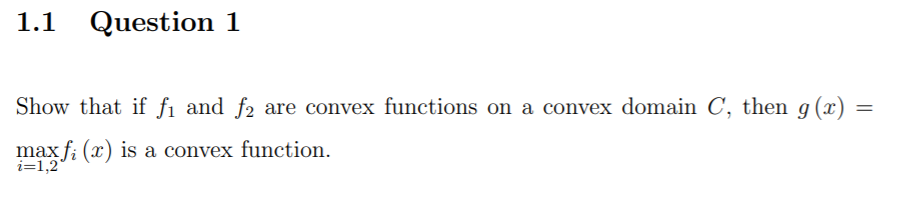
**INTRODUCTION TO OPTIMIZATION**

HOMEWORK 2

DUE DATE : 20/05/2021

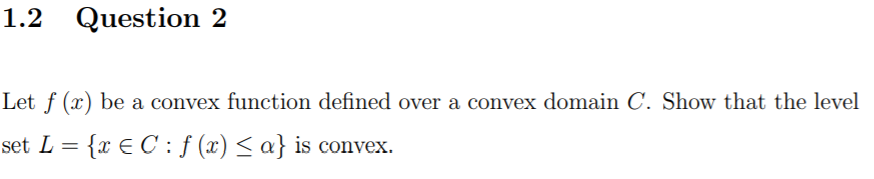


Answer

For

So we get:

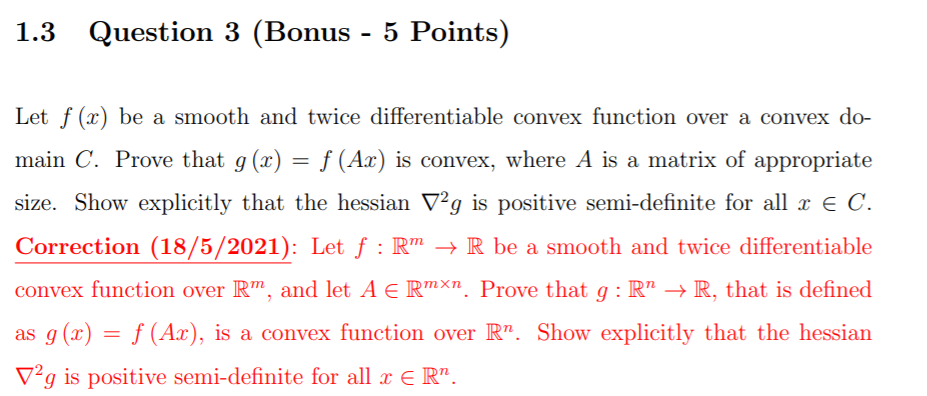
We conclude that



Answer

For

Meaning



Answer

For

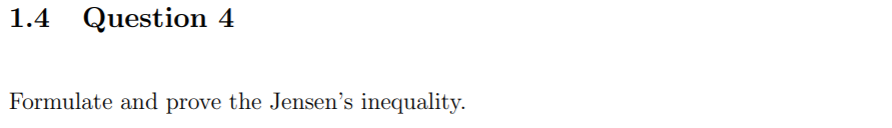
Now we want to show that the hessian is positive:

In the first homework we concluded that for function g(x)=f(Ax) the hessian can be expressed as following:

*Now we'll show that*

We'll multiply \* by an arbitrary vector z and we'll get:

This is true for every vector y(that is different from ) including



Answer

Jensen inequality :

Let be a convex function defined on a convex set . Then for any and the following holds:

Proof by using induction:

For n=2:

Given

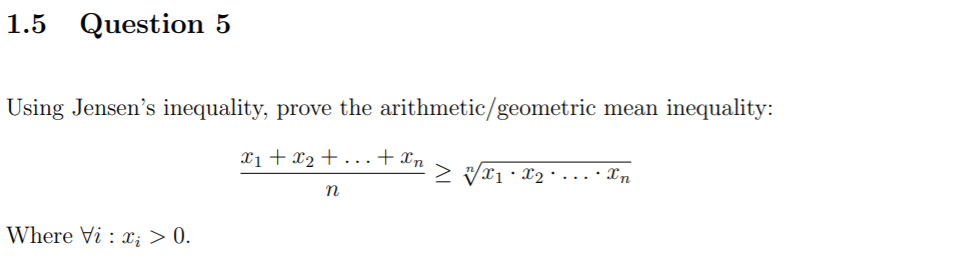
Assume it's true also for some n (hypothesis )

Now we generalize it by assuming having , then by convexity quality we get:

Now we want to prove that it's true also for n+1

Given

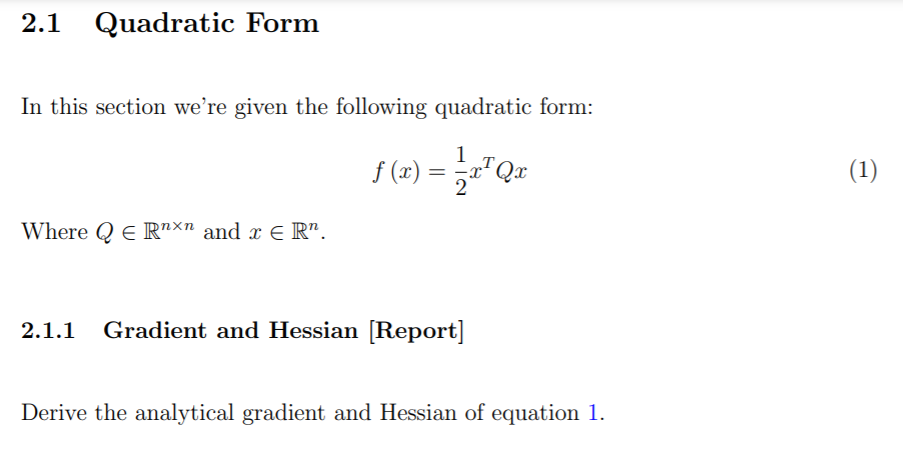
There's at least one , we'll assume it's

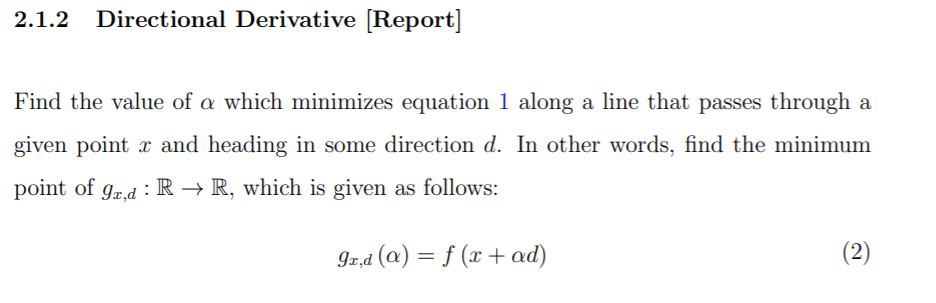


Answer

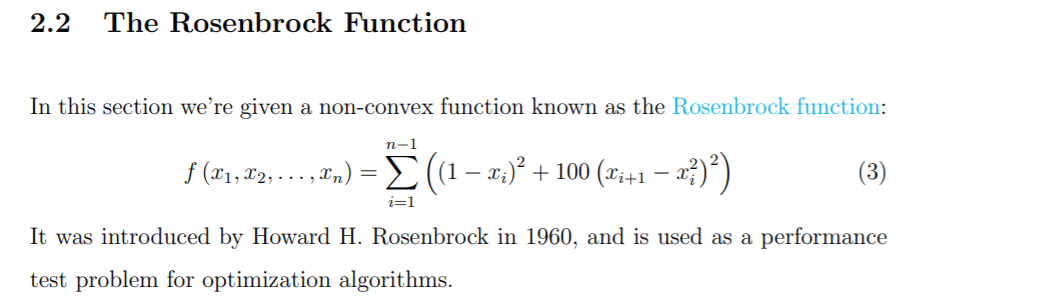
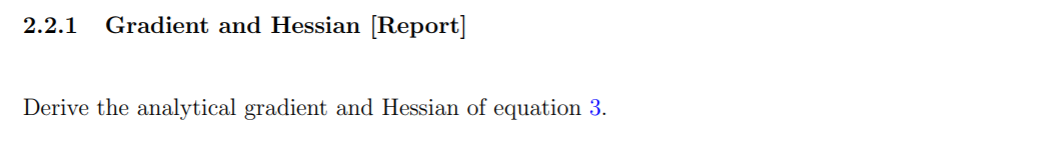
We define f(x)=-log(x) which is a convex function, and we apply it on left side of the inequality and then we apply the Jensen inequality and develop the expression to get to the right side.

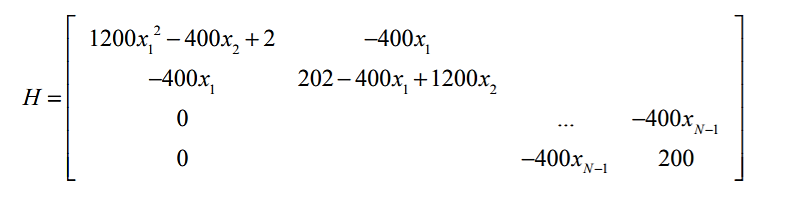
Log is a monotonically increasing function and thus;

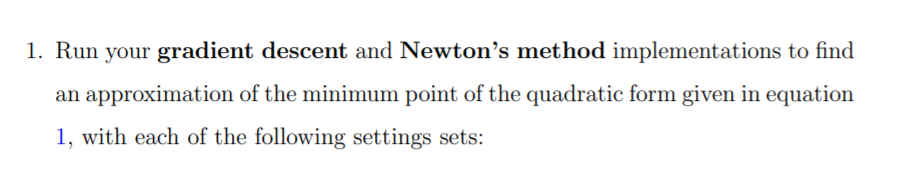
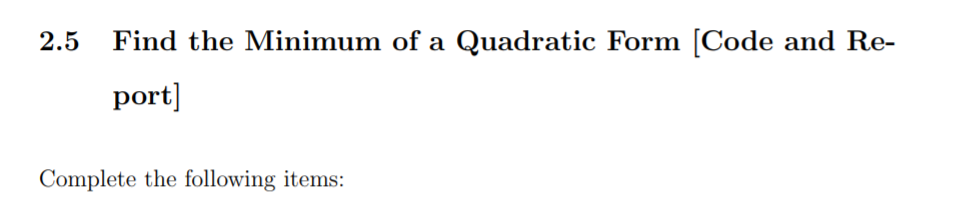
* 



Using the definitions of gradient and hessian matrix we get:







Conclusions:

Exact search is not very practical for most cases especially when using newton's method, since it requires a lot of iterations to find the minimum .(comparing set 4 vs set 1 for example)

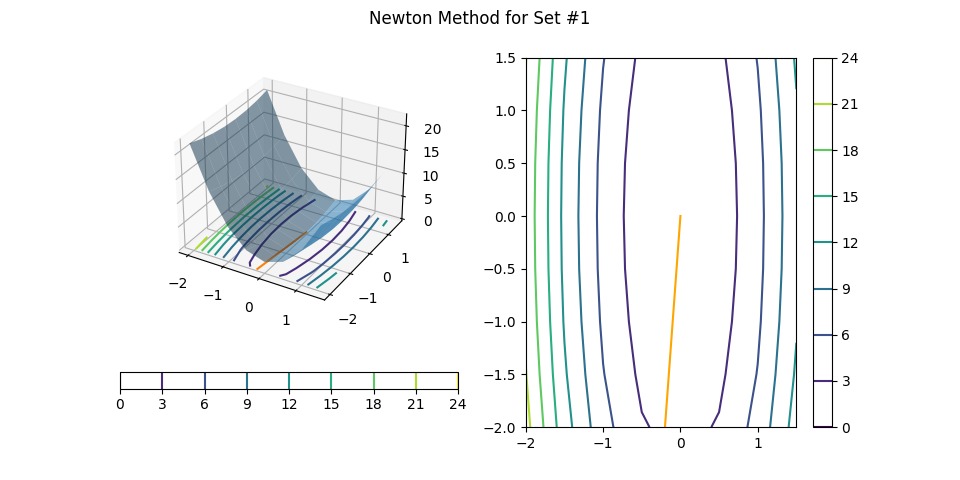
In most cases , using newton's method with inexact search resolves the problem quicker than gradient descent .

From set 3 and 5 we learn that if we have a symmetric matrix , using gradient descent with exact search (but not newton's method )or using newton's method with inexact search(but not gradient's method ) would preform the best .

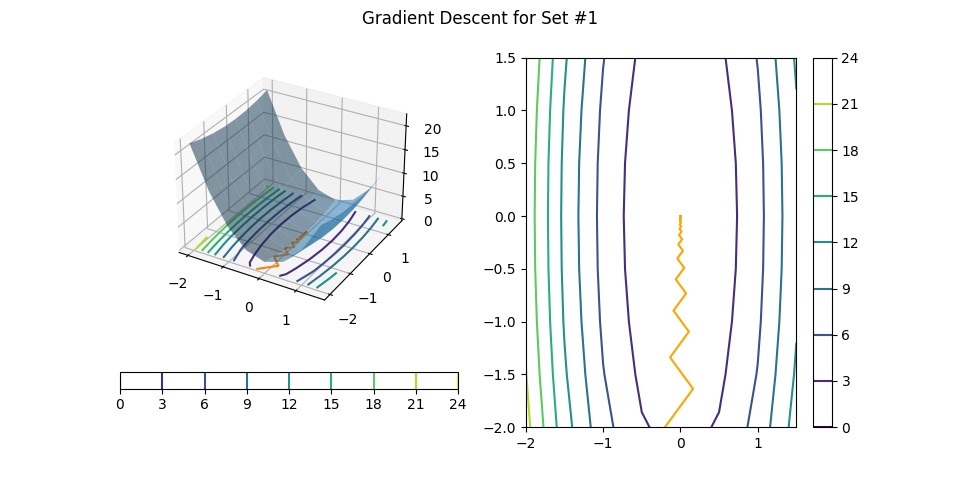
Form the rest of the sets that are inexact search, we learn that if we have an asymmetrical matrix, using newton's method is fastest and can solve the problem the quickest, and we also notice that even the gradient descent with inexact search would be faster than solving it gradient descent with exact search.

We notice that newton's method preforms badly when its exact search, and is best when it's inexact search.

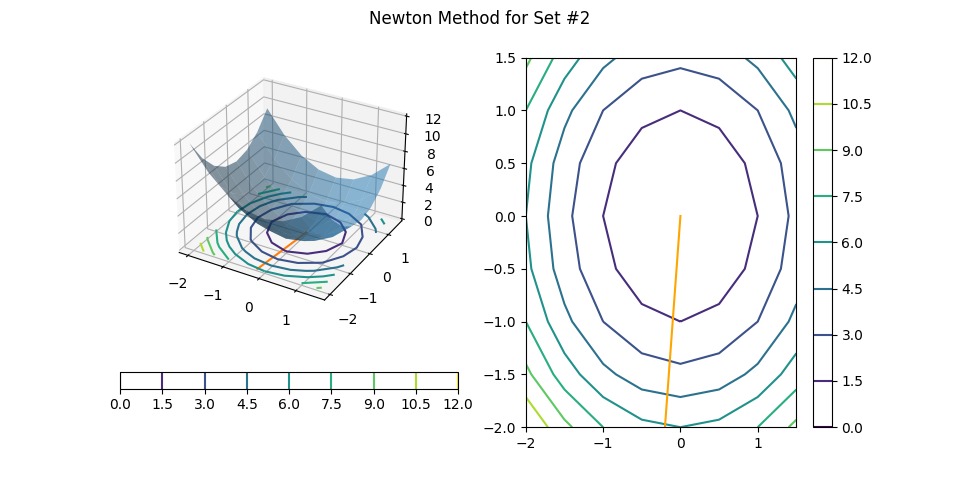
* + - * Graphs:
      * Set 1
* Newton method



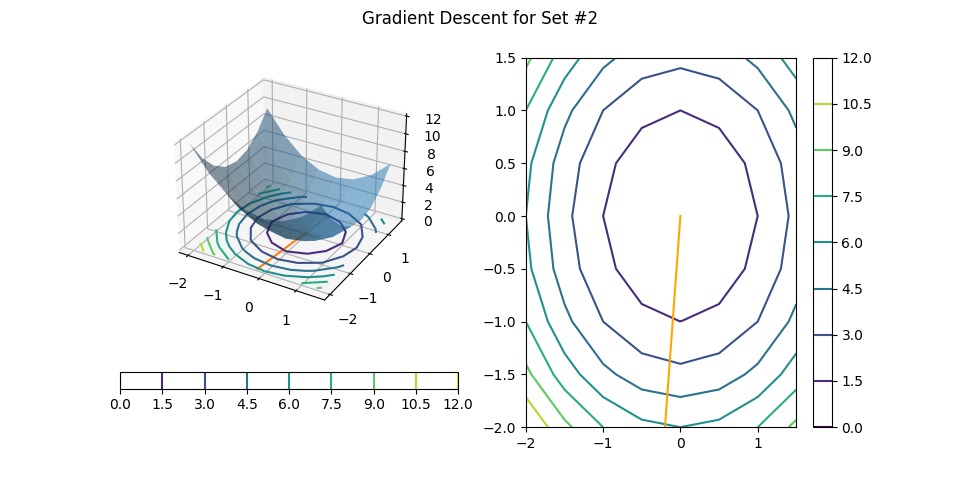
* gradient method



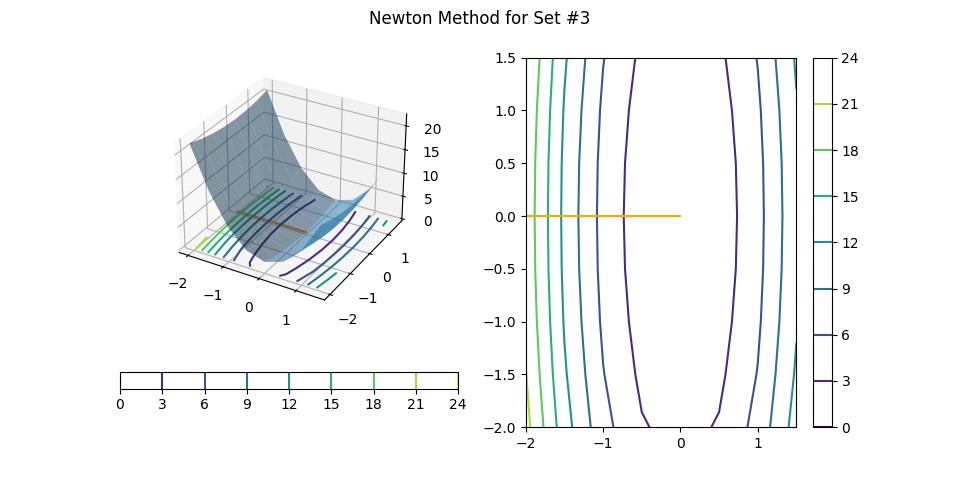
* + - * Set 2
* Newton method



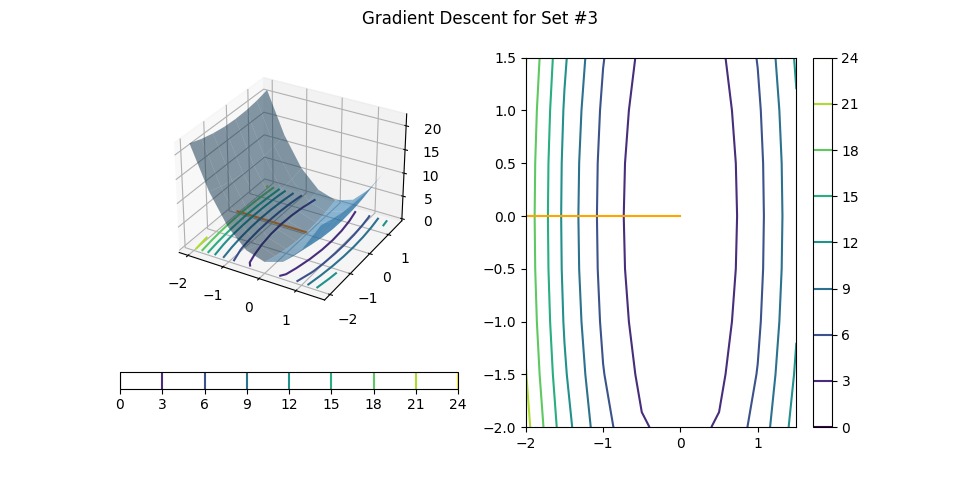
* gradient method



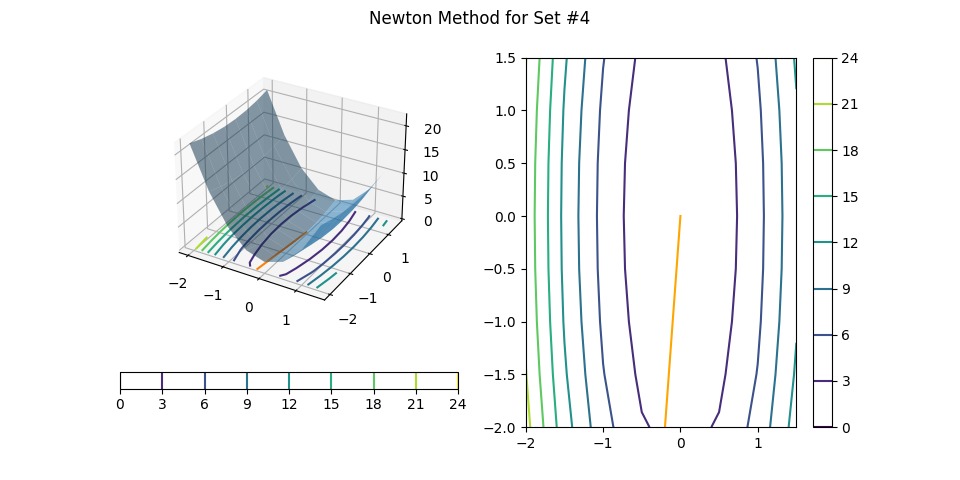
* + - * Set 3
* Newton method



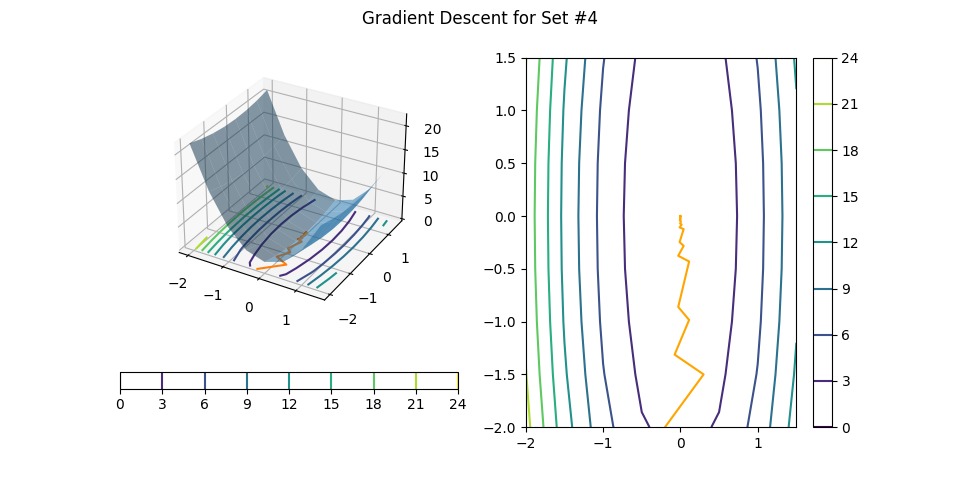
* gradient method



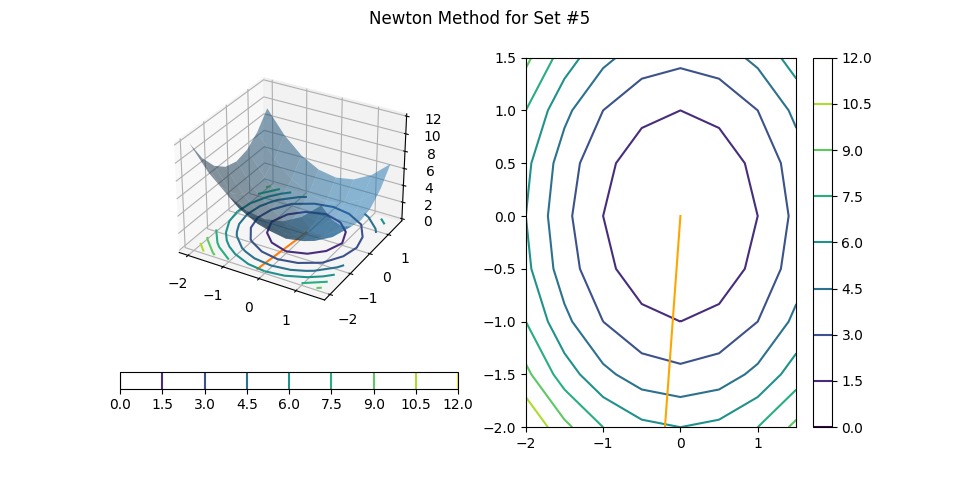
* + - * Set 4
* Newton method



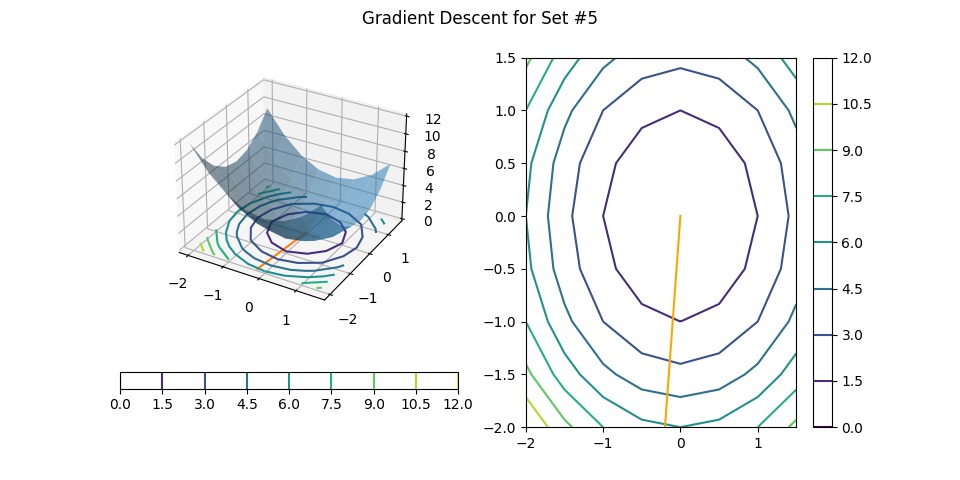
* gradient method



* + - * Set 5
* Newton method

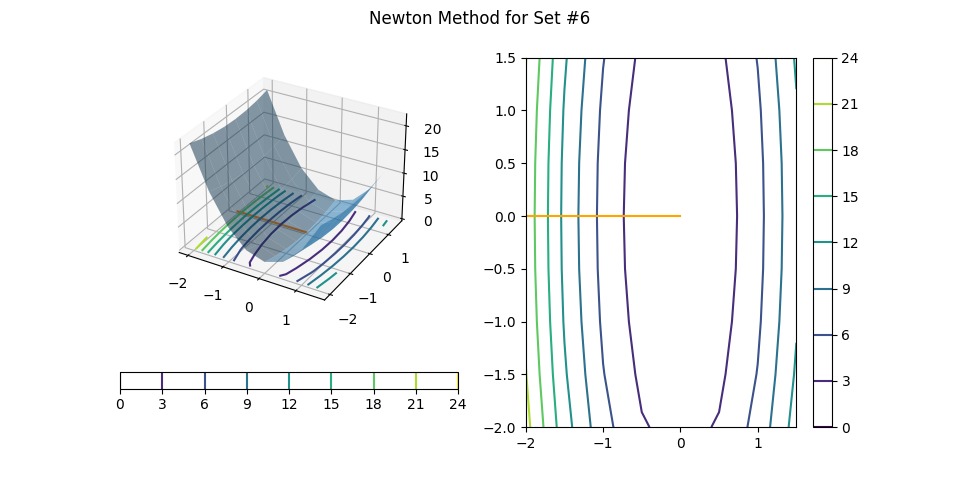


* gradient method

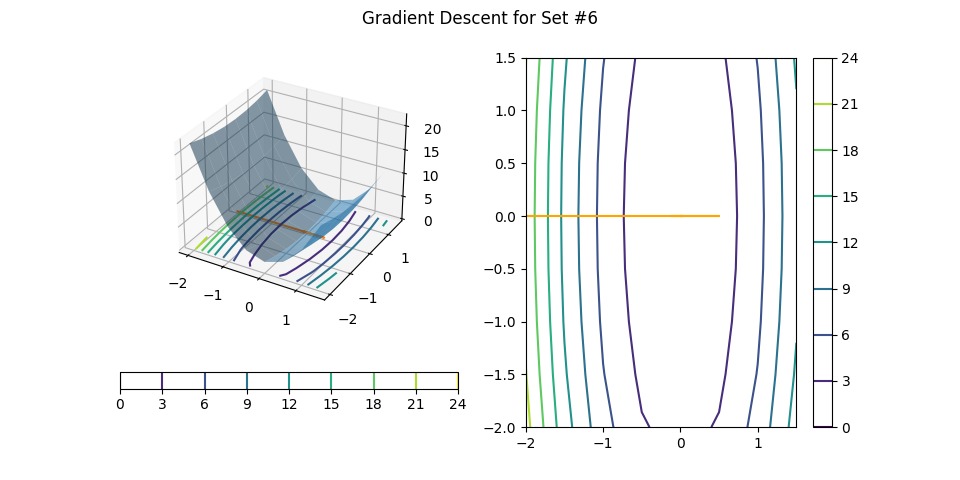


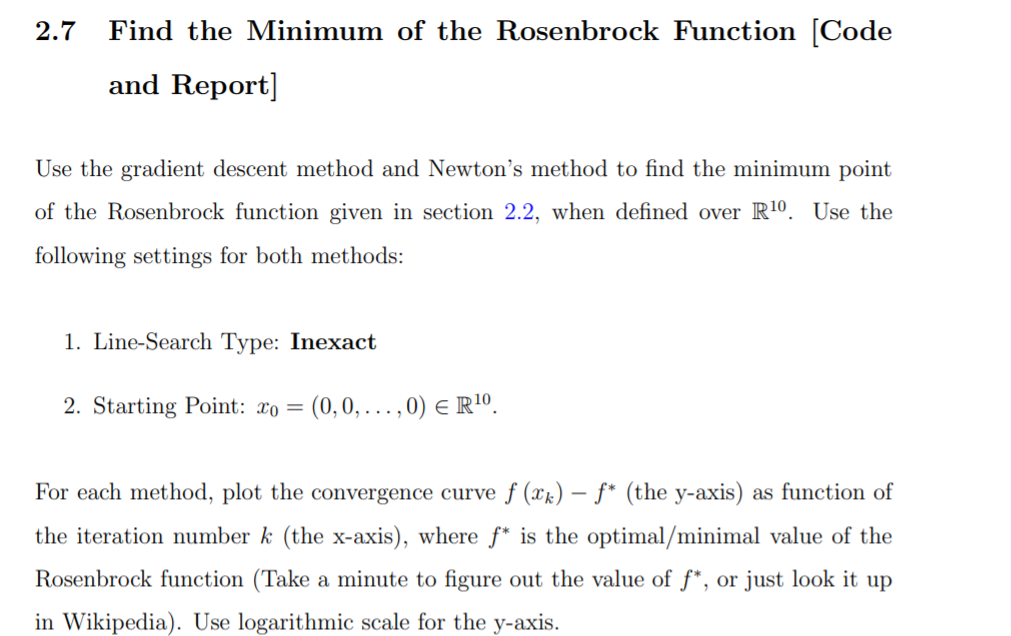
Set 6

* Newton method



* gradient method









We can see from the graphs that using newton's method leads to a faster convergence to the optimal function where it takes 25 iterations for a really small error margin, while the inexact gradient descent will take about 17 thousand iterations to converge to the optimal function within the range of that same error margin.

Conclusion, newton's method is far more efficient in this case .