

CR 20. INTRODUCCION A SCILAB

JUAN CARLOS HENAO LÓPEZ

Ingeniero Electricista
Universidad Católica Popular del Risaralda
Estudiante de Maestría en Ingeniería
Juank_henao@hotmail.com

JAMES ANDRES BARRERA MONCADA

Ingeniero Electricista
Universidad Católica Popular del Risaralda
Estudiante de Maestría en Instrumentación Física
James.barrera@ucpr.edu.co

RESUMEN: Scilab es un software de cálculo científico orientado a la computación numérica que posee una extraordinaria versatilidad y capacidad para resolver problemas de matemática aplicada, física, ingeniería, procesamiento de señales y otras muchas aplicaciones. Su base la constituye un sofisticado intérprete formado por diferentes rutinas de cálculo matricial, análisis numérico y visualización gráfica.

Algunas de las aplicaciones que puede desarrollar este lenguaje, se centran en el manejo de constantes, variables, y funciones, al igual que su graficación, que sumando con herramientas de análisis, se convierte en un software muy útil para cualquier ingeniería.

CONSTANTES

Son valores reales o complejos que se guardan en literales y que son útiles al momento de realizar manipulaciones algebraicas.

En pantalla escribir

```
--> a=14;
```

```
--> A=16
```

A partir de estas asignaciones, se pueden realizar diferentes operaciones

Suma

```
-->a+A
```

Resta	-->a-A
Multiplicación	-->A*a
División	-->a/A
Potencia	-->a**2
Raíz Cuadrada	-->sqrt(A)

De igual forma, SCILAB maneja constantes propias

%i	Número imaginario
%e	Número exponencial
%pi	Número Pi = 3,14159...
%eps	Número muy cercano a 0
%inf	Número muy grande
%t	Valor booleano verdadero
%f	Valor booleano falso

Estas cantidades guardan entre si relaciones de orden, especialmente útiles al momento de programar.

Comparador	Significado
>	Mayor
>=	Mayor o Igual
<	Menor
<=	Menor o igual
==	Igual
<>	No igual

Scilab permite calcular el valor de funciones trascendentales por medio de palabras reservadas en su sistema, algunas de estas funciones son:

Función	Prompt
Seno	-->sin(x)
Arco seno	-->asin(x)
Coseno	-->cos(x)
Arco coseno	-->acos(x)
Tangente	-->tan(x)
Arco tangente	-->atan(x)
Exponencial	-->exp(x)
Logaritmo Natural	-->log(x)
Logaritmo Decimal	-->log10(x)
	-->floor(x)
	-->ceil(x)

La precisión o forma en la cual se presentan por pantalla o se manipulan datos numéricos se pueden controlar por medio de los siguientes comandos.

- >format(14) 14 elementos
- >format('e') formato científico o exponencial, coma flotante
- >format('v') formato variable (por defecto)
- >format('v',20) formato variable con 20 dígitos
- >format('e',15) formato científico con 15 dígitos

Existe también la posibilidad de operar cantidades complejas.

So numeros de la forma $a+bi$ donde i es la parte imaginaria, para lo cual a manera de ejemplo escribir:

--> $a=2-2i$

--> $b=1+4i$

FUNCIÓN

Una función es una relación compuesta por variables, constates y operaciones

$$f(x) = x^2 - 3x - 4$$

$$g(x) = \text{sen}(x + 1)$$

$$h(m) = \frac{\text{sen}m^2}{\sqrt{m^2 + 1}}$$

En Scilab se pueden definir de diversas formas:

De manera on-line

6. **function [arg sal]=nombre(arg ent), función, endfunction**

-->function [z]=juan(x), z=x^2+1, endfunction

--> juan(3)

O también

-->deff('[z]=juan(x,y)',z=sqrt(x^2+y^2))

-->juan(3,4)

VECTORES

Vector es un arreglo de objetos que puede representarse en forma de fila o de columna.

-->a=[1, 2, 3, 4]

-->a=[1;2;3;4]

Traspuesta de un vector

--> a=[1;2;3;4]'

Ejercicio: Una persona tiene tres cuentas bancarias (A, B y C) y hace tres consignaciones en cada banco en tres momentos diferentes. Usando tres vectores, para cada día, representar la información:

	Día 1	Día 2	Día 3
Banco A	US\$250	\$1.345.000	EUR 200
Banco B	US\$300	\$700.000	EUR 400
Banco C	US\$450	\$125.000	EUR 180

MATRICES

Las matrices son arreglos de varias filas y varias columnas, que se introduce en SciLab de la siguiente manera

```
-->a=[1,2,3;3,4,5;6,7,8];
```

```
-->A=[0,2,3;5,0,-1;3,-3,-5];
```

Ejercicios: Desarrollar en SciLab los siguientes problemas

```
-->a+A          -->A*a
```

```
-->A+a          -->a*A
```

```
-->A-a          -->a.*A
```

```
-->a-A          -->A.*a
```

```
-->2+A          -->1/A
```

```
-->2*A
```

```
-->A/2
```

Algunas matrices muy útiles

```
[]              matriz vacía
```

```
diag(x)         Matriz de ceros con diagonal x
```

```
diag(X)         Vector con diagonal de la matriz X
```

eye (m,m)	matriz identidad
Ones(m,n)	Matriz of unos
Rand (m,n)	Matriz aleatoria de distribución
Zeros(m,n)	Matriz de ceros

Las operaciones, entre otras que se pueden realizar con estas matrices son:

- > sum(A) suma de las componentes de la matriz A
- > sum(A,1), suma de los elementos de columna de A
- > sum(A,2), suma de los elementos de cada fila de A
- > trace(A) traza de A
- > prod(A) producto de las componentes de la matriz A
- > prod(A,1), producto elementos de cada columna de A
- > prod(A,2), producto de los elementos de cada fila de A
- > max(A) máximo de las componentes de la matriz A
- > det(A) determinante de la matriz cuadrada A
- > rank(A) rango de la matriz
- > inv(A) inversa de la matriz A
- > lu(A) factorización LU de la matriz A

POLINOMIOS

Es una expresión algebraica de la forma

$$p(x) = \sum_{i=0}^{n-1} c_i x^{n-1}$$

A manera de ejemplo, considérese el siguiente polinomio, al cual se le obtiene sus raíces.

- $f(x) = 2 - x + 2x^2 - 3x^3$
- `-->poly([2,-1,2,-3],"x","coeff");`
- `→roots(p);`

GRAFICAS CON SCILAB

Otras de las potencialidades que tiene SCILAB, son sus herramientas y facilidades para representar de forma gráfica, diferentes tipos de funciones.

Comando `plot(arg,arg);`

`-->A=[1,2,3,4];`

`-->B=[3,1,4,3];`

`-->plot(A,B);`

`-->plot2d(A,B);`

Ejemplo: graficar la función $y = \cos(x)$

`-->x=linspace(-1,4)';`

`-->y=cos(x)`

`-->plot2d(x,y);`

Ejemplo: graficar la función $y = \cos(x) * \cos(2x)$

`-->x=linspace(-1,4)';`

`-->y=cos(x).*cos(2*x)`

$$r = \sin(2\theta) \cos(3\theta)$$

`-->plot2d(x,y)`

`-->plot2d(y)`

Ejemplo: graficar la función $y = \cos(x) * \cos(2x)$ $y = \cos(2x)$

```
-->x=linspace(-1,4)';
-->y=cos(x).*cos(2*x);
-->z=cos(2*x)
-->w=[y,z];
-->plot2d(x,w)
-->plot2d(y)
```

Para graficar en coordenadas polares, se sigue un procedimiento similar al expuesto anteriormente

Se usa el comando -->polarplot2d(arg1,arg2)

Ejemplo: Graficar en coordenadas polares

$$r = \text{sen}(2\theta)\cos(3\theta)$$

```
-->th=[0:0.1:2*pi]';
-->r=sin(2*th).*cos(3*th);
-->polarplot2d(th,r)
-->x=linspace(0,2*pi)
-->y=sin(x);
-->plot2d2(x,y)
-->plot2d3(x,y)
-->plot2d4(x,y)
```

Para curvas paramétricas se tiene

```
--> linspace(0,8*pi);
--> param3d(t.*sin(t),t.*cos(t),3*t)
```

Para graficar funciones en tercera dimensión, se sigue el procedimiento

Comando -->plot3d(x,y,z)

Ejercicio: Dibuja la superficie definida por la función $z=f(x,y)$.

x : vector de dimensión n

y : vector de dimensión m

z : matriz de dimensión nxm

x e y contienen las coordenadas de los puntos de la malla rectangular sobre la que se dibuja la función z contiene los valores de la función en los nodos: $z(i,j)=f(x(i),y(j))$

Para construir la matriz z a partir de los vectores x e y puede ser útil la función

```
-->[xm,ym]=ndgrid(x,y)
```

Ejemplo: graficar $f(x,y)=\cos x \cos y$

```
--> x=linspace(0,2*%pi);
```

```
-->y=linspace(0,4*%pi);
```

```
-->[xm,ym]=ndgrid(x,y);
```

```
-->z=cos(xm).*cos(ym);
```

```
-->plot3d(x,y,z)
```

```
-->plot3d1(x,y,z)
```

```
-->grayplot(x,y,z)
```

```
-->sgrayplot(x,y,z)
```

Las curvas de nivel, o líneas de contorno se obtiene con la serie de instrucciones

Se usa el comando `--> contour(x,y,z,nz)`

```
-->x=linspace(-1,1);
```

```
-->y=linspace(-1,1);
```

```
-->[xm,ym]=ndgrid(x,y);
```

```
-->contour(x,y,z,20)
```

```
-->contourf(x,y,z,20)
```