

# Introducción a Scilab

Andrés A. Ramírez D, \*Rodrigo Moreno G.\*\*

20 de agosto de 2009

## Índice

<b>1. Que es Scilab?</b>	<b>2</b>
<b>2. Iniciando en Scilab</b>	<b>2</b>
2.1. Descripción de la Interfaz Gráfica . . . . .	2
2.2. La ayuda, Como Usarla? . . . . .	2
<b>3. Lenguaje de Programación</b>	<b>2</b>
3.1. Comandos Básicos . . . . .	2
3.2. Vectores y Matrices . . . . .	5
3.2.1. Definiciones . . . . .	5
3.2.2. Operaciones . . . . .	5
3.3. Estructuras Básicas de Programación . . . . .	6
3.3.1. For . . . . .	6
3.3.2. While . . . . .	6
3.3.3. If . . . . .	6
3.4. Graficar . . . . .	6
3.4.1. plot . . . . .	6
3.4.2. plot2d . . . . .	7
3.4.3. plot3d . . . . .	7
3.5. Construcción de Funciones . . . . .	7
3.5.1. pwd . . . . .	7
3.5.2. chdir . . . . .	7
3.5.3. exec . . . . .	7
3.5.4. SciPad . . . . .	7
3.5.5. function . . . . .	7
<b>4. Ejemplos</b>	<b>7</b>
<b>5. Ejercicios</b>	<b>9</b>

---

\*aaramirezdu@unal.edu.co

\*\*rmorenoga@unal.edu.co

## 1. Que es Scilab?

Scilab es un paquete de software científico para computación numérica, en especial para el la simulación de sistemas de control y aplicaciones en procesamiento de señales.

Scilab fue desarrollado en el INRIA, Institut National de Recherche en Informatique et Automatique, un instituto francés de investigación. Posteriormente colaboró la escuela de ingenieros ENPC, Ecole Nationale de Ponts et Chaussées.

Actualmente hay un gran consorcio con empresas como Renault, Peugeot-Citroen, CEA Commissariat à l’Energie Atomique, CNES Centre National d’Etudes spatiales, Dassault Aviation, EDF Electricité de France, Thales...

Sus principales características son:

- Software para cálculo científico.
- Interactivo
- Programable
- De libre uso, con la condición de siempre hacer referencia a sus autores.
- Disponible para diferentes plataformas: Windows, Linux, Sun, Alpha,...

## 2. Iniciando en Scilab

### 2.1. Descripción de la Interfaz Gráfica

Ver figura 2.1:

### 2.2. La ayuda, Como Usarla?

Para utilizar la ayuda simplemente podemos hacer clic en el signo de interrogación que aparece en la barra de menú de la parte de arriba de la interfaz de Scilab y a continuación en **Scilab help** o simplemente podemos escribir la palabra help en la consola. Luego de esto aparece una ventana con el contenido de la ayuda como la de la figura 2.2. Aquí aparece un listado de los comandos de cada uno de los paquetes instalados con Scilab, cada uno con una descripción de las entradas y salidas y ejemplos de como usar cada comando. También si se necesita información de un comando en especial se puede digitar la palabra help seguida del nombre del comando en la ventana principal de Scilab.

## 3. Lenguaje de Programación

### 3.1. Comandos Básicos

- Comentarios:

```
-->//Tradicionalmente: este es un comentario.
```

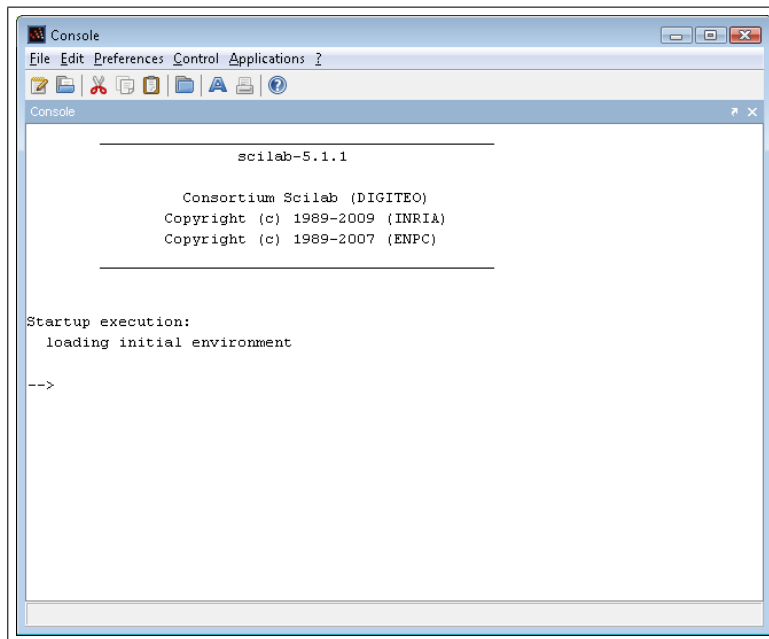


Figura 1: Interfaz Gráfica

■ Constantes:

```
-->//Var tendra valor de 5
-->Var=5
Var=
5.
```

Scilab discrimina MAYUSCULAS y minúsculas.

```
-->var
!--error 4
Undefined variable: var
-->var=3
var=
3.
```

■ Operaciones Básicas:

```
-->Var+var;
-->Var-var;
-->Var*var;
-->Var/var;
-->Var^var;//0 Var**var
-->sqrt(var);
```



Figura 2: Interfaz de ayuda

- La función exponencial  $e^x$  y logaritmo:

```
-->exp(Var);
-->log(ans);// ans toma la respuesta inmediatamente anterior sin asignar.
-->log2(1000);
-->log10(1024)
```

- Eliminación de variables:

```
-->clear Var
-->clear
```

- Funciones Trigonómicas:

```
-->Var=2*%pi;
```

Función	Comando	Ejemplo
seno	<code>sin()</code>	<code>-- &gt; sin(Var)</code>
coseno	<code>cos()</code>	<code>-- &gt; cos(Var)</code>
arcoseno	<code>asin()</code>	<code>-- &gt; asin(Var)</code>
arcocoseno	<code>acos()</code>	<code>-- &gt; acos(Var)</code>
tangente	<code>tan()</code>	<code>-- &gt; tan(Var)</code>
arcotangente	<code>atan()</code>	<code>-- &gt; atan(Var)</code>
cotangente	<code>cotg()</code>	<code>-- &gt; cotg(Var)</code>

- Numeros Complejos:

```
-->%i;
-->1+%i;
```

## 3.2. Vectores y Matrices

### 3.2.1. Definiciones

- Vector Fila:

```
-->Vec1=[2 4 6]
Vec1 =
! 2. 4. 6. !

-->Vec2=[5,%pi,2*%pi];
-->Vec3=1:2:10
Vec3 =
! 1. 3. 5. 7. 9. !
```

- Vector Columna:

```
-->Vec4=[1
-->2
-->3];
-->Vec5=[2;4;6;8;10];
-->Vec6=Vec3';
```

- Matrices:

```
-->B=[1 2 3;4 5 6;7 8 9];
-->C=[Vec1;Vec2];
-->C=[C,[2;3]];
```

### 3.2.2. Operaciones

Operación	Comando
Suma	<code>+</code>
Resta	<code>-</code>
Multiplicación	<code>*</code>
Multiplicación elemento a elemento	<code>.*</code>
Potencia e. a e.	<code>^</code>
Inversa	<code>inv()</code>
Transpuesta Conjugada	<code>'</code>

### 3.3. Estructuras Básicas de Programación

#### 3.3.1. For

La instrucción de ciclo For es una estructura básica de programación para ejecutar acciones repetitivas o ciclos, esta instrucción ejecuta una o varias instrucciones de acuerdo al número de veces que se especifique, en Scilab un ejemplo de su sintaxis es el siguiente:

```
//-----  
for j = 2:n-1  
  a(j,j) = j  
end  
//-----
```

También podemos escribir toda la instrucción en una sola línea:

```
//-----  
for j = 2:n-1, a(j,j) = j; end;  
//-----
```

#### 3.3.2. While

Esta es otra instrucción para ejecutar instrucciones cíclicas, esta estructura ejecuta una o varias instrucciones verificando si se cumple, o no, una condición antes de empezar cada ciclo, un ejemplo de su estructura es:

```
//-----  
while norm(a-(a+e),1) > %eps, e=e/2; k=k+1; end  
//-----
```

#### 3.3.3. If

La estructura condicional If sirve para verificar condiciones de verdad con el objetivo de ejecutar o no una instrucción, su sintaxis en Scilab es la siguiente:

```
//-----  
if i == j then  
  a(i,j) = 2;  
elseif abs(i-j) == 1 then  
  a(i,j) = -1;  
else a(i,j) = 0;  
end  
//-----
```

Cabe anotar que, como en cualquier lenguaje de programación, estas instrucciones se pueden anidar entre sí.

### 3.4. Graficar

#### 3.4.1. plot

```
plot(x,y,<LineSpec>,<GlobalProperty>)  
plot(<axes_handle>,...)
```

### 3.4.2. plot2d

```
plot2d([x],y)
plot2d([x],y,<opt_args>)
```

**Ejemplo:**

```
plot2d(x, [y z sin(2*x)], leg="sen(x)@cos(x)@sen(2x)" )
plot2d( [x x], [sin(x) cos(x)], [-1 -2] )
```

### 3.4.3. plot3d

```
plot3d(x,y,z,<opt_args>)
plot3d(xf,yf,zf,<opt_args>)
```

## 3.5. Construcción de Funciones

### 3.5.1. pwd

Al arrancar, Scilab tiene definido un subdirectorio (o carpeta) preferencial, actual o por defecto. Para saber el nombre de esta carpeta, se da la orden *pwd*

### 3.5.2. chdir

Para decir a Scilab que cambie la carpeta por defecto, se usa la orden *chdir*, por ejemplo:

```
chdir 'c:\algebra\matrices\'
```

### 3.5.3. exec

Este comando se utiliza para cargar una función (script) en el entorno de Scilab, se puede digitar toda la ubicación del archivo *.sce* que se desea cargar, o si se encuentra en la carpeta correcta basta con escribir el nombre. Por ejemplo:

```
exec \WINDOWS\Escritorio\ejemplo4.sce
exec ejemplo4.sce
```

### 3.5.4. SciPad

### 3.5.5. function

```
function [res1, res2, ...] = nombrefuncion(par1, par2, ...)
...
...
endfunction
```

## 4. Ejemplos

1. Graficas en 3D:

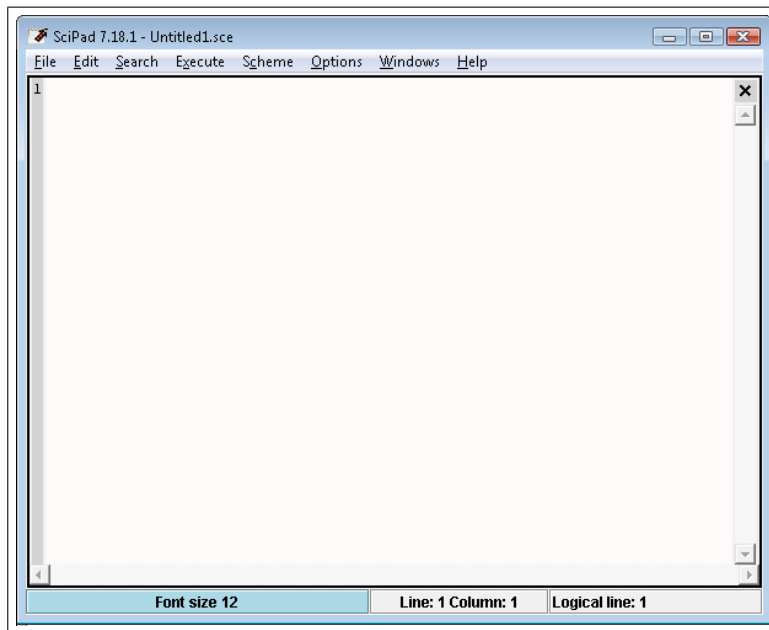


Figura 3: Interfaz SciPad

```
t=[0:0.3:2*pi]';
z=sin(t)*cos(t');
[xx,yy,zz]=genfac3d(t,t,z);
clf()
plot3d(xx,yy,zz)
```

2.

3. Matrices y variables booleanas:

```
-->a=[%f %f;%f %t];
-->a
a =
! F F !
! F T !
-->b=~a
b =
! T T !
! T F !
-->h2=[a a;a b]
h2 =
! F F F F !
! F T F T !
! F F T T !
! F T T F !
```

4. Ejemplo de Funciones:



```

//-----
function [x, y] = polarCart(r, t)
// Conversion de coordenadas polares a cartesianas.
x = r*cos(t)
y = r*sin(t)
endfunction
//-----
function [x, y] = polarCartGr(r, t)
// Conversion de coordenadas polares a cartesianas,
// el angulo esta dado en grados.
[x, y] = polarCart(r, t*pi/180)
endfunction
//-----
function fx = f(x)
fx = x(1)^2 + x(2)^2
endfunction

```

## 5. Ejercicios

1. Averiguar la en que consisten los siguiente comandos y realizar una descripción detallada.
  - a) *convol*
  - b) *dft*
  - c) *fft*
2. Hacer una función que permita graficar la identidad de Euler en un intervalo deseado.
3. Hacer una función que descomponga una señal  $x[n]$  en la suma de una señal par  $x_p[n]$  y una señal impar  $x_i[n]$  y que grafique las 3 señales.
4. Implementar por medio de una función la regla de Kramer para la solución de sistemas de ecuaciones lineales de  $3 \times 3$   
**Nota:** La regla de Kramer es un teorema en álgebra lineal, que da la solución de un sistema lineal de ecuaciones en términos de determinantes.