

Óptica Computacional

1° Cuatrimestre de 2019

Clase N° 3 Funciones

Tenemos funciones propias del Scilab
O funciones definidas por el usuario

Funciones del Scilab

Con el comando → --> `help "elementary function"`

Se accede al “menú de ayuda” donde se muestra una lista completa de las funciones que Scilab ya tiene incorporadas. Ejemplo, **funciones trigonométricas**, ***sin, cos, tan*** y **sus inversas**, ***asin, acos, atan***

```

--> sin(%pi/2)
ans =
    1.
--> sin(90)
ans =
    0.8939967
--> sin(%pi/3)
ans =
    0.8660254
--> asin(ans)
ans =
    1.0471976

```

Argumento de la función "seno"
Se expresa en "radianes"

$\pi/2(\text{rad}) = 90^\circ$

$\sin(90) \neq \sin(\%pi/2)$

Recuerde que en la variable "ans" siempre se guarda el último resultado y que puede usarse como variable de entrada en el siguiente comando inmediato

```

--> %pi/3
ans =
    1.0471976

```

Ejemplos.

1- En la "consola", calcule:

1- $\tan(0.5)$ and $\sin(0.5)/\cos(0.5)$;

2- $\log_{10}(2)$, $\log(2)$, $\log(10)$ and $\log(2)/\log(10)$.

Es posible definir una variable y luego operar sobre ella. Por ejemplo:

```
-->x = 2
```

```
x =
    2.
```

```
-->x^2+3*x+7
```

```
ans =
    17.
```

Ejercicio: Asigne el valor 0.25 a la variable y. Luego calcule $y^2 - 2y + 3$

Puedo definir una variable que tome valores equidistantes en un dado rango

```
x=linspace(-2*pi,2*pi,300)
y=sin(x)
plot(x,y);
```

```
z=[-2*pi,%pi/150,2*pi]
w=sin(x)
plot(w,z);
```

Ejercicio: Graficar la $\tan(x)$ en el rango $-\pi/2, \pi/2$

Scilab Help >> Elementary Functions > Floating point > int

int

round towards zero

Syntax

```
y=int(X)
```

Arguments

x real matrix
y integer matrix

Description

int(X) returns the integer part of the real matrix x. Same as fix.

Examples

```
int([1.3 1.5 1.7 2.5 3.7])
// Notice the result for x negative
int([-1.3 -1.5 -1.7 -2.5 -3.7])
```

Algunas funciones operan sobre el formato numérico
Este comando devuelve la "parte entera" de una matriz de variables reales

Funciones definidas por el usuario

```
function f = myquadratic ( x )  
f = x^2  
endfunction
```

Definamos valores en el intervalo [1, 10].

```
xdata = linspace ( 1 , 10 , 50 );  
ydata = myquadratic ( xdata );  
plot ( xdata , ydata )
```