

Universidad del Valle  
Escuela de Ingeniería de Sistemas y Computación  
Curso: Métodos Numéricos  
Docente: Daniel Barragán Calderón

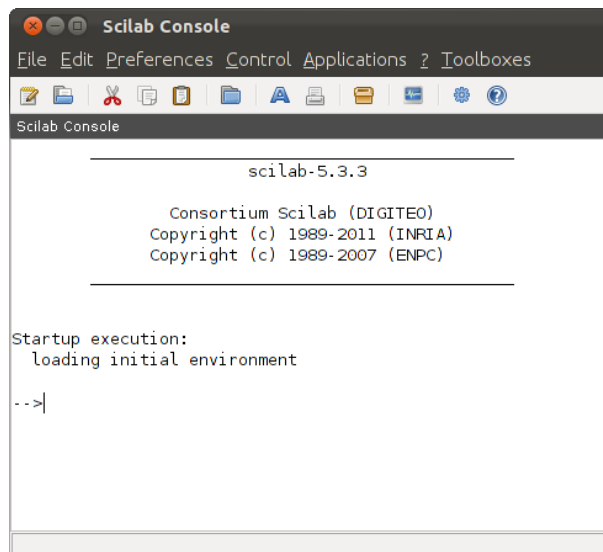
## Clase Introducción a la Programación con Scilab (Parte 5)

### Introducción

Scilab fue creado en 1990 por investigadores en INRIA (*Institut national de recherche en informatique et en automatique*) y ENPC (*École nationale des ponts et chaussées*). Scilab es un lenguaje de programación de código abierto, multiplataforma, orientado a cálculos numéricos. Puede ser usado para procesamiento de señales, análisis estadístico, tratamiento de imágenes, simulación de fluidos, optimización y modelamiento y simulación de sistemas dinámicos.

Enlace de descarga:

<http://www.scilab.org/>



Interfaz Gráfica Scilab

## Guía Básica

En esta guía se abordaran otros aspectos de la programación y visualización de funciones con Scilab.

### Sistemas Lineales

Scilab permite crear una matriz Identidad por medio de la función **eye**.

```
--> eye(3,3)
ans =
  1.  0.  0.
  0.  1.  0.
  0.  0.  1.
```

Scilab permite obtener la transpuesta de una matriz o vector por medio del operador '.

```
--> b = [588.6; 686.7; 784.8]
b =
  588.6
  686.7
  784.8
```

```
--> btransp = b'
btransp =
  588.6  686.7  784.8
```

Scilab permite obtener la inversa de una matriz cuadrada por medio de la función **inv**.

```
--> A = [150 -100 0; -100 150 -50; 0 -50 50]
A =
  150. - 100.  0.
 - 100.  150. - 50.
   0.  - 50.  50.
```

```
--> AI = inv(A)
AI =
  0.02  0.02  0.02
  0.02  0.03  0.03
  0.02  0.03  0.05
```

Scilab permite obtener la pseudoinversa de una matriz no cuadrada por medio de la función **pinv**

```
--> A = [1 3; 3 8; 1 9]
A =
  1.  3.
  3.  8.
  1.  9.
```

```
-->API = pinv(A)
API =
    0.1155779    0.4371859 - 0.4271357
   -0.0075377 - 0.0502513    0.1582915
```

Scilab permite encontrar el determinante de una matriz por medio de la función ***det***.

```
--> A
A =
    150. - 100.    0.
   -100.    150. - 50.
     0.   -50.    50.
```

```
--> D = det(A)
D =
    250000.
```

Scilab permite concatenar matrices como en el siguiente ejemplo:

```
--> P = eye(3,3)
P =
     1.    0.    0.
     0.    1.    0.
     0.    0.    1.

--> A = [150 -100 0;-100 150 -50; 0 -50 50]
A =
    150. - 100.    0.
   -100.    150. - 50.
     0.   -50.    50.

--> Aextend = [A P]
Aextend =
    150. - 100.    0.    1.    0.    0.
   -100.    150. - 50.    0.    1.    0.
     0.   -50.    50.    0.    0.    1.
```

Scilab permite realizar cálculos matriciales de forma ágil. El siguiente ejemplo muestra como encontrar el valor de x1 en un sistema de ecuaciones lineales, empleando la ***Regla de Cramer***.

```
-->A = [150 -100 0;-100 150 -50; 0 -50 50]
A =
    150. - 100.    0.
   -100.    150. - 50.
     0.   -50.    50.
```

```
-->D = det(A)
```

```
D =  
250000.
```

```
-->A(:,1) = [588.6; 686.7; 784.8]
```

```
A =  
588.6 - 100. 0.  
686.7 150. - 50.  
784.8 - 50. 50.
```

```
-->x1 = det(A)/D
```

```
x1 =  
41.202
```

Scilab permite encontrar la solución de un sistema de ecuaciones lineales de la forma  $Ax=b$  por medio del operador \ (barra invertida)

```
--> A = [150 -100 0;-100 150 -50; 0 -50 50]
```

```
A =  
150. - 100. 0.  
- 100. 150. - 50.  
0. - 50. 50.
```

```
--> b = [588.6; 686.7; 784.8]
```

```
b =  
588.6  
686.7  
784.8
```

```
--> x = A\b
```

```
x =  
41.202  
55.917  
71.613
```

Scilab permite obtener la factorización LU de una matriz A por medio de la función **lu**.

```
--> A = [150 -100 0;-100 150 -50; 0 -50 50]
```

```
A =  
150. - 100. 0.  
- 100. 150. - 50.  
0. - 50. 50.
```

```
--> [L,U] = lu(A)
U =
 150. - 100.    0.
  0.  83.333333 - 50.
  0.   0.    20.
L =
 1.    0.  0.
-0.6666667  1.  0.
 0.    -0.6  1.
```

Para generar la solución a partir de la factorización LU de A, se hacen las siguientes operaciones:

```
--> d = L\b
d =
 588.6
1079.1
1432.26
```

```
--> x = U\d
x =
 41.202
 55.917
 71.613
```

Scilab permite realizar la factorización de Cholesky para matrices simétricas por medio de la función **chol**.

```
--> A = [150 -100 0;-100 150 -50; 0 -50 50]
A =
 150. - 100.  0.
-100.  150. - 50.
  0.  -50.  50.
```

```
--> U = chol(A)
U =
12.247449 - 8.1649658  0.
 0.    9.1287093 - 5.4772256
 0.    0.    4.472136
```

Scilab permite encontrar la norma p (  $p = 1, 2, \dots, \infty$  ) de un vector o matriz por medio de la función **norm**.

```
--> A = [150 -100 0;-100 150 -50; 0 -50 50]
A =
 150. - 100.  0.
-100.  150. - 50.
  0.  -50.  50.
```

```
--> norm(A,1)
ans =
    300.
```

```
--> norm(A,2)
ans =
    256.24427
```

```
--> norm(A,%inf)
ans =
    300.
```

```
--> norm(A,'fro')
ans =
    269.25824
```

Scilab permite encontrar el número de condición de una matriz por medio de la función ***cond***.

```
--> A = [150 -100 0;-100 150 -50; 0 -50 50]
A =
    150. - 100.   0.
   -100.   150. - 50.
     0.   -50.   50.
```

```
--> cond(A)
ans =
    21.493142
```

Scilab permite encontrar los eigenvalores y los eigenvectores de una matriz por medio de la función ***spec***.

```
--> A = [150 -100 0;-100 150 -50; 0 -50 50]
A =
    150. - 100.   0.
   -100.   150. - 50.
     0.   -50.   50.
```

```
--> [v, d] = spec(A)
d =
    11.922141   0.   0.
     0.   81.833588   0.
     0.   0.   256.24427
v =
 -0.4018081  0.6188664 -0.6749479
 -0.5548080  0.4218590  0.7170935
 -0.7285179 -0.6626005 -0.1738457
```

Scilab permite resolver sistemas de ecuaciones no lineales por medio de la función ***fsolve***.

Para calcular la solución de:

$$2 \operatorname{sen}(x+2)+3x=1$$

```
--> xini = 0;  
--> deff('y=f(x)', 'y=2*sin(x+2)+3*x-1')  
--> x = fsolve(xini, f)  
x =  
- 0.3300595
```

Para calcular la solución de:

$$\begin{aligned} x^3 + y^3 &= 3 \\ x^2 + y^2 - 2y &= 0 \end{aligned}$$

```
--> xini = [1 1];  
--> deff('[y]=g(x)', 'y=[x(1)^3+x(2)^3-3, x(1)^2+x(2)^2-2*x(2)]');  
--> [x,v]=fsolve(xini,g)  
v =  
0. 0.  
x =  
0.9587068 1.2843962
```

## Problemas

1. Resuelva manualmente empleando los métodos vistos en clase un ejercicio por cada uno de los anexos problemasCap8.pdf a problemasCap12.pdf
2. Realice la implementación en Scilab de alguno de los métodos iterativos, como se explica en la sección de Métodos Iterativos (Iterative Methods) del libro Applied Numerical Methods with MATLAB for Engineers and Scientists del autor Steven C. Chapra.