|  |
| --- |
| By Cliff Rodriguez |
| Optimization |
| Date: August 15, 2018 |

|  |
| --- |
| Cliff Rodriguez  [Date] |

Contents

# Question 1

MatLab

syms x; % declare choice variables

% Problem 1a

a = -(x +13)^4; % Set function

rootsa = solve(a); % find roots of function

a1 = diff(a); % Take derivative of function

disp('first derivative');

pretty(a1); % display derivative in readable form

crit\_ptsa = solve(a1); % solve for critical points

disp('critical points');

disp(crit\_ptsa);

% Plot line Template

fplot(a,[-20, 20], 'b'); % plot line

hold on

% standard lines for each image

legend('show','Location','best'); % add legend to graph

% comment out to remove critical points

plot(double(crit\_ptsa), double(subs(a,crit\_ptsa)),'ro'); % plot critical points

title('Optimization Graph');

% customize for each graph

text(-12,100, 'critical point (-13,0)');

saveas(gcf,'Figure 1a.png');

hold off

Console Output

>> HW3\_1a

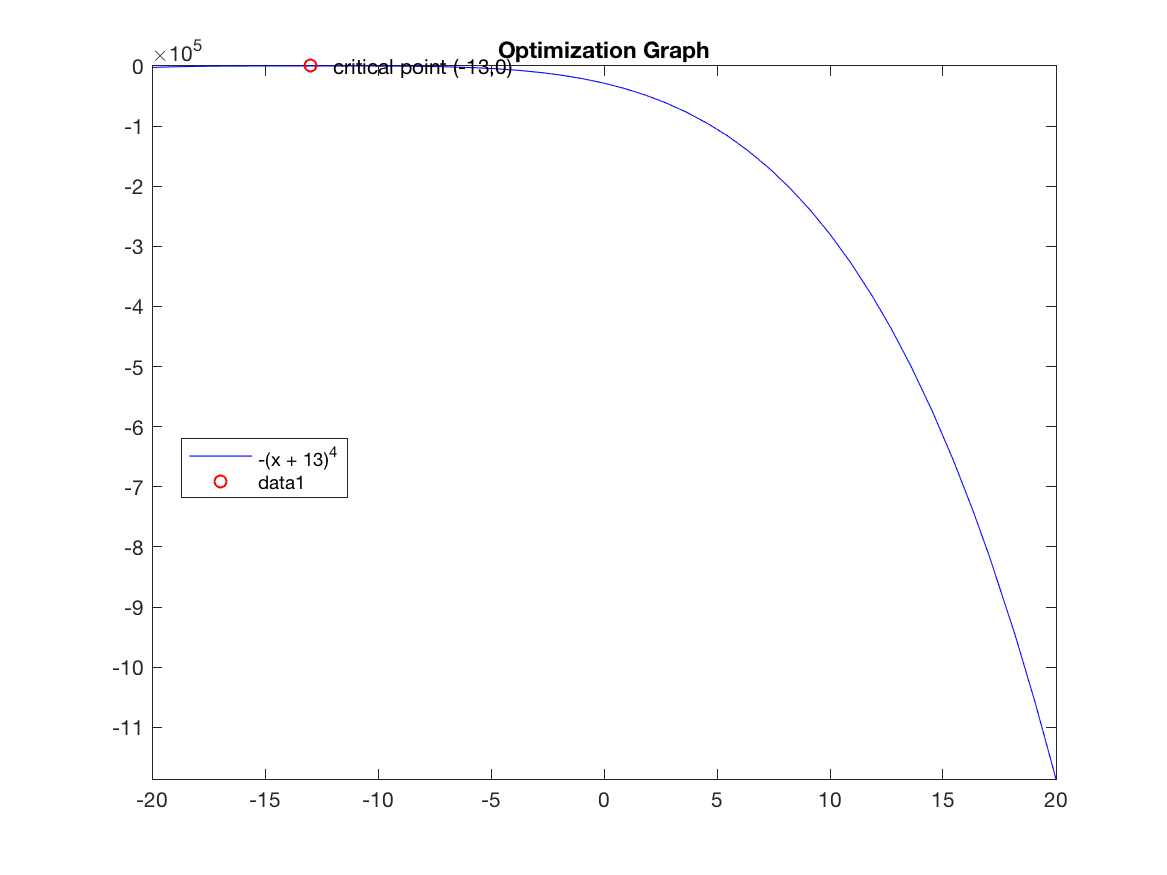
first derivative

-4 (x + 13)^3

critical points

-13

Graph



MatLab

% Problem 1b

clear;

syms x;

b = (-3\*x^4 - 20\*x^3 + 144\*x^2 +17); % Set function

rootsb = solve(b); % find roots of function

b1 = diff(b); % Take derivative of function

pretty(b1); % display derivative in readable form

crit\_ptsb = solve(b1); % solve for critical points

% disp(rootsb);

disp('roots b');

pretty(crit\_ptsb);

% Plot line Template

fplot(b, 'b'); % plot line

hold on

fplot(b1); % plot derivative

% comment out to remove critical points

plot(double(crit\_ptsb), double(subs(b,crit\_ptsb)),'ro'); % plot critical points

text(17,0, 'critical point (17,0)');

text(530,3, 'critical point (530,3)');

text(-11247,-8, 'critical point (-11247,-8)');

% add more text lines if more critical points exist

% standard lines for each image

% legend('show','Location','best'); % add legend to graph

% customize for each graph

title('Optimization Graph');

saveas(gcf,'Figure 1b.png');

console output

>> HW3\_1b

3 2

- 12 x - 60 x + 288 x

roots b

/ -8 \

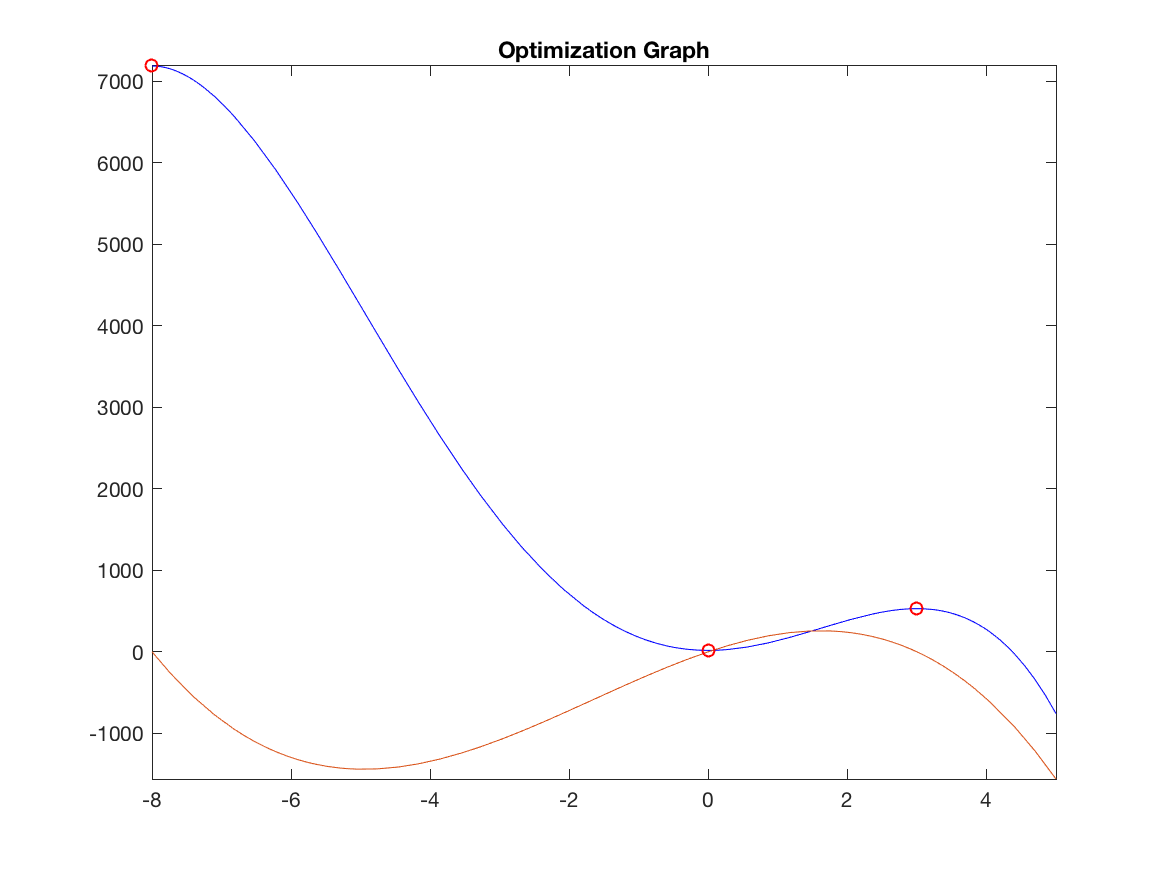
| |

| 0 |

| |

\ 3 /

Graph



MatLab

% Problem 1c

clear;

syms x;

c = 4\*x\*exp(3\*x); % Set function

rootsc = solve(c); % find roots of function

c1 = diff(c); % Take derivative of function

c2 = diff(c1);% Take second derivative

disp('first derivative');

pretty(c1); % display derivative in readable form

disp('second derivative');

pretty(c2); % display second derivative in readable form

crit\_ptsc = solve(c1); % solve for critical points

% disp(rootsb);

disp('roots c');

pretty(crit\_ptsc);

% Plot line Template

fplot(c, 'r'); % plot line

hold on;

% comment out to remove critical points

plot(double(crit\_ptsc), double(subs(c,crit\_ptsc)),'bo'); % plot critical points

text(.5,0, 'critical point (-1/3,0)');

% add more text lines if more critical points exist

% standard lines for each image

legend('show','Location','best'); % add legend to graph

% customize for each graph

saveas(gcf,'Figure 1c.png');

title('Optimization Graph');

hold off;

Console Output

>> HW3\_1c

first derivative

exp(3 x) 4 + x exp(3 x) 12

second derivative

exp(3 x) 24 + x exp(3 x) 36

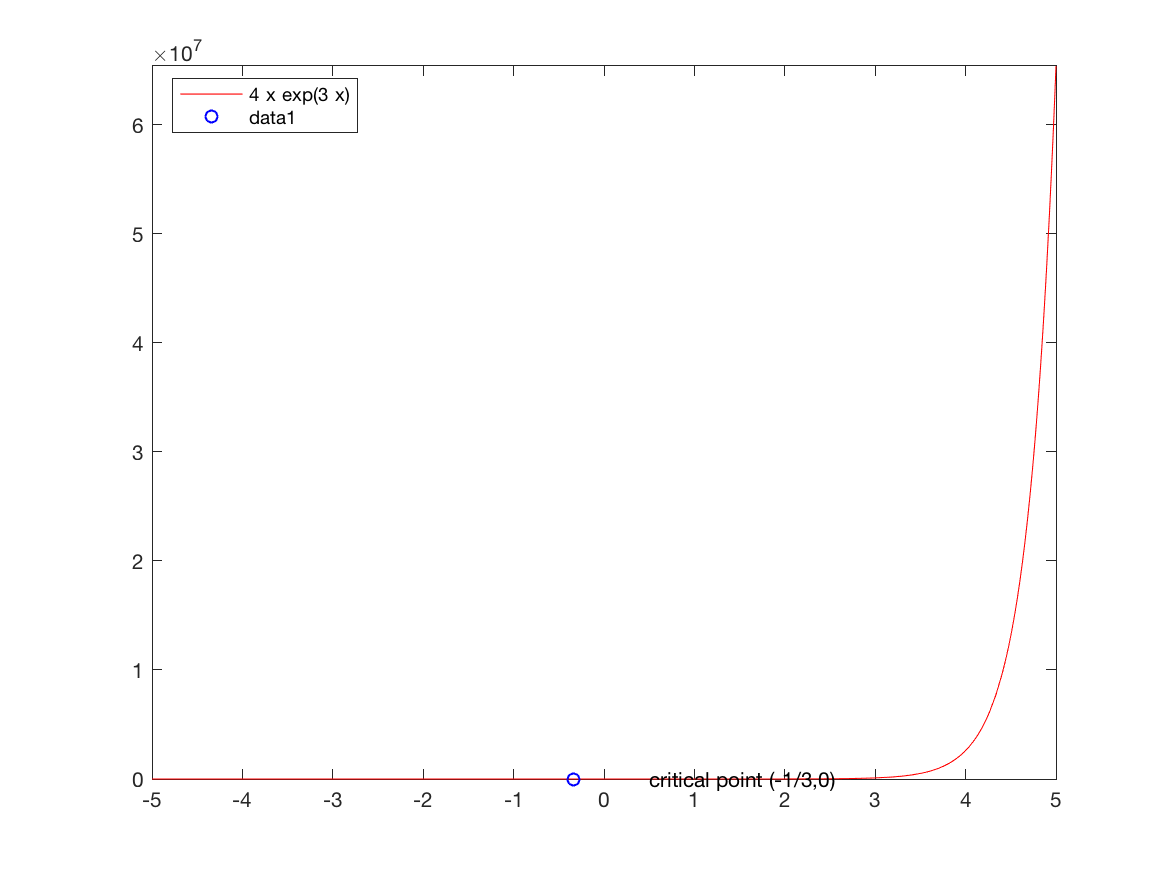
roots c

1

- -

3

Graph



MatLab

% Problem 1d

clear;

syms x;

f = log (2\*x^2 - 20\*x + 5); % Set function

roots = solve(f); % find roots of function

f1 = diff(f); % Take derivative of function

f2 = diff(f1);% Take second derivative

disp('function');

disp(f);

disp('first derivative');

pretty(f1); % display derivative in readable form

disp('second derivative');

pretty(f2); % display second derivative in readable form

crit\_pts = solve(f1); % solve for critical points

% disp(rootsb);

disp('roots');

pretty(crit\_pts);

% Plot line Template

fplot(f,[-100, 100], 'r'); % plot line

hold on;

fplot(f1, 'g');

% comment out to remove critical points

plot(double(crit\_pts), double(subs(f,crit\_pts)),'bo'); % plot critical points

text(0,5, 'critical point (0,5)');

% add more text lines if more critical points exist

% standard lines for each image

legend('show','Location','best'); % add legend to graph

% customize for each graph

saveas(gcf,'Figure 1d.png');

title('Optimization Graph');

hold off;

Console Output

>> HW3\_1d

function

log(2\*x^2 - 20\*x + 5)

first derivative

4 x - 20

---------------

2

2 x - 20 x + 5

second derivative

2

4 (4 x - 20)

--------------- - ------------------

2 2 2

2 x - 20 x + 5 (2 x - 20 x + 5)

roots

5

Warning: Imaginary parts of complex X and/or Y arguments ignored

> In HW3\_1d (line 35)

Graph



# Question 2

MatLab

%

clear;

syms q

% Total Revenue Function

TR = 5900\*q - 10\*q^2;

% Total Cost Function

TC = 2\*q^3 - 4\*q^2 + 14\*q + 845;

% Profit Function

profit = TR-TC;

disp('profit = ')

pretty(profit);

profit1 = diff(profit);

crit\_pts = solve(profit1);

disp(crit\_pts);

fplot(profit);

% fplot(@(x) profit, xinterval [0 100],'b');

hold on

grid on;

plot(double(crit\_pts), double(subs(profit,crit\_pts)),'ro');

title('Maximum and Minimum of Profit');

hold off

Console Output

>> HW3\_2

profit =

3 2

- 2 q - 6 q + 5886 q - 845

- 982^(1/2) - 1

982^(1/2) - 1

Image



# Question 3

MatLab

clear;

syms Q;

TC = (Q^3 - 5\*Q^2 + 60\*Q);

AC = (Q^2 - 5\*Q + 60);

disp('Average Cost function');

pretty(AC);

AC1 = diff(AC);

disp('first derivative');

pretty(AC1);

roots = solve(AC1);

disp('critical valye for Average cost minimum')

pretty(roots);

AC2 = diff(AC1);

disp('second derivative')

pretty(AC2);

test = roots;

if (AC2 < 0)

disp ('second derivative is negative')

elseif (AC2 == 0)

disp ('second derivative is zero')

else

disp ('second derivative is positive')

end

f = @(Q) Q^2 - 5\*Q + 60;

f(roots);

disp('The minimum average cost is:');

disp(ans);

Console Output

>> HW3\_3

Average Cost function

2

Q - 5 Q + 60

first derivative

2 Q - 5

critical valye for Average cost minimum

5

-

2

second derivative

2

second derivative is positive

The minimum average cost is:

215/4

Image



# Question 9

