Departamento de Ciencias de la Computación
UNIVERSIDAD DE CHILE

# Auxiliar 15

## Laboratorio de Matlab

Matlab es un programa de computación numérica que permite una amplia gama de operaciones matemáticas.

Matlab es la sigla de MATRIX LABORATORY y no de mathematics lab, se llama así pues se basa en la computación matricial, es decir, la representación de casi todo son matrices y las operaciones son basicamente matriciales.

Como se imaginan, veremos como comenzar a operar matrices en Matlab.

Naturalmente por lo corta de la clase, no podemos profundizar mucho. Se recomiendo mucho dirigirse a la página del laboratorio de cálculo numérico, www.dim.uchile.cl/~labma33a, en especial el pre laboratorio 1.

En Matlab los comandos ingresados se ejecutan automaticamente despues de presionar enter. Además los resultados pueden tanto arrojar como no arrojar un resultado, de cualquier tipo, dependiendo del comando ejecutado.
Podemos elegir entre que aparezca el resultado, o que no aparezca, poniendo comillas al final de un comando.

```
> 1+1
   2
> 1+1;
>
```

Para crear una matriz, asignameros a una variable (A) valores encerrados por parentesis cuadrados ([]) en donde cada valor de una fila se separa con un espacio y cada fila se separa por un punto y coma.

```
>> A=[100 1 10; 101 1 9; 102 1 9]
   A =
       100 1 10
       101 1 9
       102 1 9
```

Además podemos crear matrices especiales como una matriz de unos, una matriz de ceros o una matriz identidad.
Intenta con los comandos:

```
>> ones(3)
>> zeros(3)
>> eye(3)
```

Así como podemos definir matrices cuadradas (de NxN), podemos definir matrices de MxN. Por lo tanto también podremos definir “matrices” de 1XN, es decir, VECTORES.

```
>> [100 1 10]
```

Matlab trabaja con matrices realizando operaciones sobre ellas, para esto cuenta con comandos especiales.
Prueba los siguientes comandos y comenta con tus compañeros lo que sucede.

```
>> A=[1 1 1;1 2 3;1 3 6];
>> A./2
>> A.*A
>> A'
>> A.^2
```

Estas son operaciones comunes que se utilizarán a menudo en la resolución de problemas. Además Matlab cuenta con un catálogo muy extenso de funciones y operaciones, sobre todo para la resolución de sistemas lineales.

El tipo de representación que ocupa Matlab para los polinomios es el de un vector fila con los coeficientesde potencia mayor a menor.

```
>> p=[3 2 0 1];
```

Para evaluar un polinomio $p$ en un punto, Matlab dispone del comando polyval que se usa de la siguiente forma:

```
>> polyval(p,0)
     1
>> polyval(p,1)
     6
```

El tipo de representación que ocupa Matlab para los polinomios es el de un vector fila con los coeficientesde potencia mayor a menor.

```
>> p=[3 2 0 1];
```

Para evaluar un polinomio en un punto, Matlab dispone del comando polyval, al igual que un comando para encontrar las raices, que se usan de la siguiente forma:

```
>> polyval(p,0)
     1
>> roots(p)
```

Matlab además dispone de un entorno para la creación de funciones o métodos utilizables en el entorno. Estas funciones no se compilan y si tuviesen errores, estos aparecerian en tiempo de ejecución. Para ejecutarlas basta con escribir su nombre y entregarle los parámetros respectivos.

Estas funciones se definen en un archivo .m (M file) y tienen la siguiente estructura.

```
function x = funcion(A,b)
```

Esta es una función que se llama funcion, recibe como parámetro A y b y entrega x.

Luego del encabezado podemos escribir todas las instrucciones que queramos y usar todas las variables que queramos. Deberemos crear y modificar la variable x para poder entregarla a quien la llame.

Realizaremos una función llamada “positivizar” que recibirá una matriz y la devolverá pero con todos sus valores en módulo.

El encabezado es el siguiente:

```
function X = positivizar(A)
```

Inmediatamente se nos vendrá a la cabeza utilizar un for y tal vez un if. La sintaxis de estos comandos es como sigue.

```
for i=0:1:n
//instrucciones
end
```

La sintaxis para el comando if es el siguiente.

```
If( condicion )
//instrucciones
end
```

```
function X = positivizar(A)
    X=A;
    for i=0:1:length(A)
        if(A(i)<0)
            X(i)=A(i)*-1;
        end
    end
end
```

Como ven, utilizamos el comando length(A) que entrega el largo (o ancho) de A, o bien el tamaño si correspondiera a una matriz.

Las funciones .m pueden entregar más de un resultado, además se puede recibir uno, parte o todos los resultados de la función.

Para ver esto realizaremos la función normas, que nos devolverá dos reales, la norma euclidiana y la norma infinita calculadas como se define.

La norma euclidiana es la raiz de la suma de los cuadrados de los componentes del vector.
La norma infinita es el mayor de todos los componentes del vector.

El encabezado es el siguiente:

```
function [E,I]=normas(X)
```

```
function [e,i] = normas(X)
    Y=positivizar(X);
    i=max(Y);
    Y=X.*X;
    e=sum(Y);
    e=sqrt(e);
end
```

Reconocemos 4 funciones en esta función. Positivizar, que acabamos de crear, max que elije el máximo componente de una matriz, sum que suma todos lso componentes del vector y sqrt que saca la raiz cuadrada de un real.

Matlab también sabe graficar y para eso cuenta con el comando plot.
El comando plot crea un gráfico bidimensional a partir de dos vectores. La forma de usar este comando es:

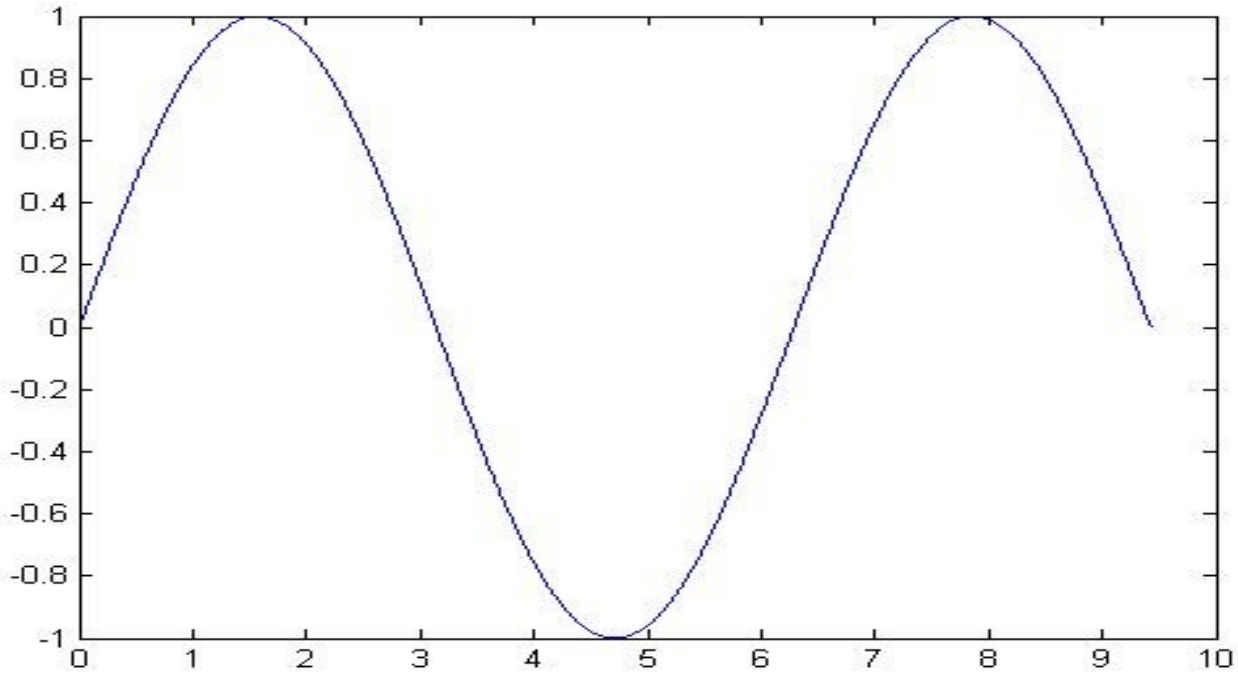
```
plot(x,y)
```

Donde x es el vector con los puntos de la abscisa e y un vector con los puntos de la ordenada.

Para crear un vector de n puntos equiespaciados en un intervalo dado [a,b], tenemos la función linspace que se utiliza como sigue.

```
linspace(a,b,n)
```

Graficaremos la función seno en el intervalo [0,3*pi] para obtener la siguiente figura.




Se le puede aplicar seno a un vector obteniendo un vector con seno componente a componente

```
>> x=linspace(0,3*pi);
>> y=sin(x);
>> plot(x,y)
```

En Matlab tenemos muchas más funciones utilizables en cualquier área de la ingeniería.

Los invitamos a investigar más sobre el programa con el comando help y probando los comandos que se les ocurran.

Intenten haciendo los laboratorios del laboratorio de cálculo numérico.


# FIN

Profesor: Andrés Muñoz

Auxiliares: Oscar Alvarez
Pedro Valencia