Report

Assignment 1

Name: Udrasht Pal

Roll No.: 2022201020

1. **Trid Function**

Find Stationary point,

i=1

to

summing of all such equations

at

Add eq. (2) & (3)

Similarly,

Now by eq. (4) we have

Using (3) & (4)

We already computed in terms of.

put , we gat

Therefore , the stationary point for this function is -

Now we need to prove that Hessian is +ve .

We need to prove that |H(X)|>0  
We prove this by induction.  
Claim: Let be the size of Hessian. If denote determinant of Hessian, the  
and

Prove: Base care.

and True.  
Induction hypothesis is:

To prove:

Consider the hessian of size .

size .

Find by expending along R1

size .

|=

=

size .

By hypothesis

Hence prove

**Hessian of tried function has +ve trace and +ve determinant**

**So Hessian is +ve definite**

For d=2

**is minima at**

1. **Three Hump Camel**

Find Stationary point.

but the value of from from equation (2) to (1)

for simple simplicity put

We need to check at which points the function is minimum. For that, we have to prove that is semi-positive definite at that point.

**It is semi definite and**

at

at

at

at

Three Hump Camel  
 **is minima at**

at

at **)**

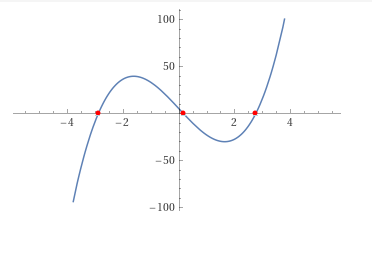
at

**Global Minima**

1. **Styblinski-Tang Function**

Find Stationary point.

By using graph method.

  
root of the equation is

similarly

. have same value

at

**It is semi definite and**

at

**It is not semi definite**

at

**It is semi definite and**

**is minima at**

Conclusion  
 is a minimizes

is the global minimizes.

1. **Rosenbrock Function**

1. **Root of Square Function**

similarly

Find Stationary point.

at

is minima at

**Global Minima**

**3. State which algorithms failed to converge and under which circumstances.**

The algorithms were run for the following test cases. For some of them they converged but from some, they not converged due to several issues:-

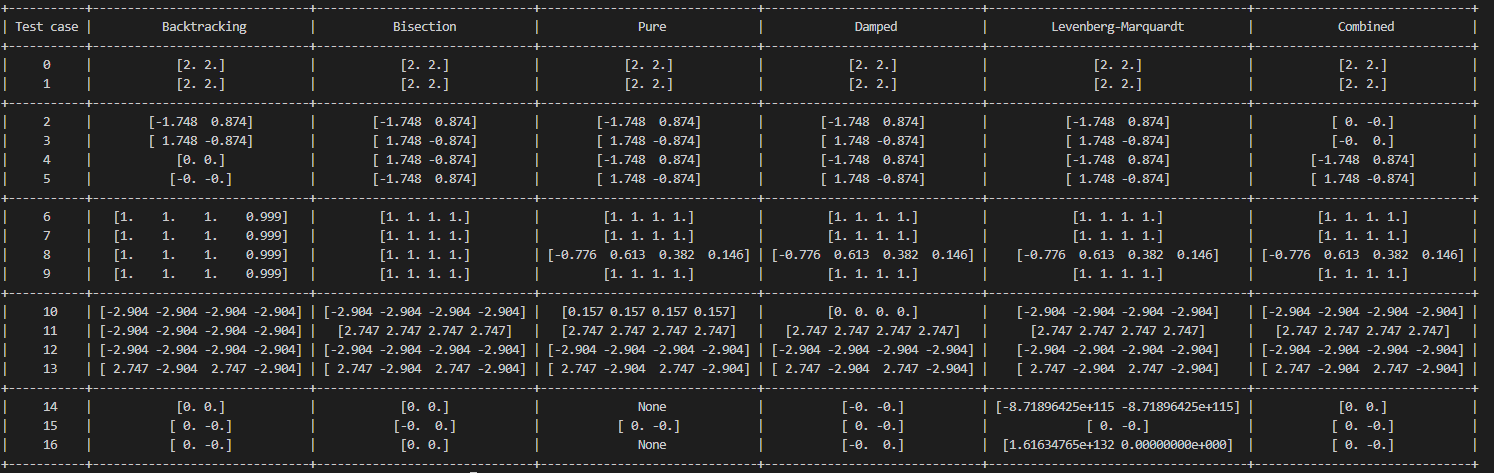
Singular Hessian matrices

Inadequate number of iterations.

Becoming trapped at local minima

Divergent series leading to overflow

Result:



Test case table

|  |  |  |
| --- | --- | --- |
| Test case | Function | Initial points |
| 0 | |  | | --- | | trid\_function | | [-2.0, -2] |
| 1 | trid\_function | [-2.0, -2] |
| 2 | |  | | --- | | three\_hump\_camel\_function | | [-2.0, 1] |
| 3 | |  | | --- | | three\_hump\_camel\_function | | [2.0, -1] |
| 4 | |  | | --- | | three\_hump\_camel\_function | | [-2.0, -1] |
| 5 | |  | | --- | | three\_hump\_camel\_function | | [2.0, 1] |
| 6 | rosenbrock\_function | [2.0, 2, 2, -2] |
| 7 | rosenbrock\_function | [2.0, -2, -2, 2] |
| 8 | rosenbrock\_function | [-2.0, 2, 2, 2] |
| 9 | rosenbrock\_function | [3.0, 3, 3, 3] |
| 10 | styblinski\_tang\_function | [0.0, 0, 0, 0] |
| 11 | styblinski\_tang\_function | [3.0, 3, 3, 3] |
| 12 | styblinski\_tang\_function | [-3.0, -3, -3, -3] |
| 13 | styblinski\_tang\_function | [3.0, -3, 3, -3] |
| 14 | func\_1 | [3.0, 3] |
| 15 | func\_1 | [-0.5, 0.5] |
| 16 | func\_1 | [-3.5, 0.5] |

**Three Hump camel function:**

(0,0) is the global minima for the function.

(-1.7474, 0.8737) and (1.7474, -0.8737) are local minima for the function

As show in the result table their are many test case where global minima is not occur due to stuck at local minima.

**Only 4 test case where the global minima achieve and 20 test case where the global minima is not achieve**

**Fail to achieve global minima**

* Fail at test case 2, 3 at conditions Backtracking, bisection, pure, Damped ,L-M
* Fail at test case 4, 5 at conditions bisection ,pure, Damped ,L-M,combine

**Rosenbrock function:**

Minima for the function is [1,1,1,….1]

As show in table there are many test cases where the minima is not occur because iteration required for convergence exceeded 10000.

**8 test case where the global minima is not achieve**

**Fail to achieve global minima**

* Fail at test case 6,7,9 at conditions Backtracking
* Fail at test case 8, at conditions Backtracking, bisection ,pure, Damped ,L-M,combine

**Styblinski Tang function:**

* X1 = [-2.90353 -2.90353 -2.90353 … -2.90353] 🡺 THIS IS THE GLOBAL MINIMA
* X3 = [2.7468 2.7468 2.7468 …. 2.7468] 🡺 THIS IS A LOCAL MINIMA

Test case 10 damp method stuck on initial point and pure method is also stuck on point

Test case 11 stuck in local minima for bisection, damped, pure L-M combine

In Newton's method, starting from a point far from the actual minimum can cause the algorithm to become stuck and fail to converge. Likewise, the table illustrates additional test cases that failed to converge due to one of the aforementioned reasons.

**Fail to achieve global minima**

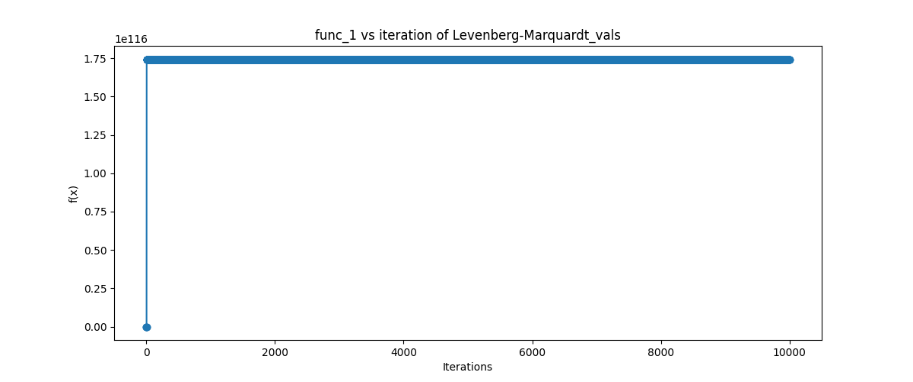
* Fail at test case 10 at conditions pure and damped
* Fail at test case 11 at conditions bisection ,pure, Damped ,L-M,combine
* Fail at test case 12 at conditions Backtracking ,bisection ,pure, Damped ,L-M,combine

**Root of square function:**

One stationary point and it is global minima that is (0,0)

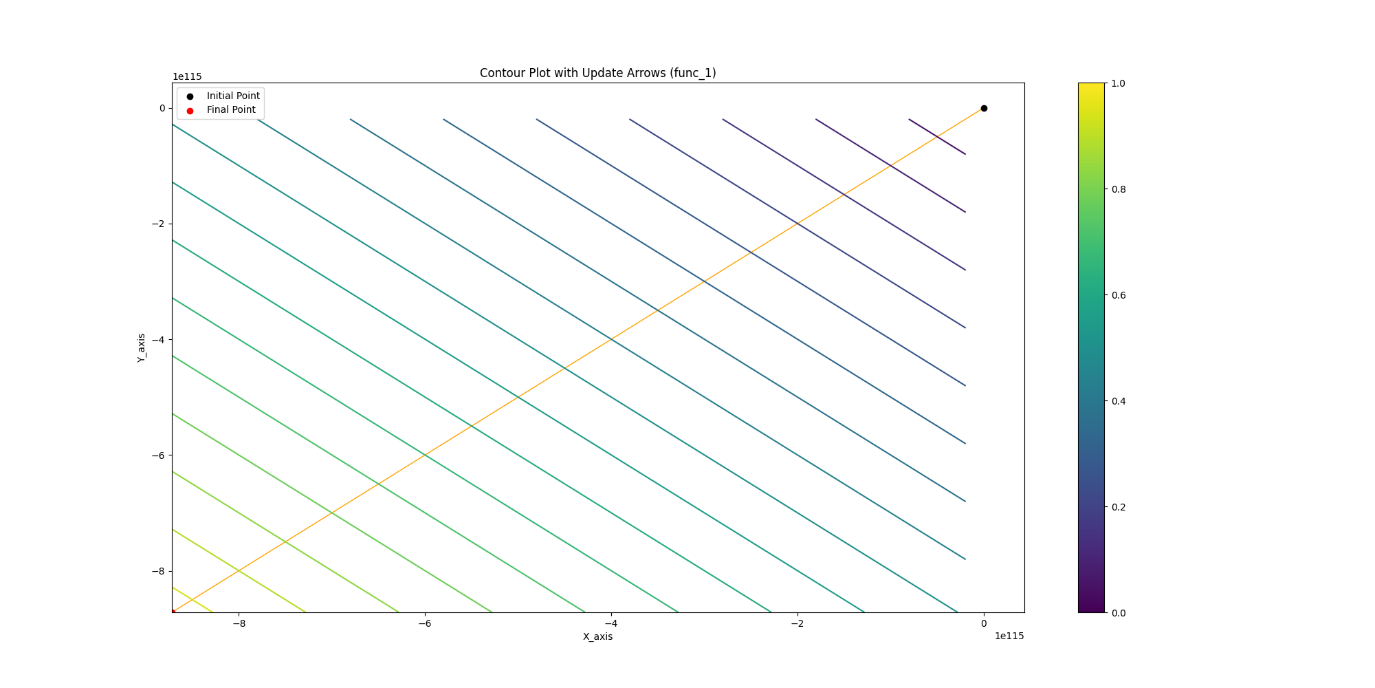
I test case 14,16 the pure condition the overflow encountered RuntimeWarning: overflow encountered in power return np.diag(1 / ((1 + point\*\*2) \*\* 1.5))

This is the reason for test case 14 and test case 16, you see such huge values in Levenberg-Marquard column.



initial point [-3, 3] Levenberg-Marquardt

condition the function is not Converge



initial point [-3, 3] Levenberg-Marquardt

condition the function is not Converge

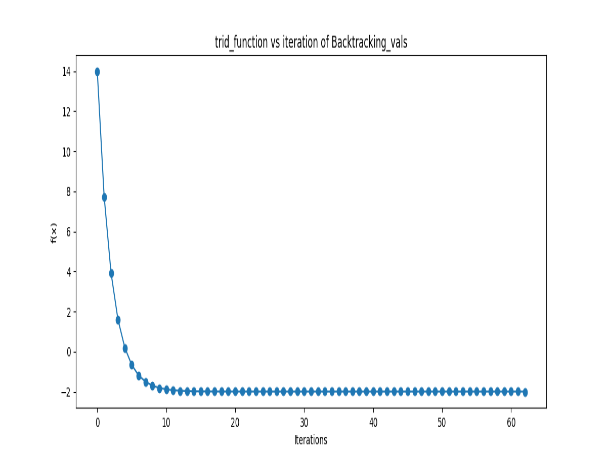
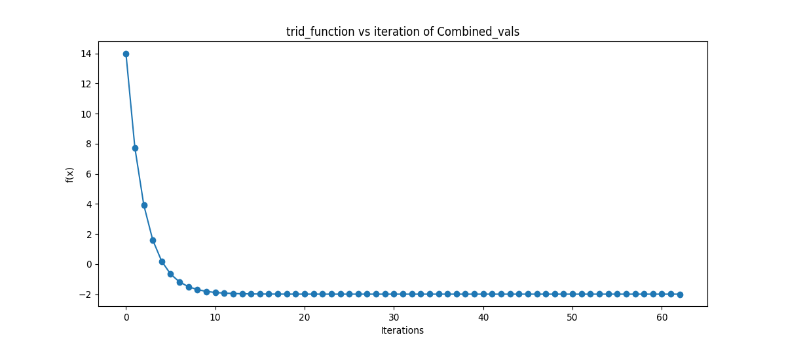
**Fail to achieve global minima**

* Fail at test case 14 at conditions pure and L-M
* Fail at test case 16 at conditions pure and L-M

**Plot f(x) vs iterations and |f′(x)| vs iterations**

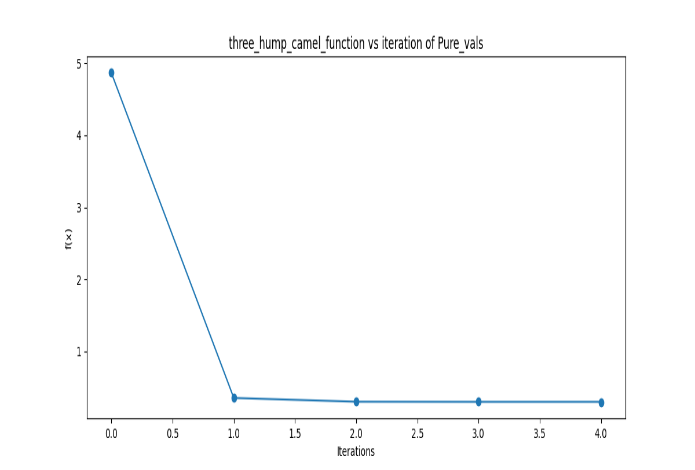
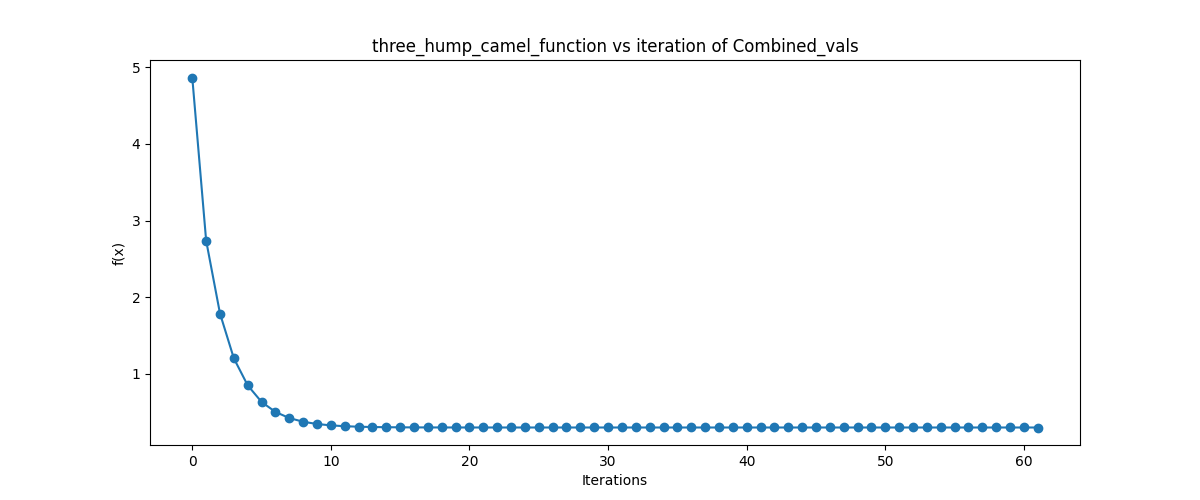
**f(x) vs iterations**

**Plot f(x) vs iterations of Trid Function.**

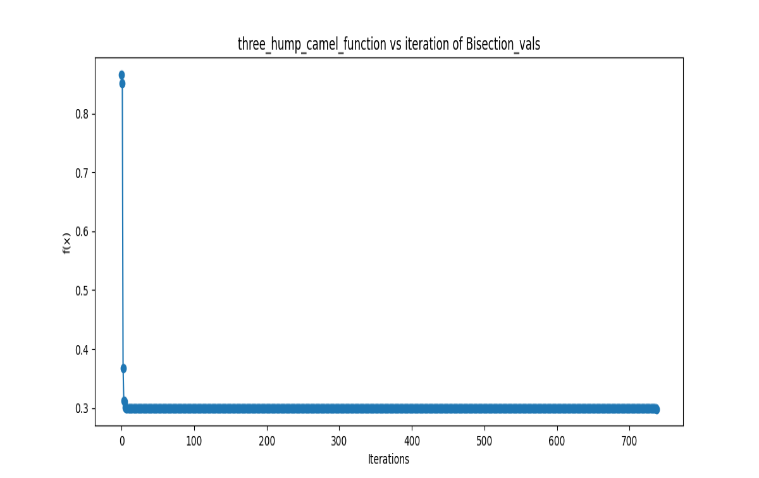


initial point [-2,-2] initial point [-2,-2]

Plot f(x) vs iterations of Three hump camel function

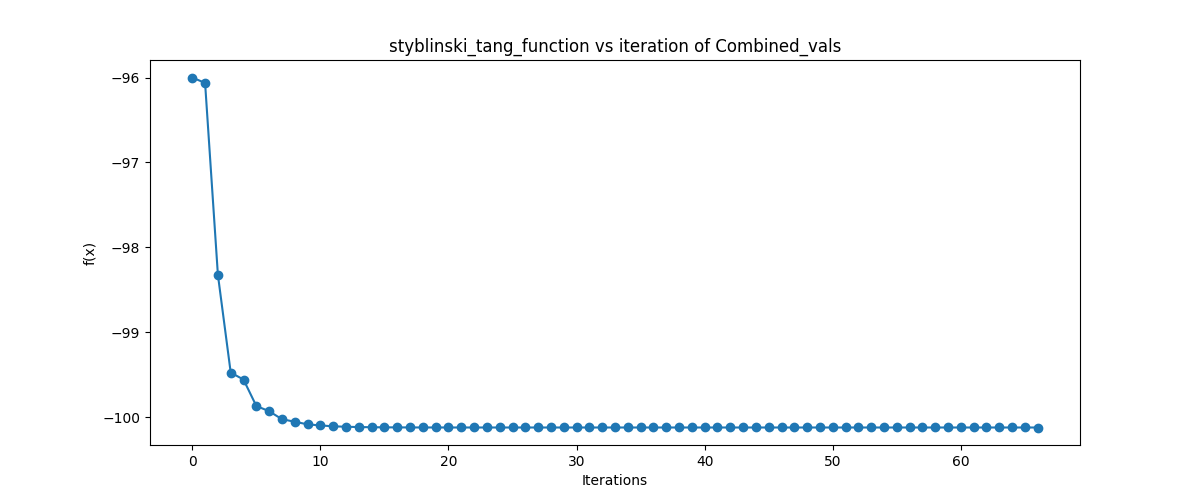
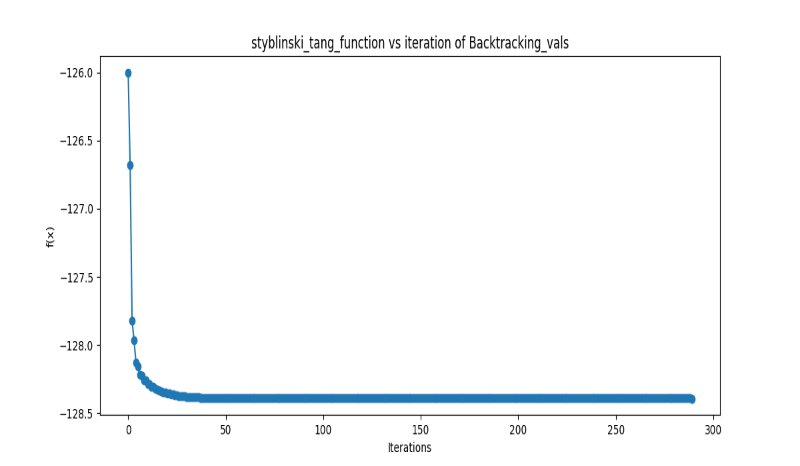
.

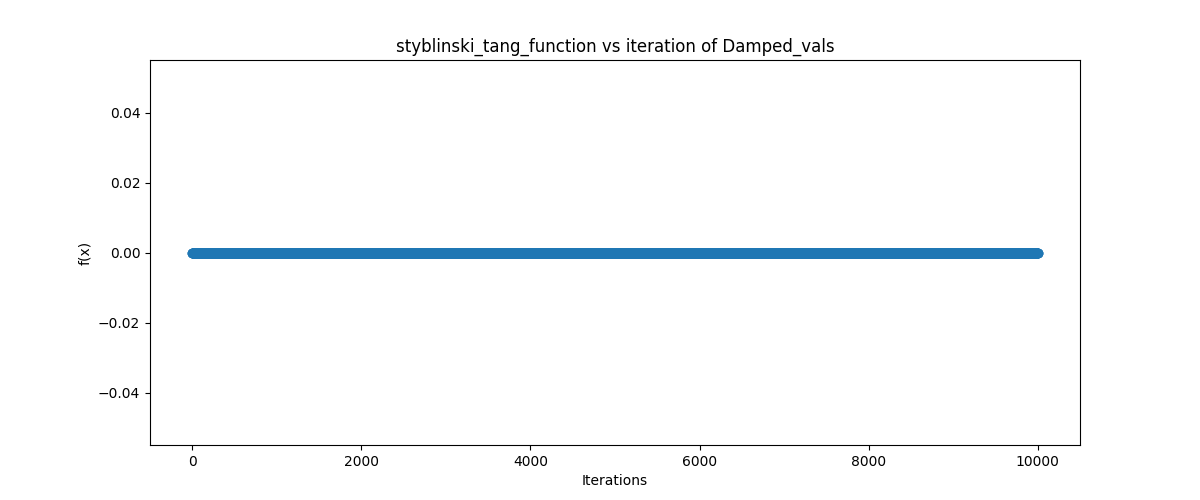
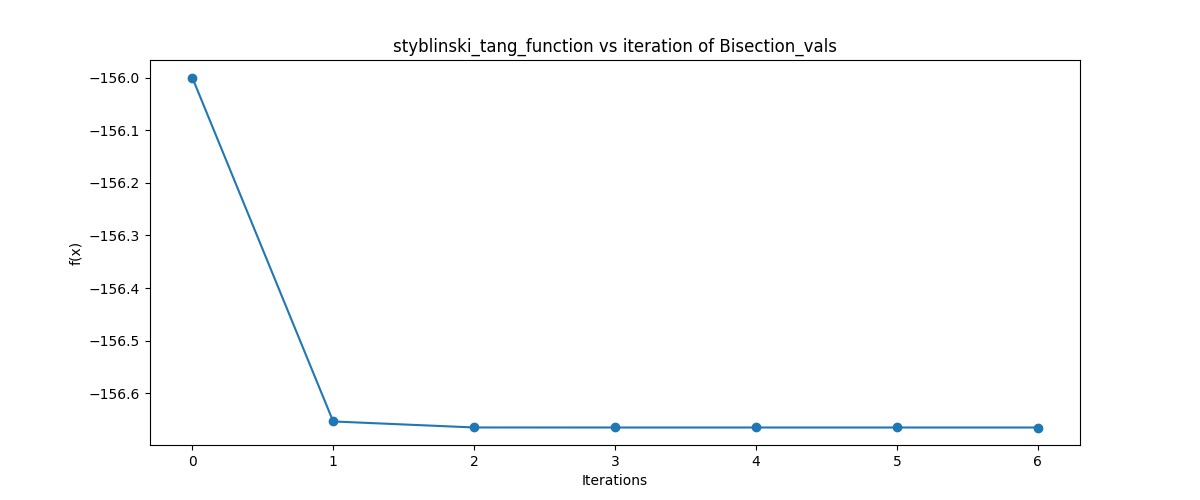
initial point [2,1] initial point [-2,-1]



initial point [2,- 1]

Plot f(x) vs iterations Styblinski-Tang Function

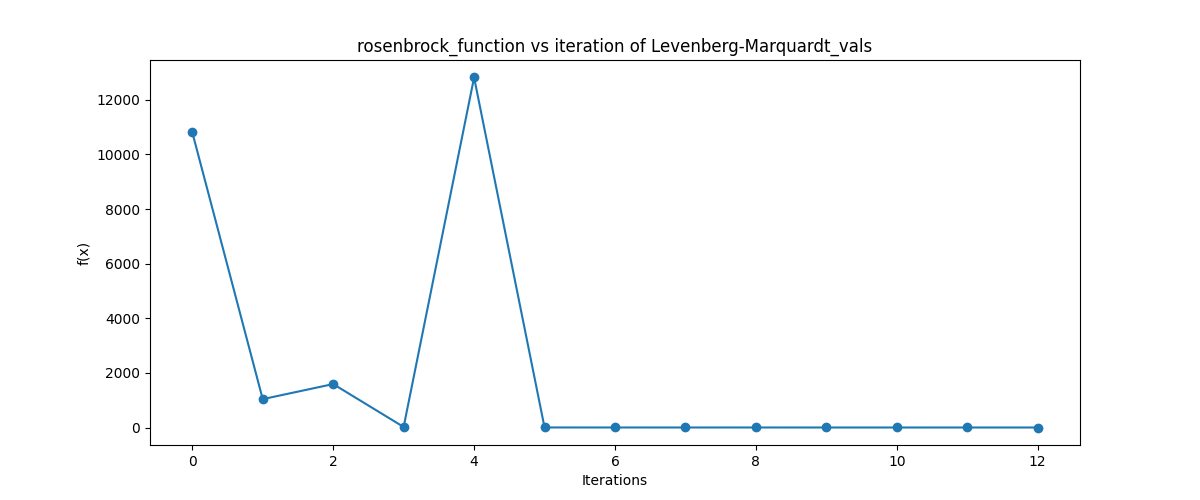
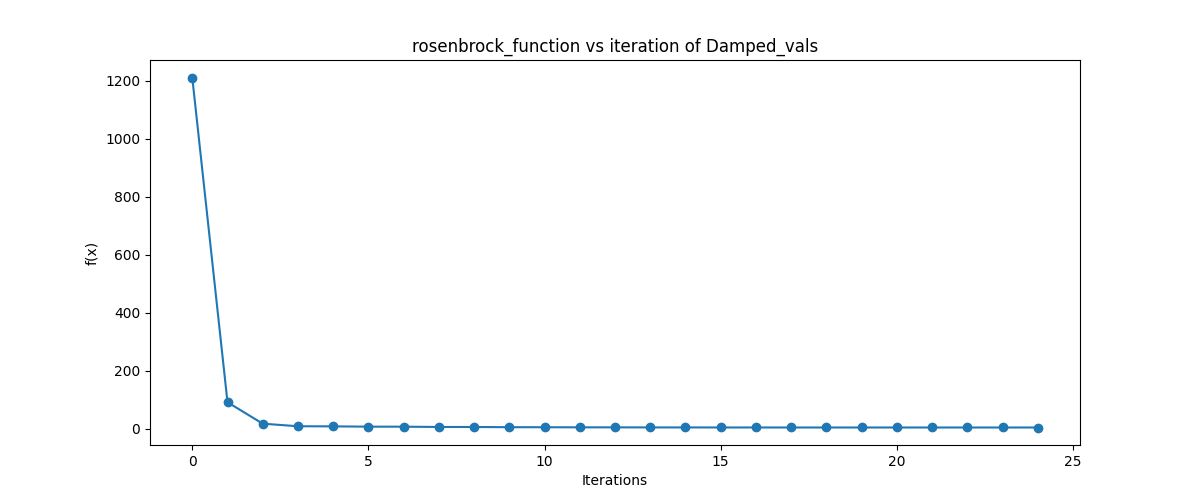


initial point [3,-3,3,-3] initial point [3,3,3,3] 

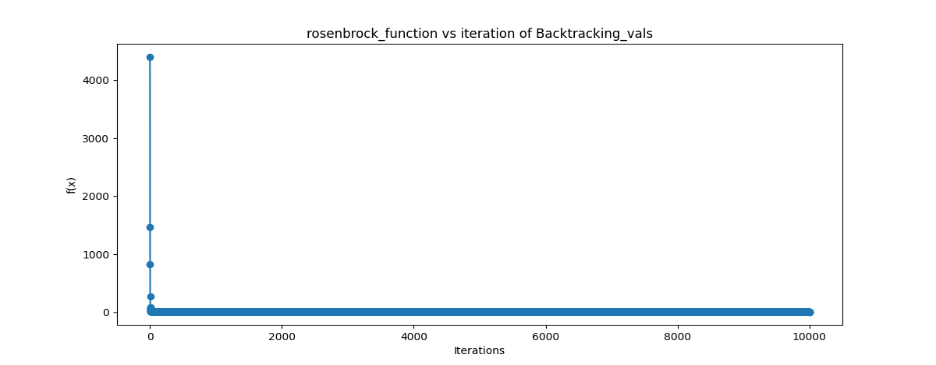
initial point [0,0,0,0] initial point [-3,-3,-3,-3]

damped condition the f(x) is not Converge

Plot f(x) vs iterations Rosenbrock Function



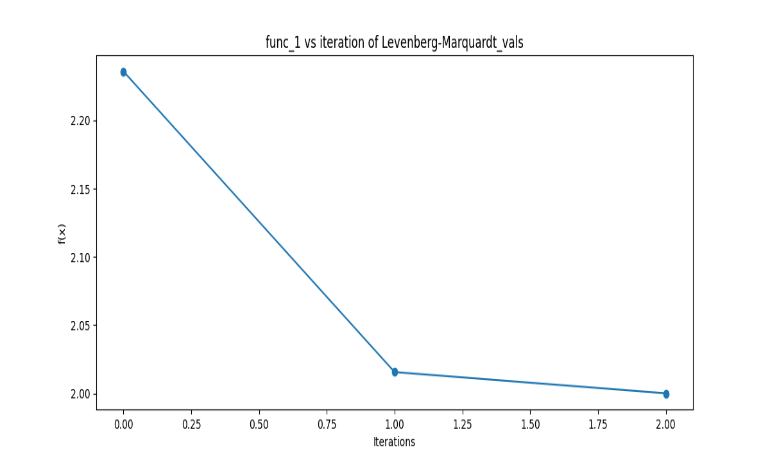
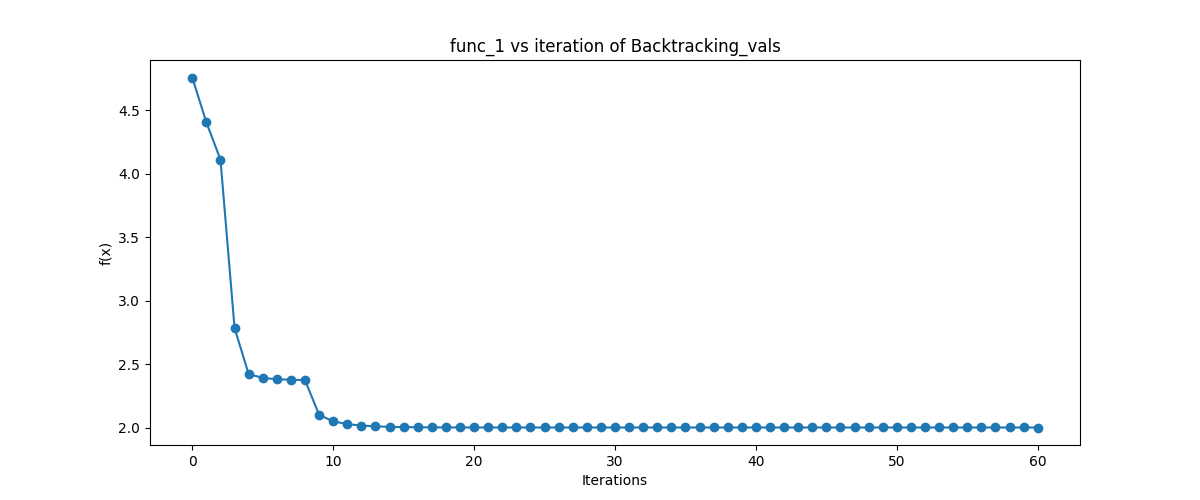
initial point [-2,2,2,2] initial point [3,3,3,3]

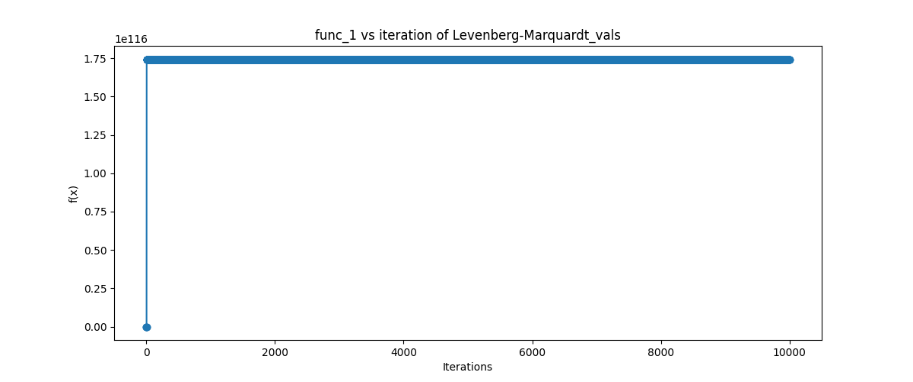


At initial point [2,2,2,2]

backtracking condition the function is not Converge

Plot f(x) vs iterations Root of Square Function(func\_1)



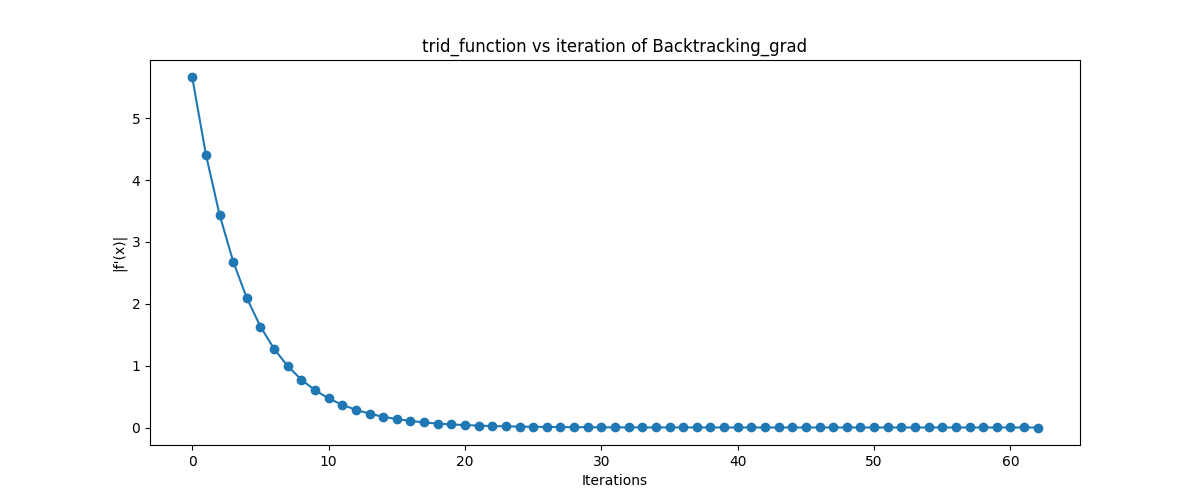
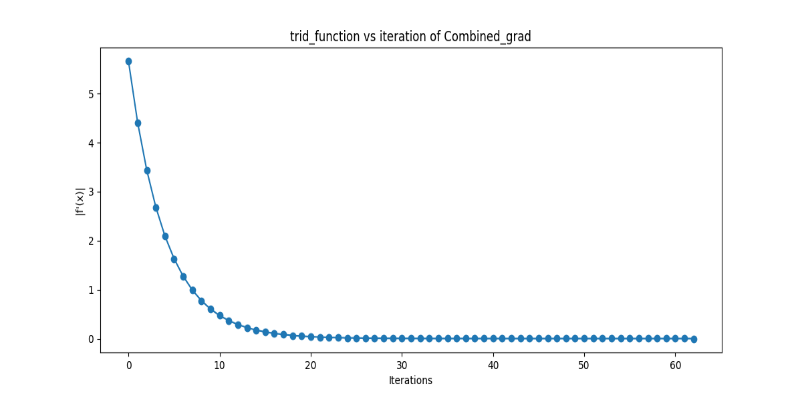
initial point [-0.5, 0.5] initial point [-3.5, 0.5]

At initial point [-3, 3] Levenberg-Marquardt

condition the function is not Converge

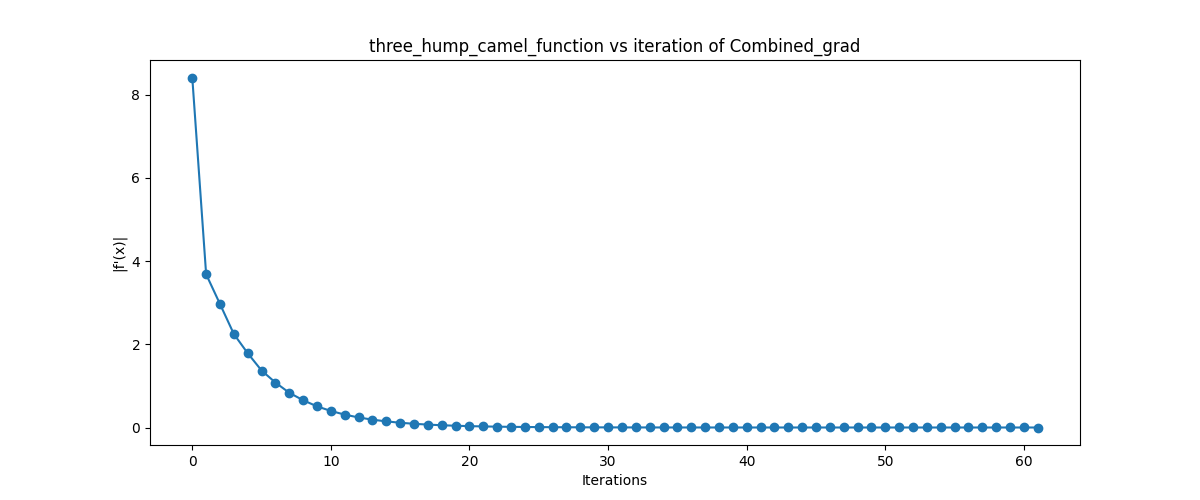
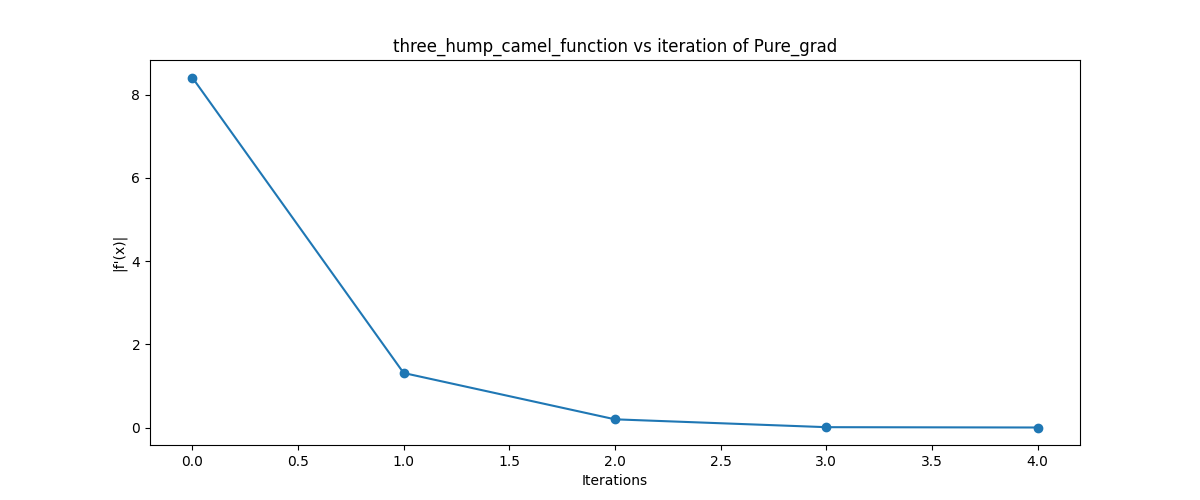
**|f′(x)| vs iterations**

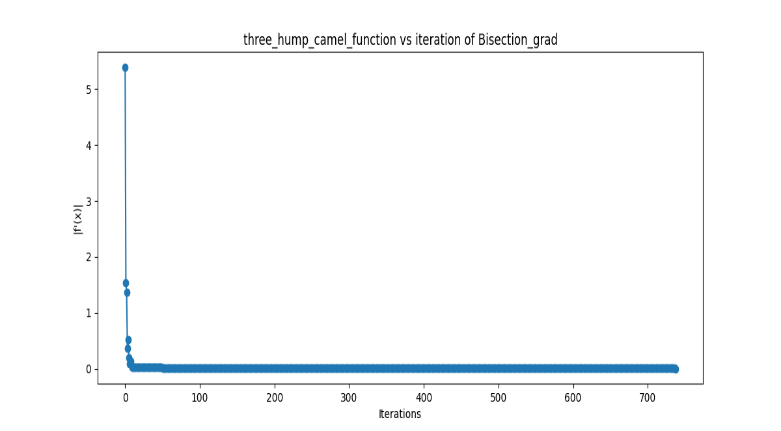
Plot |f′(x)| vs iterations of Trid Function.



initial point [-2,-2] initial point [-2,-2]

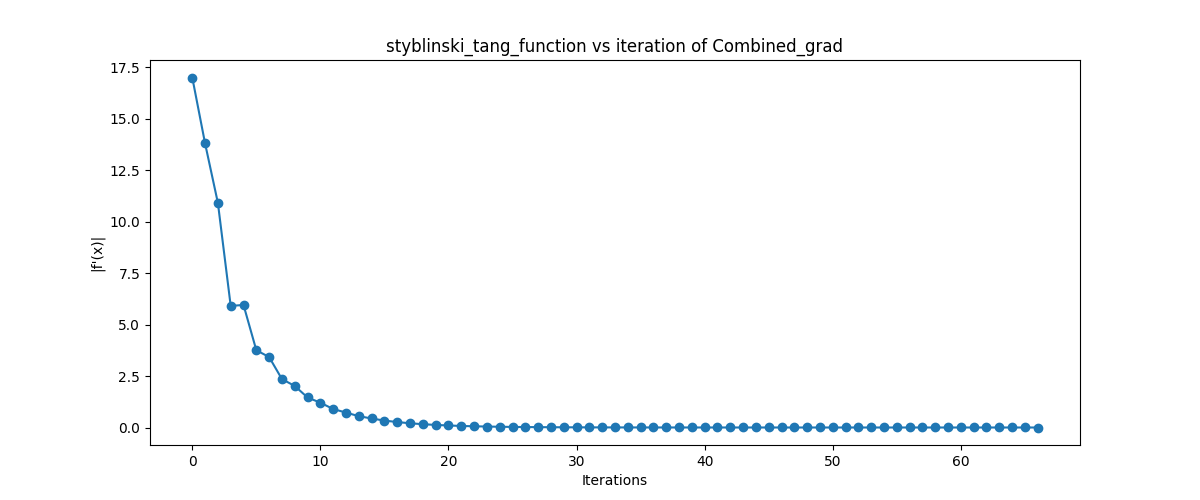
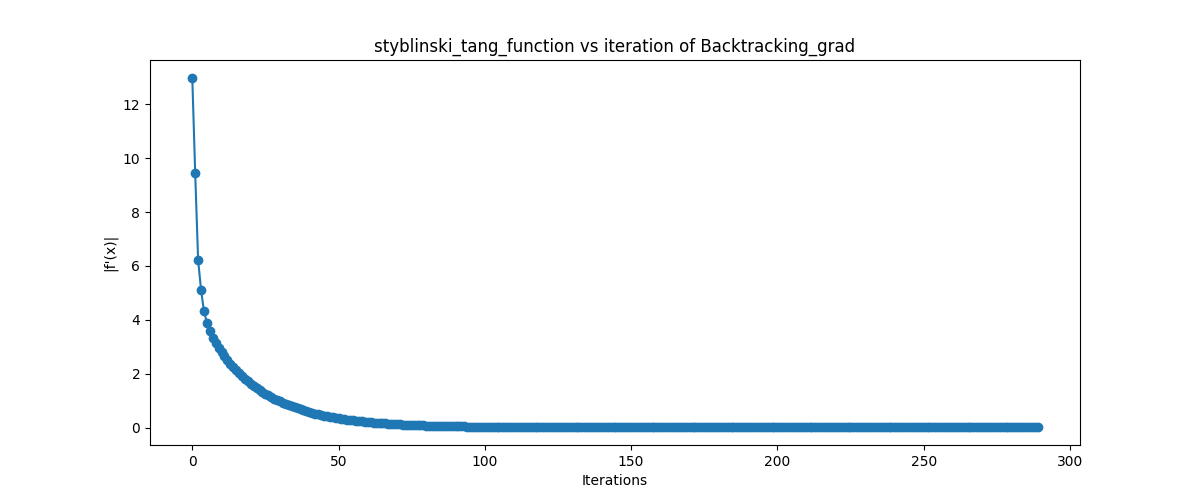
Plot |f′(x)| vs iterations of Three hump camel function



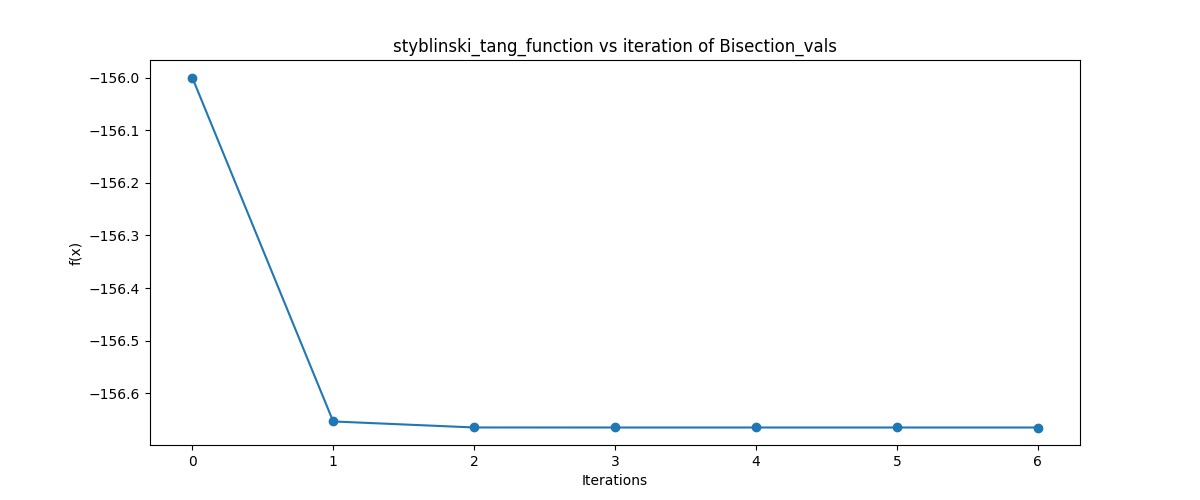
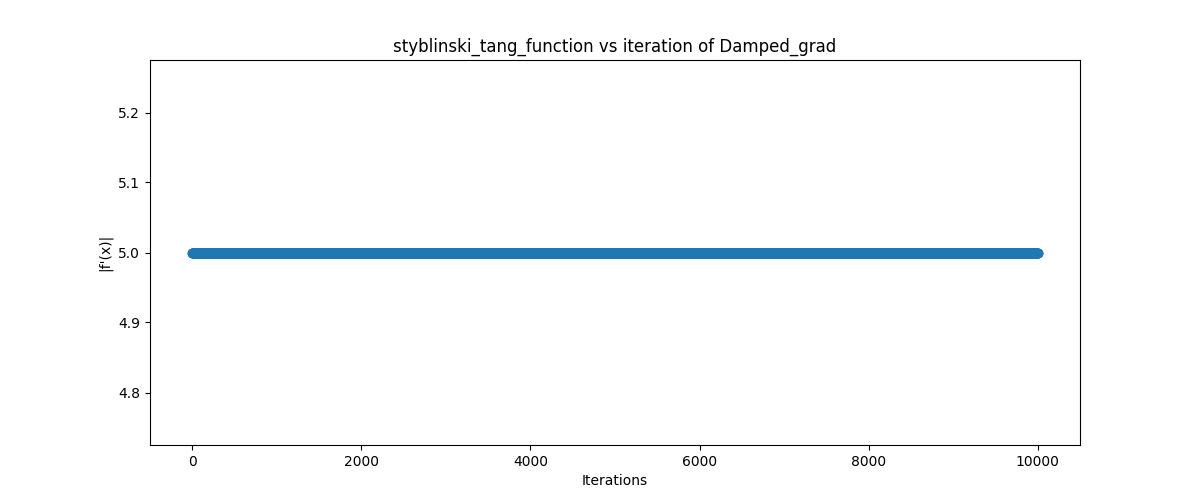
initial point [2,1] initial point [-2,-1] 

initial point [2,-1]

Plot |f′(x)| vs iterations Styblinski-Tang Function



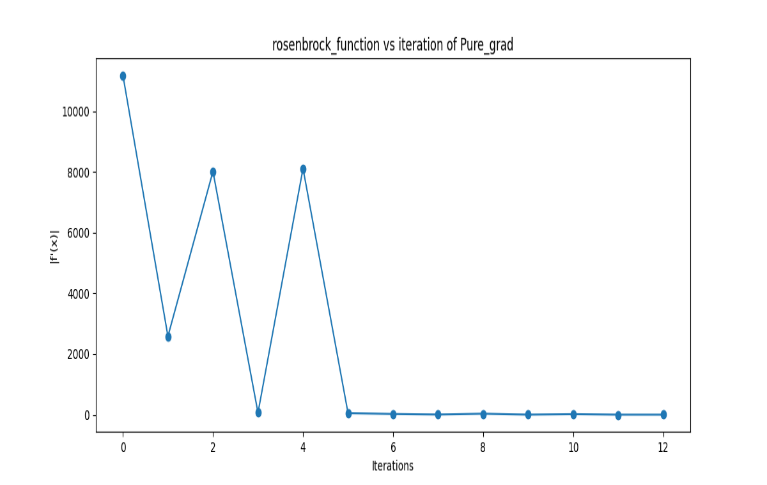
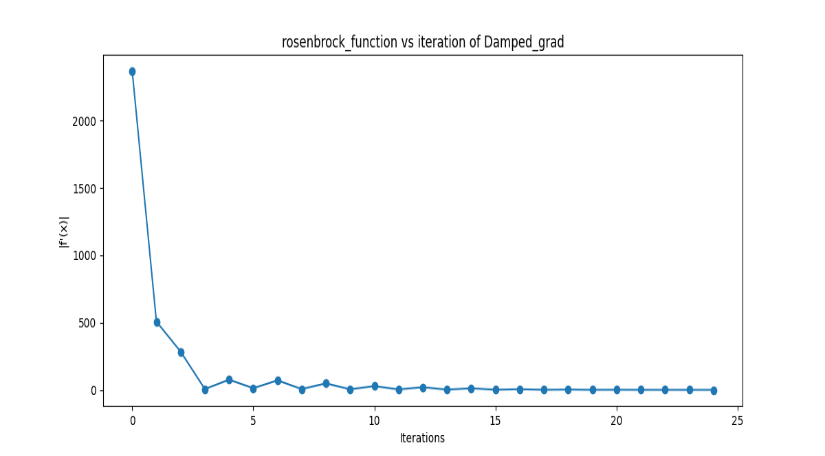
initial point [3,-3,3,-3] initial point [3,3,3,3]



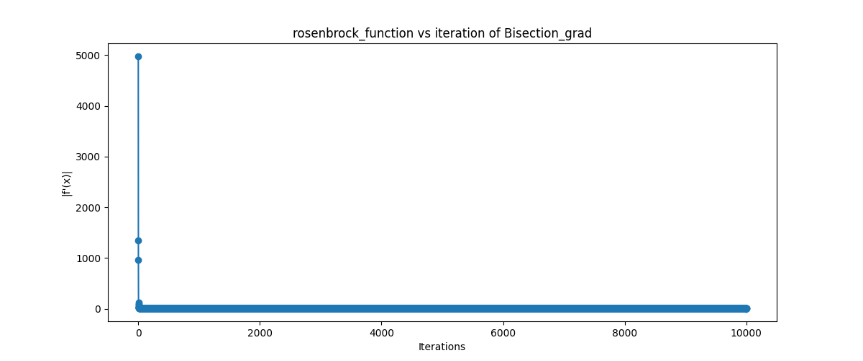
initial point [0,0,0,0] at initial point [-3,-3,-3,-3]

damped condition the f(x) is not Converge

Plot |f′(x)| vs iterations Rosenbrock Function



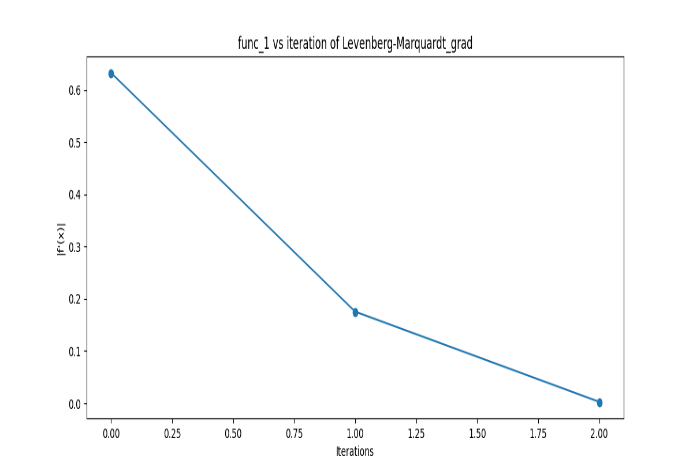
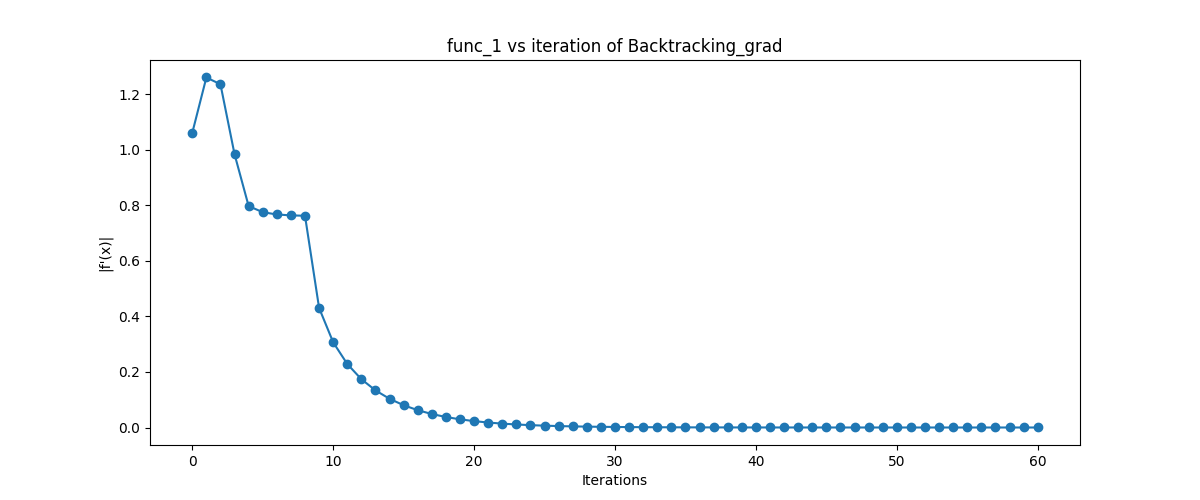
initial point [-2,2,2,2] initial point [3,3,3,3]



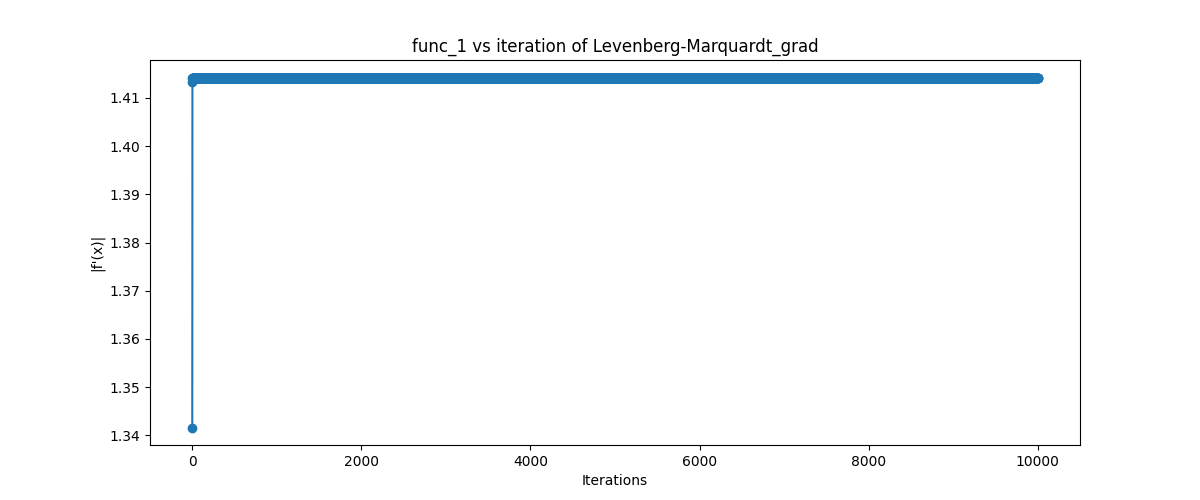
initial point [2,2,2,2]

at backtracking the function is not Converge

Plot |f′(x)| vs iterations Root of Square Function(func\_1)



initial point [-3.5, 0.5] initial point [-0.5, 0.5]



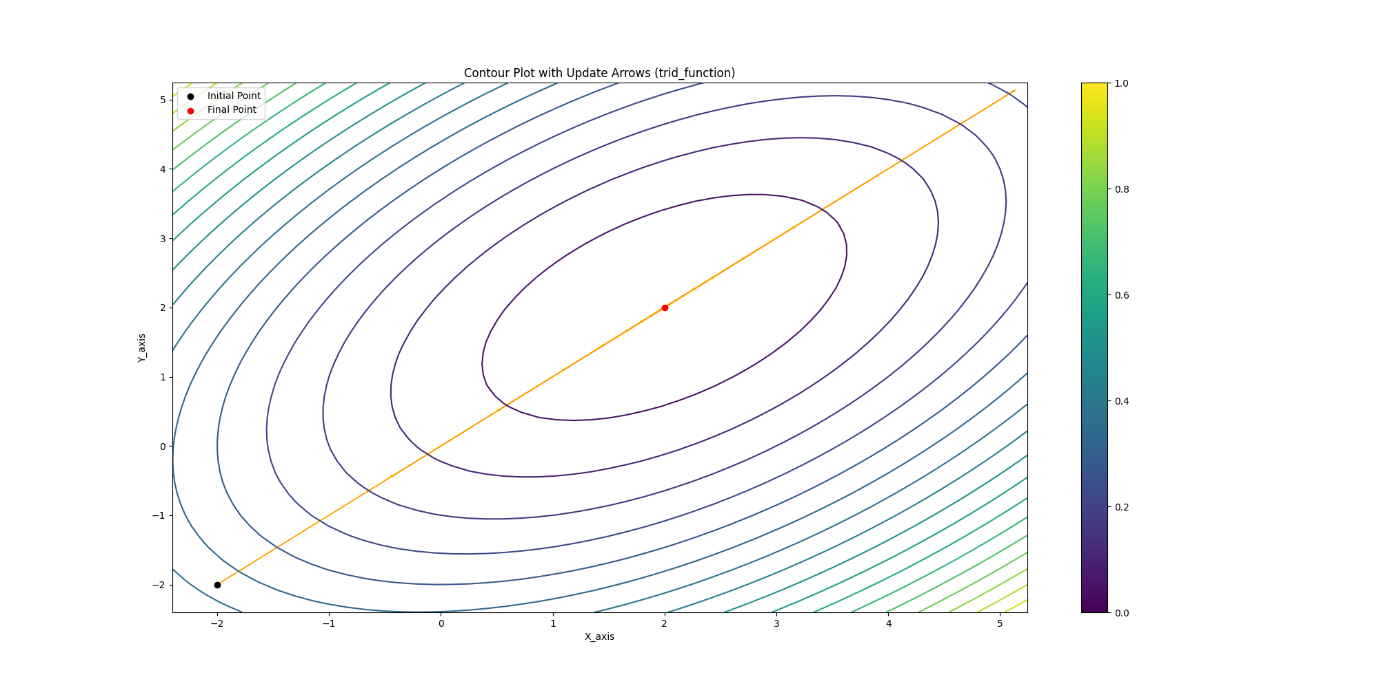
initial point [-3, 3] Levenberg-Marquardt

condition the function is not Converge

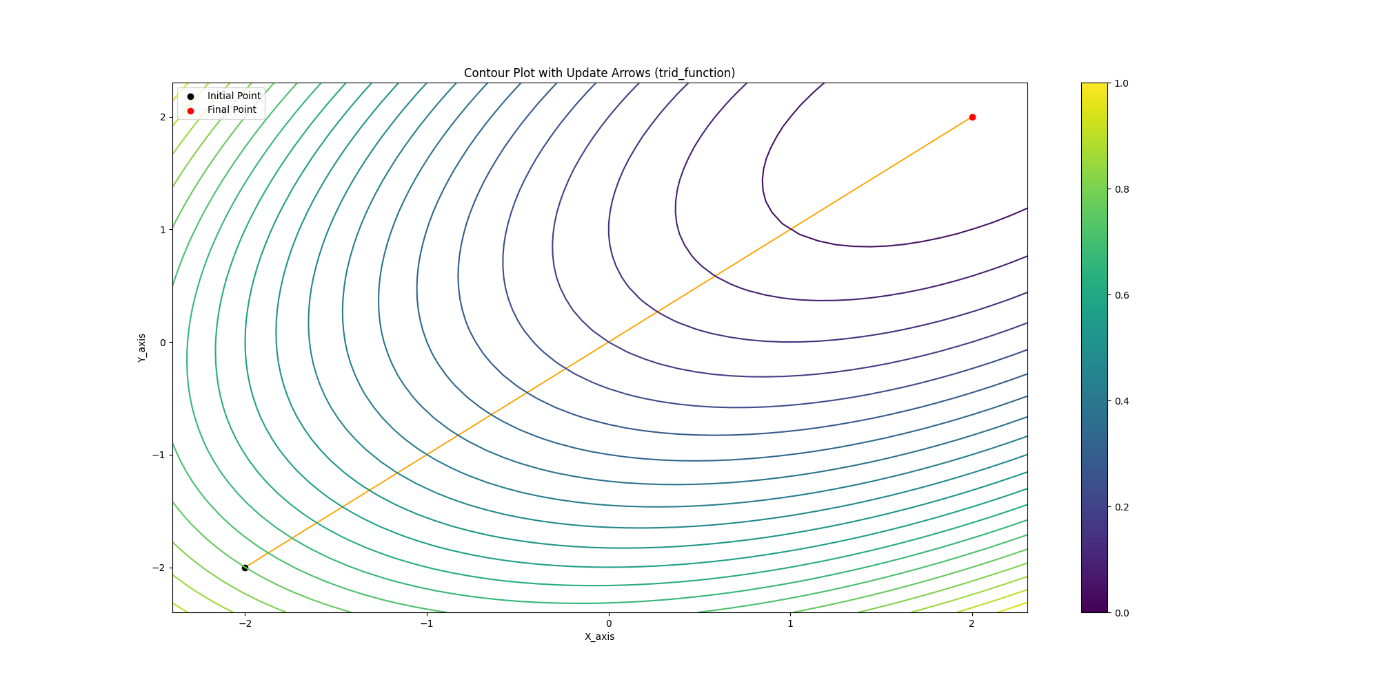
**Make a contour plot with arrows indicating the direction of updates for**

**all 2-d functions.**

**Trid Function**

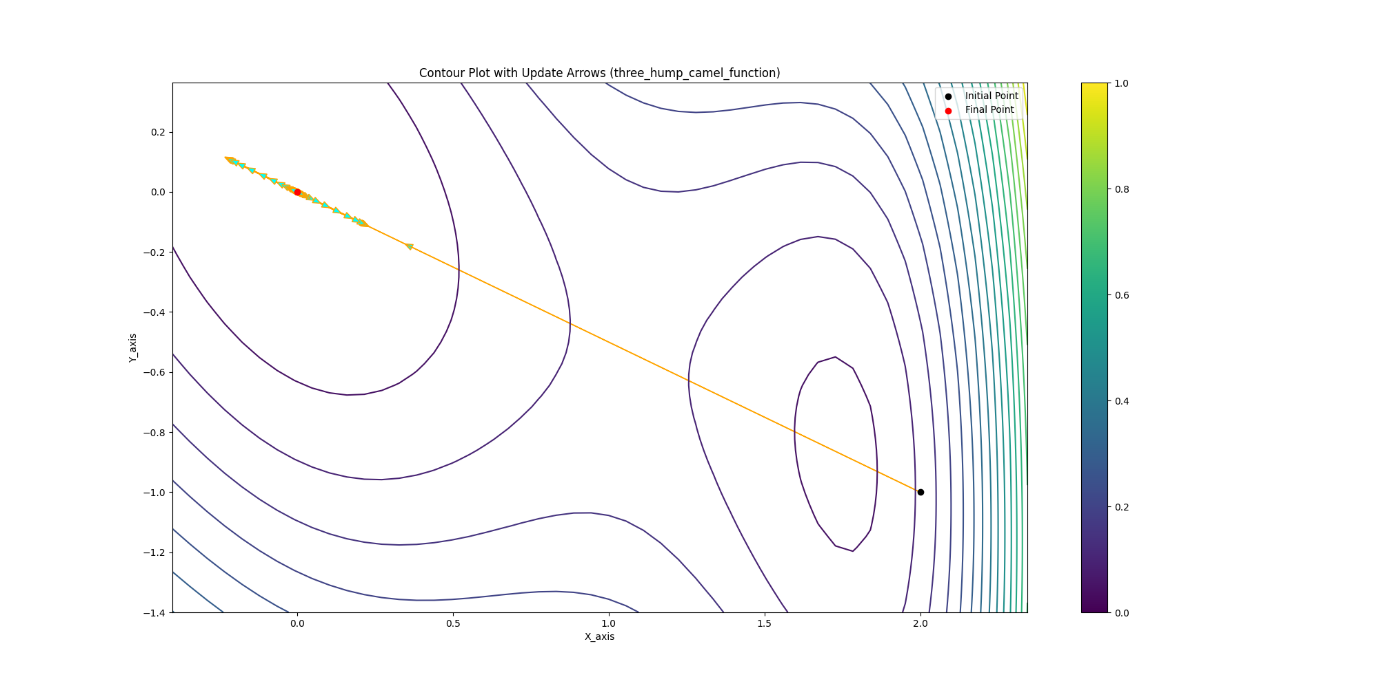
****

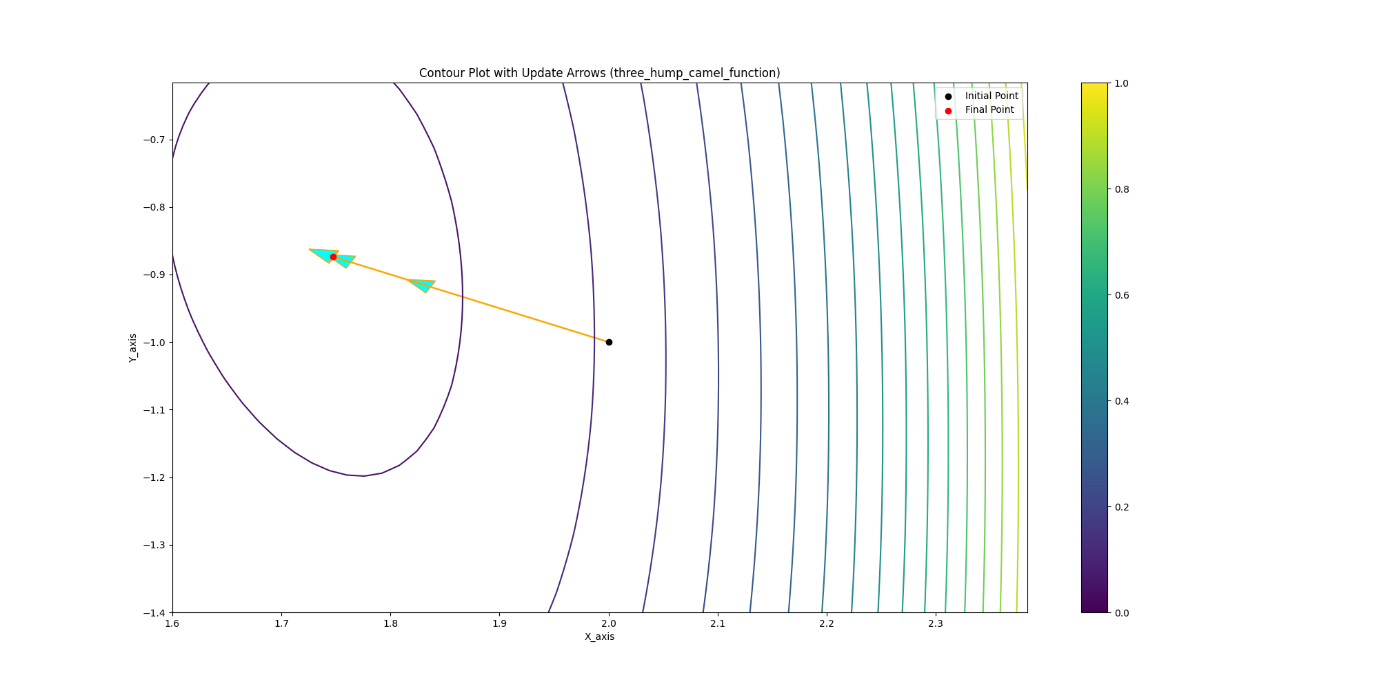
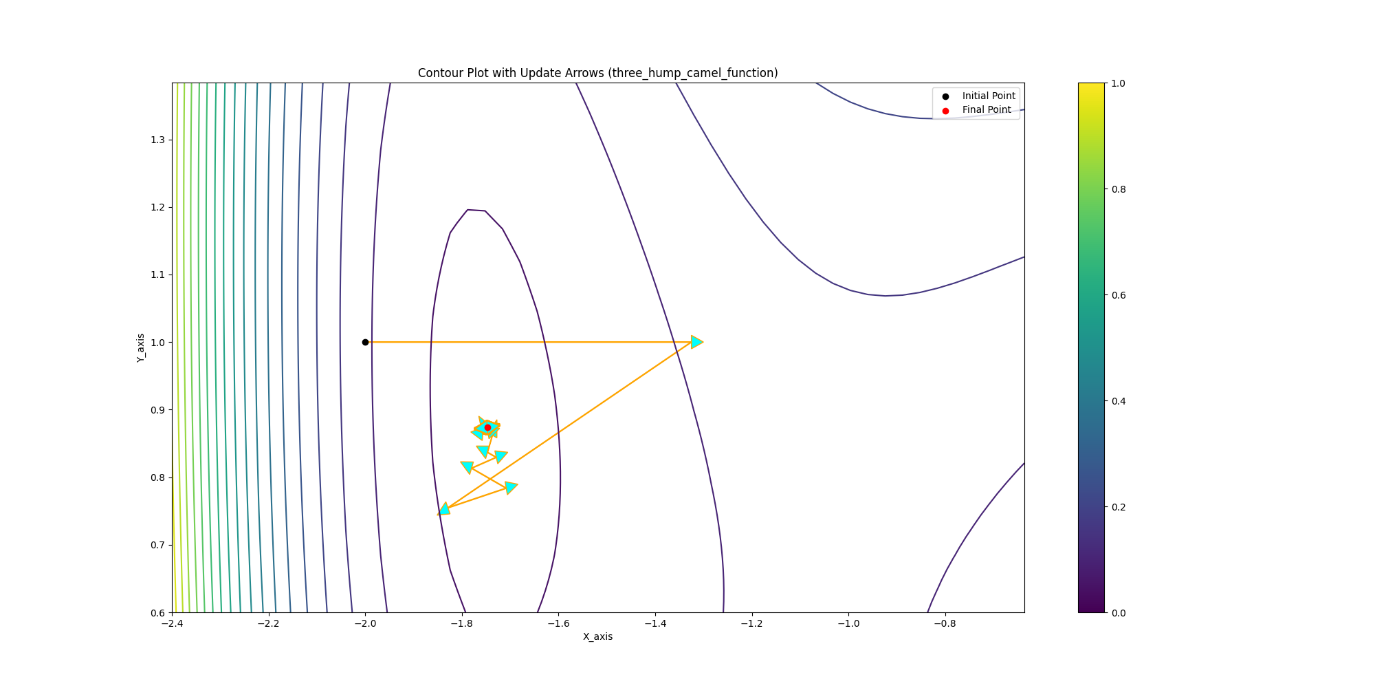
Initial points [-2,-2] at combined condition, final point [2,2]

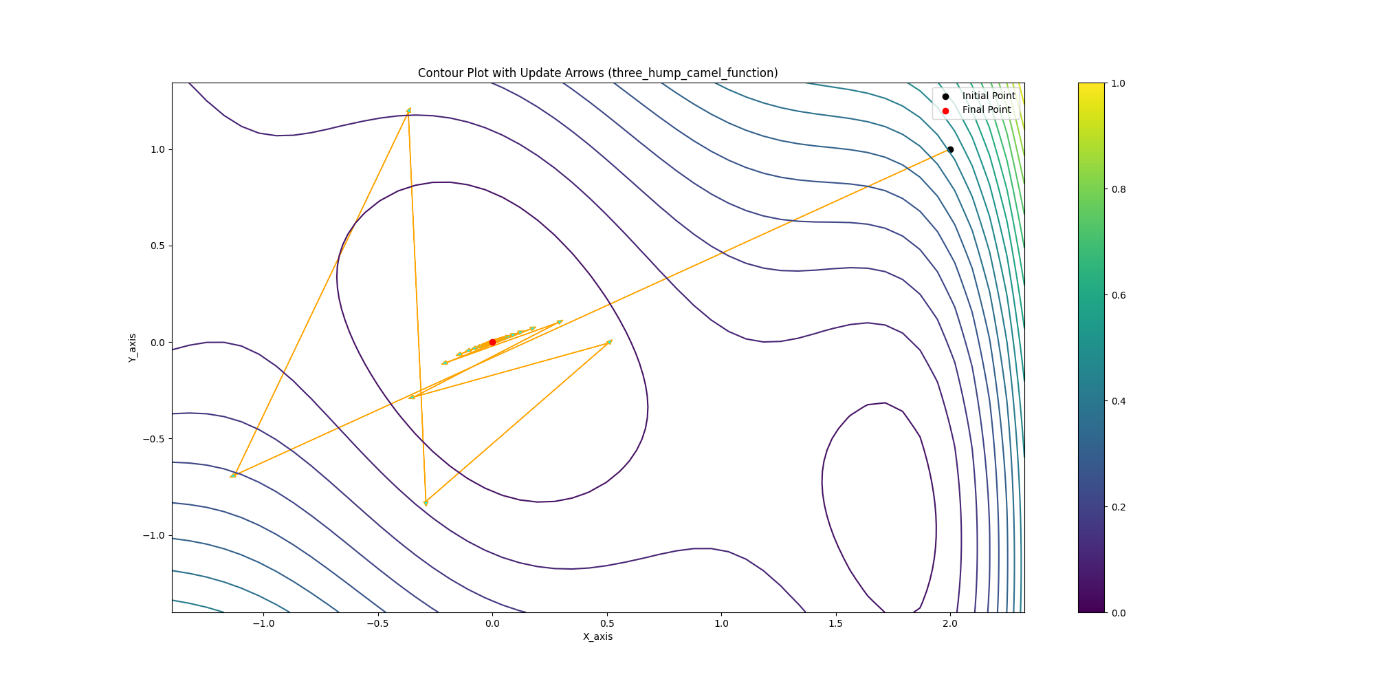


Initial points [-2,-2] at bisection condition, final point [2,2]

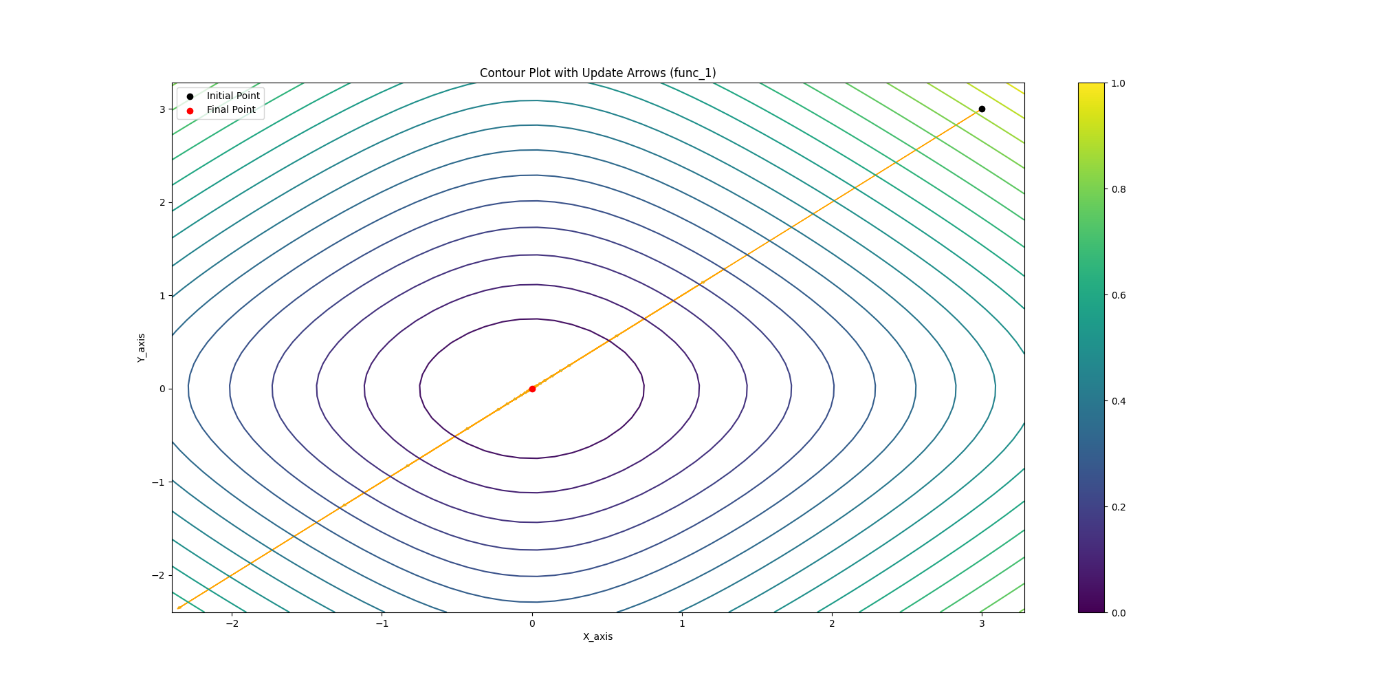
**Three Hump Camel**

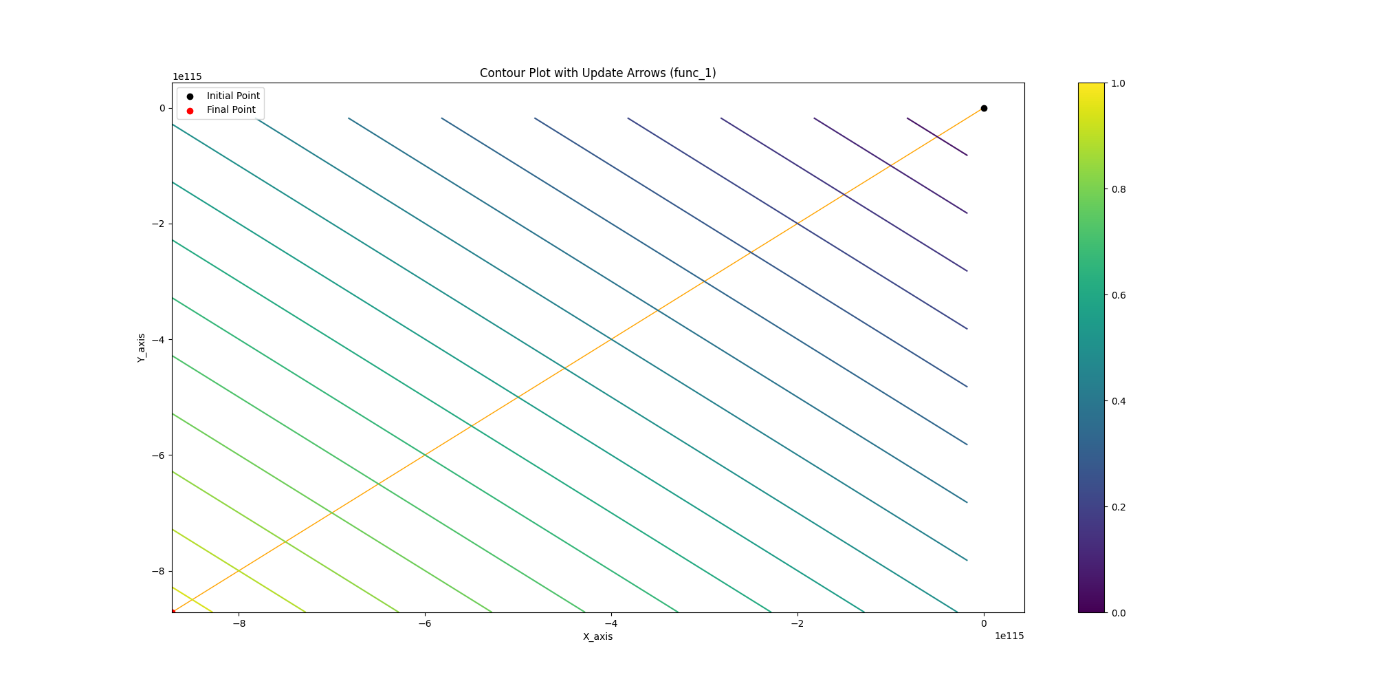


Initial points [2,-1] at combined condition, final point [0,0] Initial points [2,-1] at damped condition, final point [1.748,-0.874] Initial points [-2,1] at Bisection condition, final point [0.874, -1.748]

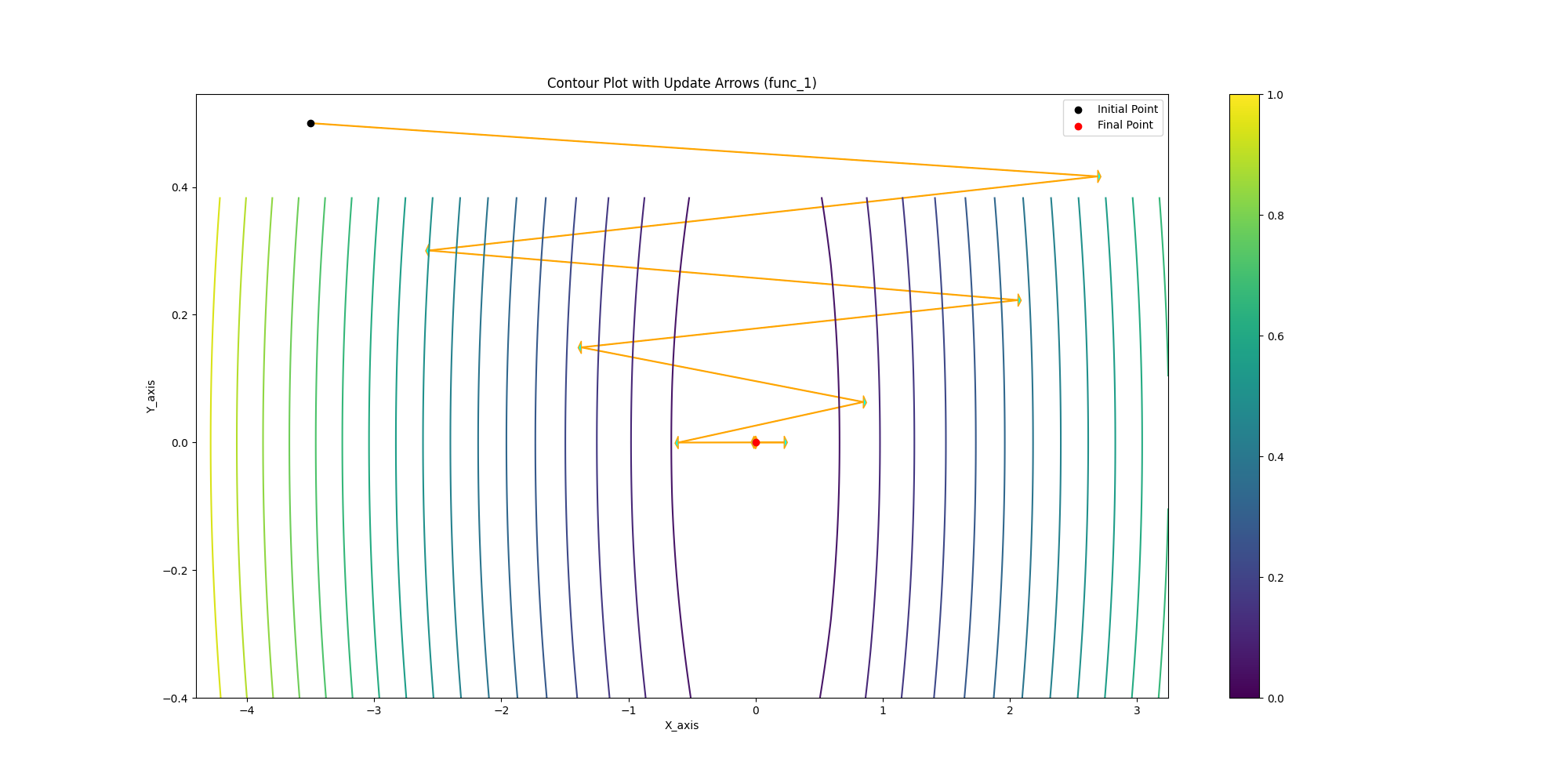


Initial points [2,1] at Backtracking condition, final point [0, 0]

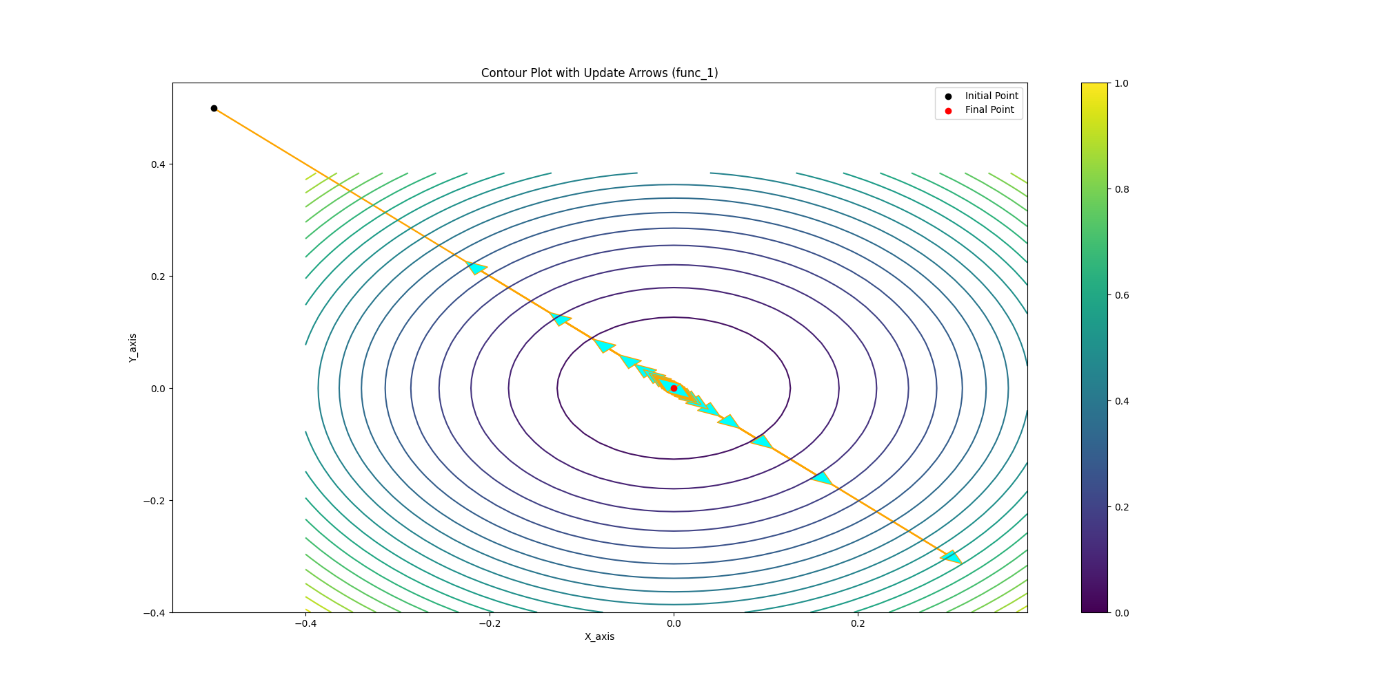
**Root of Square Function(func\_1)**

Initial points [3,3] at combined condition, final point [0, 0]

Initial points [-3.5,0.5] at Levenberg\_Marquart condition

****

Initial points [-3.5,0.5] at Levenberg\_Marquart condition , final point [0,0]



Initial points [-0.5,0.5] at Backtracking condition , final point [0,0]