Introducción a MATLAB

Funciones matemáticas top

Table of Contents

1.Biblioteca de funciones predefinidas

Para facilitarnos la vida MATLAB incorpora una serie de funciones matemáticas. Vamos a revisar algunas de las más usadas. Algunas os sonarán de scripts anteriores, como **`find`**, **`conv`** o **`deconv`**:

Operaciones con matrices/números:

- **`abs:`** Si se aplica a un número real calcula su valor absoluto y si se aplica a un número complejo calcula su módulo.
- **`all:`** Devuelve verdad (uno) si todos los elementos del vector al que se aplica esta función son verdad (distintos de cero).
- **`any:`** Devuelve verdad (uno) si algún elemento del vector al que se aplica esta función es verdad (distinto de cero).
- **`exp:`** Aplica la función exponencial.
- **`find:`** Devuelve un vector con los índices de todos los elementos no nulos del vector al que se aplica. find(a>100) devuelve los índices de todos **`los`** elementos del vector a que son mayores que 100.
- **`log:`** Calcula el logaritmo neperiano.
- **`log10:`** Calcula el logaritmo decimal.
- **`max:`** Calcula el valor máximo de los elementos de un vector. Si se desea, permite conocer en qué posición del vector se encuentra el elemento de valor máximo.
- **`min:`** Calcula el valor mínimo de los elementos de un vector. Si se desea, permite conocer en qué posición del vector se encuentra el elemento de valor mínimo.
- **`mod / rem:`** Calcula el resto de una división.
- **`sqrt:`** Calcula la raíz cuadrada.

Operaciones con polinomios:

- **`conv:`** Calcula el producto de dos polinomios.
- **`deconv:`** Calcula el cociente y el resto de la división de dos polinomios.
- **`poly:`** Genera el polinomio característico de una matriz. Genera el vector de coeficientes de un polinomio a partir de un vector que contiene sus raíces.
- **`roots:`** Calcula las raíces de un polinomio a partir de su vector de coeficientes.

Trabajando con funciones:

- **`fminsearch:`** Calcula el valor de la variable independiente para el cual una función matemática alcanza el valor mínimo. Hay que especificarle como parámetro de entrada el punto en el que empieza a buscar el mínimo.
- **`quad:`** Obtiene la integral definida de una función matemática por aproximación cuadrática (método de Simpson). La función considerada debe indicarse entre comillas simples.
- **`quadl:`** Obtiene la integral definida de una función matemática por aproximación cuadrática (método de Lobatto). La función considerada debe indicarse entre comillas simples.

Trigonometría:

- **`cos:`** Calcula el coseno de una cantidad expresada en radianes.
- **`sin:`** Calcula el seno de una cantidad expresada en radianes.

Tarea 1: Calcular el seno de 60º.

Pista: MATLAB, en su eterna sabiduría, nos provee de los comandos **rad2deg** y **deg2rad**. Puedes emplear **help** para comprobar como se comportan.

```
% Tu código aquí
p = deg2rad(60)
```

```
p = 1.0472
```

```
q = sin(p)
```

```
q = 0.8660
```

```
help rad2deg
```

```
 rad2deg - Convert angle from radians to degrees
    This MATLAB function converts angle units from radians to degrees for
    each element of R.

    Syntax
      D = rad2deg(R)

    Input Arguments
      R - Angle in radians
        scalar | vector | matrix | multidimensional array

    Output Arguments
      D - Angle in degrees
        scalar | vector | matrix | multidimensional array

    Examples
      pi in Degrees
      Spherical Distance

    See also deg2rad

    Introduced in MATLAB in R2015b
    Documentation for rad2deg
    Other uses of rad2deg
```

Resultado esperado:

```
s_rad = 0.8660
```

Tarea 2: El siguiente código define una función `fun` con dos variables independientes `x(1)` y `x(2)`, y un punto de partida `x0`. Calcula cual es el valor de dichas variables para el cual la función alcanza el valor mínimo.

```
fun = @(x)100*(x(2) - x(1)^2)^2 + (1 - x(1))^2;
x0 = [-1.2,1];
% Tu código aquí
help fminsearch
```

```
 fminsearch - Find minimum of unconstrained multivariable function using derivative-free method
    Nonlinear programming solver.

    Syntax
      x = fminsearch(fun,x0)
      x = fminsearch(fun,x0,options)
      x = fminsearch(problem)
      [x,fval] = fminsearch(___)
      [x,fval,exitflag] = fminsearch(___)
      [x,fval,exitflag,output] = fminsearch(___)

    Input Arguments
      fun - Function to minimize
        function handle | function name
      x0 - Initial point
        real vector | real array
      options - Optimization options
        structure such as optimset returns
      problem - Problem structure
        structure

    Output Arguments
      x - Solution
        real vector | real array
      fval - Objective function value at solution
        real number
      exitflag - Reason fminsearch stopped
        integer
      output - Information about the optimization process
        structure

    Examples
      Minimize Rosenbrock's Function
      Monitor Optimization Process
      Minimize a Function Specified by a File
      Minimize with Extra Parameters
      Find Minimum Location and Value
      Inspect Optimization Process

    See also fminbnd, optimset, Optimize

    Introduced in MATLAB before R2006a
    Documentation for fminsearch
```

```
x = fminsearch(fun,x0)
```

```
x = 1×2
```

```
     1.0000    1.0000
```

```

```

Resultado esperado:

```
x = 1×2
     1.0000    1.0000
```

2. Variables predefinidas/Constantes

MATLAB también incorpora una serie de variables predefinidas que se podrían interpretar como valores constantes, y que pueden ser directamente añadidos a expresiones. Juega un poco con ellas ejecutando estas celdas de código:

```
% La famosa constante pi
pi
```

```
ans = 3.1416
```

```
% Valor infinito
Inf
```

```
ans = Inf
```

```
% Valor no un número (not a number)
NaN
```

```
ans = NaN
```

```
% Parte imaginaria de un número complejo
1i
```

```
ans = 0.0000 + 1.0000i
```

```
% Parte imaginaria de un número complejo
1j
```

```
ans = 0.0000 + 1.0000i
```

Tarea 3: Calcula la circunferencia de un círculo que tiene como diámetro 4 metros, empleando la constante `pi`.

```
% Tu código aquí
c = 2 * pi * 2
```

```
c = 12.5664
```

```

```

Resultado esperado: `c = 12.5664`