

Parte 2: Gráficos y Funciones

Matrices Elementales

```
» zeros(1,3)
ans = 0 0 0

» ones(3,2)
ans =
    1 1
    1 1
    1 1

» zeros(2)
ans =
    0 0
    0 0

% Matriz identidad (Eye -> I, badum ts)
» eye(3)
ans =
    1 0 0
    0 1 0
    0 0 1

% pseudorandom numbers
% rand genera numeros entre 0 y 1 con distribucion uniforme
% randn genera numeros con distribucion normal

%tamaño
»size(b)

%longitud
» length(c)
```

Funciones Matematicas Elementales

```
» sin(2)
» cos([2 4])
» tan(B)
» exp([3 -1 j*pi/2])
» log([3 ; 6])
» sqrt([3 ; j^2 ; -1])

% Funciones utiles con numeros complejos
» abs(1+j)
» angle(1+j)
» conj(1+j)
```

```

» real(1+j)
» imag(1+j)

% Otras funciones
» fix % redondea un numero al entero mas cercano hacia cero
» floor % redondea un numero al entero mas cercano hacia menos infinito
» ceil % redondea un numero al entero mas cercano hacia mas infinito
» rounds % redondea un numero al entero mas cercano
» mod(x,y) % mod(x,y) = x - y .* floor(x ./ y)
» rem(x,y) % si y no es cero rem(x,y) = x - y.*fix(x./y)

```

Funciones de Matematica Avanzada

Algunas funciones útiles es procesamiento de señales son la *funcion error*, *funcion error complementario*, *minimo comun multiplo* y *maximo comun divisor*.

```

%Error y complemento del error
» erf(-1)
» erfc(-1)

% Minimo comun multiplo: Acepta matrices y devuelve el LCM (least common
multiple) de los elementos
» lcm([3 5 9 5 12]);
» gcd(15 25 35 80);

```

Funciones Matematicas con Matrices

En procesamiento de señales es comun la inversion de matrices y el calculo de eigenvalores.

```

% Matriz invertida (debe ser cuadrada y su determinante /= de 0)
» inv(A)

inv(A)*A % devuelve identidad

» eig(B) % encuentra los eigenvalores de una matriz cuadrada

» max(C) % retorna el valor maximo en un vector
» min(C) % igual que max, pero retorna el minimo
» mean(A) % para una matriz retorna un vector con el valor medio de las
columnas de A
» std(B) % retorna la desviacion estandar de un vector, y para una matriz
retorna un vector con la desviacion estandar de las columnas
» sort(C) % Ordena las columnas de un vector en orden ascendente

```

Para funciones en tiempo discreto (SVID) hay cinco funciones que pueden ser útiles.

```
% Retorna la suma de los elementos de un vector, o un vector suma de las
columnas en una matriz
» sum(B)

% igual que sum pero con el producto
» prod(B)

% Retorna un vector donde cada elemento es la diferencia del elemento del
vector original menos el de su posicion anterior. El vector resultado tiene
un elemento menos que el vector de input. Para una matrix, devuelve una
matriz con la diferencia en las columnas.
» diff(A)

% Cumulative Sum: Retorna un vector donde cada elemento es la suma de los
elementos previos. En una matriz aplica lo mismo en columnas
» cumsum(C)

% Opera como cumsum pero con el producto
» cumprod(C)
```

Otras funciones muy importantes son histograma, covarianza, convolucion y Transformada de Fourier rápida.

```
% hist grafica un histograma de un vector de datos.
» hist(randn(1000,1))

% cov encuentra la covarianza de un vector o la matriz de covarianza en una
matriz, considerando cada columna que representa una observacion y cada
columna contiene datos de una variable.
» cov(randn(5,5))

% conv convoluciona dos vectores. La longitud del vector de retorno es la
suma de las longitudes de los vectores de convolucion, menos uno.
» conv(ones(1,5),ones(1,5))
ans =
    1    2    3    4    5    4    3    2    1

%La funcion fft computa la Transformada de fourier discreta de un vector
» fft(cos(2*pi(0:15)'/4))
```

Interpolación y Polinomios

En procesamiento de señales es importante encontrar las raices de las ecuaciones.

```
% roots encuentra todas las raices de una ecuacion algebraica
% La ecuacion se asume de forma polinomial  $ax^n + \dots + bx^2 + cx^3 + dx^4 = 0$ 
donde el vector input son los coeficientes a , ... , c , b , c , d
» roots([3 9 8 1])
```

```
% Polynomial curve fitting. Recibe:
% Un vector X que contiene los valores de una variable independiente
% Un vector Y contiene los valores de la variable dependiente
% Un escalar N, grado del polinomio usado para ajustar la curva a los
datasets.
% Retorna los coeficientes, en orden ascendente de un polinomio de grado N
que tiene el minimo error medio cuadrado para los valores de entrada
» X = 1:10;
» Y = 5 + 3*x + randn(1,10)*2 ;
» polyfit(X, Y, 2)
ans =
    0.0170    2.4341    6.4467
```

Gráficos en dos dimensiones

```
% plot acepta dos vectores como argumento y opcionalmente algun comando de
formato
% Grafica el segundo vector verticalmente vs el primero horizontalmente.
» t = 0:1/32:2 ; x = sin(2*pi*t) ;
» plot(t,x);
» plot(t,x,'k--')
```

- Si agregamos un tercer argumento, controla el estilo y el color. Se determina con string

color	marker	line style
"y" yellow	. point	- solid
"m" magenta	o circle	: dotted
r red	x x-mark	-. dashdot
g green	+ plus	-- dashed
b blue	s square	
w white	d diamond	
b black	v triangle	
	^ trianle	
	< triangle	
	> triangle	
	p pentagram	
	h hexagram	

Linspace

La funcion linspace es una herramienta util para producir un vector **equiespaciado**. Acepta dos o tres argumentos escalares.

- Si ingresan dos escalares d1 y d2, linspace retorna un vector de 100 valores equally-spaced entre d1 y d2, incluyendo ambos puntos.
- Si tres escalares d1, d2 y d3 ingresan, linspace devuelve un vector de N valores entre d1 y d2
- ejemplo `linspace(0,5,11)`

Logspace

- Utilizado para producir un vector equally-logarithmically-spaced.
- `logspace(d1, d2)` devuelve un vector de 50 equally-logarithmically-spaced entre 10^{d1} y 10^{d2} .
- `logspace(d1, d2, N)` devuelve un vector de N valores

Loglog

- `loglog` grafica un vector vs otro, pero con escala logaritmica vertical y horizontal. La sintaxis funciona igual que `plot`
- Ejemplo: `loglog(f,abs(H));`

Semilog

- `semilogx` produce un gráfico con escala logaritmica en "x" y lineal en "y"
- `semilogy` logaritmico en "y", lineal en "x"

Stem

- Se utiliza para SVID. Cada punto es indicado por un círculo al final de la linea
- Ejemplo: `stem(t,x,'k','filled')` es el formato clásico utilizado en bibliografía.

Múltiples gráficos

A veces es útil mostrar múltiples ploteos simultaneamente. Se realiza con el comando `subplot`.

- subplot recibe 3 argumentos:
 - numero de filas del grafico
 - numero de columnas del grafico
 - qué posición ocupará el próximo plot en el subplot

```
» subplot(2,2,1) ; plot(t,x,'k') ;  
» subplot(2,2,2) ; stem(t,x,'k','filled') ;  
» x = cos(2*pi*t);  
» subplot(2,2,4) ; plot(t, x, 'k--') ;
```

Axis, Zoom y Grid

Para controlar la escala que se muestra en los plots se utiliza el comando axis.

- 1 argumento de entrada: vector de cuatro elementos

- min x
- max x
- min y
- max y

El comando `zoom` alterna la opcion de zooming. `zoom on` activa el zoom y `zoom off` lo desactiva.

Grid agrega líneas cuadriculadas al gráfico. Los comandos `grid`, `grid on` y `grid off` funcionan como zoom.

Hold

el comando `hold` hace que cualquier curva que se plotee sea en el mismo gráfico y misma escala del ultimo plot graficado, hasta que se ejecute `hold off`

```
» plot(t,x) ;
» hold on;
» plot(t/2,x/2);
» hold off;
```

Title, xlabel, ylabel y text

```
» plot(t,x);
» title('Este es el título del gráfico');
» xlabel('Eje temporal');
» ylabel('Tensión [ V ] (por ejemplo);
```

Gráficos en Tres Dimensiones

```
% plot3:
» plot3(t,cos(2*pi*t),sin(2*pi*t));
» grid on;

% mesh:
% Se utiliza para funciones de dos variables independientes en una
superficie sobre un plano.

% meshgrid
% Toma el valor de dos vectores como argumentos y devuelve dos matrices
donde las columnas de X y Y son copias de los vectores x y y
respectivamente

» x = 0: 0.1 : 2;
» y = -1: 0.1 : 1;
» [ X , Y ] = meshgrid(x,y);
» z = (X .^ 2) .* (Y .^ 3);
» mesh(x,y,z);
» xlabel('x') ; ylabel('y') ; zlabel('x^2.y^3');
```

Stairs

`stairs` acepta dos argumentos como `plot` y `stem`, pero plotea un grafico escalonado.

Image

Acepta una matriz de elementos especificando la intensidad del color.

Manejo de Gráficos

en MATLAB cada gráfico es un objeto, cuya apariencia puede ser modificada utilizando el comando `set`.

```
h = plot(t, sin(2*pi*t));  
set(h, 'LineWidth', 2);
```

El comando `get` devuelve una lista con las propiedades de un objeto.

Ventana de Gráficos

Cuando se utiliza un comando de ploteo, MATLAB abre una ventana llamada `figure` con propiedades por default, y el plot es graficado dentro de `figure`. El usuario puede manejar esta ventana con distintos comandos.