MATLAB

El software MatLab se desarrolló como un " Laboratorio de matrices", pues su elemento básico es una matriz. Es un sistema interactivo y un lenguaje de programación de cómputos científico y técnico en general.

Comandos

Algunos comandos para tener en cuenta en las operaciones son:

- **clear** borra toda la pantalla.
- **clc** borra toda la pantalla pero deja internamente el valor de las variables.
- **who** enumera todas las variables usadas hasta el momento.
- **help (**tema**)** proporciona ayuda sobre el tema seleccionado.
- ↑ Con este botón se pueden recuperar sentencias anteriormente usadas.
- **syms** sirve para declarar variables.
- **round(**operación**)** redondea al entero más cercano:

```
>> round(9/4)
ans =
     2
```

- **sqrt** calcula raíz cuadrada.
- **solve** resuelve una ecuación o sistema de ecuaciones.

1) Introducir una matriz

Si se quiere introducir por ejemplo la matriz $A = \begin{pmatrix} 4 & 2 \\ 3 & 3 \end{pmatrix}$, se puede escribir:

```
>> A=[4,2;3,3]
A =
    4    2
    3    3
```

También se puede escribir A=[4 2;3 3]. Si se agrega un punto y coma al final (A=[4,2;3,3]**;**), no sale la matriz quedando en la memoria del programa.

2) Operaciones matriciales básicas :

- **Adición (sustracción)** — **A+B** ó **A-B**
- **Multiplicación** — **A*B**
- **Producto por un escalar** — α***A**
- **Cálculo de la inversa** — **inv(A)** ó **A^(-1)**
- **Cálculo del determinante** — **det(A)**

3) Cálculo del polinomio característico:

Se calcula el polinomio característico asociados a la matriz A dada.

p=poly(A)

```
>> poly(A)
ans =
    1   -7    6
```

El resultado son los coeficientes del polinomio característico ordenado de acuerdo a las potencias decrecientes de la variable λ, es decir:

$P(\lambda) = \lambda^2 - 7\lambda + 6$

Otra forma de calcular el polinomio característico es usando el comando: **vpa(polynsym(p)),** donde " **n**" indica el número de cifras decimales con que se quiere obtener los coeficientes del polinomio.

>> vpa(poly2sym(p))

ans =

x^2-7.*x+6.

Expresa el polinomio característico en la variable x.

4) Cálculo de los autovalores.

Los comandos que se pueden emplear para el cálculo de los autovalores son:

1- rootos(p) da las raíces del polinomio característico.

2- eig(A) da los autovalores asociados a A.

3- eigensys(A) expresa los autovalores simbólicamente.

Se efectúan los tres procedimientos para el cálculo de los autovalores de la matriz A dada.

1->> roots(poly(A))

ans =

6
1

Luego los autovalores asociados a la matriz A son $\lambda = 6$ y $\lambda = 1$.

2->> eig(A)

ans =

6

1

3->> eigensys(A)

ans =

[1]

[6]

5) Cálculo de los autovalores y autovectores. Matriz diagonal

Los comandos que se pueden usar son:

1- [Q,D]=eig(A);Q=Q proporciona la matriz Q que contiene en sus colum - nas a los **autovectores normalizados** asociados a la matriz A.

2- [Q,D]=eigs(A);D=D proporciona la matriz D diagonal que contiene a los **autovalores** asociados a A.

3- [eves,evas]=eig(A) **eves** es la matriz cuyas **columnas** son los **auto - vectores** normalizados y **evas** es la matriz diago- nal que contiene a los **autovalores.**

4- [Q,D]=eigensys(A) proporciona los autovectores y autovalores simboli- camente.

1->> [Q,D]=eig(A);Q=Q

Q =

```
    0.7071   -0.5547
    0.7071    0.8321
```

Luego los autovectores asociados a la matriz A son (0.7071 ; 0.7071) y (-0.5547 ; 0.8321).

2->> [Q,D]=eig(A);D=D

```
D =
     6     0
     0     1
```

3- >> [eves,evas]=eig(A)

```
eves =
    0.7071   -0.5547
    0.7071    0.8321

evas =
     6     0
     0     1
```

4- >> [Q,D]=eigensys(A)

```
Q=
[    1,    1]
[    1, -3/2]

D =
 [ 6, 0]
[ 0, 1]
```

6) Gráficos.

- **[x,y,z]=(x_{min} , x_{max}, y_{min}, y_{max}, z_{min}, z_{max})** indica los valores de variación de las variables x,y,z, pudiéndose agregar un rango Δ de variación entre el valor mínimo y máximo.

- **plot(x,y)** genera una gráfica en las variables x e y.

- **plot(x,y,t)** genera una gráfica en las variables x e y siendo **t** un parámetro.

- **plot3(x,y,z)** genera una gráfica en las variables x,y,z.

- **grid** agrega una grilla al gráfico**.**

1) Rectas.

En el caso de una recta expresada en forma paramétrica, se debe declarar el rango del parámetro.

Ejemplo:

(x ; y ; z) = (1; 1; -2)+ t.(1; 0 ; 2)

Se toma el parámetro **t** comprendido entre –3 y 9:

>>t=(-3:0.1:9);   El **0.1** indica el incremento de **t** a partir del –3 hasta el 9.
```
>>x=1+1*t;
>>y=1*0*t;
>>z=-2+2*t;
>>plot3(x,y,z),grid
```

2) Planos.

Para graficar un plano se despeja una de las variables.

Ejemplos:

a) x+2y -z+2=0, si se despeja z$\Rightarrow$z = x +2y +2 . Se indica el rango de variación de las variables **x** e **y** empleando el comando **meshgrid:**

```
>>[x,y]=meshgrid(x min:Δx:x máx,y min:Δy:y máx);
>>z=1*x+2*y+2;
>>plot3(x,y,t),grid
```

b) $\begin{cases} x+2y-z+2=0 \\ 3x-y-z=0 \end{cases}$

Se despeja la misma variable de las dos ecuaciones, por ejemplo **z**:

```
>>[x,y]=meshgrid(x min:Δx:x más,y min:Δy:y máx);
>>z1=x+2*y+2;
>>z2=3*x-y;
plot3(x,y,z1,x,y,z2),grid
```

3) **Cónicas**.

Las cónicas se pueden obtener como una curva de nivel utilizando el comando **contour**:

Ejemplo:

Para graficar las parábolas $\begin{cases} x^2+2xy+y^2-8x+8y=0 \quad \text{(rototrasladada)} \\ x^2-2y=0 \end{cases}$ en un mismo sistema coordenado, se procede de la siguiente manera:

```
>>[x,y]=meshgrid(x min:Δx:x máx,y min:Δy:y máx);

>>f1=x.^2+2*x.*y+y.^2-8*x+8*y;
>>contour(x,y,f1,[0,0])
>>hold on
f2=x.^2-2*y;
contour(x,y,f2,[0,0])
```

Otra forma de introducir los rangos de las variables **x** e **y** es:

```
>>xm=x min:Δx:x máx; ym=y min:Δy:y máx;
>>[x,y]=meshgrid(x m,ym);
```

4) **Cuádricas. Superficies**:

a) Paraboloide $z = x^2 + y^2$

>>[x,y]=meshgrid(x $_{min}$:Δx:x $_{máx}$,y $_{min}$:Δy:y $_{máx}$);
>>z=x.^2+y.^2;
>>surf(x,y,z)

Si en lugar de **surf** se hubiese usado el comando **contour3(z,N)** se obtendrían **N** curvas de nivel del paraboloide.

b) Paraboloide hiperbólico $z = x^2 - y^2$

>>[x,y]=meshgrid(x $_{min}$:Δx:x $_{máx}$,y $_{min}$:Δy:y $_{máx}$);
>>z=x.^2-y.^2;
>>surf(x,y,z)

c) Esfera

>>[x,y,z]=sphere(α, β, γ,radio,N);   siendo (α, β, γ) las coordenadas del centro de la esfera.
>>surf(x,y,z)

d) Elipsoide

>>>[x,y,z]=ellipsoid(α, β, γ,a,b,c);   siendo a,b,c los semidiámetros correspondientes a los ejes x,y,z respectivamente.
>>>>surf(x,y,z)

e) Hiperboloide de una hoja $x^2 + y^2 - z^2 = 1$

>>[x,y]=meshgrid(x $_{min}$:Δx:x $_{máx}$,y $_{min}$:Δy:y $_{máx}$);
>>z1=sqrt(x.^2+y.^2-1);
>>z2=-sqrt(x.^2+y.^2-1);
>>plot3(x,y,z1,x,y,z2)

f) Superficie cilíndrica circular

>>[x,y,z]=cylinder(diámetro,N)
>>surf(x,y,z)

g) Superficie cilíndrica parabólica $z=x^2$

>>[x,y]=meshgrid(x $_{min}$:Δx:x $_{máx}$,y $_{min}$:Δy:y $_{máx}$);
>>z=x.^2
>>plot3(x,y,z)

f) Superficie cónica $x^2 + y^2 - z^2 = 0$

```
>>[x,y]=meshgrid(x min:Δx:x máx,y min:Δy:y máx);
>>z1=sqrt(x.^2+y.^2);
>>z2=-sqrt(x.^2+y.^2);
>>plot3(x,y,z1,x,y,z2)
```