MatLab R2007

Vectores y matrices

Vectores y matrices

La "especialidad" de Matlab es el manejo de matrices: Matlab son las siglas de Matrix Laboratory.

Un vector se define introduciendo los componentes, separados por espacios o por comas, entre corchetes:

```
» v=[sqrt(3) 0 -2]

v =

    1.7321   0   -2.0000
```

Para definir un vector columna, se separan las filas por puntos y comas:

```
» w=[1;0;1/3]

w =

    1.0000

    0

    0.3333
```

La operación transponer (cambiar filas por columnas) se designa por el apóstrofe:

```
» w'

ans =

    1.0000   0   0.3333
```

Las operaciones matemáticas elementales pueden aplicarse a los vectores:

```
» v*w

ans =

    1.0654
» v+w'

ans =

    2.7321   0   -1.6667
```

Para crear un vector de componentes equiespaciados se emplean los dos puntos:

```
» x=4:2:10

x =

     4 6 8 10
```

(los componentes de x van desde 4 de 2 en 2 hasta 10).

Para introducir matrices, se separa cada fila con un punto y coma:

```
» M = [1 2 3 ;4 5 6 ;7 8 9]

M =

     1 2 3

     4 5 6

     7 8 9
```

Para referirse a un elemento de la matriz se hace así:

```
» M(3,1)

ans =

     7
```

Para referirse a toda una fila o a toda una columna se emplean los dos puntos:

```
» v1=M(:,2)

v1 =

     2

     5

     8
```

(v1 es la segunda columna de M).

Con las matrices también funcionan las operaciones matemáticas elementales. Así

```
» M^2

ans =

     30  36  42

     66  81  96
```

102 126 150

Si se quiere operar en los elementos de la matriz, uno por uno, se pone un punto antes del operador. Si se quiere elevar al cuadrado *cada uno de los elementos de M*, entonces

» M.^2

ans =

1 4 9

16 25 36

49 64 81

Algunas **funciones** definidas sobre matrices:

det	determinante
inv	matriz inversa
poly	polinomio característico
'	transpuesta

(Para más información: help elmat)

Poliniomios

En Matlab los polinomios se representan por vectores cuyas componentes son los coeficientes del polinomio.

Sea

$$P(x)=x^2-3x+2$$

Este polinomio se representa por un vector p

» p=[1 -3 +2]

p =

1 -3 2

Para hallar las raíces del polinomio, se hace

» roots(p)

ans =

2

1

y si se quiere hallar el valor de P(x) para un determinado valor de x (por ejemplo, para x=0)

```
» polyval(p,0)

ans =

     2
```

Gráficos

Cómo presentar datos con Matlab.

Las posibilidades de Matlab son muy grandes. Se indica a continuación cómo realizar gráficos sencillos. Para más información, o para conocer la versatilidad de Matlab: capítulo *Handle Graphics Object* del Help Desk, el manual *Using MATLAB Graphics* o la ayuda en línea help graph2d.

Veamos cómo se puede representar la función seno entre 0 y 10. Para empezar creemos una variable x que vaya de cero a 10:

```
» x=0:0.1:10;
```

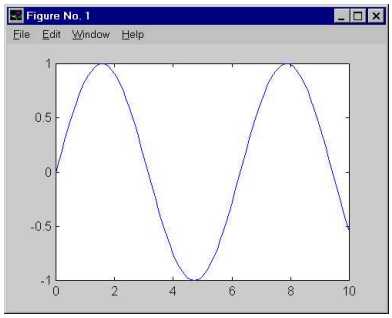
y a continuación, calculemos sin(x) almacenando el resultado en la variable y:

```
» y=sin(x);
```

Para trazar el gráfico, se emplea la función plot:

```
» plot(x,y)
```

y se obtiene en otra ventana el gráfico:



Entre los muchos comandos que se pueden utilizar para modificar los gráficos, es muy útil el empleado para cambiar la escala de los ejes. La orden es

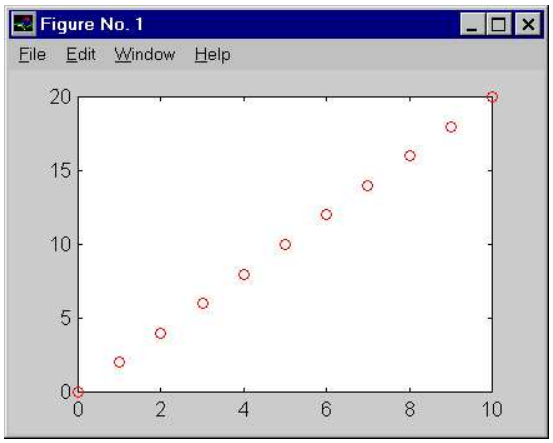
```
axis([x1 x2 y1 y2])
```

donde x1, x2 son los límites inferior y superior del eje x, e y1 e y2 los del eje y.

Para representar unos datos con símbolos de colores, se añade al comando plot, entre apóstrofes, la especificación. Vamos a crear una variable con dos filas que contenga los números del 1 al 10 en la primera fila, y el doble de esos números en la segunda, y dibujarlos con puntos rojos:

```
» x(1,:)=0:10;

» x(2,:)=2*x(1,:);

» x
```

```
x =

     0   1   2   3   4   5   6   7   8   9   10

     0   2   4   6   8   10  12  14  16  18  20

» plot(x(1,:),x(2,:),'ro')
```



(para ver las especificaciones posibles, teclear help plot. Por ejemplo, 'ro' establece un gráfico de color rojo: r y de puntos: o.) Si no se indica nada, el gráfico se traza con una línea azul.

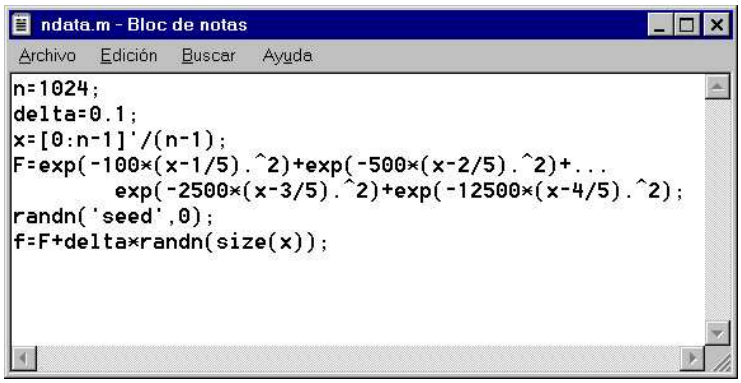
Otras funciones muy útiles: grid, que traza una cuadrícula, xlabel('títulox')e ylabel('tituloy'), que sirven para poner un título en los ejes.

Para imprimir una figura, basta seleccionar print *del menú de la figura.*

"Scripts"

Archivos de órdenes: programar en Matlab.

Realizar un programa en Matlab es fácil. Basta abrir un editor de texto (como el Bloc de Notas de Windows) y escribir los comandos uno a continuación de otro. Luego ese fichero de texto debe guardarse con la extensión .m, y a eso se le llama un *script*:



Una vez guardado el fichero (en el ejemplo, ndata.m) en el directorio actual, desde la línea de comandos de Matlab basta escribir ndata para que se ejecute el programa.

A partir de aquí, se abren las posibilidades de la programación con un lenguaje sencillo que intentaremos ver o en todo caso, os dejaré un tutorial sencillo.

Cálculo simbólico

Matemáticas en el ordenador.

Hasta ahora, las operaciones que se han mostrado se han realizado con números. El *toolbox* de cálculo simbólico permite realizar **cálculos abstractos**:

```
» diff('sin(x)')

ans =

        cos(x)
```

Las expresiones simbólicas se introducen entre apóstrofes.

A continuación se da una tabla con algunas funciones de este toolbox, junto con un ejemplo de cada una:

diff	**derivada**	diff('sin(x)')
int	**integral**	int('x^2')
solve	**resolución de ecuaciones**	solve('x^2-3*x+2=0')
ezplot	**gráficos**	ezplot('exp(x)')

Evidentemente, las expresiones pueden ser todo lo complicadas que se quiera ...

```
» solve('x=cos(x)')

ans =

        .73908513321516064165531208767387
» int('(x^4+4*x^3+11*x^2+12*x+8)/((x^2+2*x+3)^2*(x+1))')

ans =

        log(x+1)+1/8*(-4*x-8)/(x^2+2*x+3)-1/4*2^(1/2)*atan(1/4*(2*x+2)*2^(1/2))
```