MATLAB

Aplicado a telecomunicaciones

Mauricio Ortega Ruiz

Inicio

Índice

Continuar

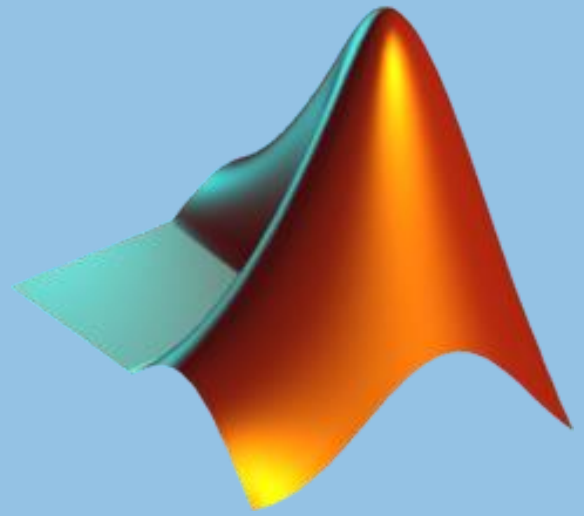
CAPITULO 1

Introducción a MATLAB

Continuar

Introducción

Matlab es un programa de cómputo basado en operaciones matriciales y orientado al cálculo numérico por computadora. Este programa resulta muy útil para ingenieros y para el área científica ya que puede resolver problemas numéricos de una manera más sencilla que utilizando lenguajes de programación de alto nivel.



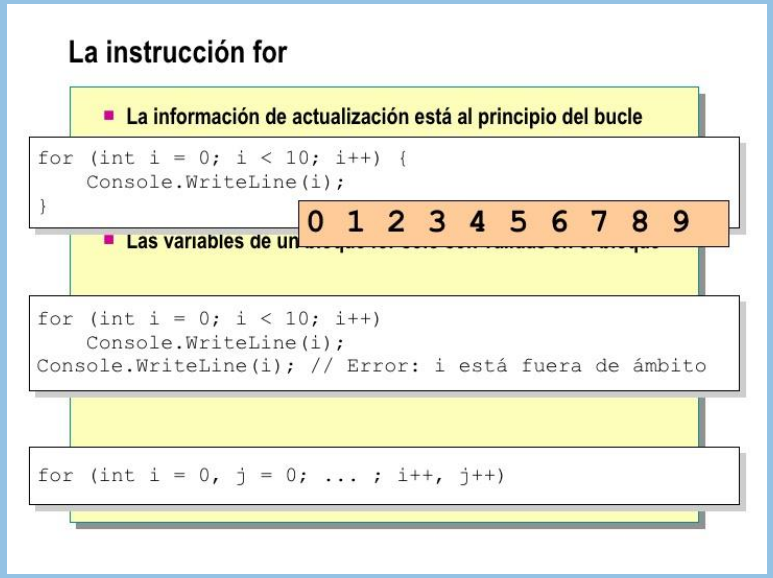
Instrucciones for, while, if.

Las declaraciones de control for, while e if operan como en la mayoría de los lenguajes de cómputo y permiten incorporar condicionales y ciclos secuenciales dentro de la programación MATLAB.

```
1
2 -   nro = input('Ingrese un número positivo: ');
3
4 -   i=0;
5
6 -   while i<=nro
7
8 -       disp(i);
9
10 -      i=i+1;
11
12 -      if i==2
13
14 -          break;
15
16 -      end;
17
18 -  end
```

Instrucción for

Mediante la instrucción for, es posible realizar ciclos de instrucciones *N* veces.



Instrucción while

Esta directiva repite un ciclo de instrucciones hasta cumplirse una cierta condición, las declaraciones se ejecutarán mientras la comparación sea verdadera.

```
clear

function h=mif(x)
    h=2*x^3+x-2;
    endfunction

N=600;
a=0;
b=1;
i=0;
while i<=N & abs(a-b)>10^(-10)& mif(x)~=0;
i=i+1;
 x=(a+b)/2;
 plot(i,x,'*')
  if mif(b)*mif(x)<0;
      a=x;

  else mif(a)*mif(x)<0;
      b=x;

  end
end
```

Instrucción if

Las declaraciones y comandos de la línea central se ejecutan únicamente si la comparación es verdadera. La forma general de la declaración if es:

If comparación

Comandos y declaraciones

end.

```
Private Class SimpleIfStm
  Inherits SimpleReduction
  Implements IContextMethod

  Public IfClause As IContextValue
  Public ThenClause As IContextMethod
  Public ElseClause As IContextMethod

  Public Function Execute() _
  As Object Implements IContextMethod.Execute
    If IfClause.Value = "True" Then
      ThenClause.Execute()
    Else
      If Not ElseClause Is Nothing Then
        ElseClause.Execute()
      End If
    End If
  End Function
End Class
```

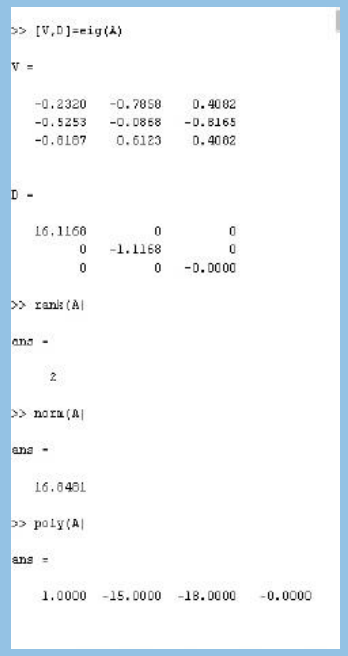
Relaciones de comparación

Las operaciones de relación generan por sí solas un resultado, dependiendo de si son verdaderas o falsa. En su uso dentro de matrices el resultado es una matriz con unos y ceros, de acuerdo con la relación entre los correspondientes elementos.

Operadores Relacionales	
>	Mayor que
<	Menor que
>=	Mayor o igual que
<=	Menor o igual que
!=	Distinto
==	Igual

Generación de vectores y matrices

MATLAB trabaja y realiza operaciones numéricas esencialmente con un solo tipo de objeto: matrices con elementos complejos. En algunas situaciones matrices de 1 por 1 se interpretan como escalares y matrices de un solo renglón o columna como vectores.



Operaciones escalares, vectoriales y matriciales

MATLAB permite la ejecución de las principales operaciones aritméticas básicas, mismas que como se ha mencionado se realizan en matrices. En caso de que el tamaño de las matrices sea incompatible con la operación por realizar, se obtendrá un mensaje de error. Todas las operaciones se aplican a escalares o matrices de 1x1, y a matrices.

$$\vec{A}\cdot\vec{B} = \left(A_x\hat{i} + A_y\hat{j} + A_z\hat{k}\right)\cdot\left(B_x\hat{i} + B_y\hat{j} + B_z\hat{k}\right) =$$
$$= A_x\hat{i}\cdot\left(B_x\hat{i} + B_y\hat{j} + B_z\hat{k}\right) + A_y\hat{j}\cdot\left(B_x\hat{i} + B_y\hat{j} + B_z\hat{k}\right) + A_z\hat{k}\cdot\left(B_x\hat{i} + B_y\hat{j} + B_z\hat{k}\right)$$
$$= A_xB_x\left(\hat{i}\cdot\hat{i}\right) + A_xB_y\left(\hat{i}\cdot\hat{j}\right) + A_xB_z\left(\hat{i}\cdot\hat{k}\right) + A_yB_x\left(\hat{j}\cdot\hat{i}\right) + A_yB_y\left(\hat{j}\cdot\hat{j}\right) + A_yB_z\left(\hat{j}\cdot\hat{k}\right) +$$
$$+ A_zB_x\left(\hat{k}\cdot\hat{i}\right) + A_zB_y\left(\hat{k}\cdot\hat{j}\right) + A_zB_z\left(\hat{k}\cdot\hat{k}\right) =$$
$$= \left(A_xB_x\right) + \left(A_yB_y\right) + \left(A_zB_z\right)$$

Funciones vectoriales

Las funciones vectoriales operan esencialmente sobre un vector y, al aplicarlas sobre una matriz mxn, generan un vector renglón cuyos elementos son el resultado de aplicar la función a cada columna.

$$A \times (B \times C) = (C \times B) \times A = B(A \cdot C) - C(A \cdot B)$$
$$\nabla(uv) = u\nabla v + v\nabla u$$
$$\nabla(A \cdot B) = A \times (\nabla \times B) + (A \cdot \nabla)B + B \times (\nabla \times A) + (B \cdot \nabla)A$$
$$\nabla \cdot uA = u\nabla \cdot A + A \cdot \nabla u$$
$$\nabla \cdot (A \times B) = B \cdot \nabla \times A - A \cdot \nabla \times B$$
$$\nabla \times (uA) = u\nabla \times A - A \times \nabla u$$
$$\nabla \times (A \times B) = (B \cdot \nabla)A + A(\nabla \cdot B) - (A \cdot \nabla)B - B(\nabla \cdot A)$$
$$\overline{(\nabla \cdot A)B} = (A \cdot \nabla)B + B(\nabla \cdot A)$$
$$\nabla \times (\nabla \times A) = \nabla(\nabla \cdot A) - (\nabla \cdot \nabla)A$$

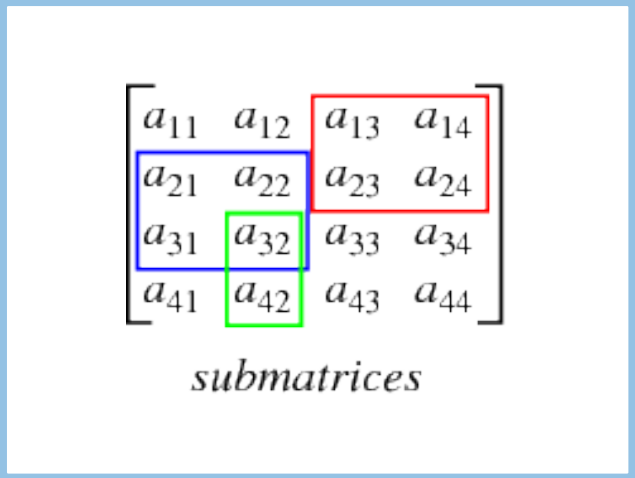
Funciones matriciales

Gran parte de la potencia de MATLAB proviene de las funciones matriciales y operaciones con matrices.

- **Complex.**
- abs - Absolute value.
- angle - Phase angle.
- complex - Construct complex data from real and imaginary parts.
- conj - Complex conjugate.
- imag - Complex imaginary part.
- Rounding and remainder.
- fix - Round towards zero.
- floor - Round towards minus infinity.
- ceil - Round towards plus infinity.
- round - Round towards nearest integer.
- mod - Modulus (signed remainder after division).
- rem - Remainder after

Submatrices y uso del índice

Utilizar submatrices y vectores en operaciones permite minimizar el uso de ciclos for, end y así hacer más eficiente MATLAB.



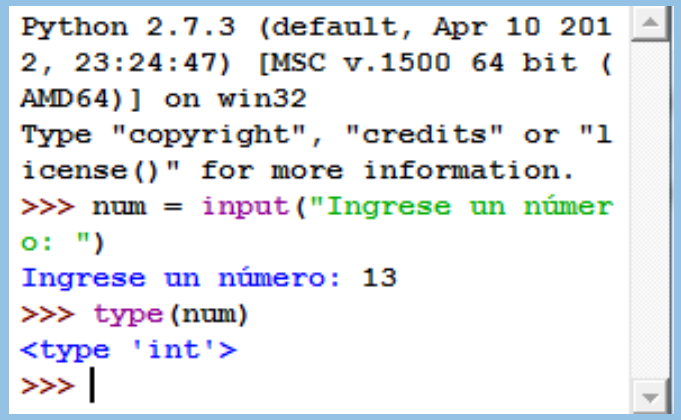
Generación de archivos y funciones tipo m

Todos los comandos y funciones analizadas pueden introducirse a MATLAB desde el teclado y ser ejecutados en línea, sin embargo existen archivos con extensión .m que pueden contener todos los comandos ejecutados por MATLAB.

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & a_{m3} & \cdots & a_{mn} \end{bmatrix}$$

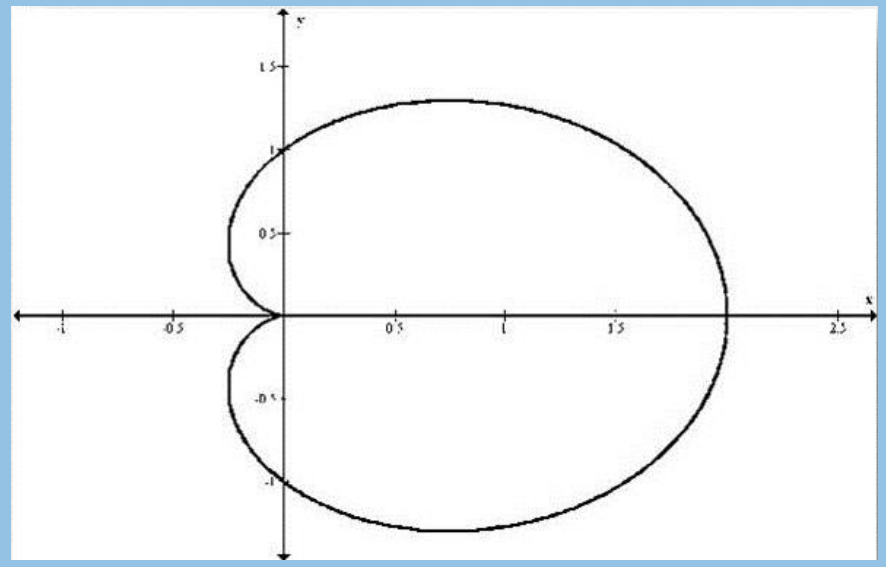
Entrada de datos y de texto

Al encontrar dentro de la función la instrucción de error se detendrá la ejecución del archivo. La entrada de datos se realiza con la función input, y de igual formal la ejecución se detendrá hasta introducir el dato solicitado.



Curvas planas

El comando plot genera gráficas lineales x-y. Los vectores X y Y deben ser del mismo tamaño. El comando plot (X,Y) grafica X vs. Y, donde X corresponde a las ordenadas y Y a las abscisas.



Gráficas en 3 dimensiones

El comando para gráficas en 3 dimensiones es plot3 similar a plot. Si x, y, z son vectores del mismo tamaño entonces plot2 produce una gráfica en perspectiva que genera una curva lineal, pasando a través de los puntos cuyas coordenadas son los respectivos elementos de x, y, z. De igual forma, el título y los ejes se pueden añadir y en el caso del eje z se emplea zlabel.