

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

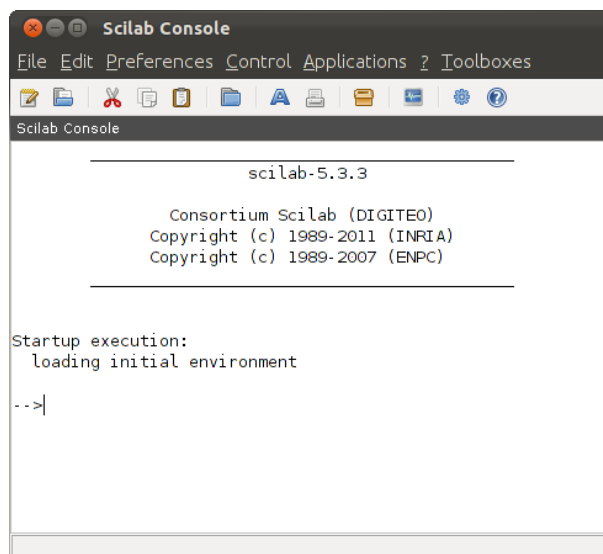
Clase Introducción a la Programación con Scilab (Parte 4)

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.

Raíces y Optimización

Scilab permite por medio de la función **roots** encontrar las raíces de polinomios. Scilab transforma el problema de encontrar la raíces de un polinomio en un problema de eigenvalores. La forma de emplear el comando roots es: $x = \text{roots}(c)$, Donde c es un vector con los coeficientes del polinomio.

Por ejemplo para calcular las raíces del polinomio $5x^4 + 4x^3 + 2x^2 + 8x + 10 = 0$ el comando roots se debe ejecutar de la siguiente manera:

```
x = roots([5, 4, 2, 8, 10])
```

Scilab permite por medio de la función **fminsearch** encontrar el mínimo de una función multidimensional. La solución se basa en el método de Nelder-Mead, el cual es un método directo. La forma de emplear el comando es: $[x, fval] = \text{fminsearch}(\text{function}, x0)$. Donde $xmin$ y $fval$ es la posición y el valor del mínimo, function es el nombre de la función a evaluar, $x0$ puede ser un escalar, un vector o una matriz

Digite las siguientes instrucciones en un script de nombre **mifuncion1** y almacenelo en la carpeta **tallerscilab**. Ejecute el script.

```
function y = mifuncion1(x)
y = 2 + x(1) - x(2) + 2*x(1)^2 + 2*x(1)*x(2) + x(2)^2
endfunction
```

En una consola de Scilab digite el siguiente comando

```
[x, fval] = fminsearch(mifuncion1, [-0.5, 0.5])
```

Compruebe gráficamente el valor para el mínimo devuelto por la función.

En problemas de optimización multidimensional es posible aplicar métodos gráficos para encontrar el valor óptimo. Scilab permite la visualización de funciones como gráficas 3D y contornos 2D.

Digite las siguientes instrucciones en un script de nombre **optimizacion1** y almacenelo en la carpeta **tallerscilab**. Ejecute el script.

```
function z=f(x)
    z = 2+x(1) -x(2) +2*x(1)^2+2*x(1) .* x(2) +x(2)^2;
endfunction

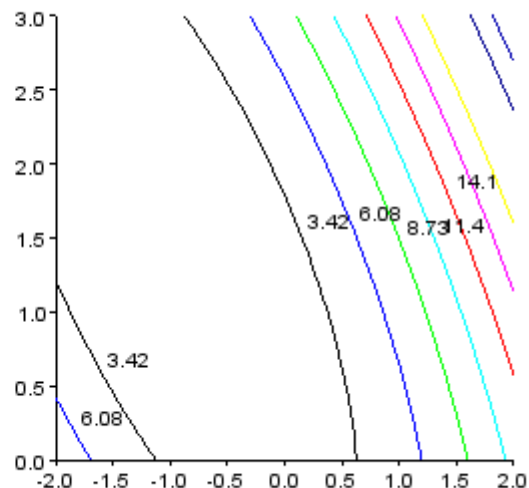
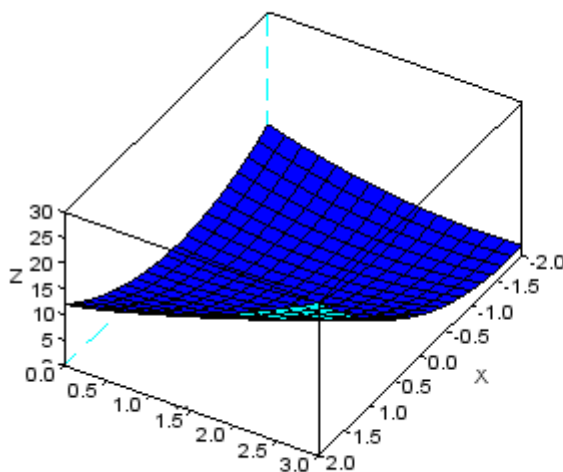
opt=optimset("TolX",1.e-2); // determina un error aproximado para la respuesta menor a 1.e-2
[x,fval,exitflag,output]=fminsearch(f,[-1.2 1]);
disp(x,"The minimum=")
```

```

disp(fval,"The minimum function value=")
deff('z=f(x,y)','z= 2 + x - y + 2*x^2 + 2*x.*y + y^2')
x= [-2:0.2:2];
y= [0:0.2:3];
z=feval(x,y,f);

subplot(1,2,1)
fplot3d(x,y,f,alpha=5,theta=31) // plot3d(x,y,z)
subplot(1,2,2)
contour2d(x,y,z,10);

```



Problemas

1. Resuelva manualmente empleando los métodos de raíces y optimización vistos en clase un ejercicio del anexo problemaCap5.pdf, uno del anexo problemaCap6.pdf y uno del anexo problemaCap7.pdf
2. Implementando en Scilab los métodos de raíces y optimización vistos en clase ó empleando los scripts del archivo **anexo.rar**, seleccione y resuelva un ejercicio del anexo problemaCap5.pdf, uno del anexo problemaCap6.pdf y uno del anexo problemaCap7.pdf
3. Modifique los scripts del archivo **anexo.rar** para exportar la información de cada iteración en formato de tabla de LaTeX.
4. Realice la implementación en Scilab del método de Brent, como se explica en la sección de Métodos Abiertos (Open Methods) del libro Applied Numerical Methods with MATLAB for Engineers and Scientists del autor Steven C. Chapra.