

Construcción de funciones en MATLAB

Matlab tiene muchas *funciones construidas*, pero nosotros podemos construir también funciones.

Ejemplos de funciones construidas que trae Matlab.

Funciones trigonométricas:

sin(x)
cos(x)
tan(x)
asin(x)
acos(x)
atan(x)
exp(x)
log(x)
log10(x)

Funciones matemáticas diversas:

abs(x) : Retorna el valor absoluto de x.
fix(x) : Retorna la parte entera de x.
rem(x,y) : Retorna el resto (*remainder*) de dividir x entre y.
round(x) : Redondea x al más próximo entero.
sign(x) : Retorna -1 si x es menor que 0, 0 si x es 0, +1 si x es mayor que 0.
sqrt(x) : Retorna la raíz cuadrada de x.
plot(x,y) : Pinta un punto en la posición (x,y).

Los conceptos *argumento* y *retorno* en programación.

En el comando `t = sin(x)`, **x es el argumento**. A la hora de ejecutar la función en el argumento se especificará un valor concreto del que deseamos hallar su seno.

Una función puede tener, uno, dos, o varios argumentos.

El “valor de retorno” es el número resultante de calcular el seno del argumento, lo cual se asigna a t.

sin(1) → 0.8414,
sin(2) → 0.9092

se habla que la función sin(x) retorna un valor. Pero debe tenerse en cuenta que las funciones no tienen obligatoriamente sólo un valor de retorno. Puede haber funciones con dos, tres o más valores de retorno:

EJEMPLOS

$[r] = f(x, y) \rightarrow$ dos argumentos, un valor de retorno.

$[r1, r2] = f(x, y, z) \rightarrow$ tres argumentos, dos valor de retorno.

$[u, v, w] = f(x, y, z, p, q, r) \rightarrow$ seis argumentos, tres valor de retorno.

ESTRUCTURA GENERAL DE UNA FUNCIÓN

function [v1, v2] = nombre(x, y, z)

INICIO

BLOQUE GENERAL

FINAL

INDICACIONES:

* En Matlab **abrir un fichero nuevo** del tipo ‘función’:

File \rightarrow New \rightarrow Function M-file

* Dentro del Editor de Matlab, aparece la primea línea que hay que **completar** con el nombre de la función, argumentos, y variables de retorno.

* Luego escribir el código (INICIO, BLOQUE GENERAL y FINAL) y **salvarla** con el mismo nombre que se le ha dado a la función.

* Para ejecutarla -desde la ventana de comandos de Matlab- hay que escribir por ejemplo (para el caso de la función de arriba):

$[a, b] = \text{nombre}(1, 5, -3)$

Es decir, indicando dos nombres de variables de retorno, y dando los correspondientes valores a cada uno de los argumentos.

EJEMPLO:

Introducción

Construiremos una función para calcular raíces por el “método de Newton”. Pero previamente vamos a ver cómo se calcula $\sqrt{5}$ por el método de Newton:

- * 1º: Se toma la primera aproximación “la mitad de 5” : x_0 .
- * 2º. Se toma la mitad de $(x_0 + 5/x_0)$, como nueva aproximación a la raíz: x_1 .
- * 3º. Se toma la mitad de $(x_1 + 5/x_1)$, como nueva aproximación a la raíz: x_2 .
- * 4º. Se toma la mitad de $(x_2 + 5/x_2)$, como nueva aproximación a la raíz: x_3 .
- * Etcétera. El proceso se repite una cantidad N de veces.

1º) Se toma $x_0 = \frac{5}{2}$ (semilla inicial)

2º) Se toma $x_1 = \frac{x_0 + \frac{5}{x_0}}{2}$

3º) Se toma $x_2 = \frac{x_1 + \frac{5}{x_1}}{2}$
etc., etc.

Nº) Se toma $x_n = \frac{x_{n-1} + \frac{5}{x_{n-1}}}{2}$ es la raíz aproximada.

Hagamos primeramente un programa raiz5.m, para calcular la raíz cuadrada de 5 mediante 20 pasos de este tipo.

```
=====
% raiz5.m
% Cálculo de la raíz de 5 por el método de Newton
% Autor: Juan C. Gorostizaga (17/10/2009)

x0 = 2.5; % semilla inicial
T = 20; % cantidad total de iteraciones

for i = 1:T
    x = 1/2 * (x0 + 5/x0);
    x0 = x;
end;
disp('Raíz de 5 = '), disp(x)
=====
```

Al ejecutar este programa nos da: 2.23606797749979

Construyamos ahora una función que nos sirva para calcular raíces cuadradas de cualquier número positivo.

Construcción de una función:

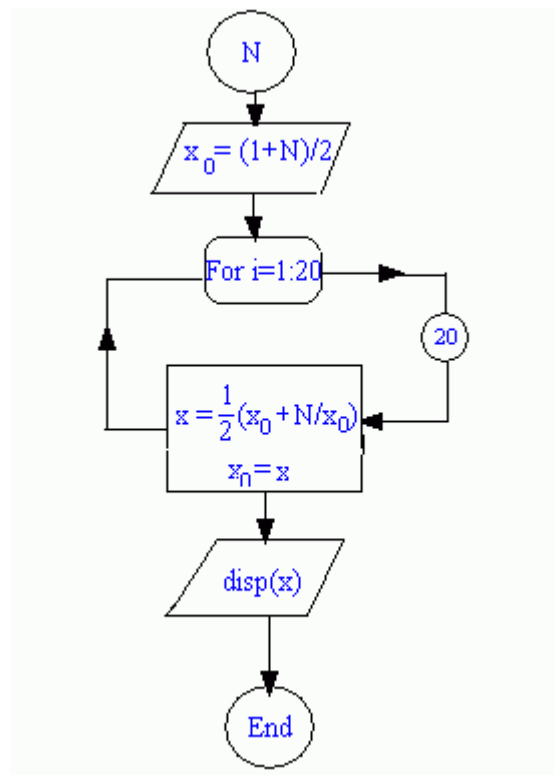
```
function [v1] = raiznewton(n)
% raiznewton(n) computa la raíz cuadrada por el método de Newton
% autor: Pepito Ramirez (10/02/2011)

% Comprobar si n es positivo
if n < 0
    error('En raiznewton(n) : el argumento n debe ser positivo');
end;

x0 = (1+n)/2; % aproximación inicial

for i = 1:20
    x = (x0 + n/x0)/2;
    if abs(x - x0)/x < eps,
        break,
    end;
    x0 = x
end
v1 = x;
```

Diagrama de flujo



La manera de utilizar esta función desde la ventana de comandos, o bien desde el interior de otro programa es la siguiente:

```
r = raiznewton(1257)
```

```
ans =
```

```
r =  
35.45419580247167
```

```
>>raiznewton(-2459)
```

```
??? Error using ==> raiznewton
```

En raiznewton(n) : el argumento n debe ser positivo