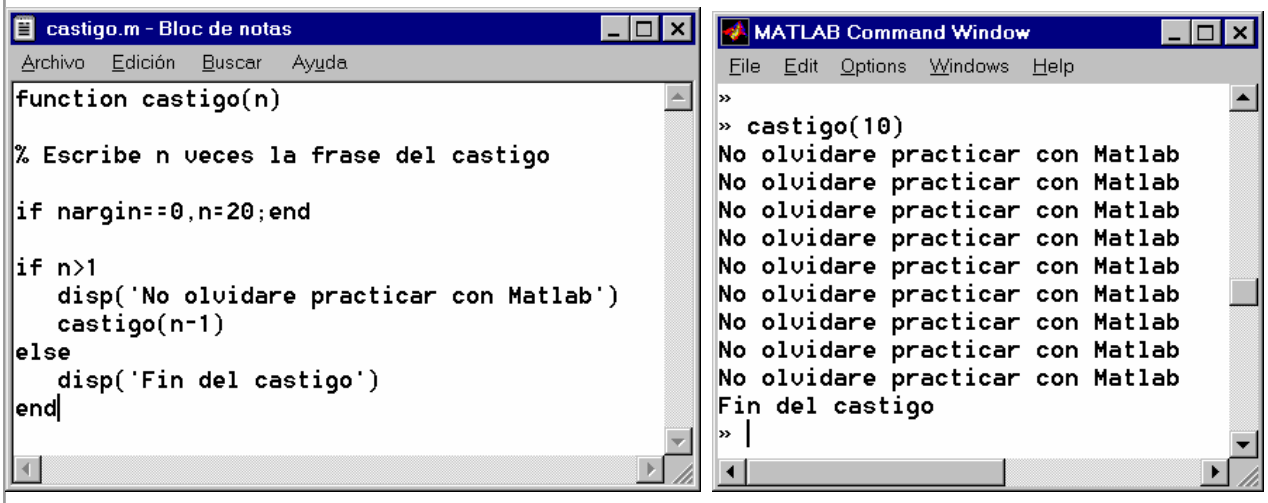
- Las funciones pueden tener cero o más argumentos de entrada y cero o más argumentos de salida.
- Las funciones pueden ser llamadas con menos argumentos de los especificados. Si se llaman con más se produce un error.
- Cuando una función tiene mas de una variable de salida, las variables de salida deben encerrarse entre corchetes. Ej: [v,d]=eig(A)
- El número de argumentos de entrada y de salida utilizados en la llamada a la función se encuentra accesible en el interior de la función en las variables **nargin** y **nargout**. Esto puede ser de utilidad cuando se pretende utilizar una variable para realizar operaciones difernetes según el número de argumentos con la que sea llamada.

Reglas y propiedades II

- Las funciones tienen su propio entorno de trabajo distinto del accesible desde la ventana de comandos. Las únicas conexiones entre el entorno de trabajo y la función son los argumentos de entrada y salida.
- Si una variable predefinida, por ejemplo ‘pi’ es redefinida dentro de una función, eso no afecta a posteriores operaciones realizadas en el entorno de trabajo de la ventana de comandos.
- Cuando se llama a una función los argumentos de entrada no se copian en el entorno de la función si no que se hacen legibles desde ese entorno. Si una variable de entrada es cambiada es entonces cuando se hace una copia para no modificar su valor en el entorno de trabajo de la ventana de comandos.
- Las funciones pueden compartir variables con otras funciones. Esto se hace declarando las variables como ‘global’ en ambas funciones. Ej.: la variable TICTOC en las funciones tic y toc.

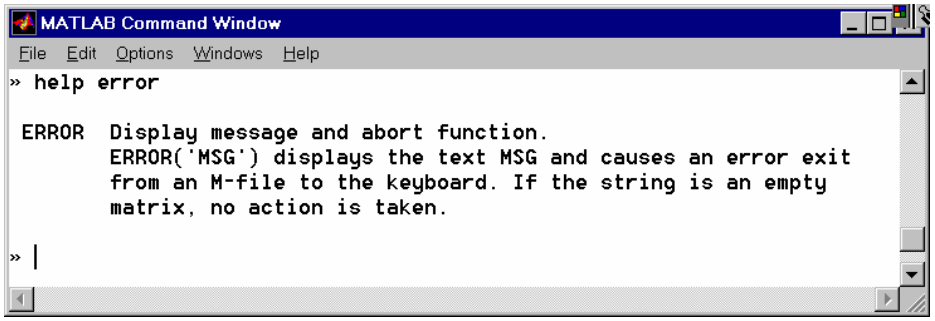
Reglas y propiedades III

- Debe evitarse en lo posible el uso de las variables globales. Si se usan es recomendable utilizar nombres largos en mayúsculas y claramente indicativos del uso de la variable.
- Matlab busca las funciones de ficheros M igual que lo hace para los ficheros M de comandos (como se vio anteriormente).
- Si se llama a un fichero de comandos desde una función el fichero de comandos comparte entorno con la función y no con la ventana de comandos.
- Las funciones pueden ser llamadas recursivamente, es decir: una función puede llamarse a sí misma.

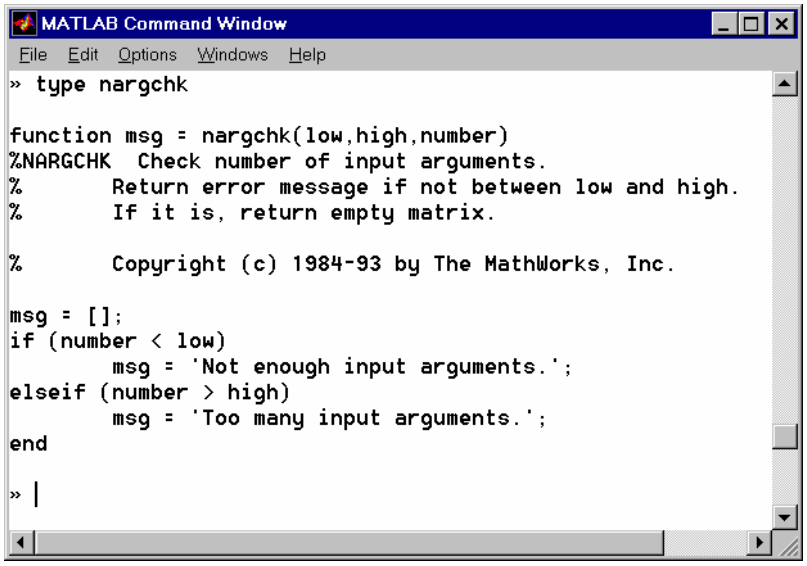


Reglas y propiedades IV

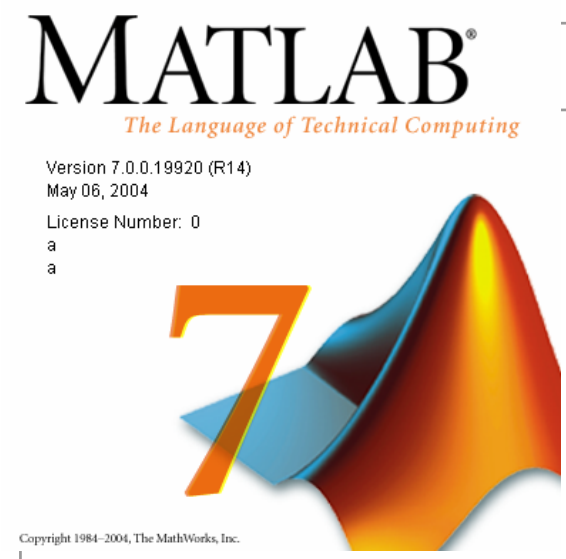
- Una función M termina cuando se alcanza el final del fichero-M en el que se encuentra o cuando se alcanza el comando ***return***
- La función ***error*** de Matlab permite hacer tratamiento de errores



- La función ***nargchk*** proporciona una respuesta uniforme cuando los argumanetos de entrada están fuera de los límites especificados



4. Entrada/Salida



- Entrada por teclado y Salida por pantalla
- Cargar y guardar ficheros.
  load save
- Abrir y cerrar ficheros
  fopen fclose
- Entrada/Salida de datos binarios.
  fread fwrite
- Conversión de cadenas.
  sprintf sscanf

Entrada por teclado y Salida por pantalla

- INPUT

Entrada por teclado: Para introducir información por teclado que es asignada a variables numéricas o alfanuméricas se utiliza el comando input

Var_num = input('prompt');

Var_string = input('prompt','s‘

- DISP

disp(X) muestra un array con el valor de X

Cargar y guardar ficheros

- LOAD

  Recupera variables desde el disco.

  LOAD fname recupera variables desde el fichero 'fname'.

  LOAD, por si solo carga desde el fichero 'matlab.mat'.

  LOAD xxx.yyy lee el fichero ASCII xxx.yyy, que debe contener una matriz rectangular de datos numéricos dispuestos den m líneas con n valores en cada una. El resultado será una matriz de mxn llamada xxx.

- SAVE

  Salva las variables del espacio de trabajo en disco.

  SAVE fname salva en el fichero ‘fname’.

  SAVE, por si solo salva en ‘matlab.mat’.

  SAVE fname X salva la variable X en el fichero ‘fname’.

  SAVE fname X Y Z salva las variables X Y y Z en el fichero ‘fname’.

Abrir y Cerrar ficheros

- FOPEN Abre un fichero.

  FID = FOPEN('fich',permiso) abre el fichero ‘fich’ con el permiso especificado. Donde el permiso puede ser uno de los siguientes:

'r'	lectura,
'w'	escritura (crea si es necesario),
'a'	añadir (crea si es necesario),
'r+'	lee y escribe (no crea),
'w+'	trunca o crea para lectura/escritura,
'a+'	lee y añade (crea si es necesario)

  Por defecto se crean en modo binario, si se desea abrir en modo text añadir una ‘t’ al permiso. Ej.: ‘rt’ ‘wt+’

  FID = FOPEN('filename') asume permiso 'r'.

  Si la apertura se realiza con éxito FID devuelve un valor escalar entero, el identificador de fichero, que podrá ser usado en otras funciones de Entrada/Salida.

  Si la apertura no tiene éxito FID devuelve el valor -1.

- FCLOSE Cierra un fichero.

  FCLOSE(FID) cierra el fichero de identificador FID.

  Devuelve 0 si es la operación es correcta -1 si no lo es.

Entrada/Salida de datos binarios

- FREAD Lee datos binarios de un fichero.

  [A, COUNT] = FREAD(FID,SIZE,PRECISION) lee datos binarios desde el fichero especificado y lo escribe en la matriz A. Si se utiliza COUNT devuelve el número de elementos leidos con éxito.

  El argumento SIZE es opcional, si no se especifica se lee todo el fichero. Si se especifica se puede hacer:

  N lee N elementos en un vector columna.

  inf lee hasta el final del fichero.

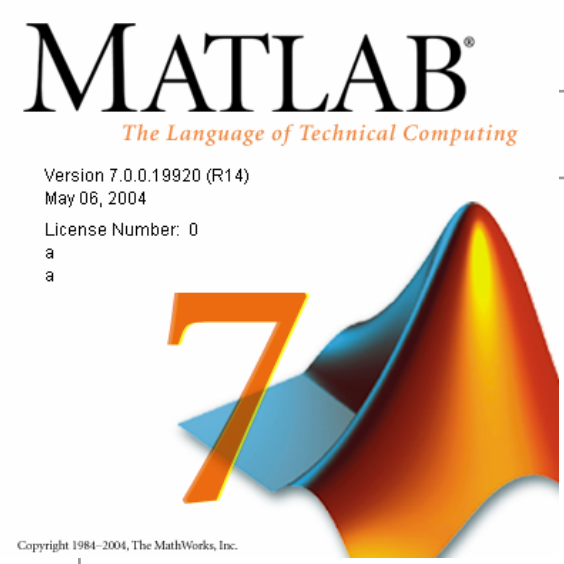
  [M,N] lee elementos hasta rellenar por columnas una matriz MxN.

  PRECISION controla la forma y tamaño del resultado.

- FWRITE Escribe datos binarios a un fichero.

  COUNT = FWRITE(FID,A,PRECISION) escribe los elementos de la matriz A (por columnas) en el fichero especificado con la precisión requerida.

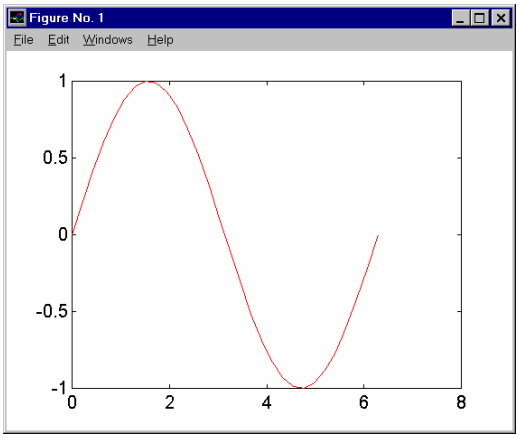
5. Gráficos 2-D



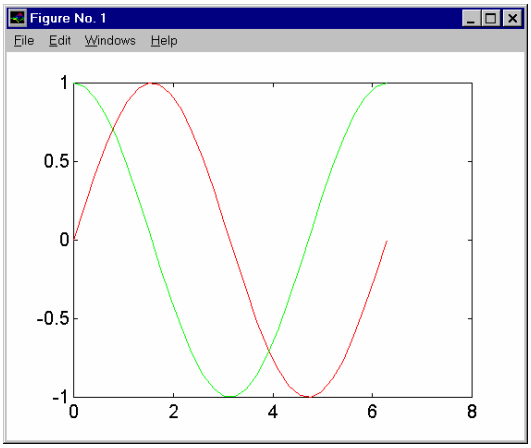
- La función “plot”
- Estilos de línea, marcadores y colores
- Cuadrículas y etiquetas
- Redefinición de los ejes
- Mantener gráficos
- Subplots
- Ventanas múltiples de figuras
- La función “ginput”
- Otros gráficos 2-D básicos
- Funciones especializadas para gráficos 2-D
- Sumario

La función “plot”

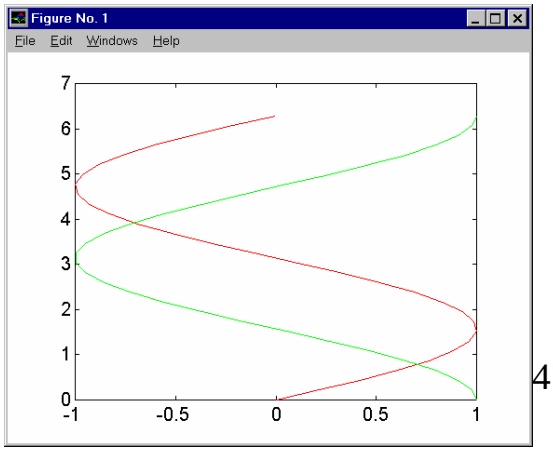
- Es el comando más común para la realización de gráficos 2-D en Matlab. Ej:
  - » x=linspace(0,2*pi,30);
  - » y=sin(x);
  - » plot(x,y)



- Y dos funciones en el mismo gráfico
  - » z=cos(x);
  - » plot(x,y,x,z)



- Si se cambia el orden de los argumentos el gráfico rota 90 grados
  - » plot(y,x,z,x)



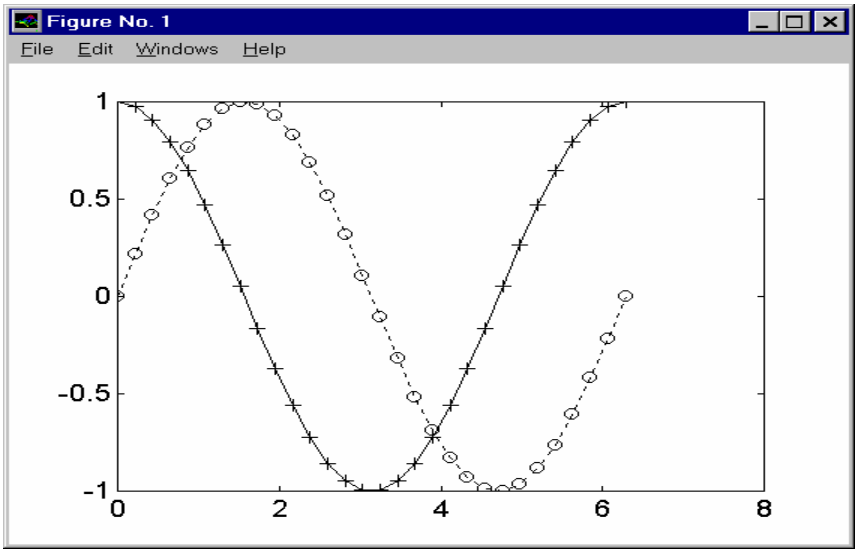
Estilos de línea, marcadores y colores

Tipos de línea y colores básicos

Símbolo	Color	Símbolo	Línea
y	amarillo	.	puntos
m	magenta	o	círculos
c	cyan	x	marcas x
r	rojo	+	signos mas
g	verde	*	asteriscos
b	azul	-	línea sólida
w	blanco	:	línea de puntos
k	negro	-.	Línea de rayas y puntos
		--	línea discontínua

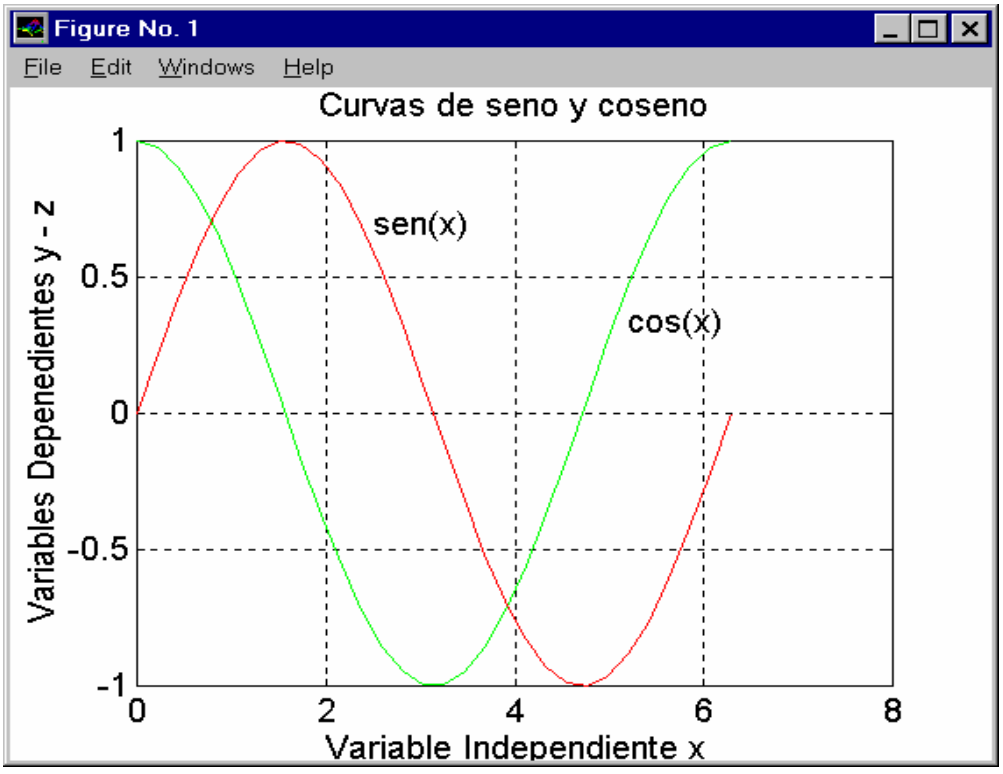
- Ejemplo:

```
» plot(x,y,'g:',x,z,'r-',x,y,'wo',x,z,'c+')
```



Cuadrículas y etiquetas

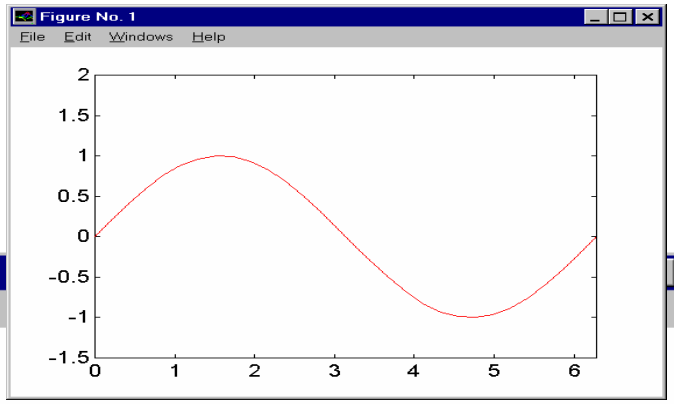
```
» x=linspace(0,2*pi,30);
» y=sin(x); z=cos(x);
» plot(x,y,x,z)
» grid  % activa la cuadrícula
» xlabel('Variable Independiente x')  % Etiqueta del eje x
» ylabel('Variables Depenedientes y - z')  % Etiqueta del eje y
» title('Curvas de seno y coseno')  % Título del gráfico
» text(2.5,0.7,'sen(x)')  % Texto en la posición indicada
» gtext('cos(x)')  % Coloca el texto con el ratón
```



Redefinición de los ejes

» x=linspace(0,2*pi,30); y=sin(x); plot(x,y)

» axis([0 2*pi -1.5 2]) % Cambio de los ejes



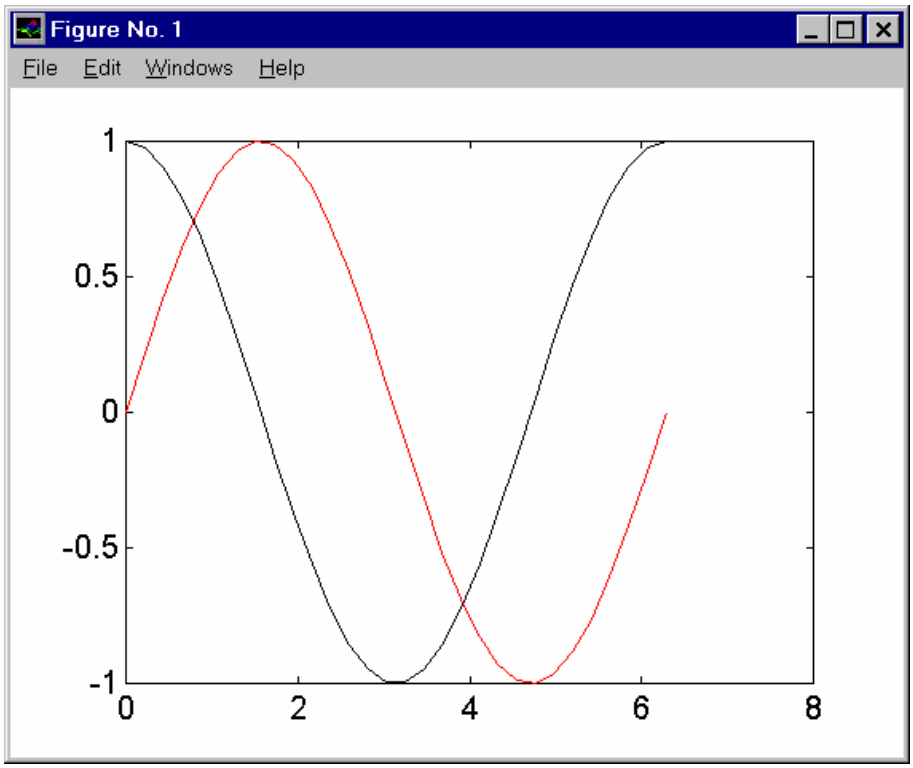
MATLAB Command Window

File Edit Options Windows Help

```
» help axis

 AXIS   Plot axis scaling and appearance.
        AXIS([XMIN XMAX YMIN YMAX]) sets scaling for the x- and y-axes
        on the current plot.

        AXIS([XMIN XMAX YMIN YMAX ZMIN ZMAX]) sets the scaling for the
        x-, y- and z-axes on the current 3-D plot.

        AXIS('auto') returns the axis scaling to its default, automatic
        mode where, for each plot, xmin = min(x), xmax = max(x), etc.

        V = AXIS returns a row vector containing the scaling for the
        current plot.  If the current plot is two-dimensional, V has
        four components; if it is three-dimensional, V has six components.

        AXIS(AXIS) freezes the scaling at the current limits, so that if
        HOLD is turned on, subsquent plots will use the same limits.

        AXIS('ij') puts MATLAB into its "matrix" axes mode.  The coordinate
        system origin is at the upper left corner.  The i axis is vertical
        and is numbered from top to bottom.  The j axis is horizontal and
        is numbered from left to right.
```

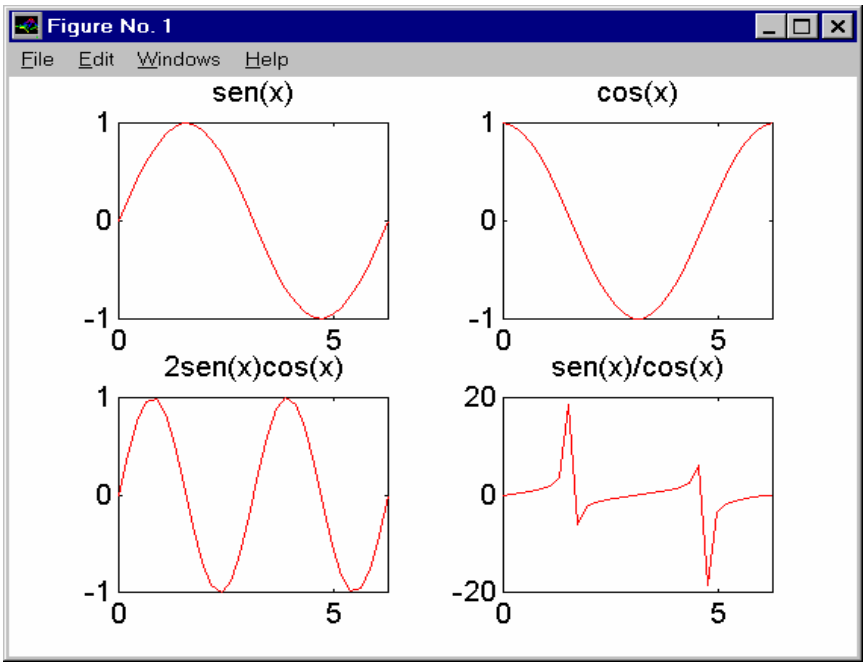
Mantener gráficos

```
» x=linspace(0,2*pi,30); y=sin(x); z=cos(x); plot(x,y)
» hold on
» ishold  % devuelve 1 si "hold" está "on"
ans =
     1
» plot(x,z,'k')
» hold off
» ishold
ans =
     0
```

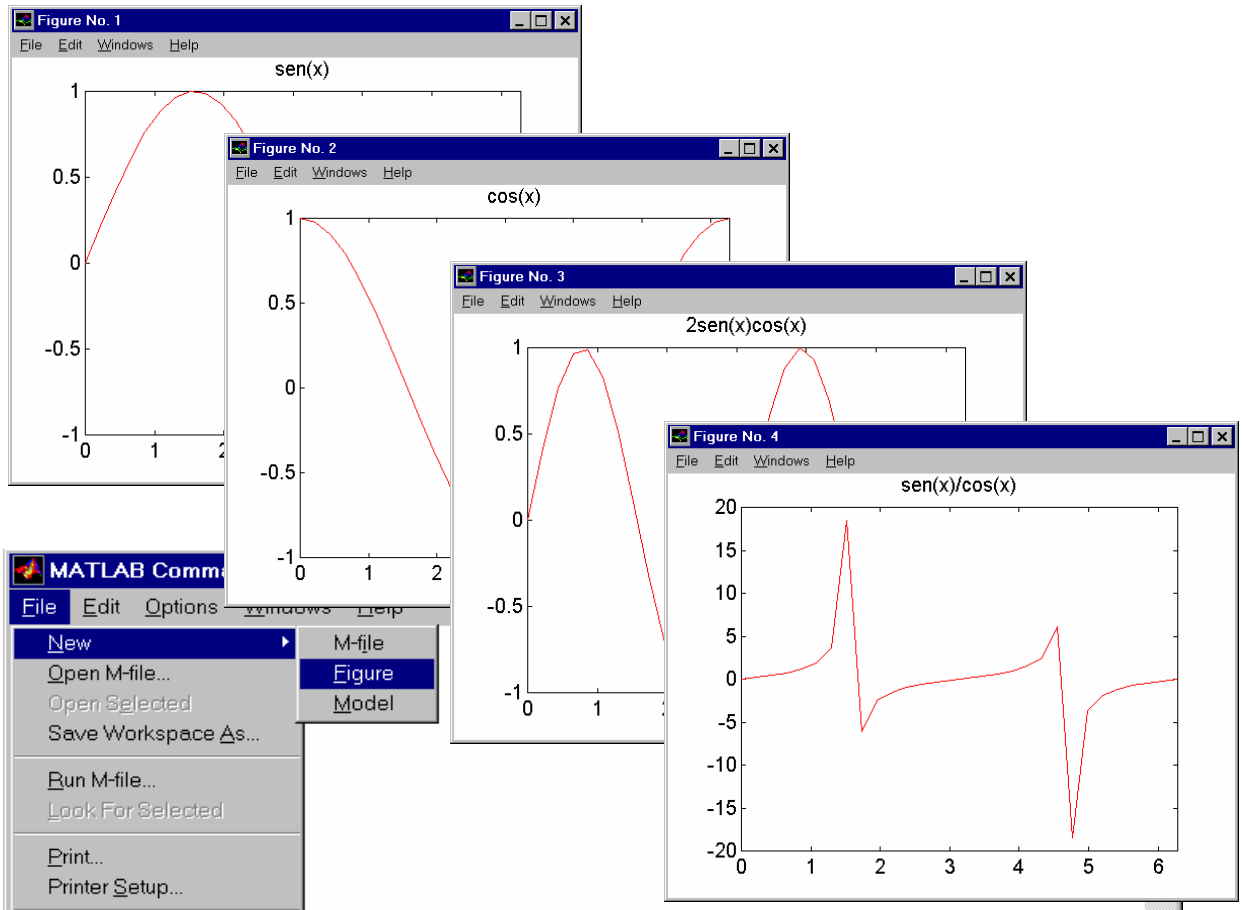


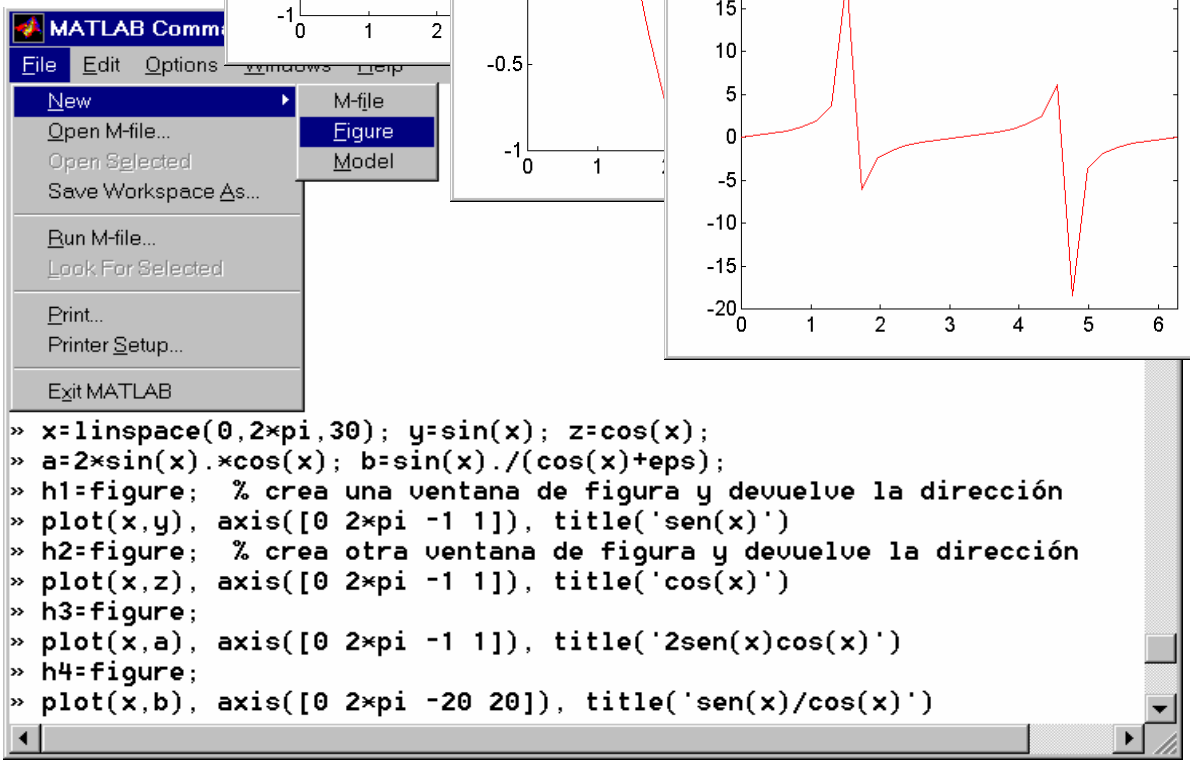
Subplots

```
» x=linspace(0,2*pi,30); y=sin(x); z=cos(x);
» a=2*sin(x).*cos(x); b=sin(x)./(cos(x)+eps);
» subplot(2,2,1)  % primera gráfica de cuatro
» plot(x,y), axis([0 2*pi -1 1]), title('sen(x)')
» subplot(2,2,2)  % segunda gráfica
» plot(x,z), axis([0 2*pi -1 1]), title('cos(x)')
» subplot(2,2,3)  % tercera gráfica
» plot(x,a), axis([0 2*pi -1 1]), title('2sen(x)cos(x)')
» subplot(2,2,4)  % cuarta gráfica
» plot(x,b), axis([0 2*pi -20 20]), title('sen(x)/cos(x)')
      (» subplot(1,1,1)  % vuelve a un solo gráfico)
```



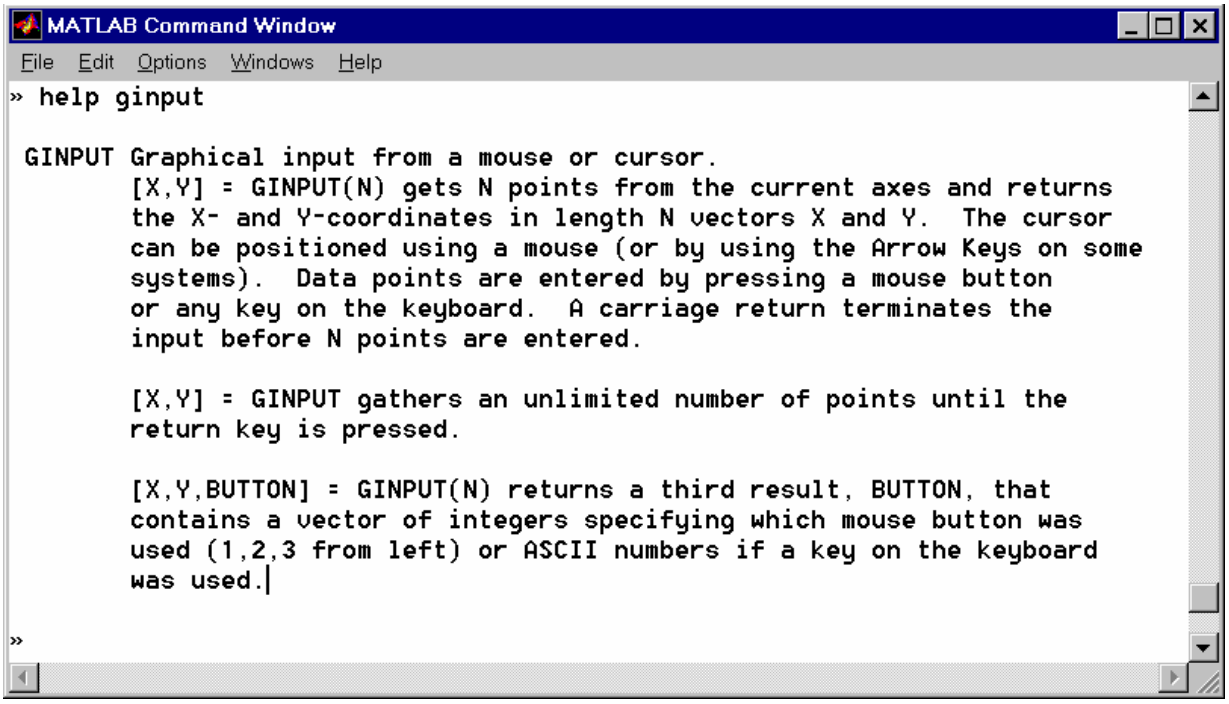
Ventanas múltiples de figuras






```
» x=linspace(0,2*pi,30); y=sin(x); z=cos(x);
» a=2*sin(x).*cos(x); b=sin(x)./(cos(x)+eps);
» h1=figure;  % crea una ventana de figura y devuelve la dirección
» plot(x,y), axis([0 2*pi -1 1]), title('sen(x)')
» h2=figure;  % crea otra ventana de figura y devuelve la dirección
» plot(x,z), axis([0 2*pi -1 1]), title('cos(x)')
» h3=figure;
» plot(x,a), axis([0 2*pi -1 1]), title('2sen(x)cos(x)')
» h4=figure;
» plot(x,b), axis([0 2*pi -20 20]), title('sen(x)/cos(x)')
```

La función “ginput”

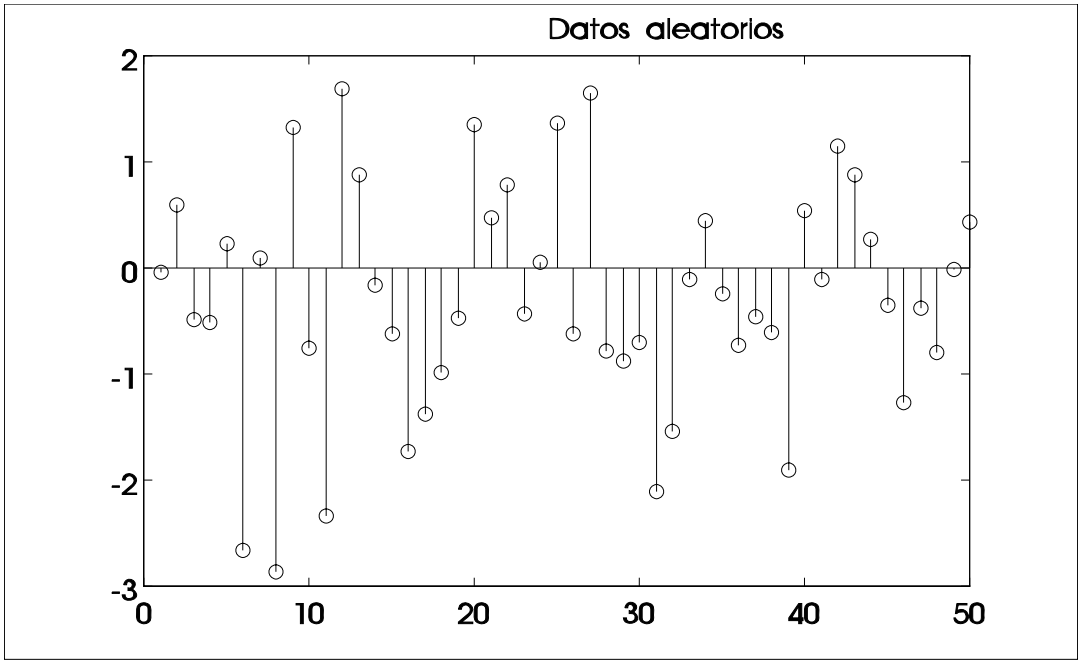


Funciones especializadas para gráficos 2-D. I

MATLAB Command Window

File Edit Options Windows Help

```
» y=randn(50,1);
» stem(y,':')
» title('Datos aleatorios')
» help stem

 STEM   Plot discrete sequence data.
        STEM(Y) plots the data sequence Y as stems from the x-axis
        terminated with circles for the data value.
        STEM(X,Y) plots the data sequence Y at the values specfied
        in X.
        There is an optional final string argument to specify a line-type
        for the stems of the data sequence.  E.g. STEM(X,Y,'-.') or
        STEM(Y,':').

        See also PLOT, BAR, STAIRS.
```

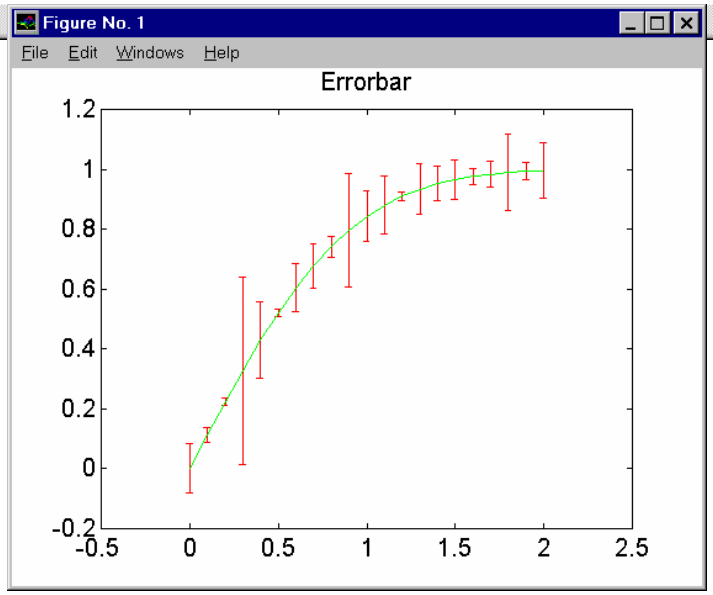


Funciones especializadas para gráficos 2-D. II

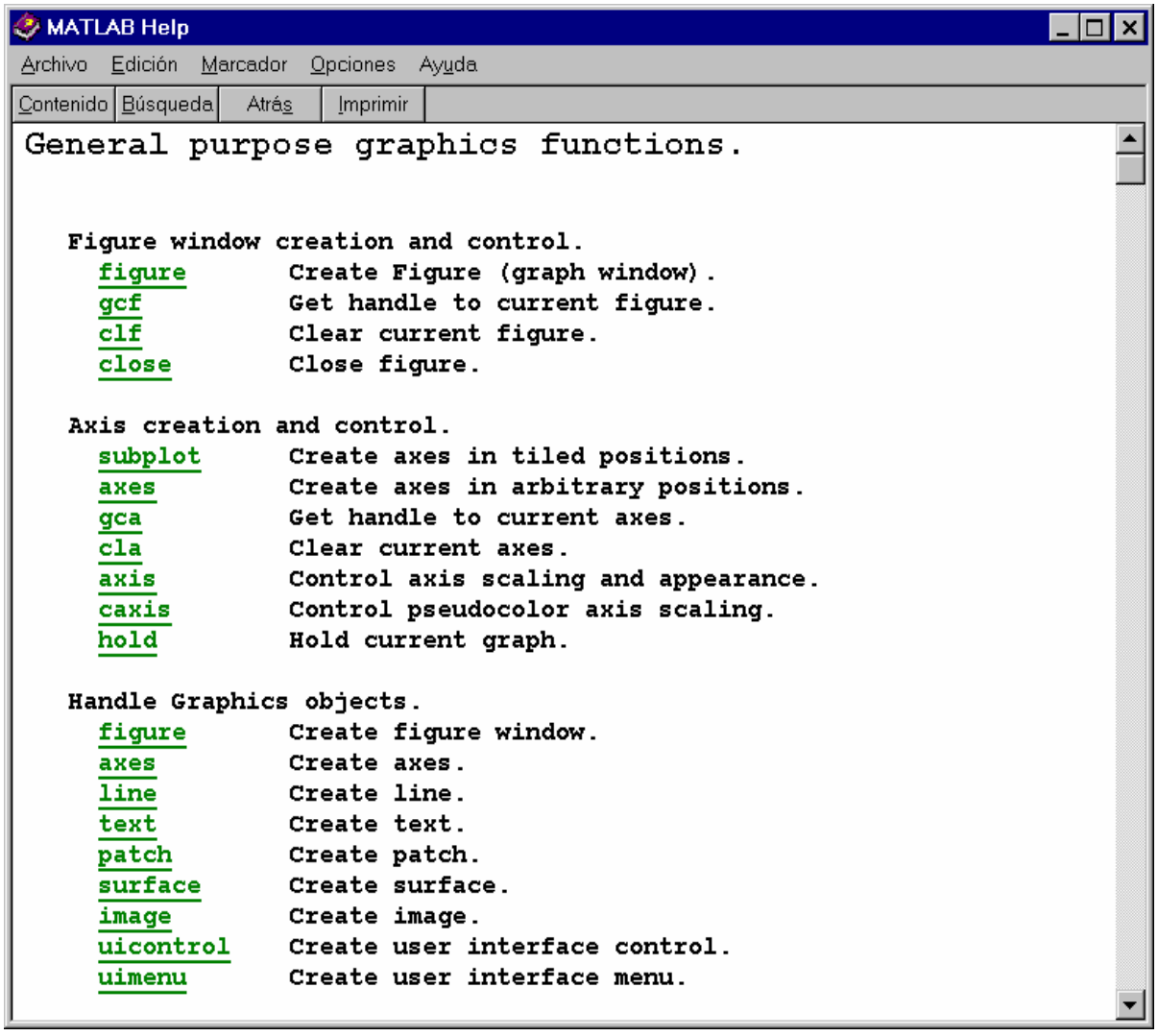
MATLAB Command Window

File Edit Options Windows Help

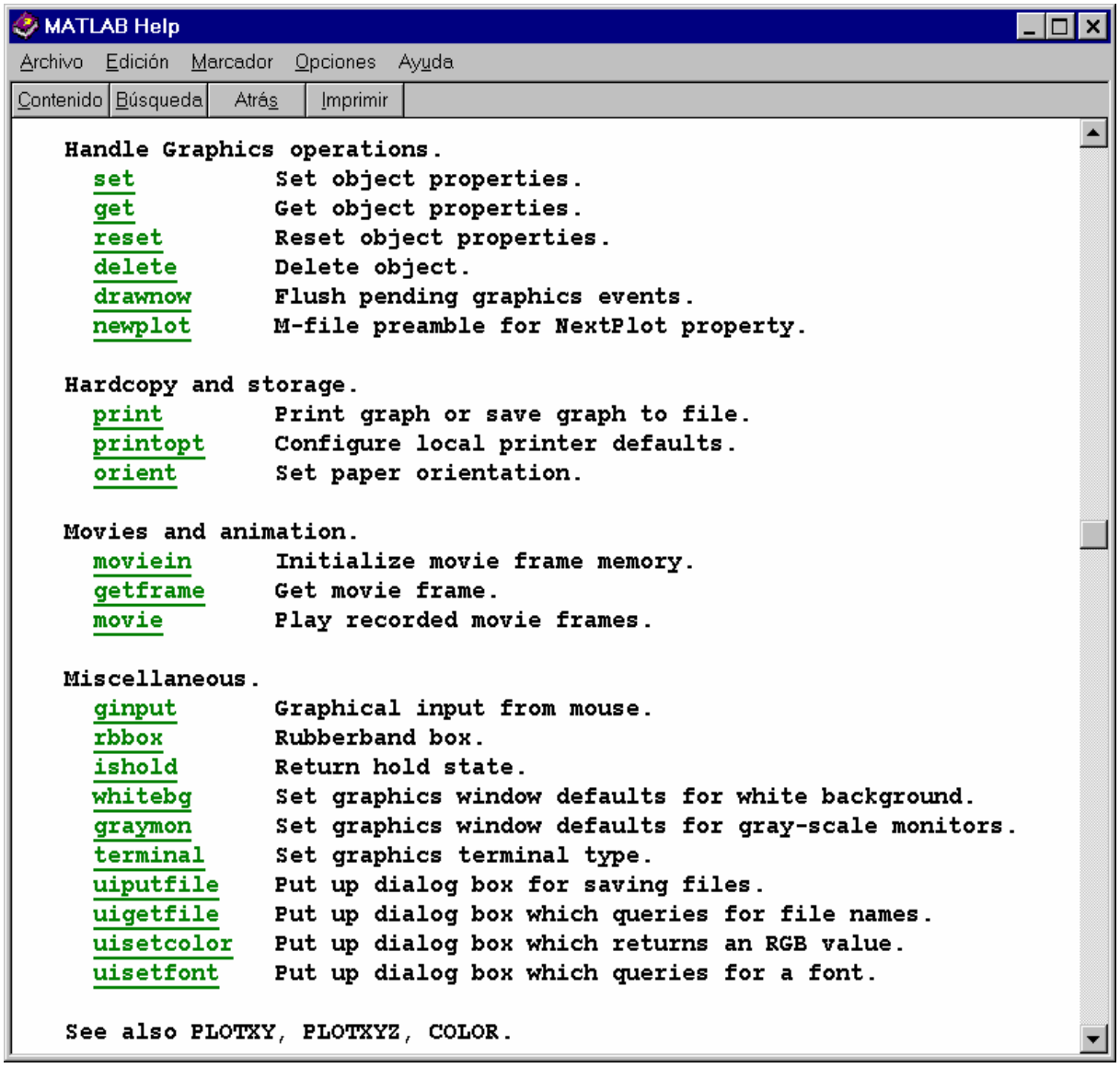
```
» x=0:0.1:2;
» y=erf(x);   % función del error
» e=randn(size(x))/10;  % valores aleatorios del error
» errorbar(x,y,e)
» title('Errorbar')
» help errorbar

 ERRORBAR Plot graph with error bars.
        ERRORBAR(X,Y,E) plots the graph of vector X vs. vector Y with
        error bars specified by the vector E.  The vectors X,Y and E must
        be the same length.  If X,Y, and E are matrices then each column
        produces a separate line.  The error bars are each drawn a distance
        of E(i) above and below the points in (X,Y) so that each bar
        is 2*E(i) long.

        ERRORBAR(Y,E) plots Y with error bars E.

        For example,

           x = 1:10;
           y = sin(x);
           e = std(y)*ones(size(x));
           errorbar(x,y,e)

        draws error bars of unit standard deviation.
```



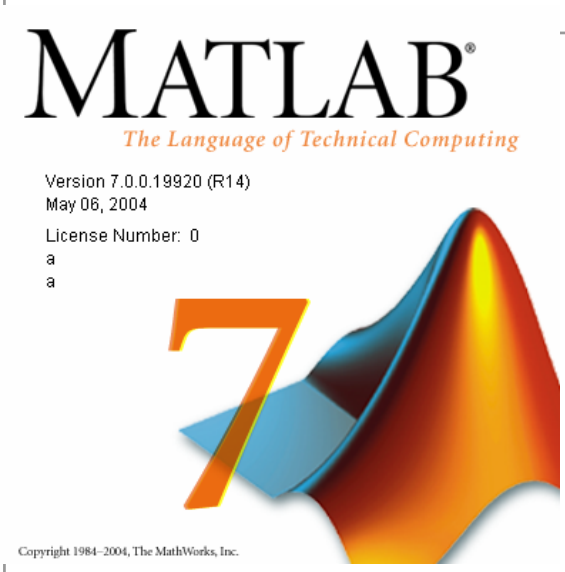
Sumario


MATLAB Help
Archivo Edición Marcador Opciones Ayuda
Contenido Búsqueda Atrás Imprimir
General purpose graphics functions.
Figure window creation and control.
figure Create Figure (graph window).
gcf Get handle to current figure.
clf Clear current figure.
close Close figure.
Axis creation and control.
subplot Create axes in tiled positions.
axes Create axes in arbitrary positions.
gca Get handle to current axes.
cla Clear current axes.
axis Control axis scaling and appearance.
caxis Control pseudocolor axis scaling.
hold Hold current graph.
Handle Graphics objects.
figure Create figure window.
axes Create axes.
line Create line.
text Create text.
patch Create patch.
surface Create surface.
image Create image.
uicontrol Create user interface control.
uimenu Create user interface menu.

Sumario II


MATLAB Help
Archivo Edición Marcador Opciones Ayuda
Contenido Búsqueda Atrás Imprimir
Handle Graphics operations.
set Set object properties.
get Get object properties.
reset Reset object properties.
delete Delete object.
drawnow Flush pending graphics events.
newplot M-file preamble for NextPlot property.
Hardcopy and storage.
print Print graph or save graph to file.
printopt Configure local printer defaults.
orient Set paper orientation.
Movies and animation.
moviein Initialize movie frame memory.
getframe Get movie frame.
movie Play recorded movie frames.
Miscellaneous.
ginput Graphical input from mouse.
rbbox Rubberband box.
ishold Return hold state.
whitebg Set graphics window defaults for white background.
graymon Set graphics window defaults for gray-scale monitors.
terminal Set graphics terminal type.
uiputfile Put up dialog box for saving files.
uigetfile Put up dialog box which queries for file names.
uisetcolor Put up dialog box which returns an RGB value.
uisetfont Put up dialog box which queries for a font.
See also PLOTXY, PLOTXYZ, COLOR.

6. Gráficos 3-D



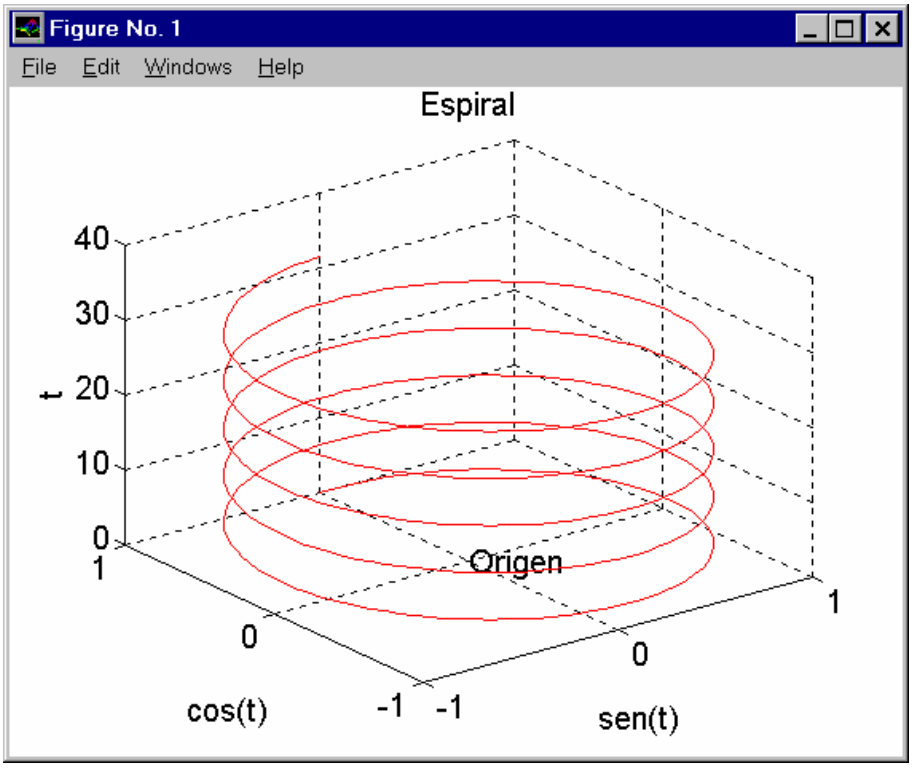
- La función “plot3”
- Puntos de vista
- Gráficas “mesh”
- Gráficas de superficies
- Gráficos de contornos
- Gráficos 2-D de datos 3-D
- Otras funciones
- Secuencias
- Sumario

La función “plot3”

MATLAB Command Window

File Edit Options Windows Help

```
» t=0:pi/50:10*pi;
» plot3(sin(t),cos(t),t)
» title('Espiral'), xlabel('sen(t)'), ylabel('cos(t)'), zlabel('t')
» text(0,0,0,'Origen')
» grid
» v=axis

v =

    -1     1    -1     1     0    40
```

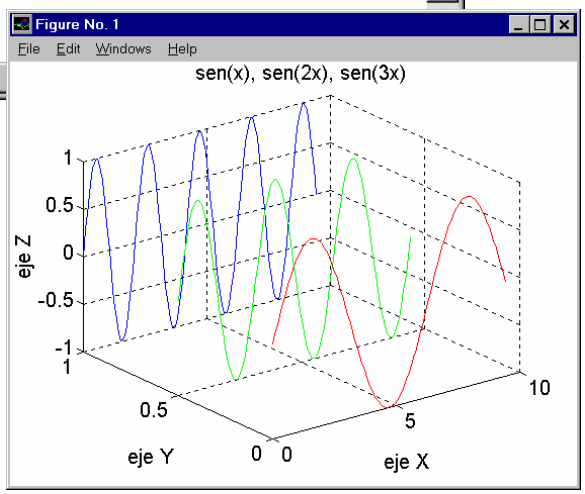


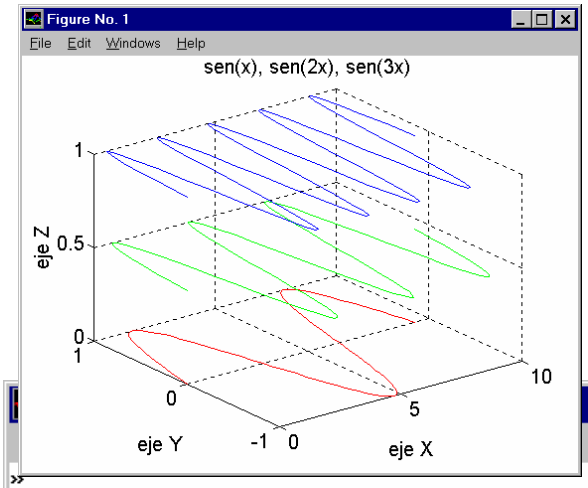
La función “plot3” II

MATLAB Command Window

File Edit Options Windows Help

```
»
» x=linspace(0,3*pi);                    % datos sobre el eje x
» z1=sin(x);                             % dibujo del plano x-z
» z2=sin(2*x); z3=sin(3*x);
» y1=zeros(size(x));                     % alarga sobre el eje y
» y3=ones(size(x)); y2=y3/2;
» plot3(x,y1,z1,x,y2,z2,x,y3,z3)
» grid, xlabel('eje X'), ylabel('eje Y'), zlabel('eje Z')
» title('sen(x), sen(2x), sen(3x)')
»
```





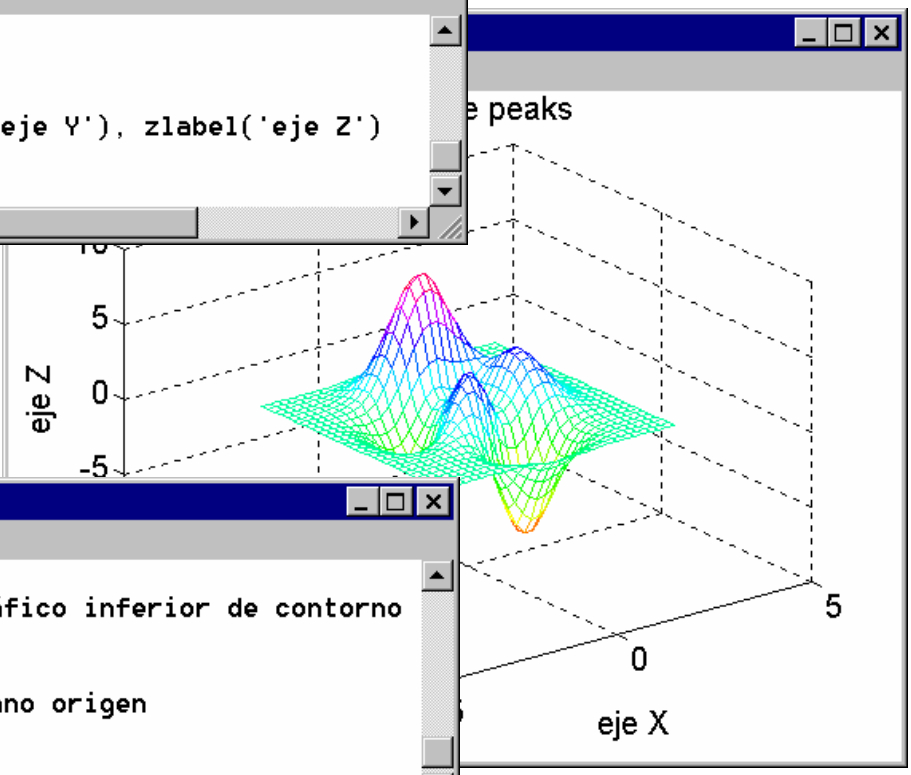

```
» plot3(x,z1,y1,x,z2,y2,x,z3,y3)
» grid, xlabel('eje X'), ylabel('eje Y'), zlabel('eje Z')
» title('sen(x), sen(2x), sen(3x)')
»
```

Gráficas “mesh”

MATLAB Command Window

File Edit Options Windows Help

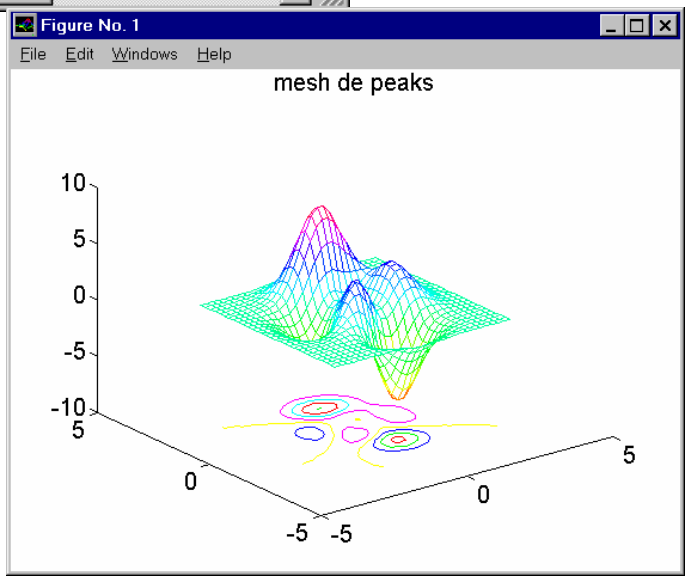
```
» [x,y,z]=peaks(30);
» mesh(x,y,z)
» grid, xlabel('eje X'), ylabel('eje Y'), zlabel('eje Z')
» title('mesh de peaks')
»
```



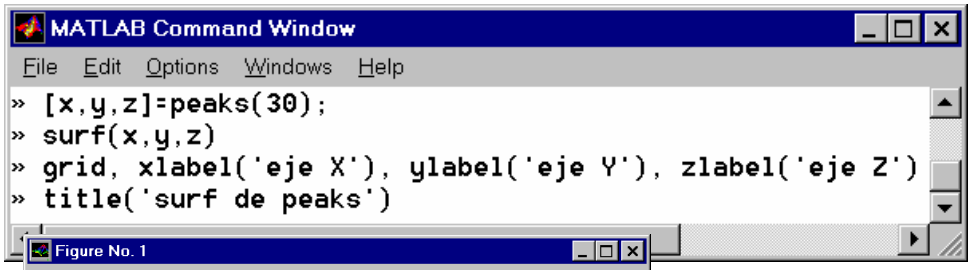
MATLAB Command Window

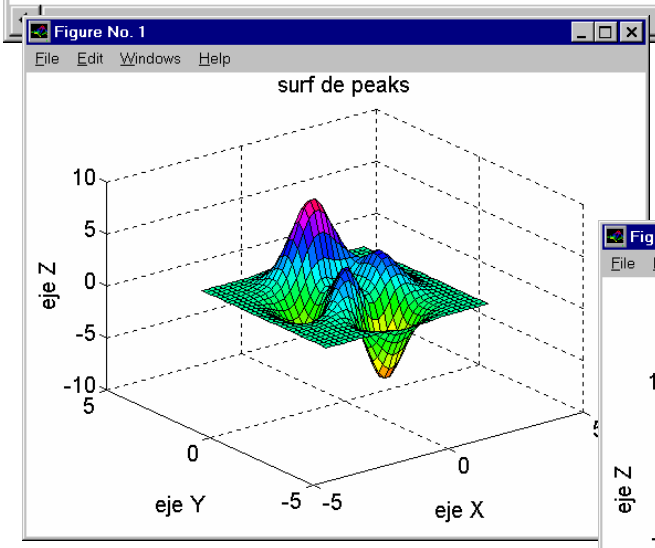
File Edit Options Windows Help

```
» [x,y,z]=peaks(30);
» meshc(x,y,z)  % mesh con gráfico inferior de contorno
» title('mesh de peaks')
» h2=figure;
» meshz(x,y,z)  % mesh con plano origen
» title('mesh de peaks')
» hidden off
```

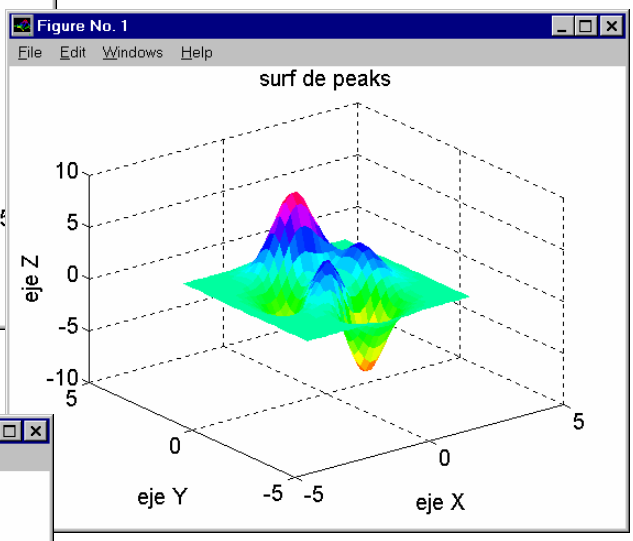


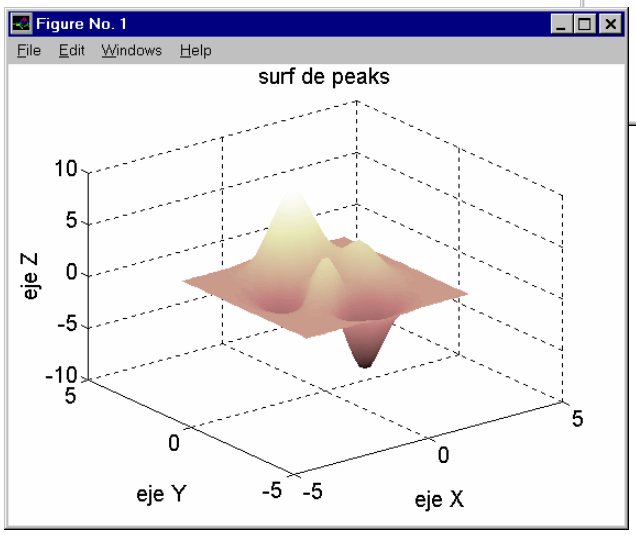
Gráficas de superficies, 'surf'





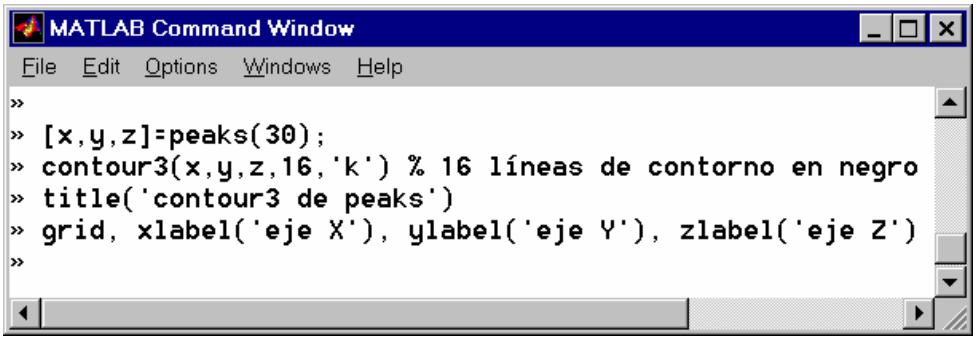
>> shading flat

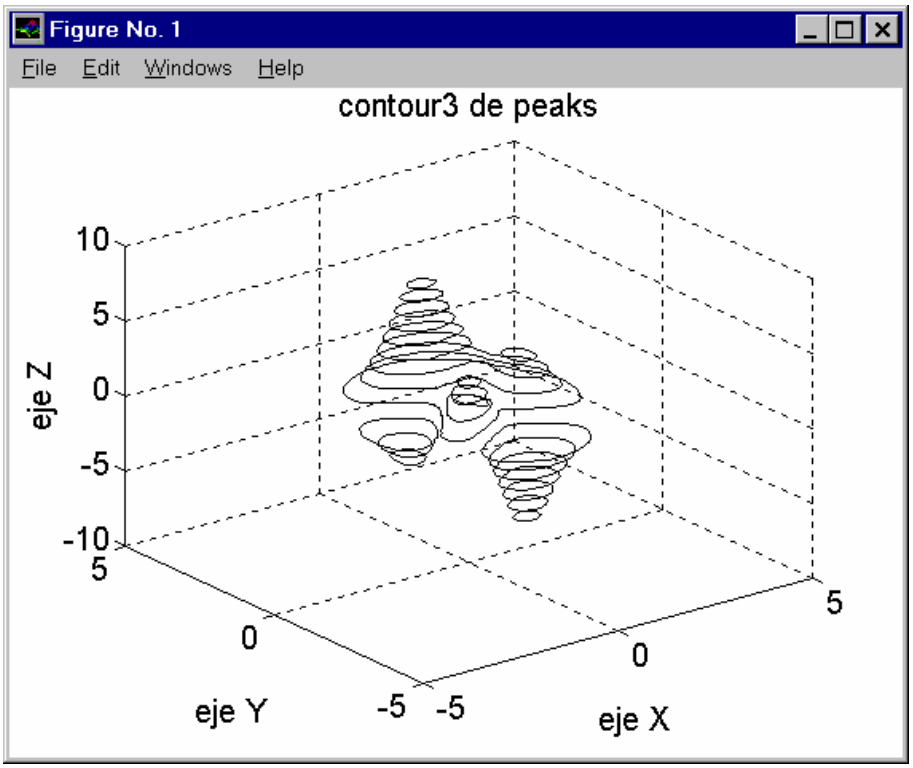




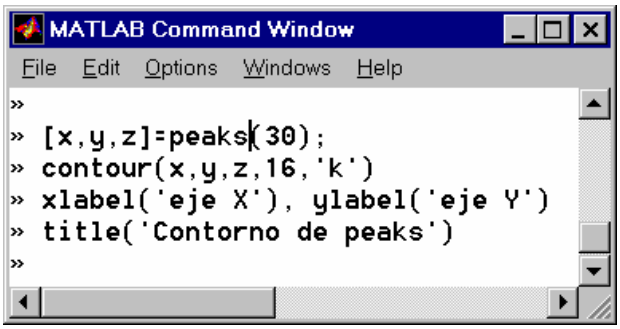
>> shading interp
>> colormap pink

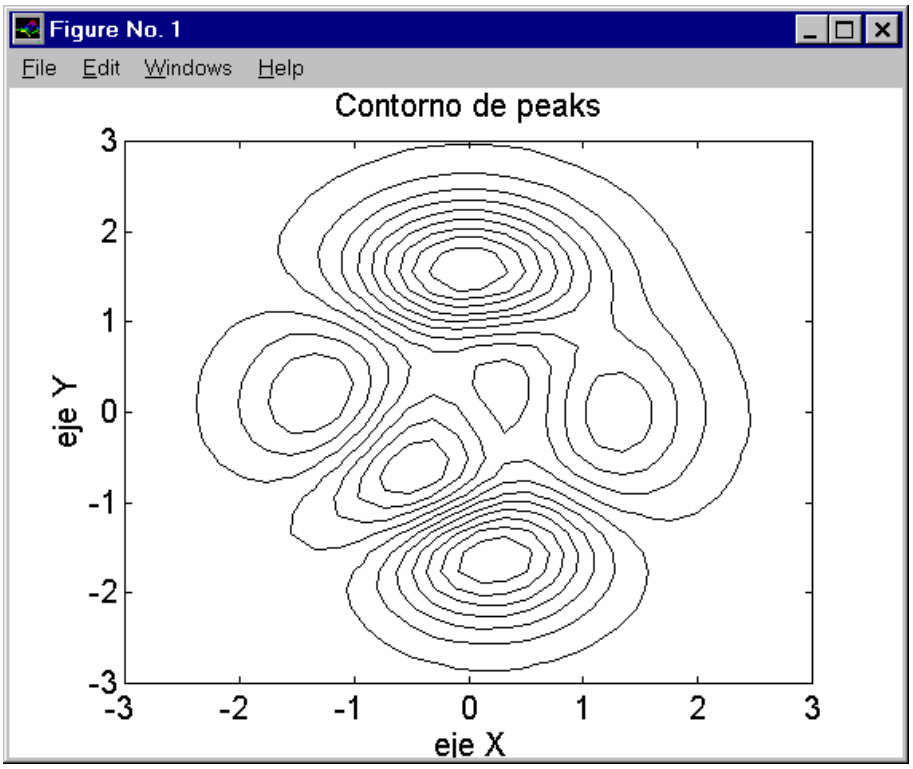
Gráficos de contornos



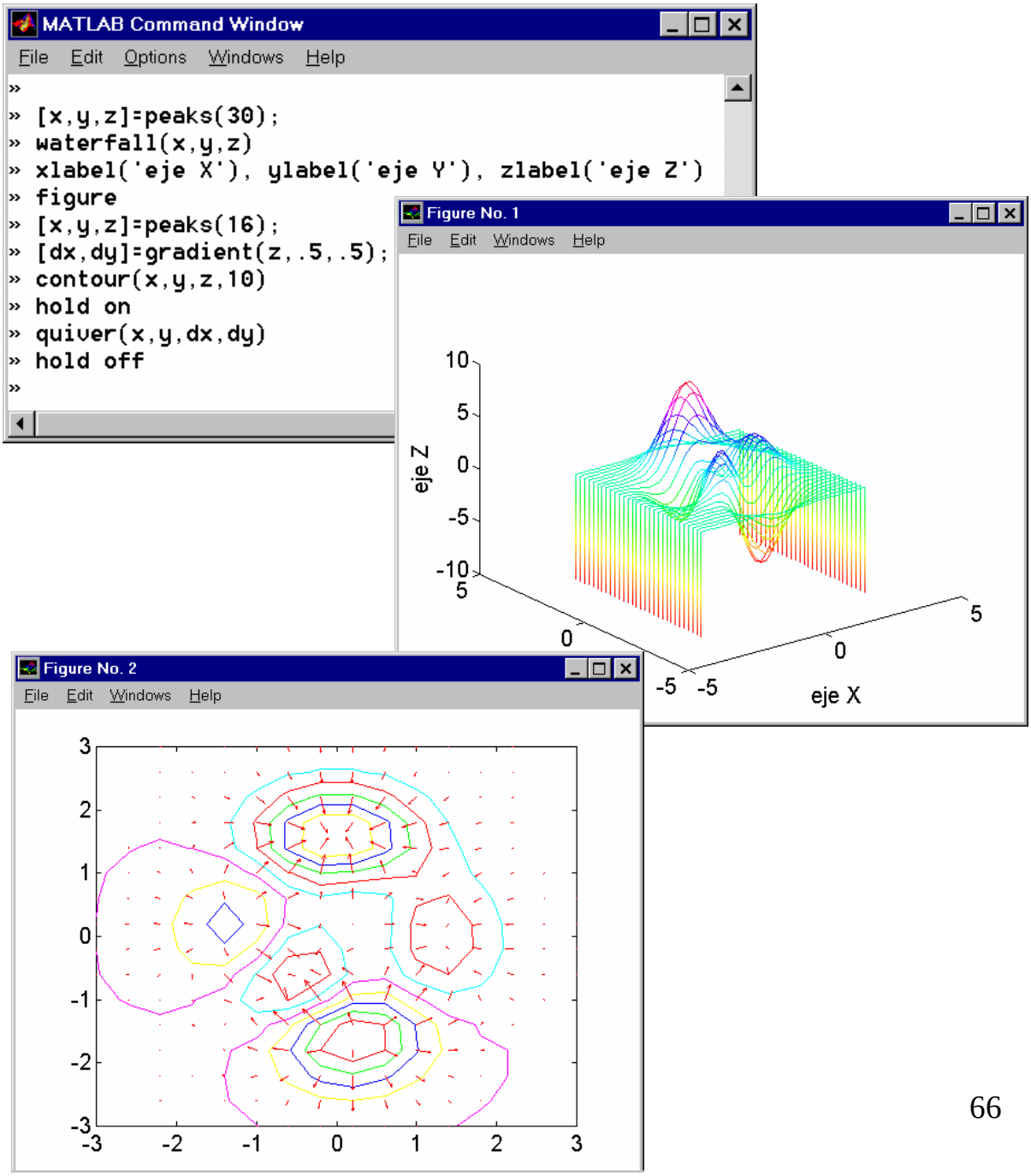


Gráficos 2-D de datos 3-D

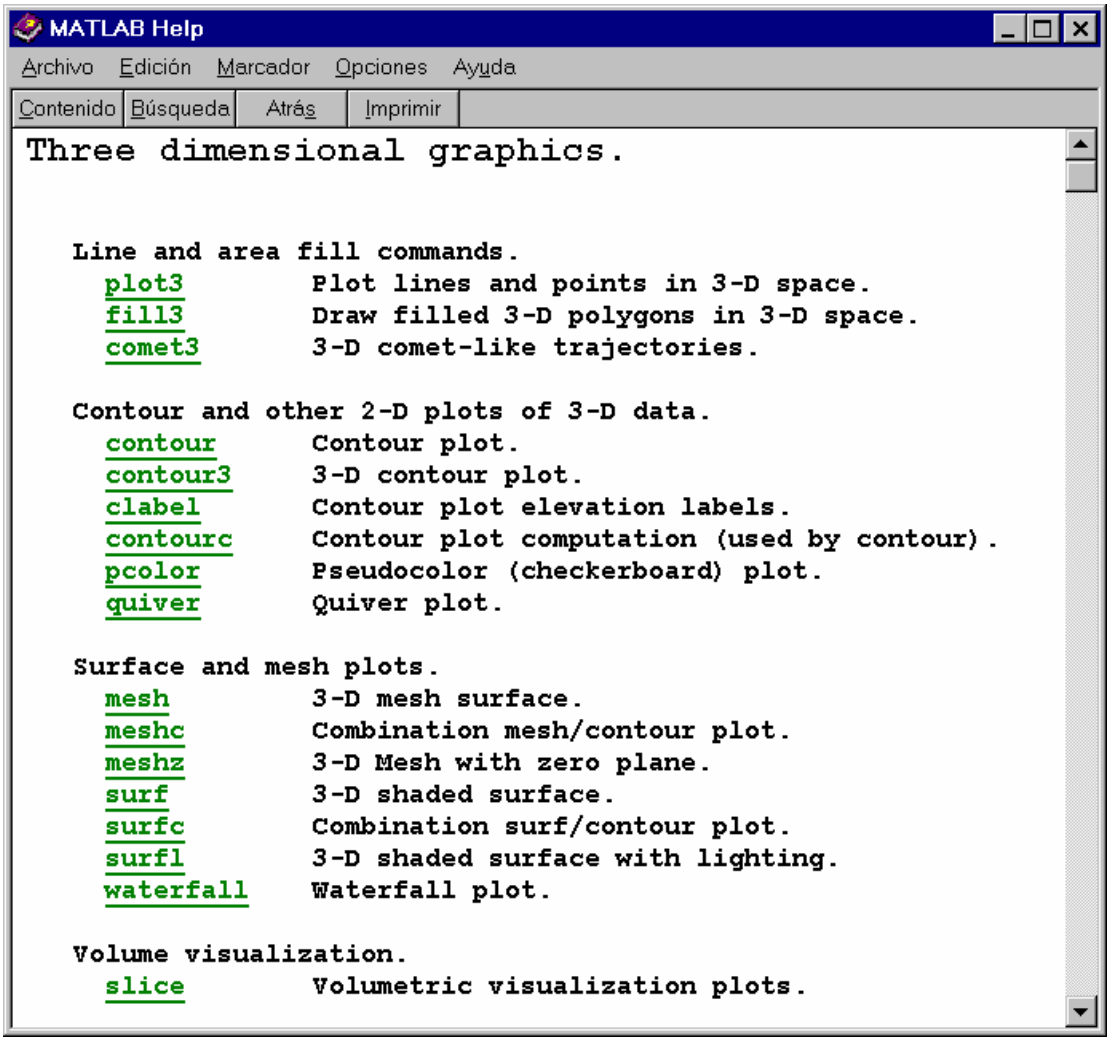




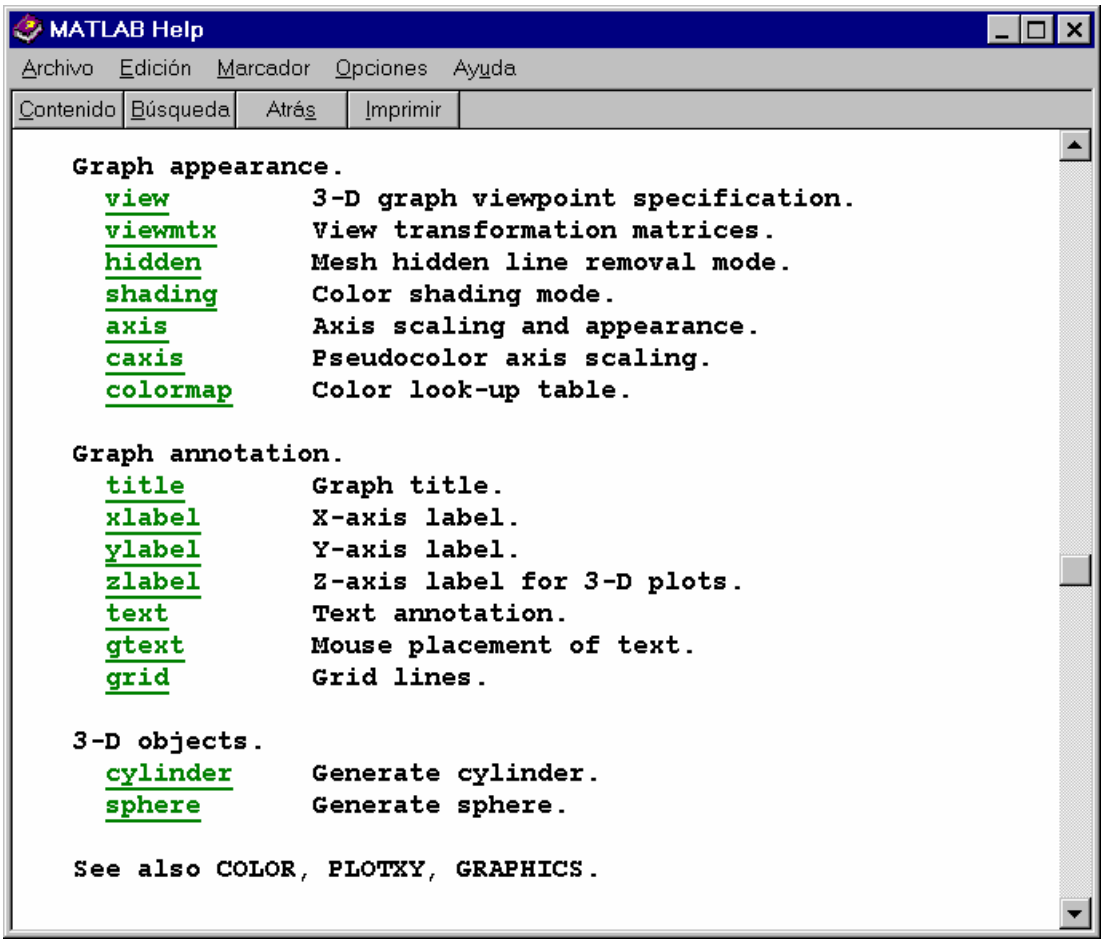
Otras funciones



Sumario


MATLAB Help
Archivo Edición Marcador Opciones Ayuda
Contenido Búsqueda Atrás Imprimir
Three dimensional graphics.
Line and area fill commands.
plot3 Plot lines and points in 3-D space.
fill3 Draw filled 3-D polygons in 3-D space.
comet3 3-D comet-like trajectories.
Contour and other 2-D plots of 3-D data.
contour Contour plot.
contour3 3-D contour plot.
clabel Contour plot elevation labels.
contourc Contour plot computation (used by contour).
pcolor Pseudocolor (checkerboard) plot.
quiver Quiver plot.
Surface and mesh plots.
mesh 3-D mesh surface.
meshc Combination mesh/contour plot.
meshz 3-D Mesh with zero plane.
surf 3-D shaded surface.
surfc Combination surf/contour plot.
surfl 3-D shaded surface with lighting.
waterfall Waterfall plot.
Volume visualization.
slice Volumetric visualization plots.

Sumario II


MATLAB Help
Archivo Edición Marcador Opciones Ayuda
Contenido Búsqueda Atrás Imprimir
Graph appearance.
view 3-D graph viewpoint specification.
viewmtx View transformation matrices.
hidden Mesh hidden line removal mode.
shading Color shading mode.
axis Axis scaling and appearance.
caxis Pseudocolor axis scaling.
colormap Color look-up table.
Graph annotation.
title Graph title.
xlabel X-axis label.
ylabel Y-axis label.
zlabel Z-axis label for 3-D plots.
text Text annotation.
gtext Mouse placement of text.
grid Grid lines.
3-D objects.
cylinder Generate cylinder.
sphere Generate sphere.
See also COLOR, PLOTXY, GRAPHICS.