Lab evaluation

Name : Nunemunthala Shiva Roll No.: 190001041

Q1 ) Q1.SCE

*// given function*

function **f**=funcval(**x**)

**f** = **x**(1)^2 + 4\***x**(1)+2\*(**x**(2)^2) + 4\***x**(2);

endfunction

*// calculating gradient of given function*

function **g**=gradient(**x**)

**g**(1)=2\***x**(1)+4;

**g**(2)=4\***x**(2)+4;

endfunction

*//one dimensional function value (for dichotomous)*

function **f**=func(**a**, **b**, **c**, **d**, **x**)

**f** = (**a**+**b**\***x**)^2 + 4\*(**a**+**b**\***x**) + 2\*(**c**+**d**\***x**)^2 + 4\*(**c**+**d**\***x**);

endfunction

*//dichotomous function for one dimensional minimization*

function **X**=dichotomous(**a**, **b**, **c**, **d**, **xl0**, **xu0**)

k = 0; xl = **xl0**; xu = **xu0**;

interval = xu - xl;uncertainty = 10^-5;

while interval > uncertainty,

x1 = (xl + xu)/2;

e = interval\*0.001;

xa = x1 - e;

xb = x1 + e;

fa = func(**a**,**b**,**c**,**d**,xa);

fb = func(**a**,**b**,**c**,**d**,xb);

if fa < fb,

xu = xb;

elseif fa > fb,

xl = xa;

else

xl = xa;

xu = xb;

end

interval = xu - xl;

k = k + 1;

end

**X** = (xl + xu)/2

F = func(**a**,**b**,**c**,**d**,**X**)

endfunction

*//main program*

iter=5;

x0=[0 0];

for i=1 : iter

disp('Iteration Number:')

disp(i)

g= -1\*gradient(x0);

a=x0(1);

b=g(1);

c=x0(2);

d=g(2);

alpha = dichotomous(a,b,c,d,-10,10);

x0=x0+alpha\*g';

disp('Value of x')

disp(x0)

f=funcval(x0);

disp('Functioin value at x')

disp(f)

end

output:

exec('/Users/nunemunthalashiva/Documents/CS357\_IITI/lab6/lab\_eval/Q1.sce', -1)

"Iteration Number:"

1.

"Value of x"

-1.3333413 -1.3333413

"Functioin value at x"

-5.3333333

"Iteration Number:"

2.

"Value of x"

-1.77777 -0.8888966

"Functioin value at x"

-5.9259259

"Iteration Number:"

3.

"Value of x"

-1.9259286 -1.0370397

"Functioin value at x"

-5.9917695

"Iteration Number:"

4.

"Value of x"

-1.9753084 -0.9876546

"Functioin value at x"

-5.9990855

"Iteration Number:"

5.

"Value of x"

-1.9917696 -1.0041152

"Functioin value at x"

-5.9998984

Q2)

Code:

*// for sake of simplicity lets assume x=[p;r]*

function newton(**x**)

*// caluclating function value at given point*

function **z**=f(**x1**, **x2**)

**z**= **x**(1)^4 + 2\*(**x**(1)^2)\*(**x**(2)^2) + **x**(2)^4;

endfunction

*// returning gradient at given point*

function **z**=grad(**x**)

fx1 = 4\*(**x**(1)^2 + **x**(2)^2)\***x**(1)

fx2 = 4\*(**x**(1)^2 + **x**(2)^2)\***x**(2)

**z** = [fx1 ;fx2]

endfunction

*// return hessian inverse*

function **z**=hessian(**x**)

**z** = [8\*(**x**(1)^2)+**x**(2)^2 ,8\***x**(1)\***x**(2);8\***x**(1)\***x**(2) ,8\*(**x**(2)^2)+**x**(1)^2]

endfunction

num\_iter=0

while(grad(x)(1)~=0 && grad(x)(2)~=0)

num\_iter=num\_iter+1

x = x - inv(hessian(x)) \* grad(x)

end

printf("Num iterations = " + string(num\_iter) +" x= "+ string(x(1)) +" "+string(x(2)) + "function value=" + string(f(x(1),x(2))) )

endfunction

Q3)

Code:

x = input("Enter starting num ");

tolerance = 10^(-4);

num\_iter=0;

dx=1; *// some random value*

f = x^3 - 3\*x\*x + 2\*x

while((dx> tolerance | abs(f) >10^(-5)) & num\_iter<10)

num\_iter=num\_iter+1

derivative\_func = 3\*(x^2) +6\*x +2

xnew=x-(f/derivative\_func)

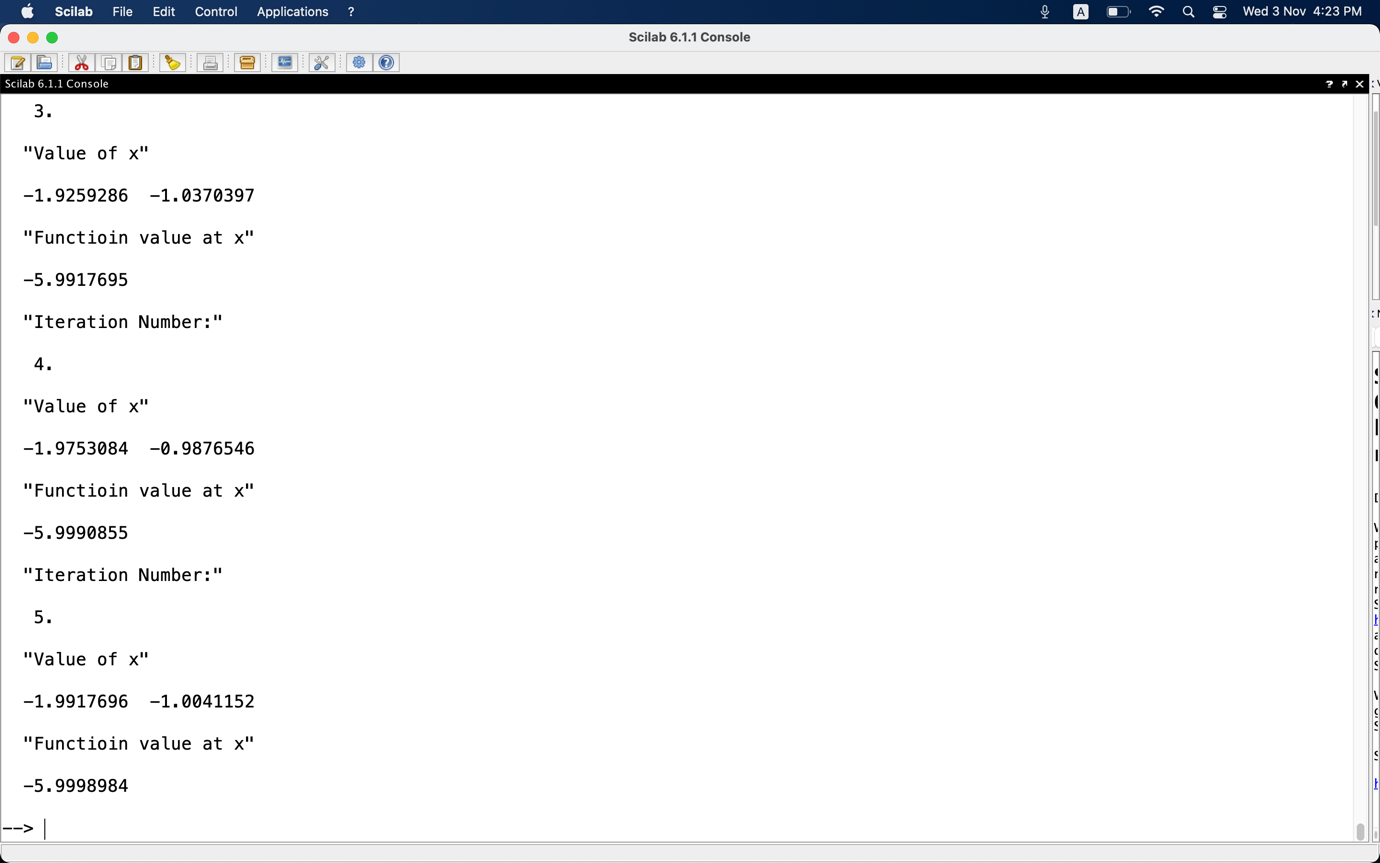
dx=abs(x-xnew)

x=xnew

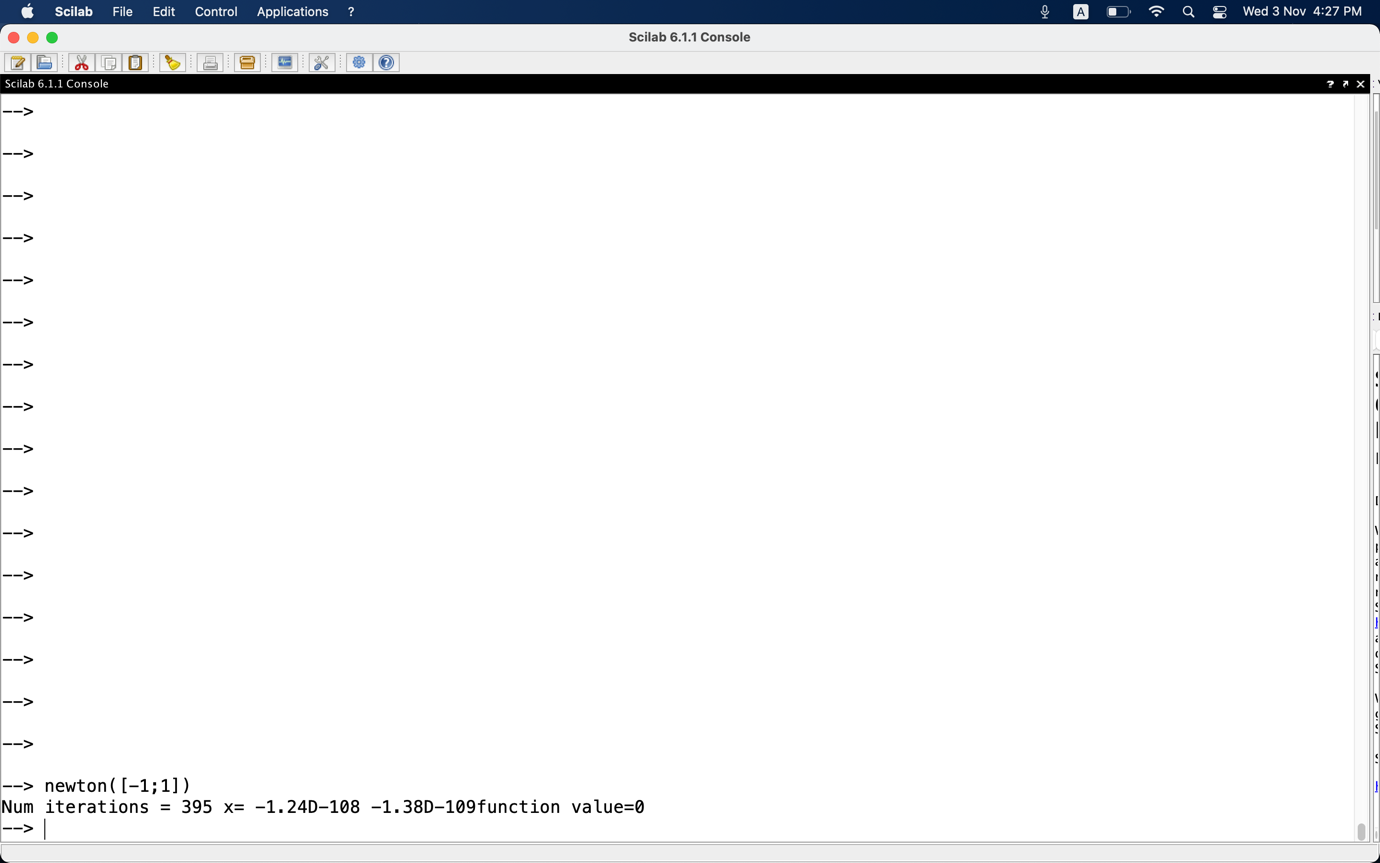
f = x^3 - 3\*x\*x + 2\*x

printf('%3i %12.4f %12.9f\n',num\_iter,x,f)

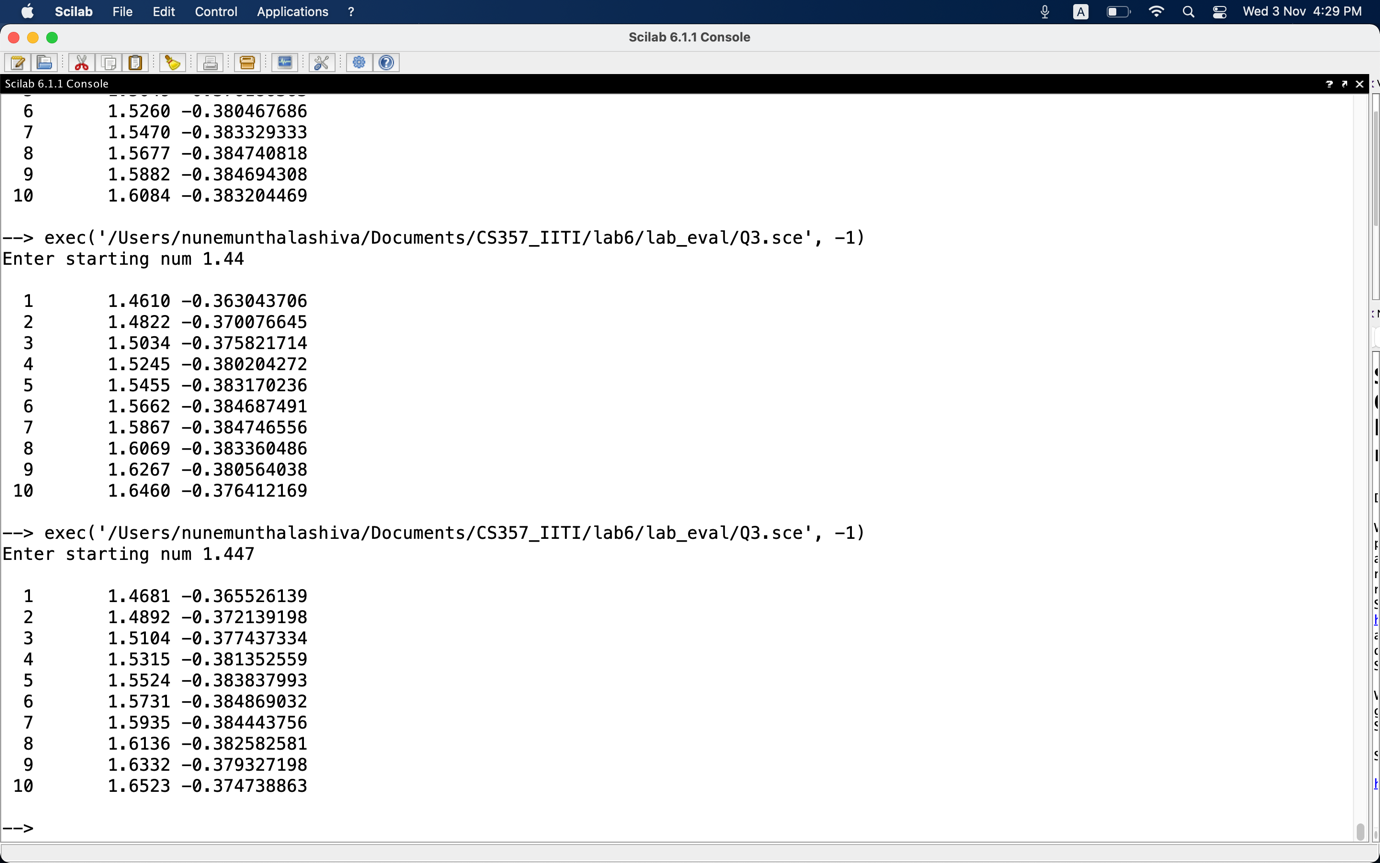
end



Q1 screenshot



Q2 screenshot



Q3 screenshot