Comparison of Gradient Descent and Newton's Type Optimisation Methods

Optimisation
Project Report

Serkan Burak Örs*

*orss19@itu.edu.tr

Contents

1 Introduction

In this project report, both gradient descent method and Newton type method implemented in order to solve given optimisation problem at Eq.[1.1]. Then the solution is visualised and results are discussed.

$$\text{minimize } F(x_1, x_2) = 4x_1^2 + 3x_2^2 - 5x_1x_2 + 8x_1 \tag{1.1}$$

$$\text{subject to: } x_1 + x_2 = 4 \tag{1.2}$$

2 Solution

By using equation[1.2], x_2 can be written in terms of x_1 as:

$$x_1 + x_2 = 4 \rightarrow x_2 = 4 - x_1 \tag{2.1}$$

And this obtained x_1 value can be substituted in equation[1.1] as:

$$f(x_1, x_2) = 4x_1^2 + 3(4 - x_1)^2 - 5x_1(4 - x_1) + 8x_1 = 12x^2 - 36x + 48 \tag{2.2}$$

Gradient Descent method:

$$\text{step size, } \alpha = 0.02 \tag{2.3}$$

$$x_{i+1} = x_i - \alpha f'(x_i) \tag{2.4}$$

Newton Type method:

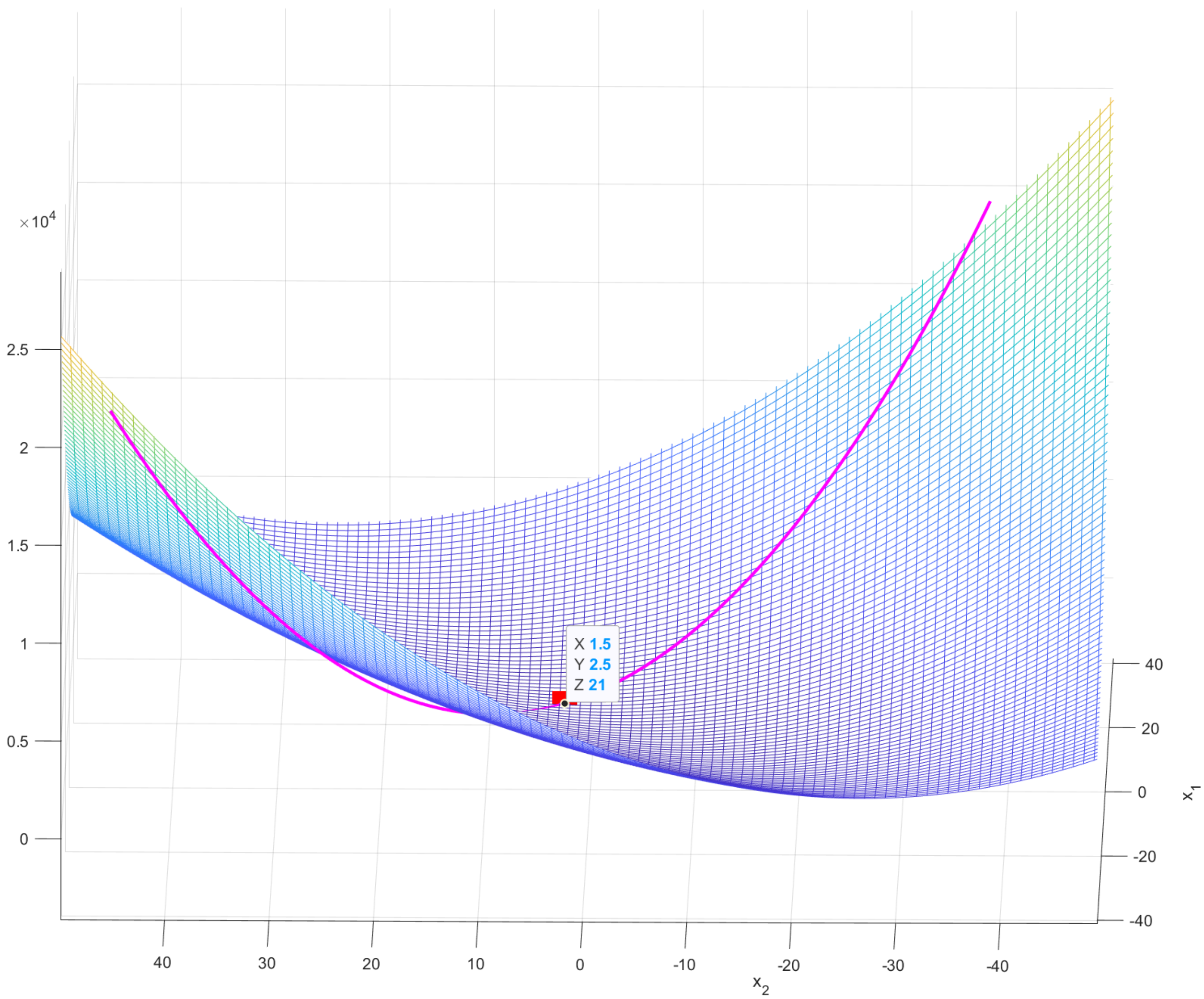
$$x_{i+1} = x_i - \frac{f'(x_i)}{f''(x_i)} \tag{2.5}$$

By solving the problem with using these methods, we can obtain results below as:

- Due to Gradient Descent method
  - $x_1 = 1.5$
  - $x_2 = 2.5$
  - $F = 21$
- Due to Newton-type method
  - $x_1 = 1.5$
  - $x_2 = 2.5$
  - $F = 21$

2.1 Visualisation

We can visualize the results as:



2.2 Discussion

For this problem, Newton type method and Gradient Descent method reached the result at same iteration. But Gradient descent method became very slow when it reached near of the result and approximate it very slowly.

3 Appendix: MATLAB Code

```
clc; clear; close all;
syms x
alpha = 0.02;
x1 = ones(1,50);
x1(1,1) = 0;
x2 = ones(1,50);
x2(1,1) = 0;
F = 12*x.^2 - 36*x + 48;
Fprime = diff(F);
FdoublePrime = diff(Fprime);

%Gradient descent
for i=1:1:50
x1(i+1) = x1(i)-(alpha*subs(Fprime,x1(i)));
end

%Newton-type
for i=1:1:50
x2(i+1) = x2(i)-((subs(Fprime,x2(i)))/(subs(FdoublePrime,x2(i))));
end

fprintf('Due to Gradient Descent method:\n x1 = %f\n x2 = %f\n F = ...
    %f\n',x1(end),4-x1(end),subs(F,x1(end)))
fprintf('Due to Newton type method:\n x1 = %f\n x2 = %f\n F = ...
    %f\n',x2(end),4-x2(end),subs(F,x2(end)))

figure(1),clf
xx=[-50:1:50]';
for  ii=1:length(xx)
for  jj=1:length(xx)
FF(ii,jj)=  4*xx(ii)^2 + 3*xx(jj)^2 - 5*xx(ii)*xx(jj) + 8*xx(ii);
end
end
mesh(xx,xx,FF);%
hold  on;%

plot3(xx,4-xx, 4*xx.^2 + 3*(4-xx).^2 - 5*xx.*(4-xx) + 8*xx ...
    ,"m-","LineWidth",2);%
xlabel("x_1");  ylabel("x_2");  %
```

```
hold  off;
hh=get(gcf,"children");

hold  on
plot3(1.50,2.50,21.00,"rs","MarkerSize",20,"MarkerFace","r")
```

4 References

[1] Kirk D.E., *Optimal Control Theory*, Dover Publications, 1998.

[2] Venkataraman P., *Applied Optimization with MATLAB Programming*, John Wiley & Sons ,2022.

[3] Xue D., *Solving Optimization Problems with MATLAB*, De Gruyter, 2020.

[4] Başpınar B., *Istanbul Technical University UCK439E - Introduction to Optimal Control Lecture Notes*, 2022.