EGR 103L – Fall 2016

Laboratory 6 - Roots and Extrema Problems

CEMAL YAGCIOGLU (cy111)
Lab Section 5D, Wednesday 11.45AM - 2.35PM
23 OCTOBER, 2016

I understand and have adhered to all the tenets of the Duke Community Standard in completing every part of this assignment. I understand that a violation of any part of the Standard on any part of this assignment can result in failure of this assignment, failure of this course, and/or suspension from Duke University.

Contents

List of Figures

1 Basic Root-Finding Problems

Function	Real Roots	Roots
$f(x) = 20e^{-4x} - 36e^{-2x} + 18e^{-x} - 1$	3	4.5651e-02, 6.3358e-01, 2,7545e+00
$f(x) = x^5 + 100\cos(2x)$	3	-7.8391e-01, 7.8691e-01, 2.1295e+00
$f(x) = \frac{10}{x-2} - 90e^{-(x/20)}, x \neq 2$	2	2.3620e+00, 7.9669e+00

2 Basic Min/Max-Finding Problems

Function	Counts	Extrema
$f(x) = 20e^{-4x} - 36e^{-2x} + 18e^{-x} - 1$	1 min, 1 max	min:$f(2.4511e-01) = -1.4596e+00$ max:$f(1.3002e+00) = 1.3421e+00$
$f(x) = x^5 + 100\cos(2x)$	2 min, 2 max	min 1:$f(-1.6682e+00) = -1.1103e+02$ min 2:$f(1.5063e+00) = -9.1415e+01$ max 1:$f(-2.4916e+00) = -6.9275e+01$ max 2:$f(4.9908e-07) = 1.0000e+02$

3 Chapra 6.16

V.m^3	10	20	30	40	50	60
h,m	8.6492e-01	1.4210e+00	1.9292e+00	2.4326e+00	2.9685e+00	3.6373e+00

4 Chapra 6.21

Angles are 37.959 and 51.532 degrees.

5 Chapra 7.23, 7.24, and 7.25(b/c)

For function 1, minimum value occurs at (x,y)=(5.6759e-01,7.5677e-01). The function”s value at this point is -6.6216e-01. For function 2, max value occurs at (x,y)=(9.6759e-01,6.5589e-01). The function”s value at this point is 4.3440e+00. For function 3, minimum value occurs at (x,y)=(3.3333e+00,-6.6670e-01). The functions value at this point is -1.7333e+01.

A Codes

A.1 Roots.m

```
%[Roots.m]
%[Cemal Yagcioglu]
%[October 23,2016]
% I have adhered to all the tenets of the
% Duke Community Standard in creating this code.
% Signed: [cy111]

clear

for EqNo=1:3 %I created
    for k=1:2%k==1 is for trial to check the limits, k==2 are the real graphs

    % Equation written with mask to not write more than once.
FirstF = @(x) ((EqNo==1).*(20.*exp(-4*x)-36.*exp(-2.*x)+18.*exp(-x)-1))...
    +((EqNo==2).*((x.^5)+100.*cos(2.*x)))...
    +((EqNo==3).*((10./(x-2))-90.*exp(-x./2)));


X2 = linspace(-100,100,1000);
Y1 = FirstF(X2);
Y2 = sign(Y1);
Rcounter = 0;
%This for loop finds all the guesses.
for i=1:999
    if Y2(i+1)~=Y2(i) %This check if sign changes if it does
                      %it writes down the x value as rootguess
        Rcounter = Rcounter +1;
        rootguess(Rcounter)=(X2(i+1)+X2(i))./2;
    end
end
%uses all the root guesses to find the roots
for rt=1:length(rootguess)
   [Fxroot(rt), Fyroot(rt)] = fzero(FirstF, rootguess(rt));

end


Allxroots(EqNo,:) = Fxroot(:);
Allyroots(EqNo,:) = Fyroot(:);




xlowlim=(min(Fxroot(:))-8); % limits for k==1 trial graphs
xuplim=(max(Fxroot(:))+8);
XFinal = linspace(xlowlim,xuplim,1000);

YFinal = FirstF(XFinal);
YSignFinal = sign(YFinal);

figure(EqNo+3*(k==1)) %figures to check boundaries are figure 4-5-6

```

```
subplot(2,1,1)
plot(XFinal,YFinal)
xlabel('x')
ylabel('f(x)')

ylowlim=min(FirstF(linspace(xlowlim,xuplim,1000)))-10;
yuplim=max(FirstF(linspace(xlowlim,xuplim,1000)))+10;
axis([xlowlim,xuplim,ylowlim,yuplim]) %% this is to adjust the limits of the trial function
% Y4 function is for finding maximum by using minumum calculator
% we take the absolute value and multiply by -1 to make sure
% maxes are now mins (this is true if local max is positive)
Y4 = @(x) -1.*(abs(((EqNo==1).*(20.*exp(-4*x)-36.*exp(-2.*x)+18.*exp(-x)-1))...
    +((EqNo==2).*((x.^5)+100.*cos(2.*x)))...
    +((EqNo==3).*((10./(x-2))-90.*exp(-x./2)))));
%this is true if local max is negative
Y5 = @(x) (abs(((EqNo==1).*(20.*exp(-4*x)-36.*exp(-2.*x)+18.*exp(-x)-1))...
    +((EqNo==2).*((x.^5)+100.*cos(2.*x)))...
    +((EqNo==3).*((10./(x-2))-90.*exp(-x./2)))));
% the following if function corrects the limits for real graphs
if EqNo==1 & k==2
    title('Plot of f(x)=20e^{-4x}-36e^{-2x}+18e^{-x}-1(cy111)')
    axis([(min(Fxroot(:))-1),(max(Fxroot(:))+1),-5,5])
    [Eq1xmin(1),Eq1ymin(1)] = fminbnd(FirstF,0,0.5);
    [Eq1NabsXmax(1),Eq1NabsYmax(1)] = fminbnd(Y4,0.6,1.5);
    Eq1xmax(1)=Eq1NabsXmax(1);
    Eq1ymax(1)=(-1).*Eq1NabsYmax(1);

elseif EqNo==2 & k==2
    title('Plot of f(x)=x^{5}+100cos(2x)(cy111)')
    axis([(min(Fxroot(:))-3),(max(Fxroot(:))+1),-200,200])
    [Eq2xmin(1),Eq2ymin(1)] = fminbnd(FirstF,-2,1);
    [Eq2xmin(2),Eq2ymin(2)] = fminbnd(FirstF,1,2);

    [Eq2NabsXmax(1),Eq2NabsYmax(1)] = fminbnd(Y5,-3,-2);
    Eq2xmax(1)=Eq2NabsXmax(1);
    Eq2ymax(1)=-1.*Eq2NabsYmax(1);

    [Eq2NabsXmax(2),Eq2NabsYmax(2)] = fminbnd(Y4,-0.5,0.5);
    Eq2xmax(2)=Eq2NabsXmax(2);
    Eq2ymax(2)=(-1).*Eq2NabsYmax(2);

elseif EqNo==3 & k==2
    title('Plot of f(x)=10/{x-2}-90e^{-(x/2)},x\neq2(cy111)')
    axis([(min(Fxroot(:))-2.5),(max(Fxroot(:))+1),-200,200])
end

subplot(2,1,2)
plot(XFinal,YSignFinal)
xlabel('x')
ylabel('sign(f(x))')
title('Plot of sign(f(x))')
% the following if function corrects the limits for real graphs
if EqNo==1 & k==2
axis([(min(Fxroot(:))-1),(max(Fxroot(:))+1),-1.2,1.2])
    print -depsc Roots1
```

```
elseif EqNo==2 & k==2
    axis([(min(Fxroot(:))-3),(max(Fxroot(:))+1),-1.2,1.2])
    print -depsc Roots2
elseif EqNo==3 & k==2
    axis([(min(Fxroot(:))-2.5),(max(Fxroot(:))+1),-1.2,1.2])
    print -depsc Roots3
end
if k==2
format short e
fprintf('X roots for Equation%0.0f are: %e %e %e\n',EqNo,Fxroot)
end
    end
end
format short e
fprintf('Max x value for Equation 1 is:%e\n',Eq1xmax)
fprintf('Max y value for Equation 1 is:%e\n',Eq1ymax)
fprintf('Min x value for Equation 1 is:%e\n',Eq1xmin)
fprintf('Min y value for Equation 1 is:%e\n',Eq1ymin)
fprintf('Max x values for Equation 2 are:%e %e\n',Eq2xmax)
fprintf('Max y value for Equation 2 are:%e %e\n',Eq2ymax)
fprintf('Min x values for Equation 2 are:%e %e\n',Eq2xmin)
fprintf('Min y value for Equation 2 are:%e %e\n',Eq2ymin)
```

A.2 HeightCylinder.m

```
%[Roots.m]
%[Cemal Yagcioglu]
%[October 23,2016]
% I have adhered to all the tenets of the
% Duke Community Standard in creating this code.
% Signed: [cy111]

clear

r=2 %m
L=5 %m
V = @(h) (2.^2*acos((2-h)./2)-(2-h).*sqrt(2.*2.*h-h.^2)).*5

h=linspace(0,4,100)



for i=1:6
    fun = @(h) (V(h)-10.*i);
    Y2(i)=fzero(fun,2)
end

%fun = @(h) (V(h,r,L));
%fplot(fun)
```

A.3 Projectile.m

```
%[Roots.m]
%[Cemal Yagcioglu]
%[October 23,2016]
% I have adhered to all the tenets of the
% Duke Community Standard in creating this code.
% Signed: [cy111]

clear

% t used to denote theta
% g=9.81
y = @(x,v0,t) ((tand(t).*x)-(9.81.*(x.^(2))./(2.*(v0.^(2))...
.*((cosd(t)).^2)))+0.8);
x=90;
v0=30;
ProjectileCalc = @(t) ((tand(t).*x)-(9.81.*(x.^(2))./(2.*(v0.^(2))...
.*((cosd(t)).^2)))+0.8);

fplot(ProjectileCalc) %to see where the zeros are aproximately
axis([0,90,-7,7])

[t1,y1] = fzero(ProjectileCalc,37)
[t2,y2] = fzero(ProjectileCalc,52)

clear x

t=[t1,t2]

ProjectileCalc2 = @(x) ((tand(t).*x)-(9.81.*(x.^(2))./(2.*(v0.^(2))...
.*((cosd(t)).^2)))+1.8);
fplot(ProjectileCalc2)
axis([-5,95,0,35])
xlabel('X distance(m)')
ylabel('Height from the floor(m)')
legend('theta=37.959','theta=51.532')
title('Projectile Motion(cy111)')
print -depsc ProjectileGraph
```

A.4 ExtremeValue.m

```
%[Roots.m]
%[Cemal Yagcioglu]
%[October 23,2016]
% I have adhered to all the tenets of the
% Duke Community Standard in creating this code.
% Signed: [cy111]

clear

y=linspace(-4,4,20);
x=linspace(-4,4,20);
[X,Y] = meshgrid(x,y);
```

```
fun = (Y.^2).*2-2.25.*X.*Y-1.75.*Y+1.5.*X.^2;
figure(1)
clf
surfc(X,Y,fun)
colormap jet
grid on
xlabel(’X’)
ylabel(’Y’)
title(’Surface Graph for Equation in Problem 7.23(cy111)’)
print -depsc Surface1

fun2 = (4.*X)+(2.*Y)+(X.^2)-(2.*(X.^4))+(2.*X.*Y)-(3.*Y.^2);
figure(2)
clf
surfc(X,Y,fun2)
colormap jet
grid on
xlabel(’X’)
ylabel(’Y’)
title(’Surface Graph for Equation in Problem 7.24(cy111)’)
print -depsc Surface2

fun3 = (-8.*X)+(X.^2)+(12.*Y)+(4.*(Y.^2))-(2.*X.*Y);
figure(3)
clf
colormap jet
surfc(X,Y,fun3)
grid on
xlabel(’X’)
ylabel(’Y’)
title(’Surface Graph for Equation in Problem 7.25(cy111)’)
print -depsc Surface3

fu1 = @(X,Y) (Y.^2).*2-2.25.*X.*Y-1.75.*Y+1.5.*X.^2;
[ MinVec1 , MinVal1] = fminsearch (@( VecD ) fu1( VecD (1) , VecD (2)) , [5 5])

fu2 = @(X,Y) (4.*X)+(2.*Y)+(X.^2)-(2.*(X.^4))+(2.*X.*Y)-(3.*Y.^2);
[ MinVec2 , MinVal2] = fminsearch (@( VecD ) fu2( VecD (1) , VecD (2)).*(-1) , [-2 2])
MinVal2final = abs(MinVal2)

fu3 = @(X,Y) (-8.*X)+(X.^2)+(12.*Y)+(4.*(Y.^2))-(2.*X.*Y);
[ MinVe3 , MinVal3] = fminsearch (@( VecD ) fu3( VecD (1) , VecD (2)) , [1 1])
```

B Figures

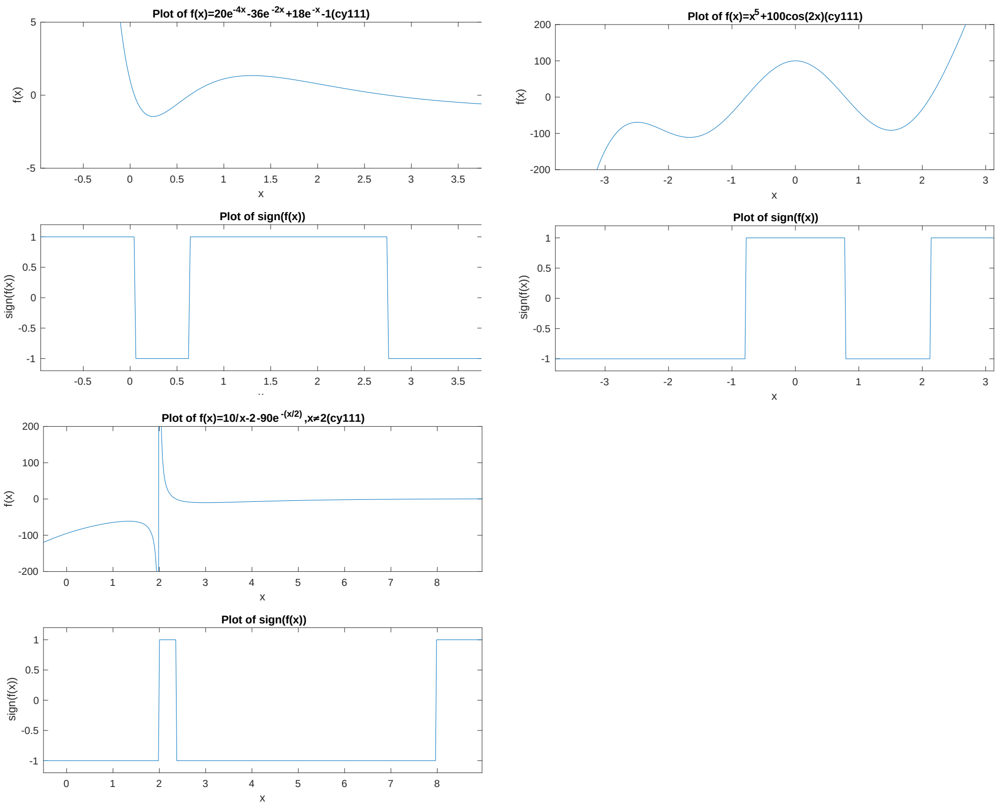

Figure 1: Basic Roots Problems

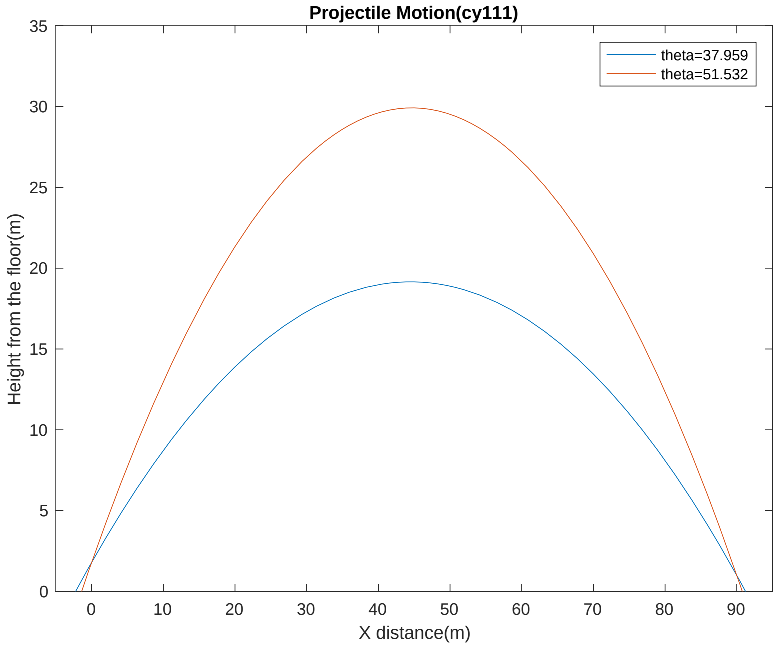

Figure 2: Projectile Motion Problem

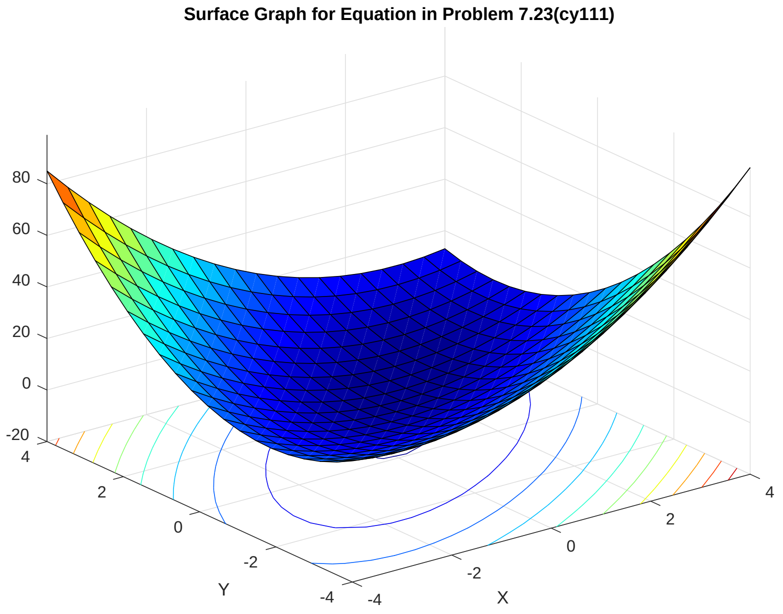

Figure 3: Max/Min Surface Graph 1

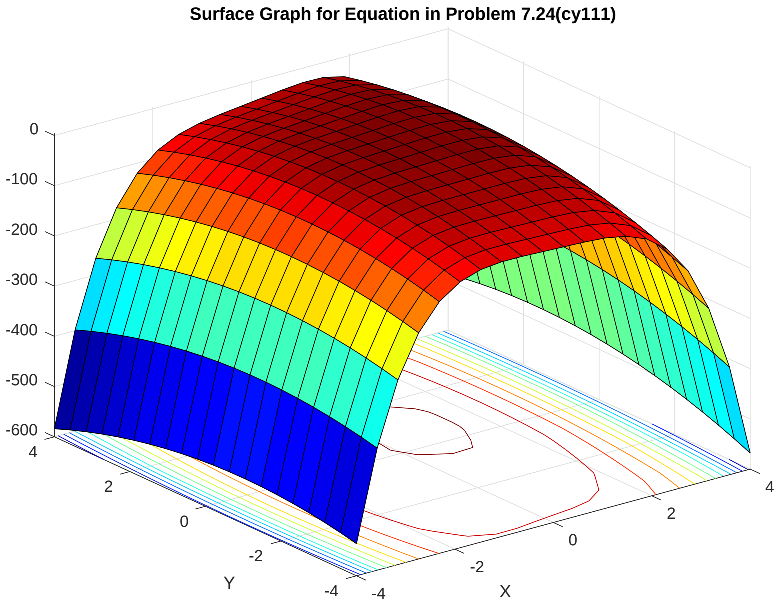

Figure 4: Max/Min Surface Graph 2

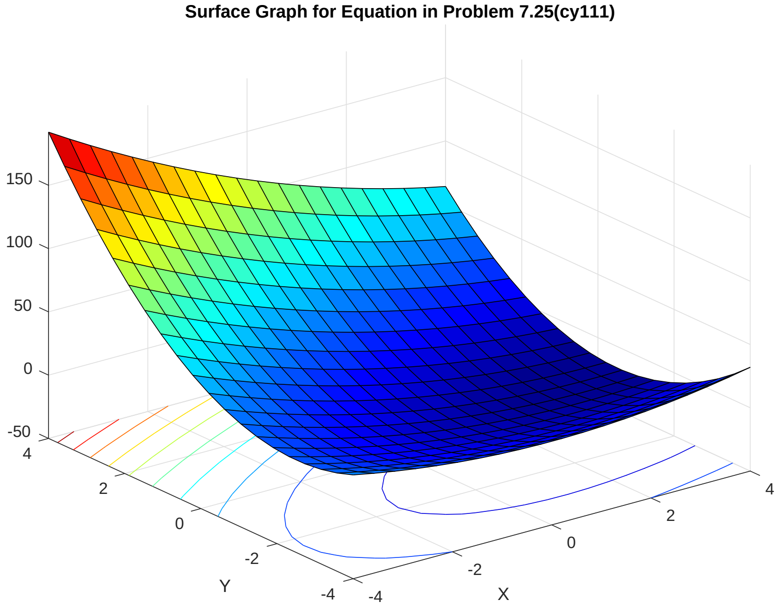

Figure 5: Max/Min Surface Graph 3