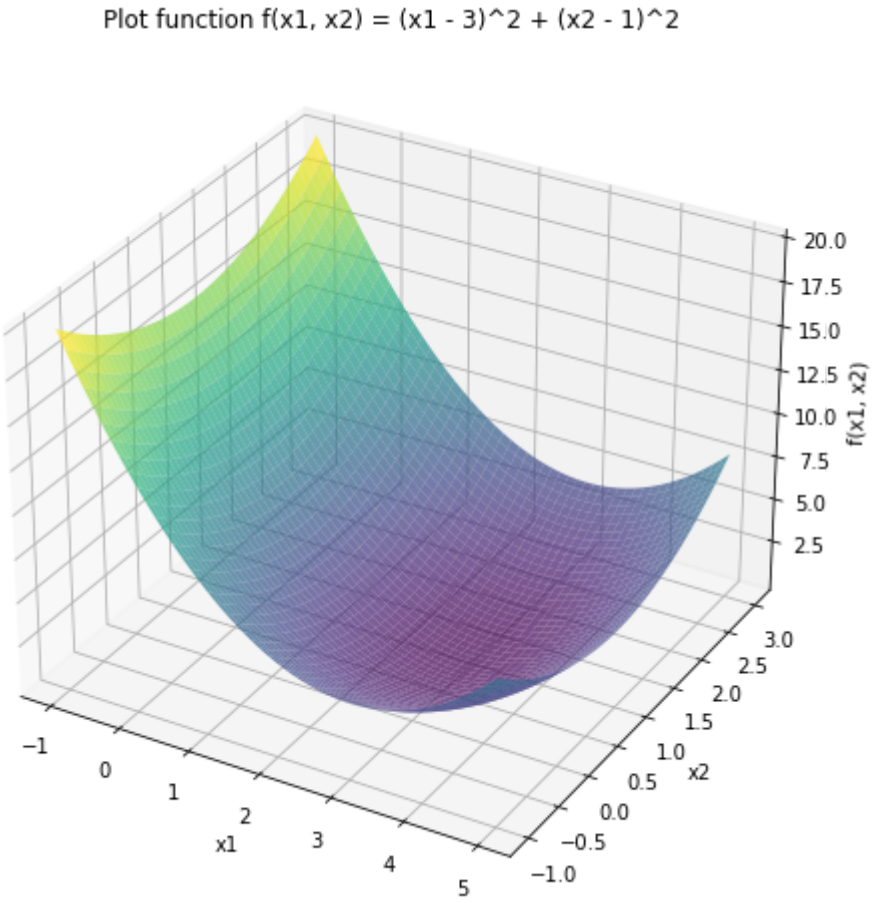
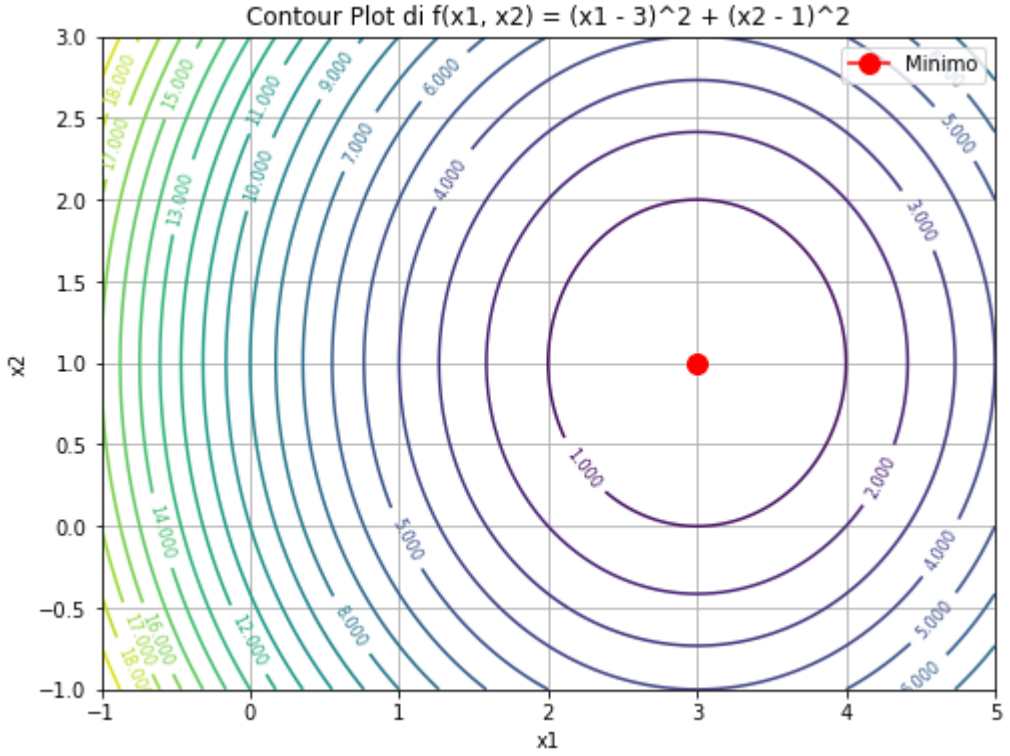
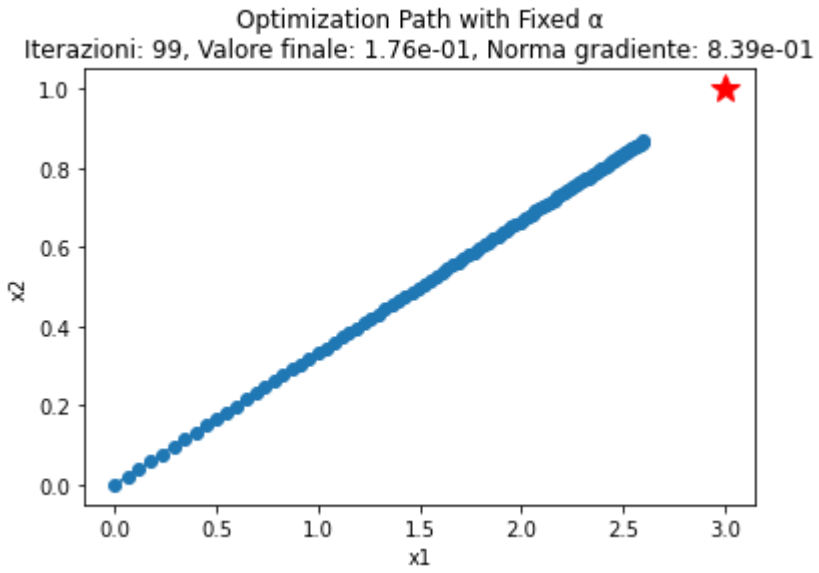
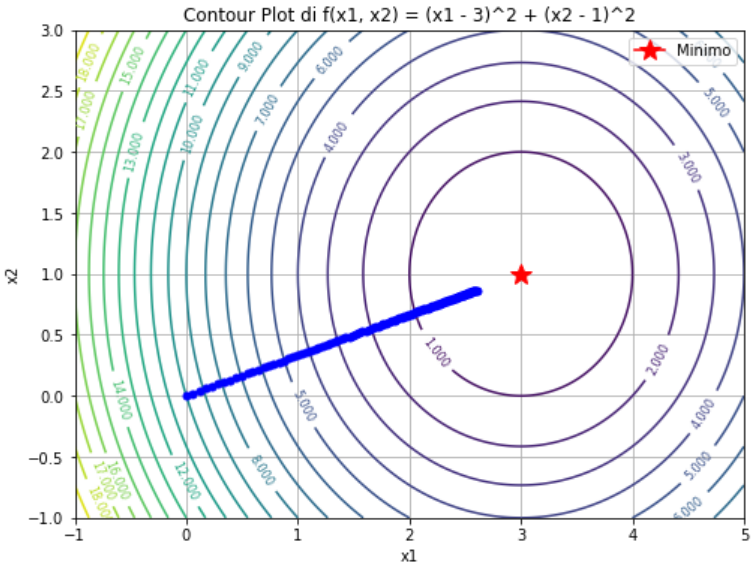
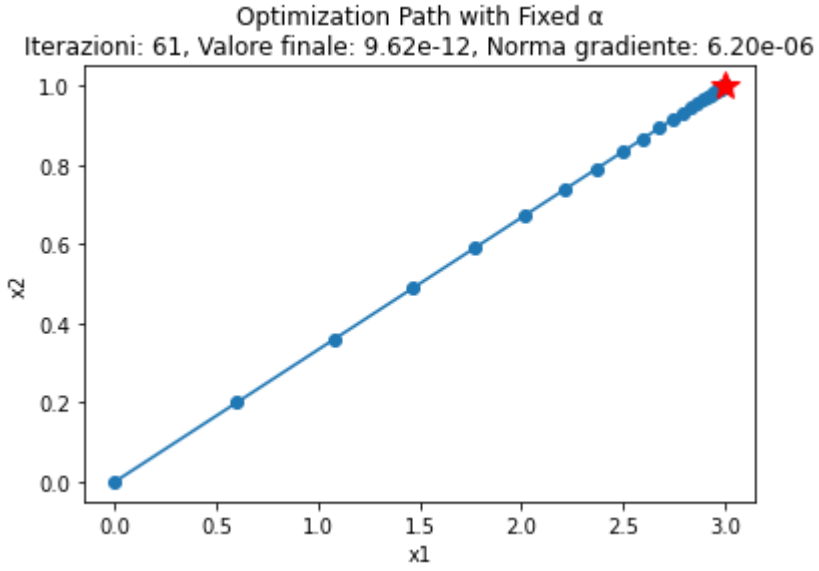
HomeWork 3_1

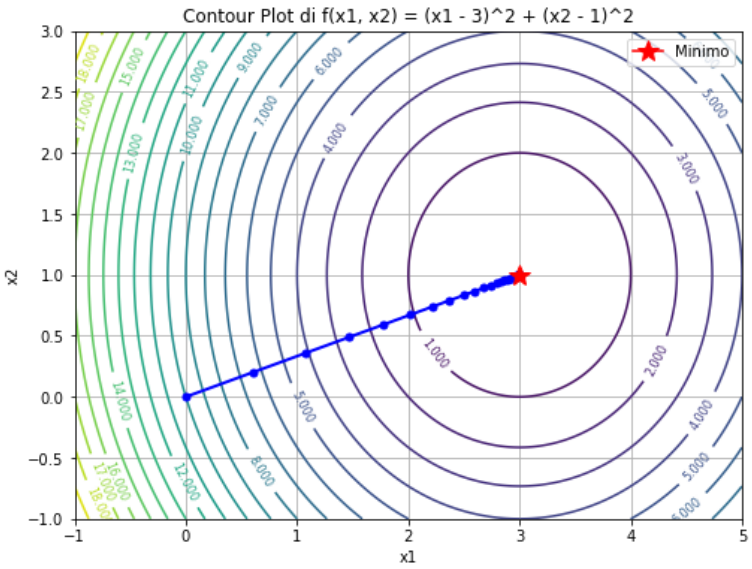

Plot function f(x1, x2) = (x1 - 3)^2 + (x2 - 1)^2
f(x1, x2)
x1
x2

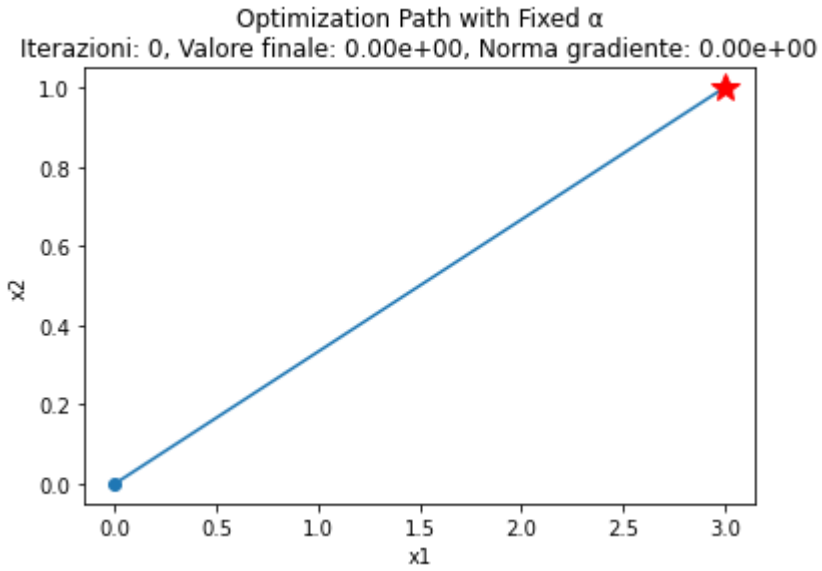

Contour Plot di f(x1, x2) = (x1 - 3)^2 + (x2 - 1)^2
Minimo
x1
x2

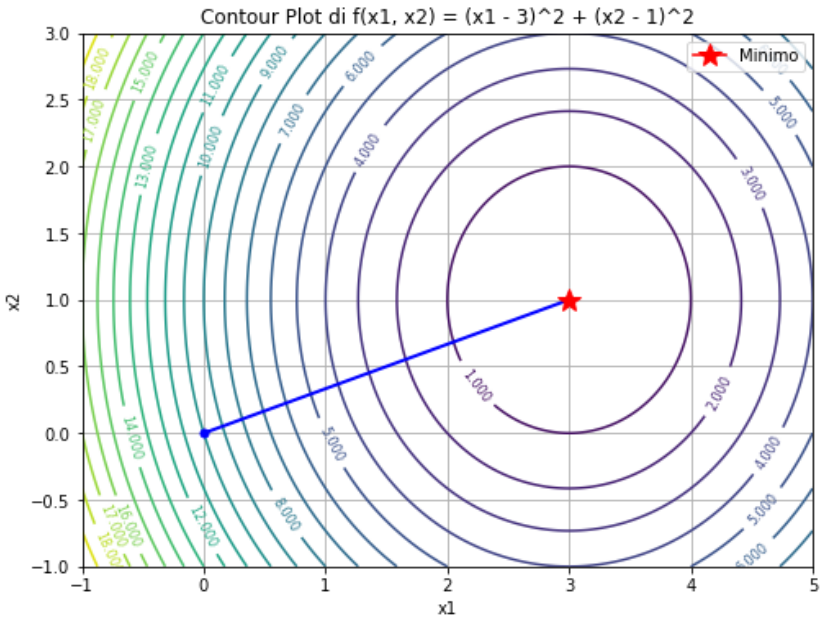

Optimization Path with Fixed α
Iterazioni: 99, Valore finale: 1.76e-01, Norma gradiente: 8.39e-01
x1
x2

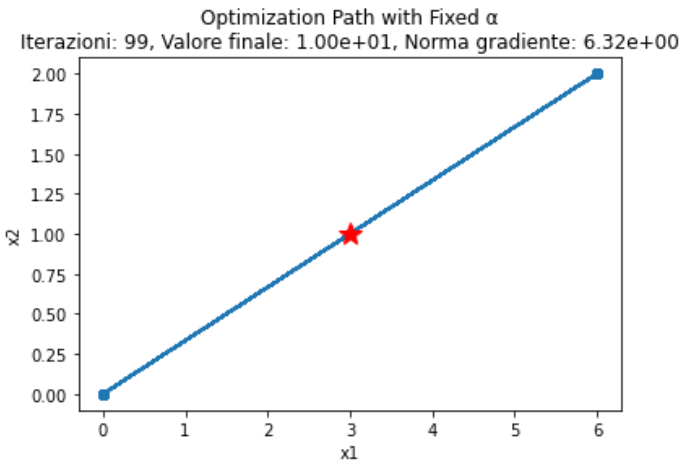

Contour Plot di f(x1, x2) = (x1 - 3)^2 + (x2 - 1)^2
Minimo
x1
x2

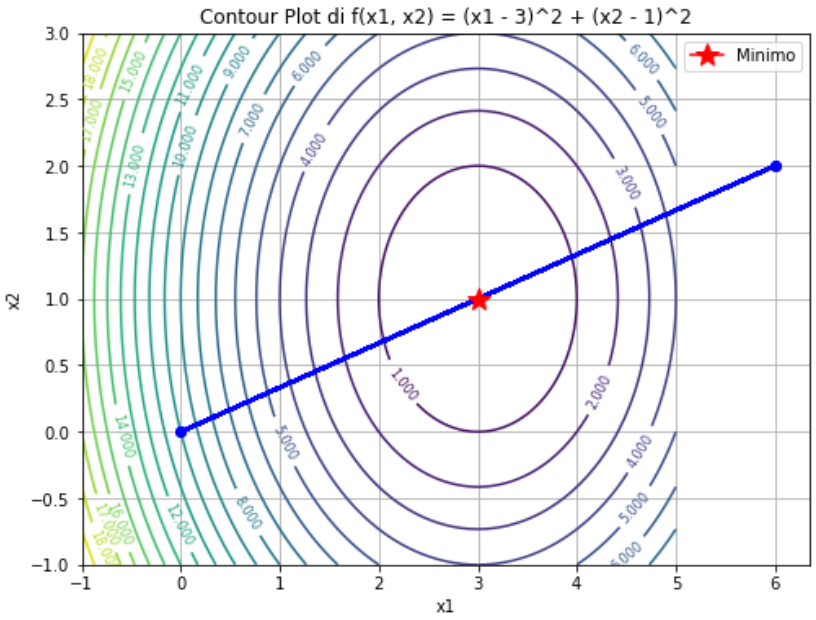

Optimization Path with Fixed α
Iterazioni: 61, Valore finale: 9.62e-12, Norma gradiente: 6.20e-06
x1
x2

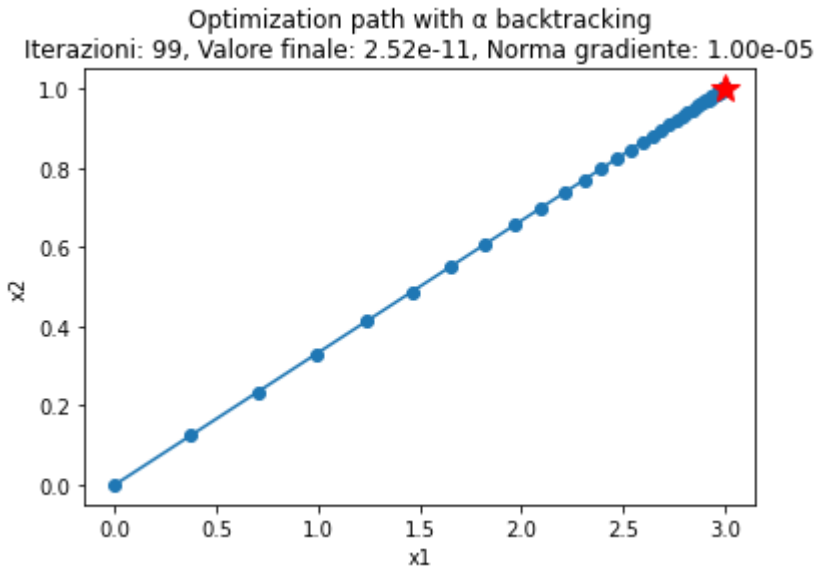

Contour Plot di f(x1, x2) = (x1 - 3)^2 + (x2 - 1)^2
Minimo
x1
x2

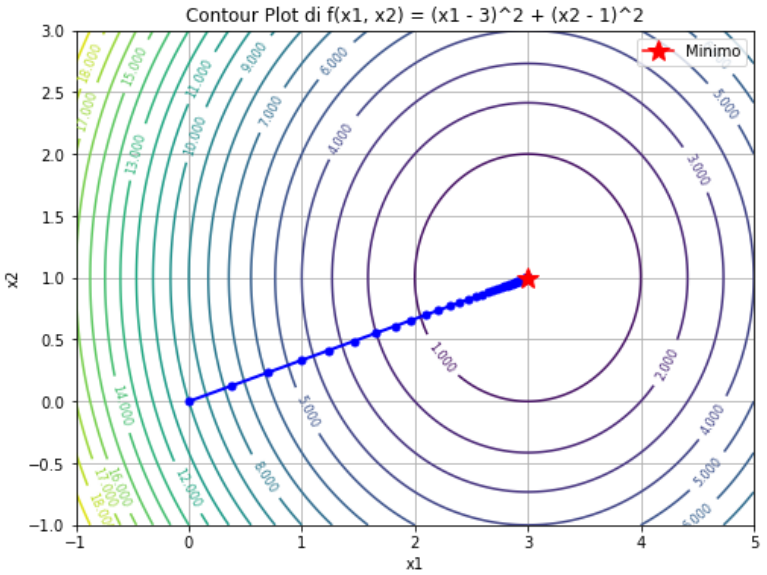

Optimization Path with Fixed α
Iterazioni: 0, Valore finale: 0.00e+00, Norma gradiente: 0.00e+00
x1
x2

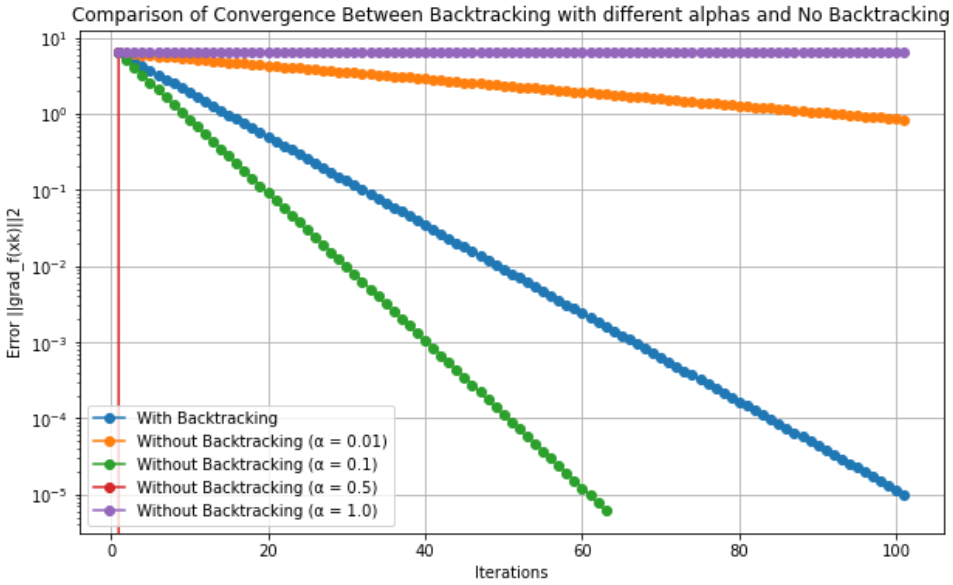

Contour Plot di f(x1, x2) = (x1 - 3)^2 + (x2 - 1)^2
Minimo
x1
x2


Optimization Path with Fixed α
Iterazioni: 99, Valore finale: 1.00e+01, Norma gradiente: 6.32e+00
x1
x2

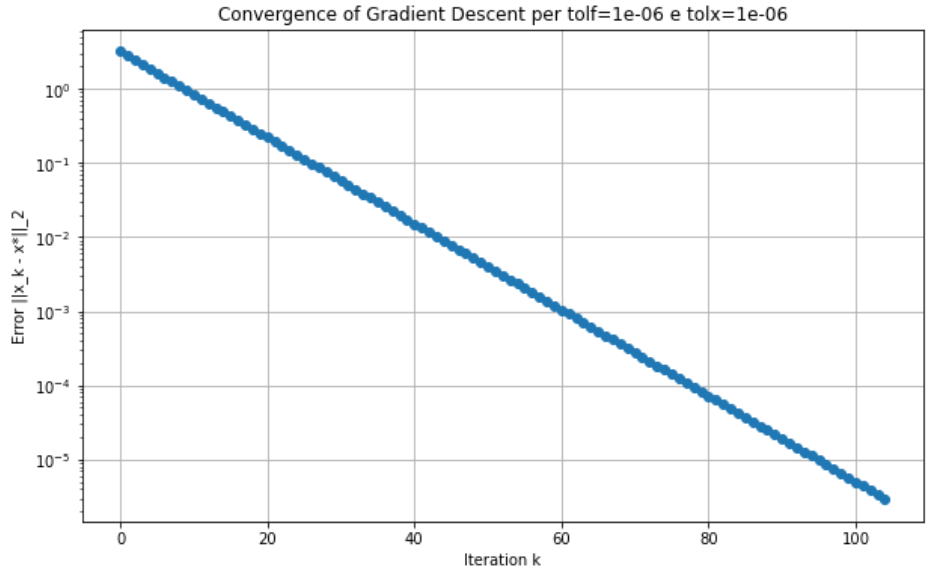

Contour Plot di f(x1, x2) = (x1 - 3)^2 + (x2 - 1)^2
Minimo
x1
x2

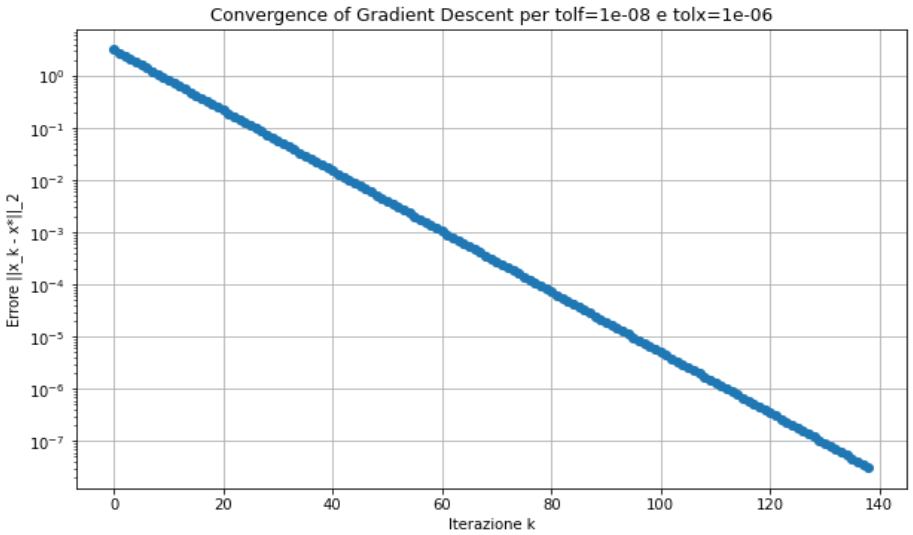

Optimization path with α backtracking
Iterazioni: 99, Valore finale: 2.52e-11, Norma gradiente: 1.00e-05
x1
x2

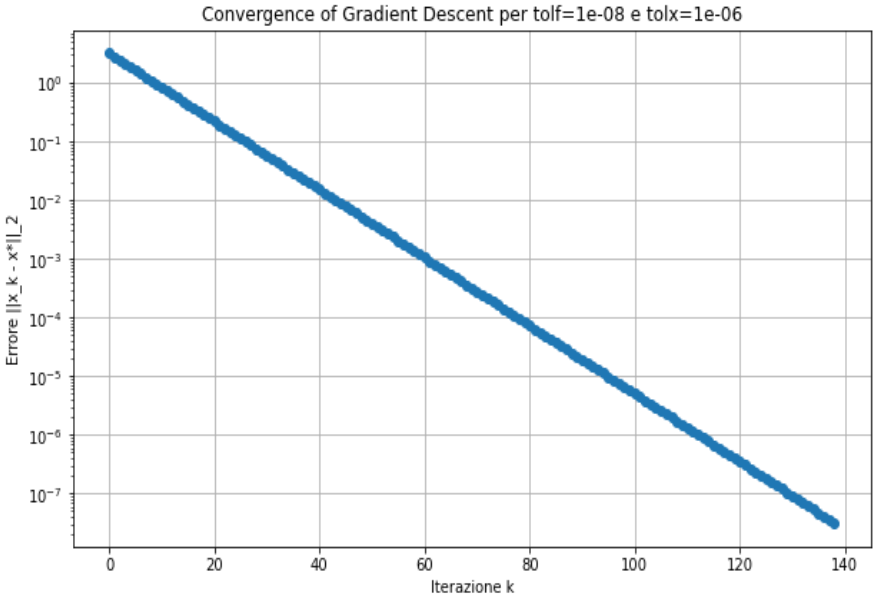

Contour Plot di f(x1, x2) = (x1 - 3)^2 + (x2 - 1)^2
Minimo
x1
x2

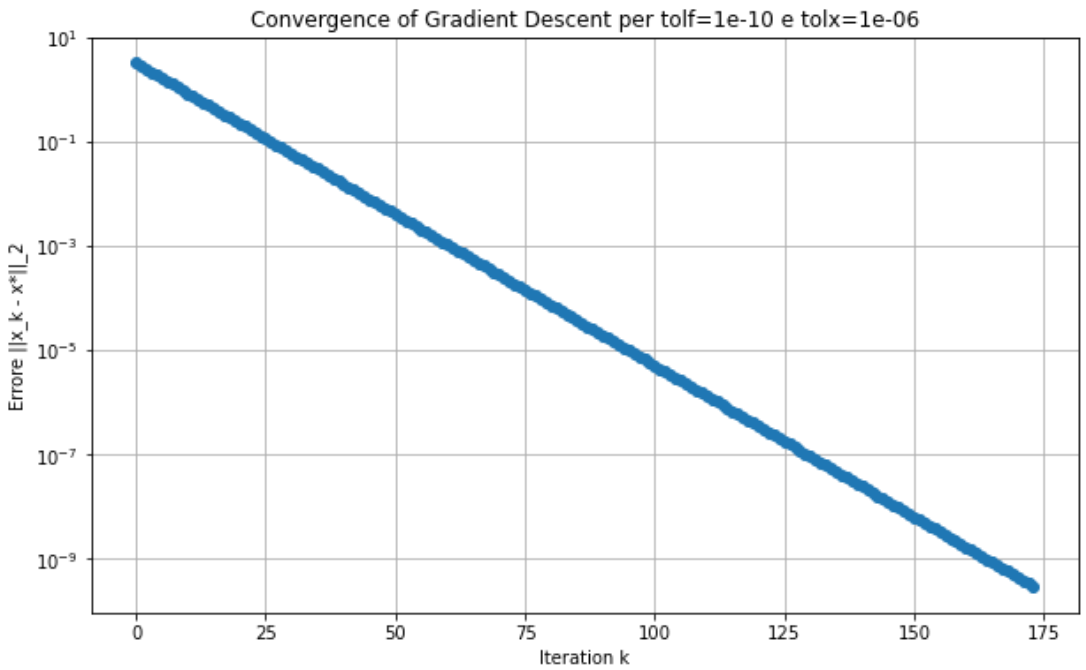

Comparison of Convergence Between Backtracking with different alphas and No Backtracking
With Backtracking
Without Backtracking (α = 0.01)
Without Backtracking (α = 0.1)
Without Backtracking (α = 0.5)
Without Backtracking (α = 1.0)
Error ||grad_f(xk)||2
Iterations

tolf: 1e-06, tolx: 1e-06 -> Iterazioni: 103, Valore finale: 8.663133929057938e-12, Cambiamento in x: 4.2047431734906344e-07
tolf: 1e-06, tolx: 1e-08 -> Iterazioni: 103, Valore finale: 8.663133929057938e-12, Cambiamento in x: 4.2047431734906344e-07
tolf: 1e-06, tolx: 1e-10 -> Iterazioni: 103, Valore finale: 8.663133929057938e-12, Cambiamento in x: 4.2047431734906344e-07
tolf: 1e-08, tolx: 1e-06 -> Iterazioni: 137, Valore finale: 9.867852111454736e-16, Cambiamento in x: 4.487591330278101e-09
tolf: 1e-08, tolx: 1e-08 -> Iterazioni: 137, Valore finale: 9.867852111454736e-16, Cambiamento in x: 4.487591330278101e-09
tolf: 1e-08, tolx: 1e-10 -> Iterazioni: 137, Valore finale: 9.867852111454736e-16, Cambiamento in x: 4.487591330278101e-09
tolf: 1e-10, tolx: 1e-06 -> Iterazioni: 172, Valore finale: 8.605713131993931e-20, Cambiamento in x: 4.190808183950403e-11
tolf: 1e-10, tolx: 1e-08 -> Iterazioni: 172, Valore finale: 8.605713131993931e-20, Cambiamento in x: 4.190808183950403e-11
tolf: 1e-10, tolx: 1e-10 -> Iterazioni: 172, Valore finale: 8.605713131993931e-20, Cambiamento in x: 4.190808183950403e-11




Convergence of Gradient Descent per tolf=1e-08 e tolx=1e-06
Errore ||x_k - x*||_2
Iterazione k


Convergence of Gradient Descent per tolf=1e-08 e tolx=1e-06
Errore ||x_k - x*||_2
Iterazione k

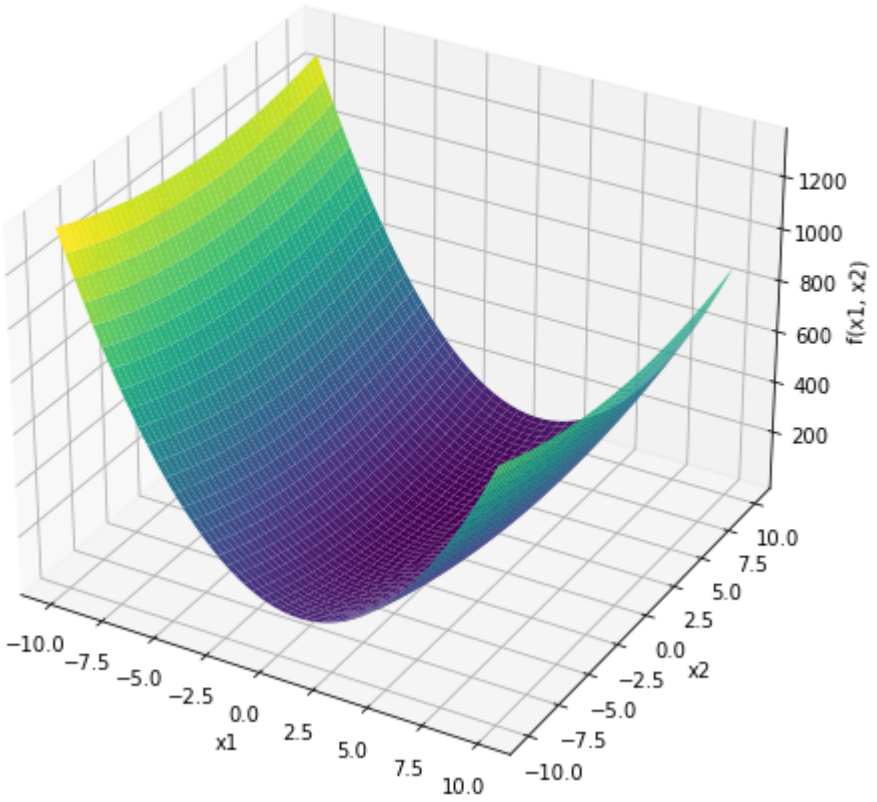


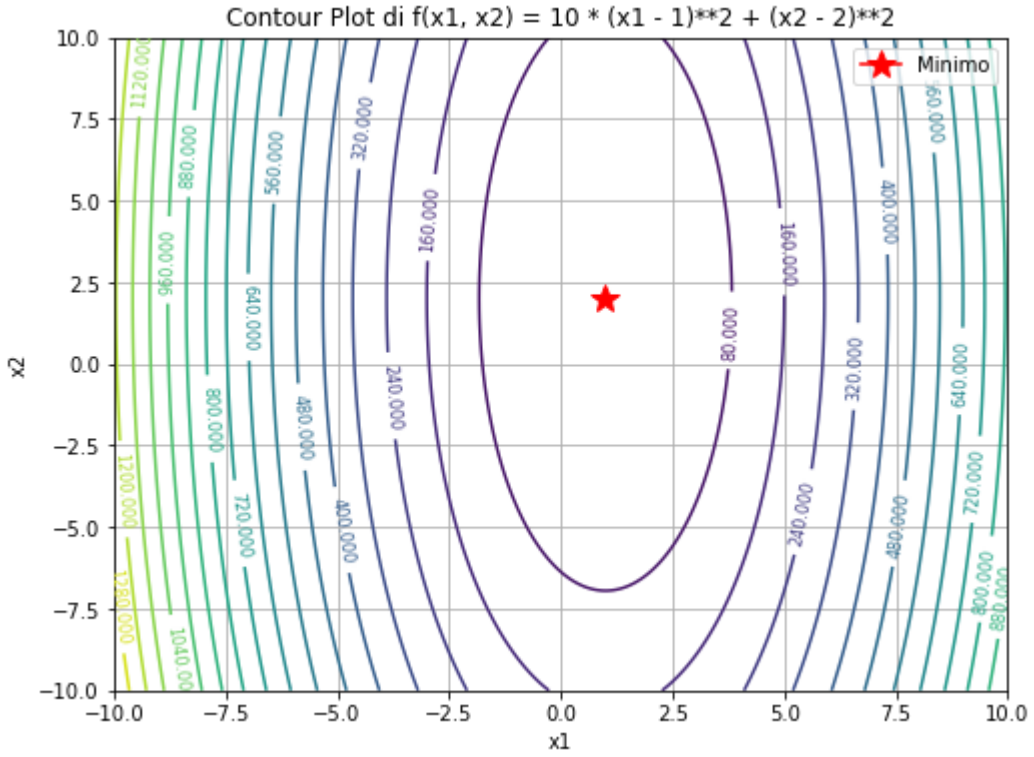
In conducting an analysis of the gradient descent algorithm's performance with varying tolerances for the objective function tolf and decision variables tolx, it was observed that the convergence of the algorithm is predominantly influenced by the tolerance on the objective function (tolf).

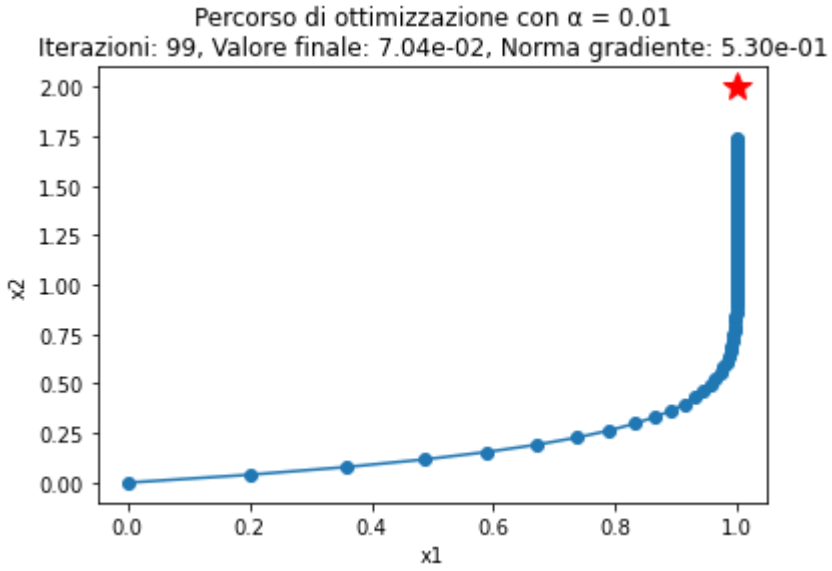
Specifically, as tolf is tightened (lowered from 1e−6 to 1e−10), the precision of the solution notably increases, as evidenced by a significant reduction in the final objective function value.
However, this increase in precision comes with a corresponding rise in the number of iterations, highlighting a classic trade-off between computational cost and solution accuracy.

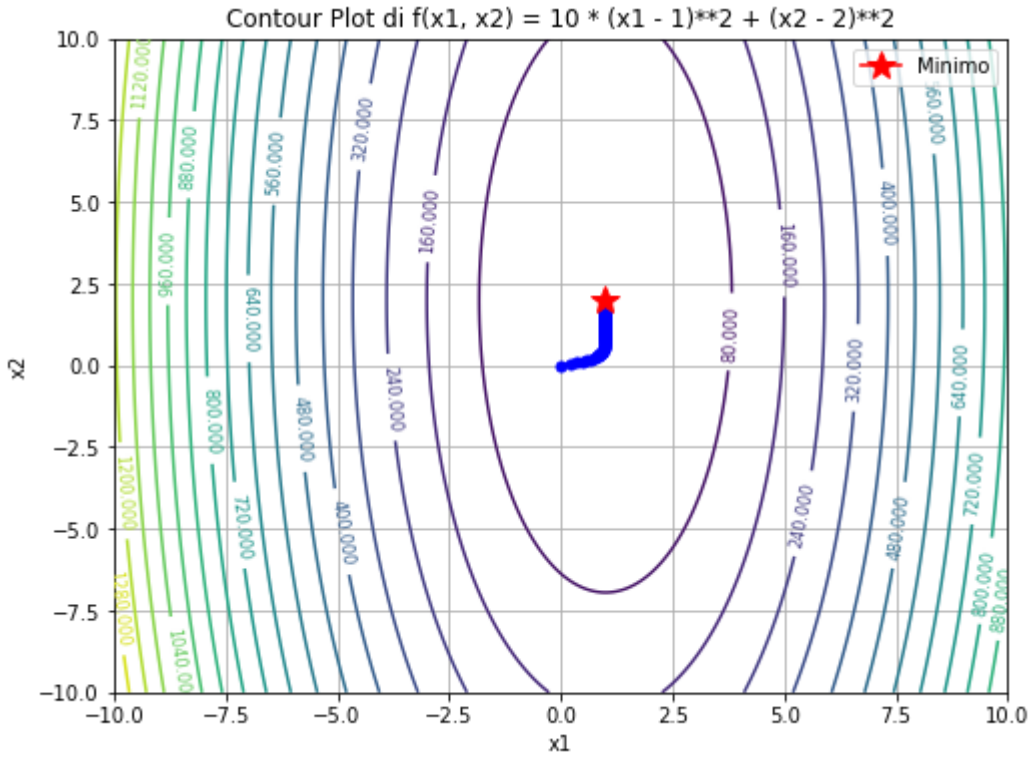
Notably, changes in tolx did not materially affect the number of iterations or the final value of the objective function, suggesting that the convergence criterion based on the gradient norm (tolf) is the primary determinant in the stopping condition of the algorithm under the tested scenarios.

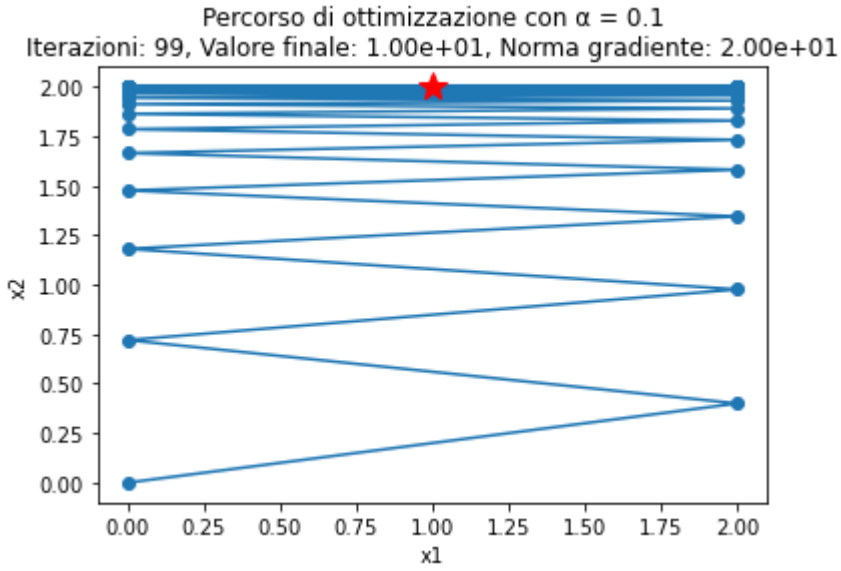
Plot of the function f(x1, x2) = 10 * (x1 - 1)**2 + (x2 - 2)**2

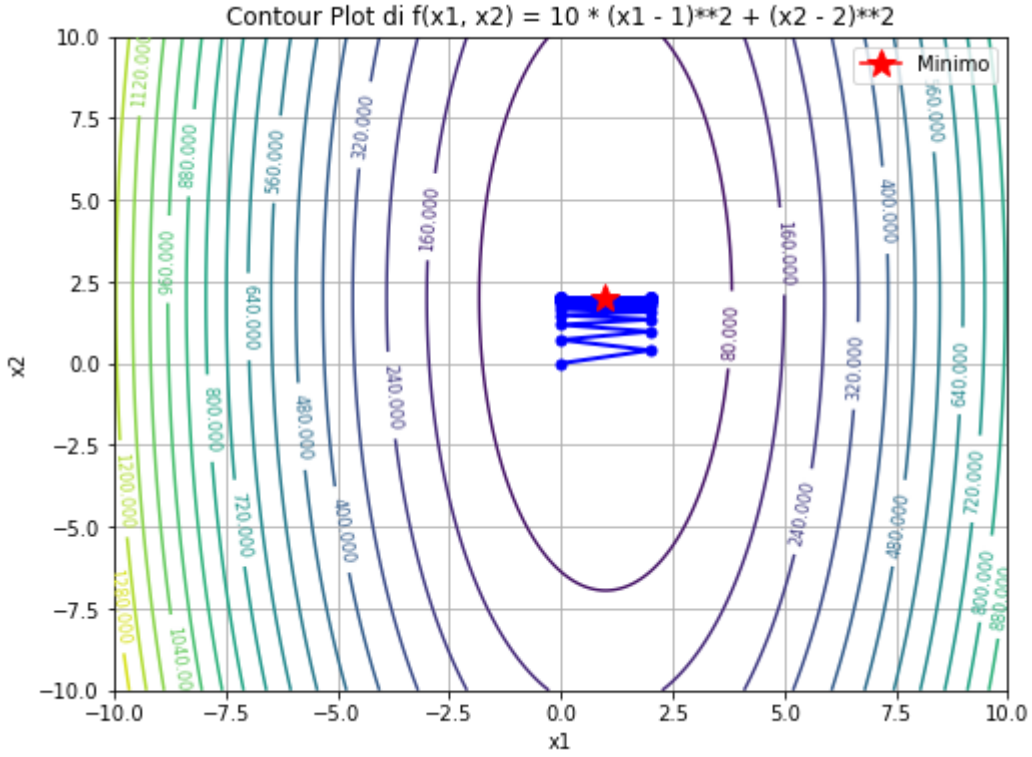


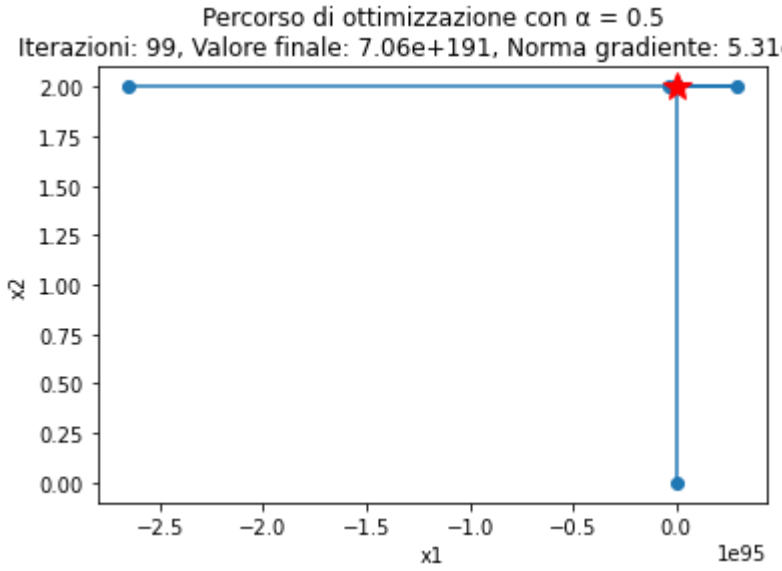


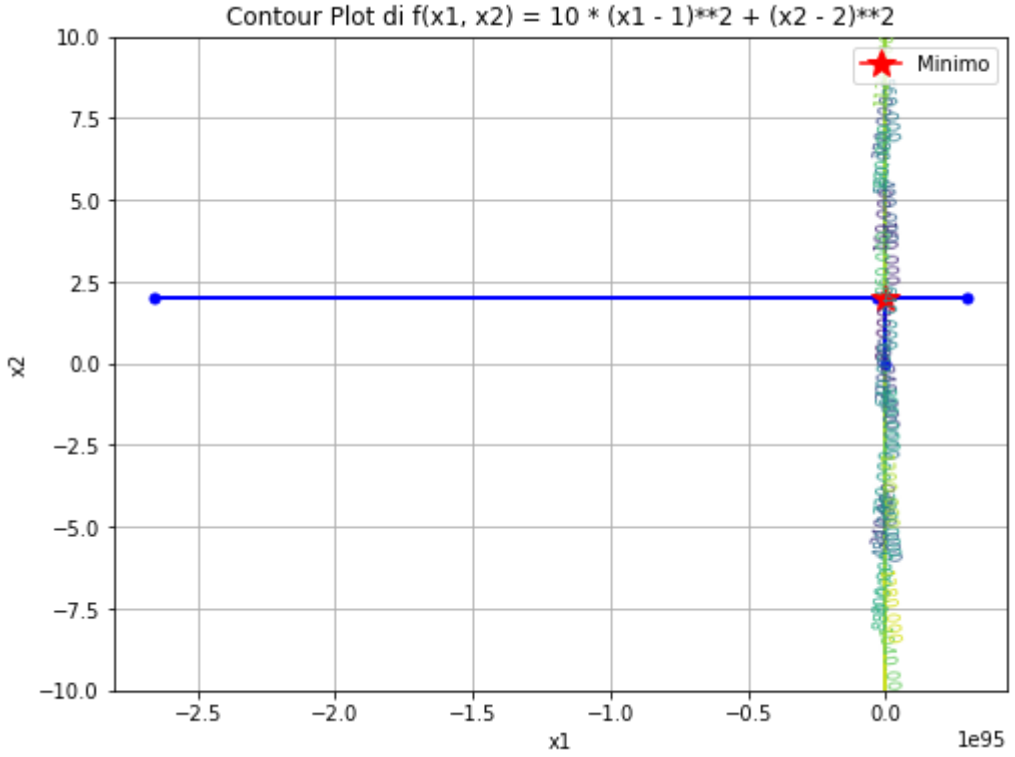

Percorso di ottimizzazione con α = 0.01
Iterazioni: 99, Valore finale: 7.04e-02, Norma gradiente: 5.30e-01
x1
x2

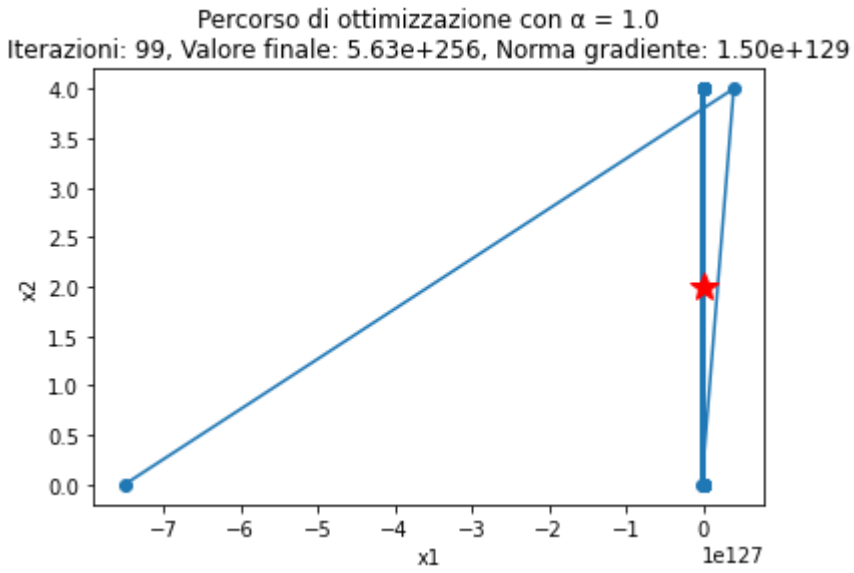

Contour Plot di f(x1, x2) = 10 * (x1 - 1)**2 + (x2 - 2)**2
Minimo
x1
x2

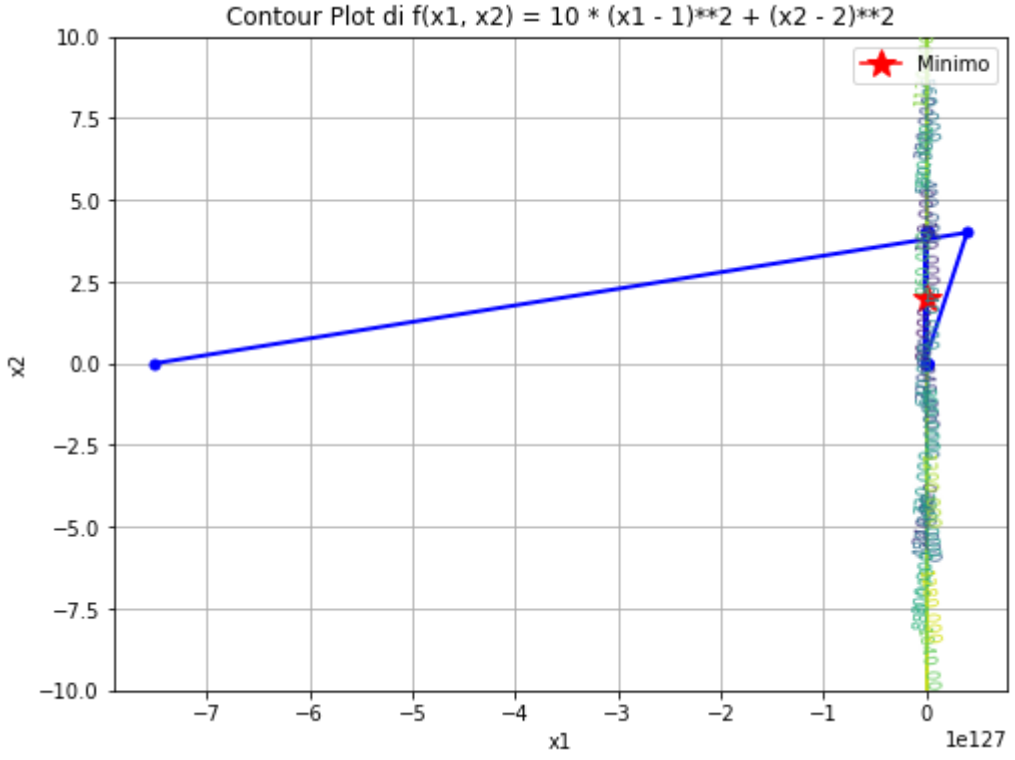

Percorso di ottimizzazione con α = 0.1
Iterazioni: 99, Valore finale: 1.00e+01, Norma gradiente: 2.00e+01
x1
x2

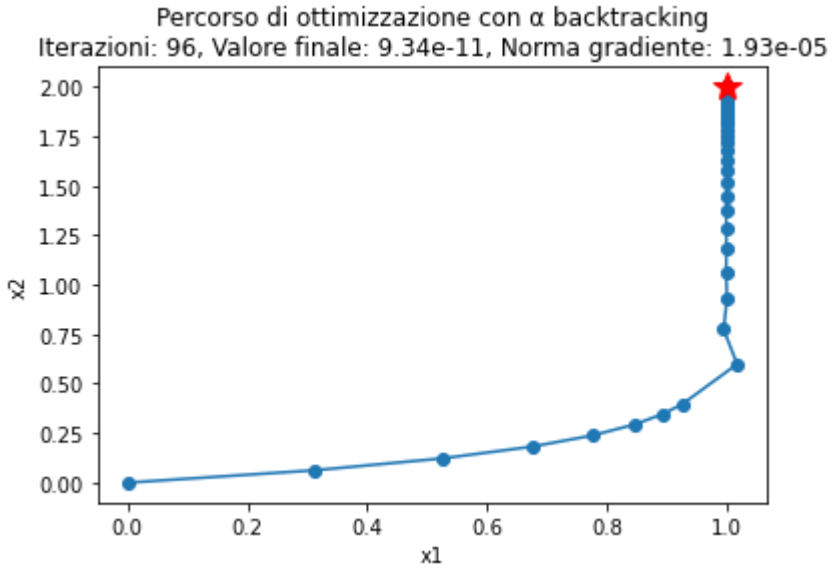

Contour Plot di f(x1, x2) = 10 * (x1 - 1)**2 + (x2 - 2)**2
Minimo
x1
x2

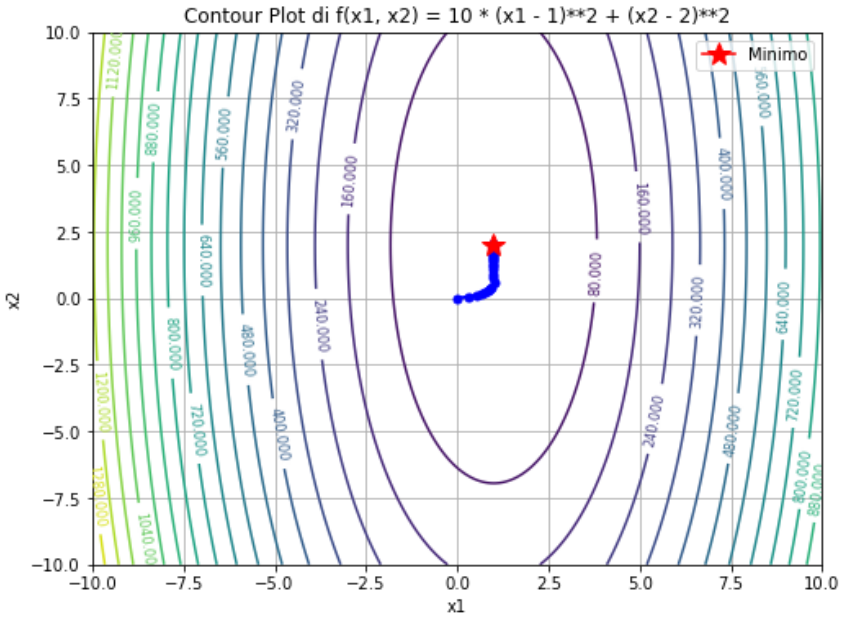

Percorso di ottimizzazione con α = 0.5
Iterazioni: 99, Valore finale: 7.06e+191, Norma gradiente: 5.31e+96
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
x2
−2.5
−2.0
−1.5
−1.0
−0.5
0.0
1e95
x1

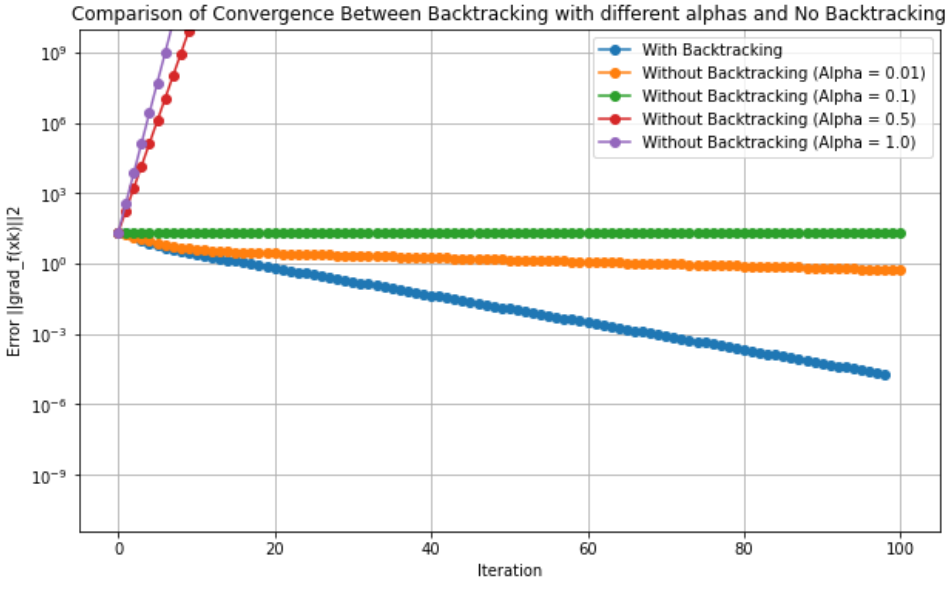

Contour Plot di f(x1, x2) = 10 * (x1 - 1)**2 + (x2 - 2)**2
Minimo
10.0
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
−10.0
x2
−2.5
−2.0
−1.5
−1.0
−0.5
0.0
1e95
x1


Percorso di ottimizzazione con α = 1.0
Iterazioni: 99, Valore finale: 5.63e+256, Norma gradiente: 1.50e+129
x2
x1
1e127

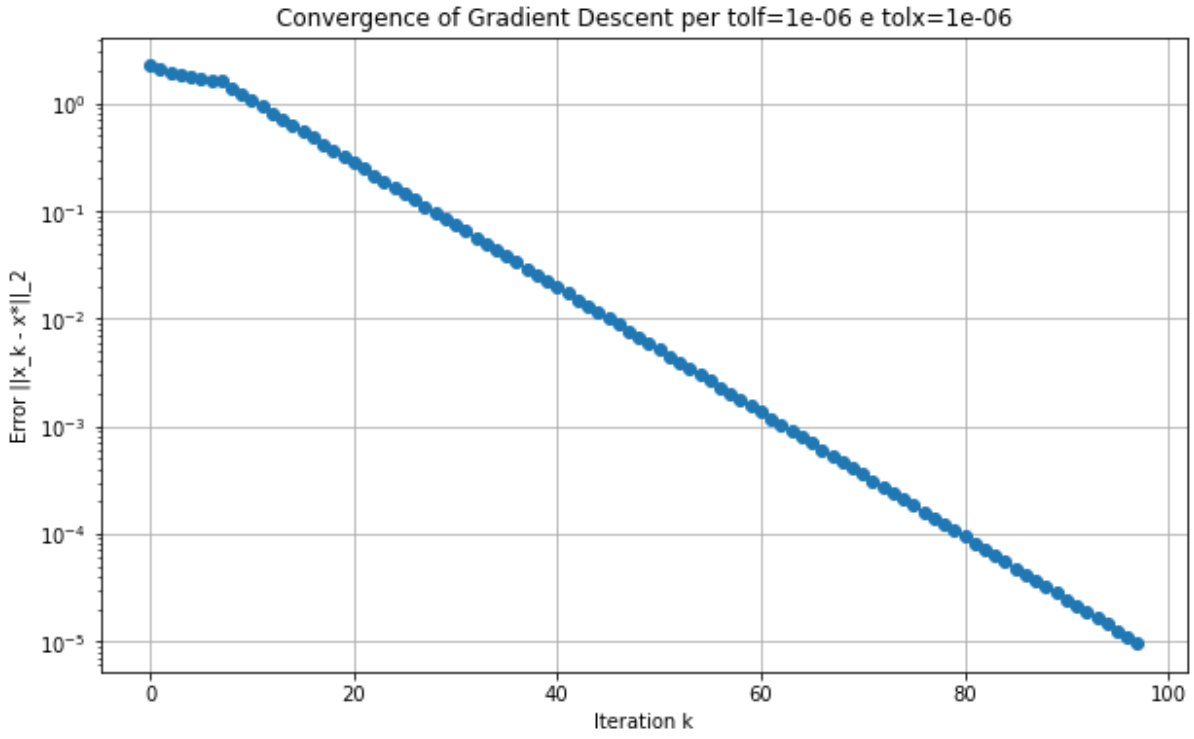

Contour Plot di f(x1, x2) = 10 * (x1 - 1)**2 + (x2 - 2)**2
Minimo
x2
x1
1e127

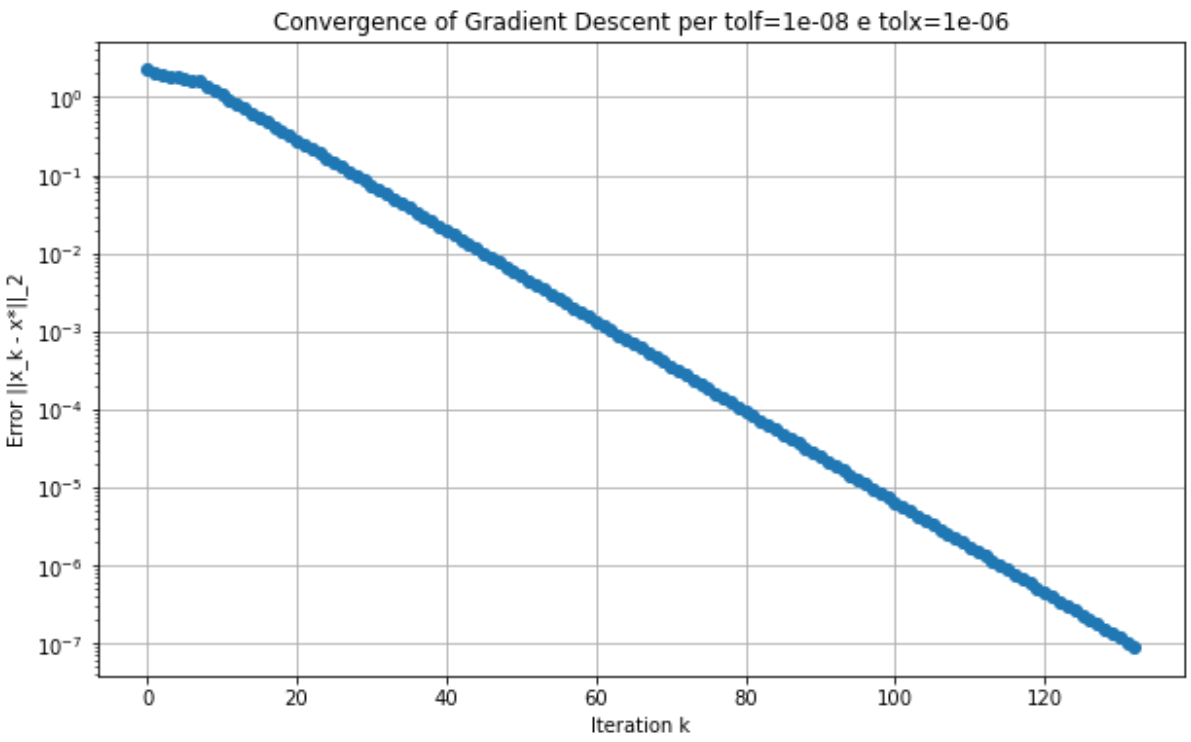

Percorso di ottimizzazione con α backtracking
Iterazioni: 96, Valore finale: 9.34e-11, Norma gradiente: 1.93e-05
x1
x2

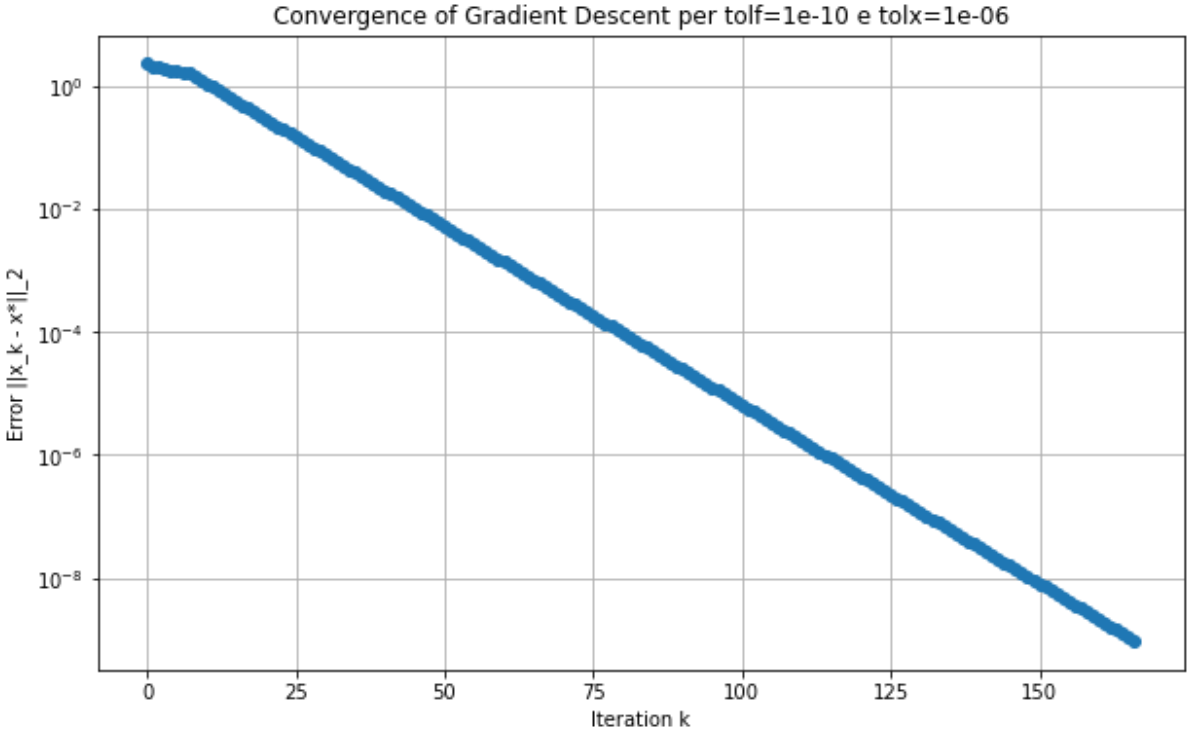

Contour Plot di f(x1, x2) = 10 * (x1 - 1)**2 + (x2 - 2)**2
Minimo
x1
x2

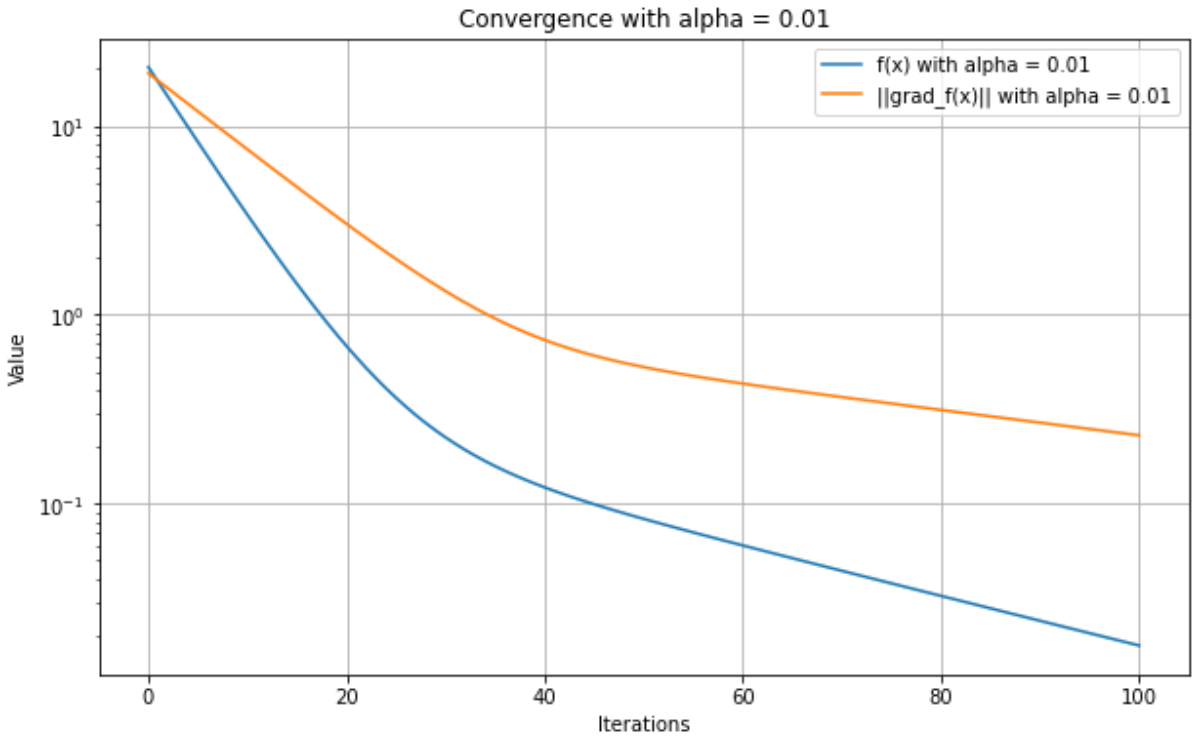

Comparison of Convergence Between Backtracking with different alphas and No Backtracking
With Backtracking
Without Backtracking (Alpha = 0.01)
Without Backtracking (Alpha = 0.1)
Without Backtracking (Alpha = 0.5)
Without Backtracking (Alpha = 1.0)
Error ||grad_f(xk)||2
Iteration

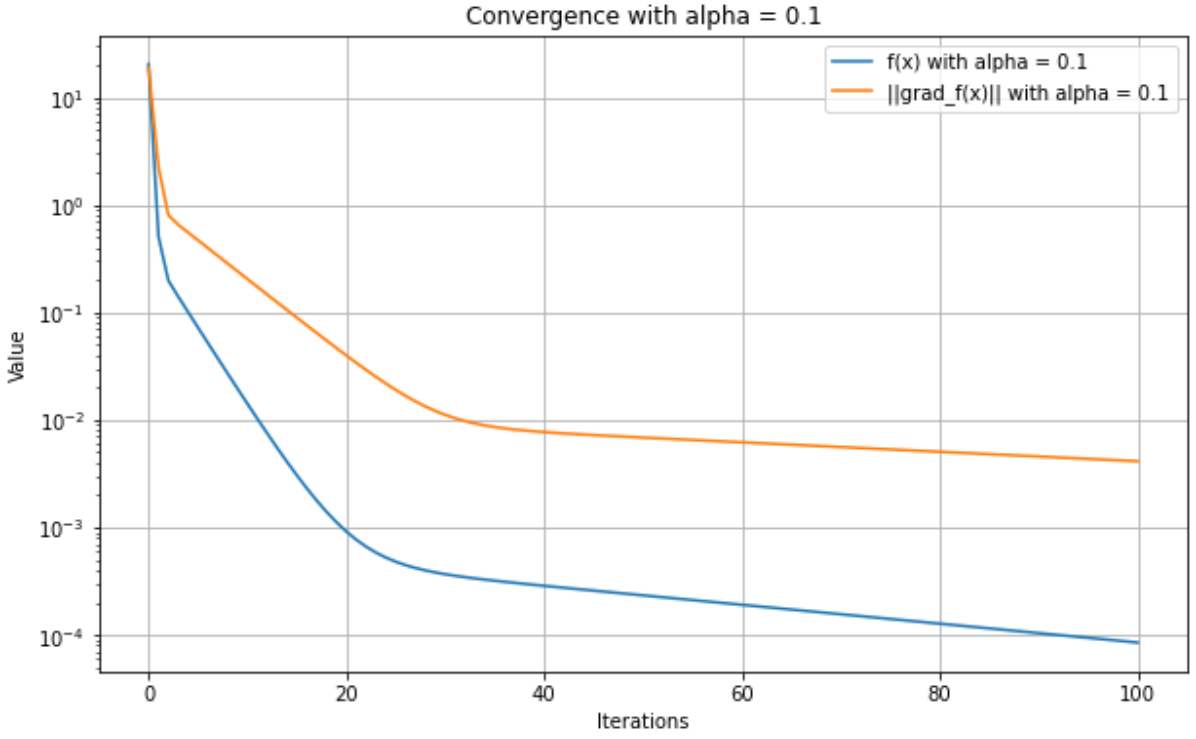
tolf: 1e-06, tolx: 1e-06 -> Iterazioni: 96, Valore finale: 9.342724716663181e-11, Cambiamento in x: 1.3808253016556193e-06
tolf: 1e-06, tolx: 1e-08 -> Iterazioni: 96, Valore finale: 9.342724716663181e-11, Cambiamento in x: 1.3808253016556193e-06
tolf: 1e-06, tolx: 1e-10 -> Iterazioni: 96, Valore finale: 9.342724716663181e-11, Cambiamento in x: 1.3808253016556193e-06
tolf: 1e-08, tolx: 1e-06 -> Iterazioni: 131, Valore finale: 8.147741909896392e-15, Cambiamento in x: 1.2894977530208962e-08
tolf: 1e-08, tolx: 1e-08 -> Iterazioni: 131, Valore finale: 8.147741909896392e-15, Cambiamento in x: 1.2894977530208962e-08
tolf: 1e-08, tolx: 1e-10 -> Iterazioni: 131, Valore finale: 8.147741909896392e-15, Cambiamento in x: 1.2894977530208962e-08
tolf: 1e-10, tolx: 1e-06 -> Iterazioni: 165, Valore finale: 9.280784314323266e-19, Cambiamento in x: 1.376241343109541e-10
tolf: 1e-10, tolx: 1e-08 -> Iterazioni: 165, Valore finale: 9.280784314323266e-19, Cambiamento in x: 1.376241343109541e-10
tolf: 1e-10, tolx: 1e-10 -> Iterazioni: 165, Valore finale: 9.280784314323266e-19, Cambiamento in x: 1.376241343109541e-10

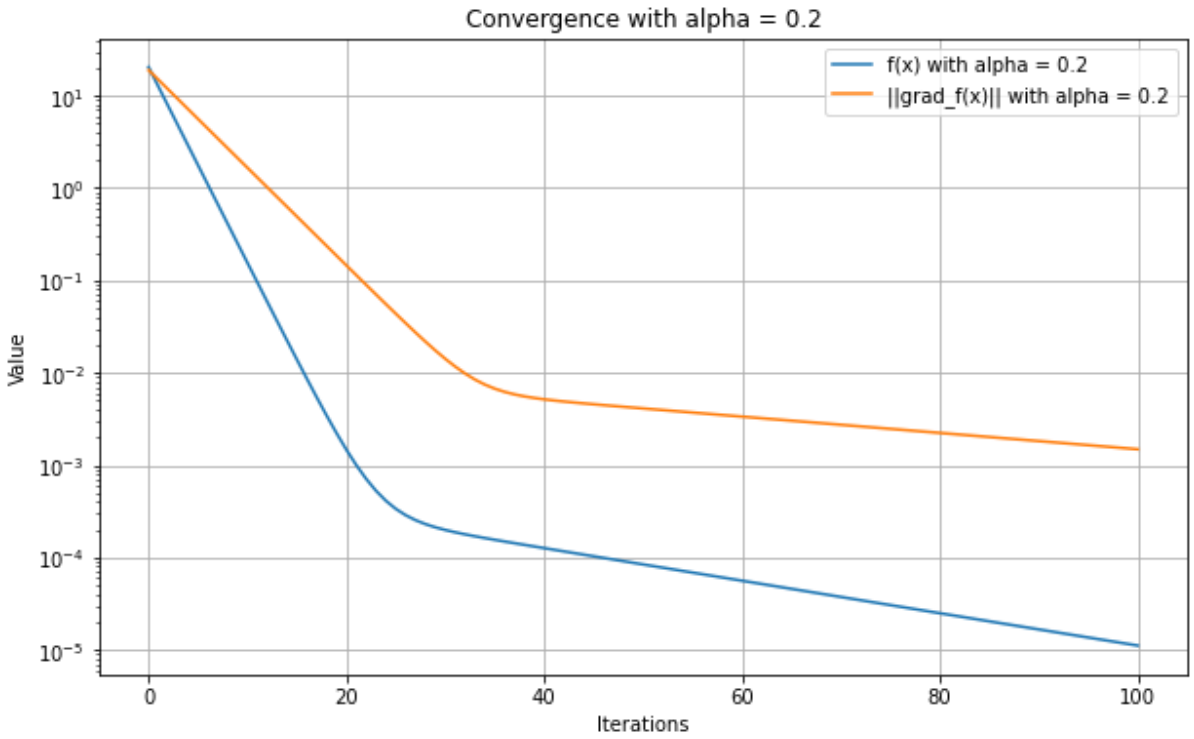


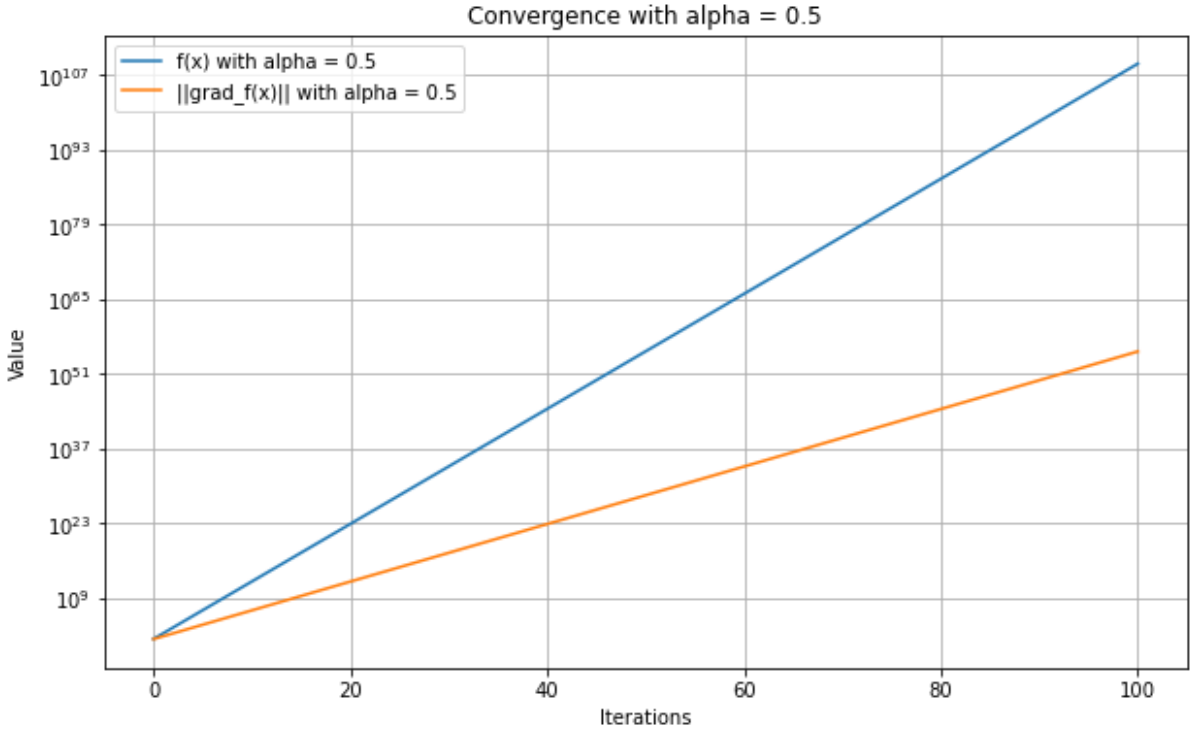

Convergence of Gradient Descent per tolf=1e-08 e tolx=1e-06
Error ||x_k - x*||_2
Iteration k
10^{0}
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
10^{-6}
10^{-7}
0
20
40
60
80
100
120

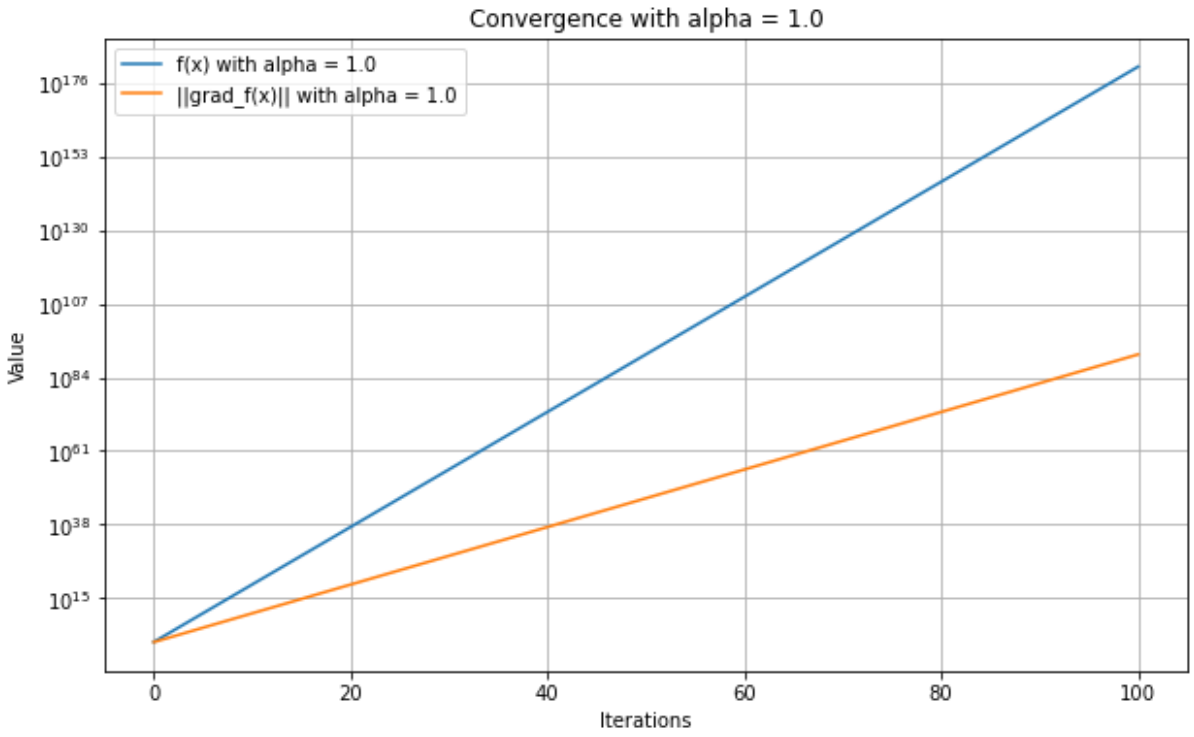

Convergence of Gradient Descent per tolf=1e-10 e tolx=1e-06
Error ||x_k - x*||_2
Iteration k
10^{0}
10^{-2}
10^{-4}
10^{-6}
10^{-8}
0
25
50
75
100
125
150

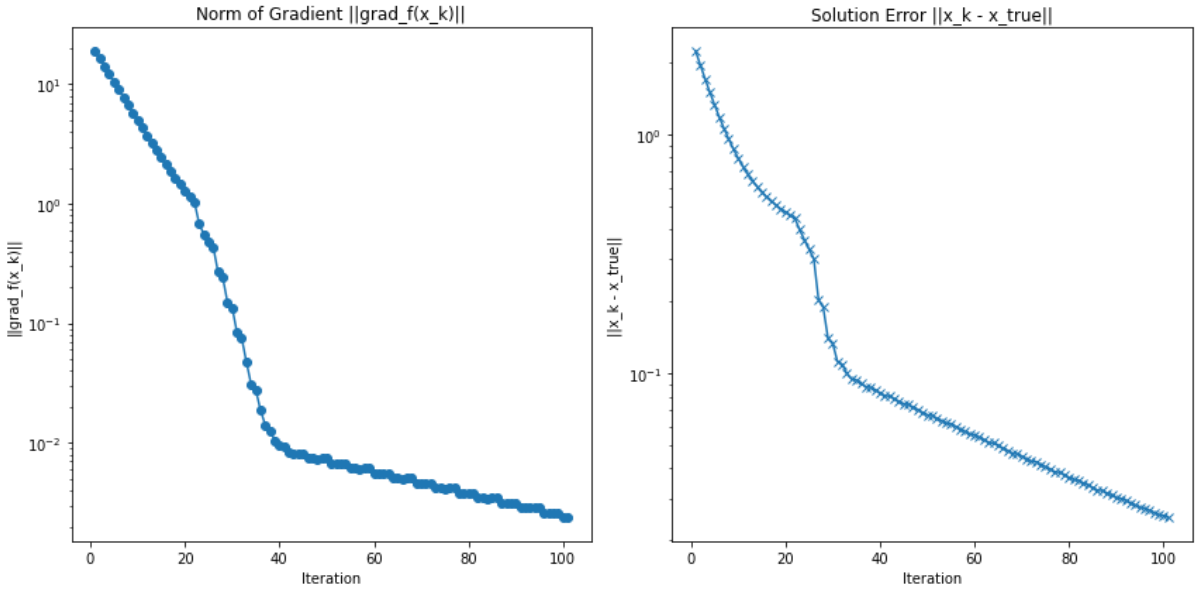

Convergence with alpha = 0.01
f(x) with alpha = 0.01
||grad_f(x)|| with alpha = 0.01
Value
Iterations


Convergence with alpha = 0.1
f(x) with alpha = 0.1
||grad_f(x)|| with alpha = 0.1
Value
Iterations


Convergence with alpha = 0.2
f(x) with alpha = 0.2
||grad_f(x)|| with alpha = 0.2
Value
Iterations

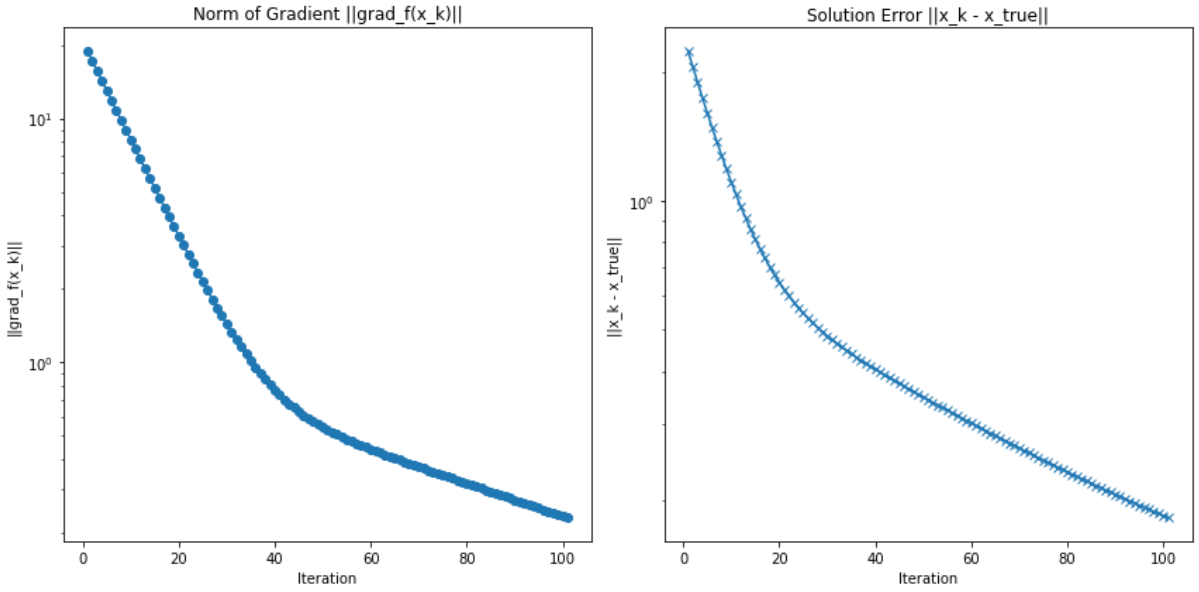

Convergence with alpha = 0.5
f(x) with alpha = 0.5
||grad_f(x)|| with alpha = 0.5
Value
Iterations

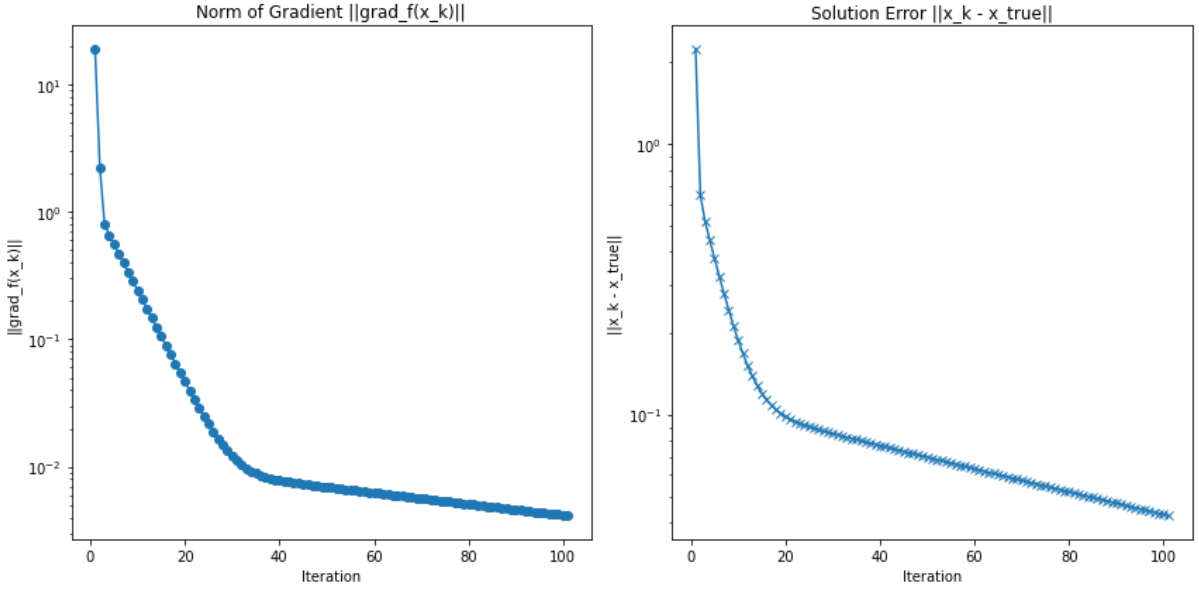


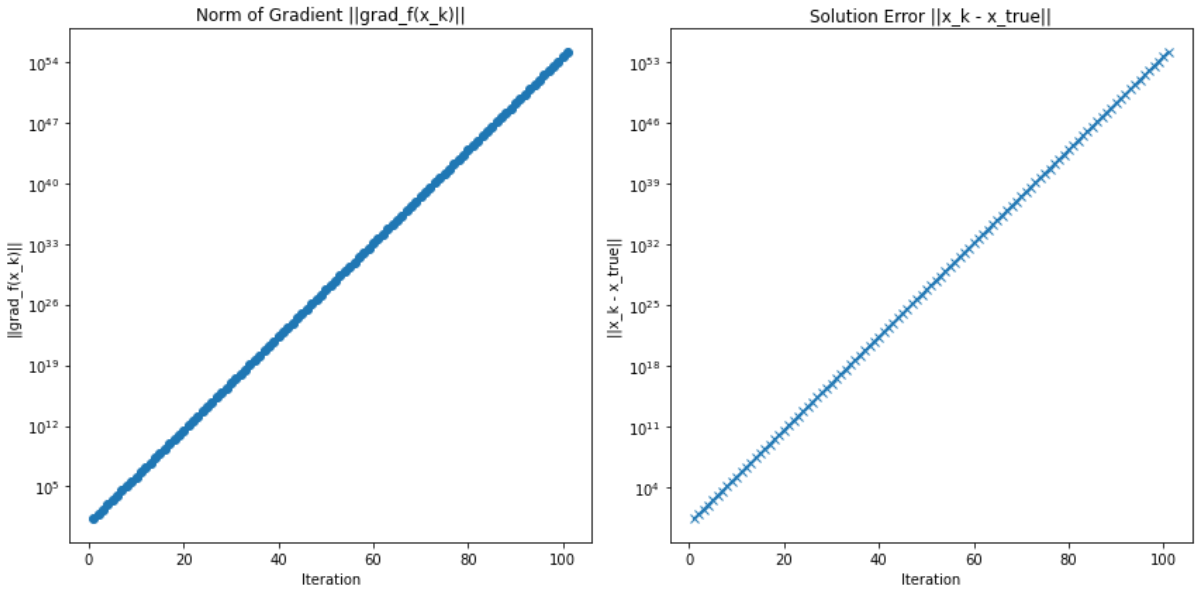


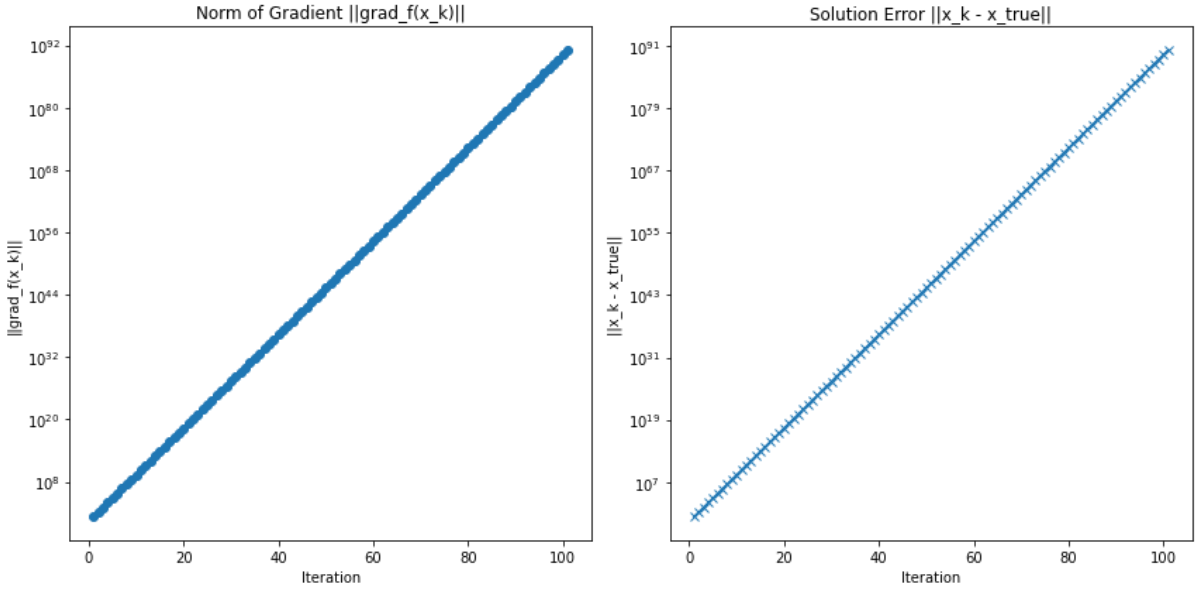
Backtracking

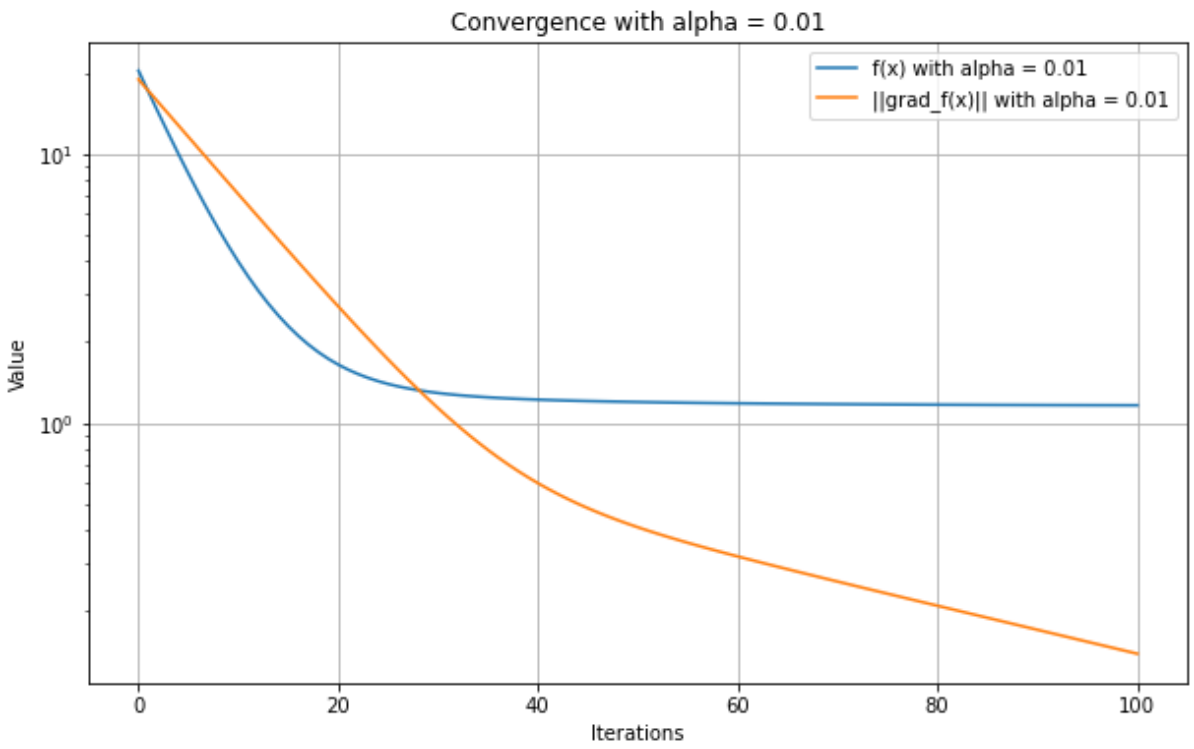
No backtracking

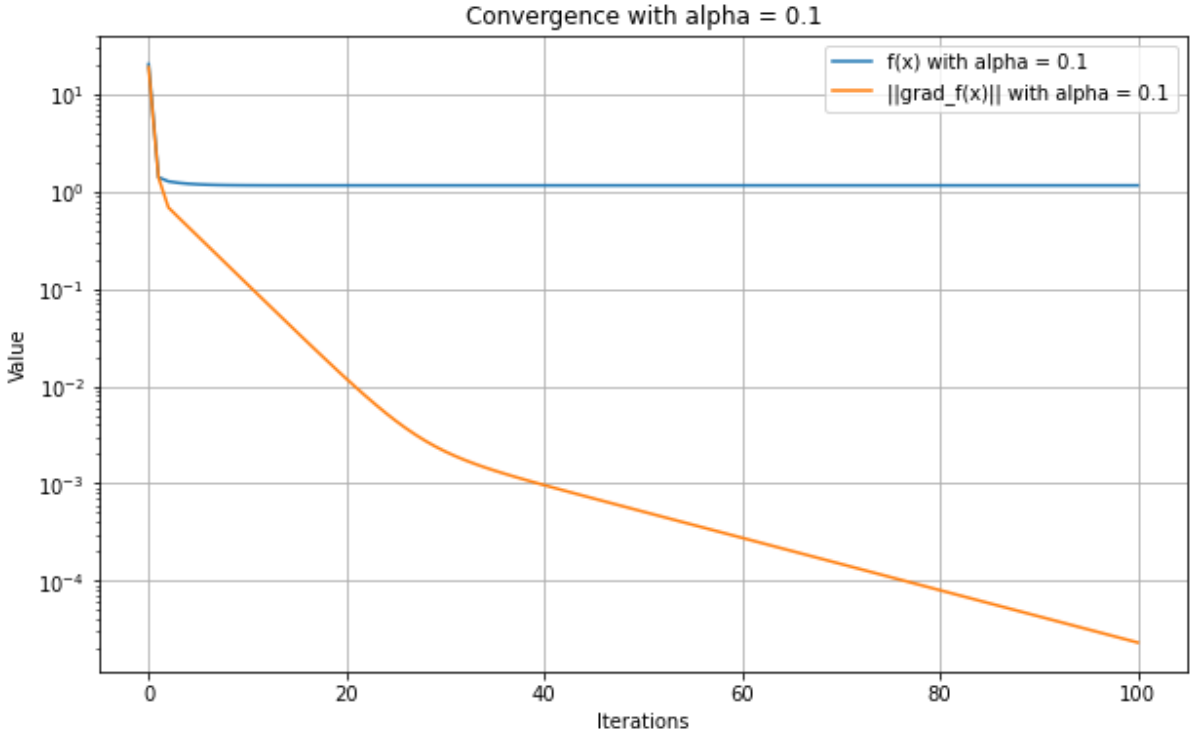


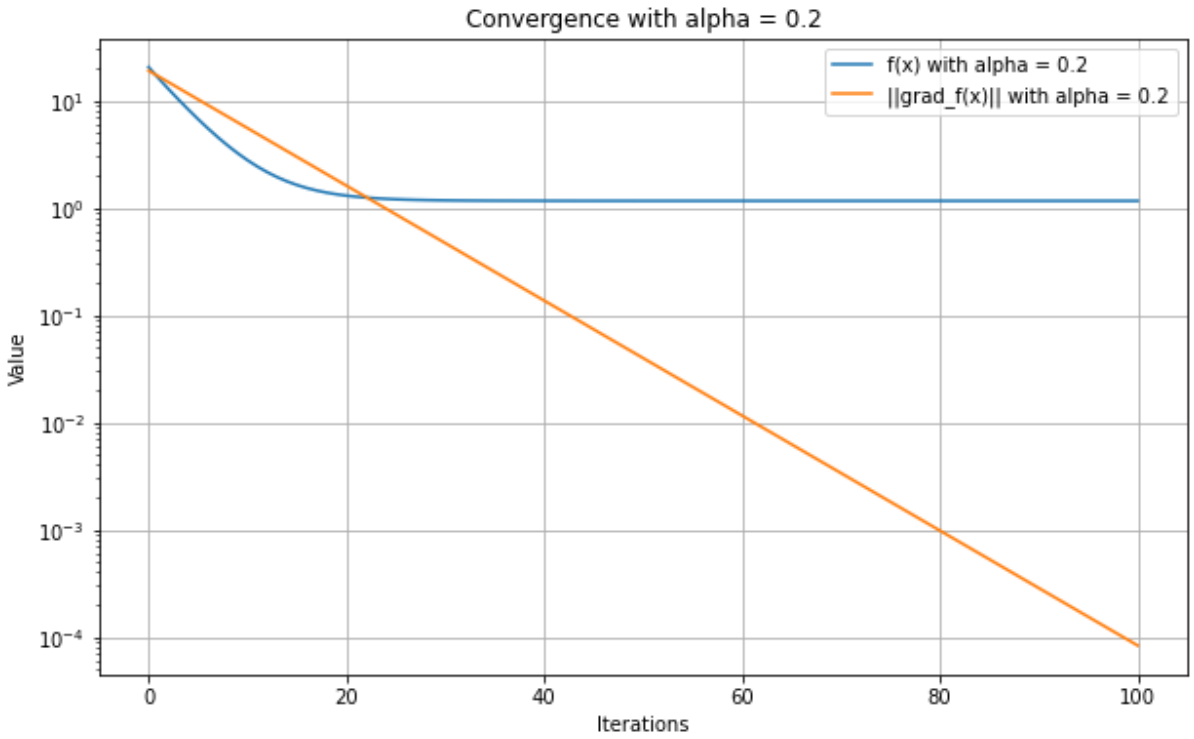


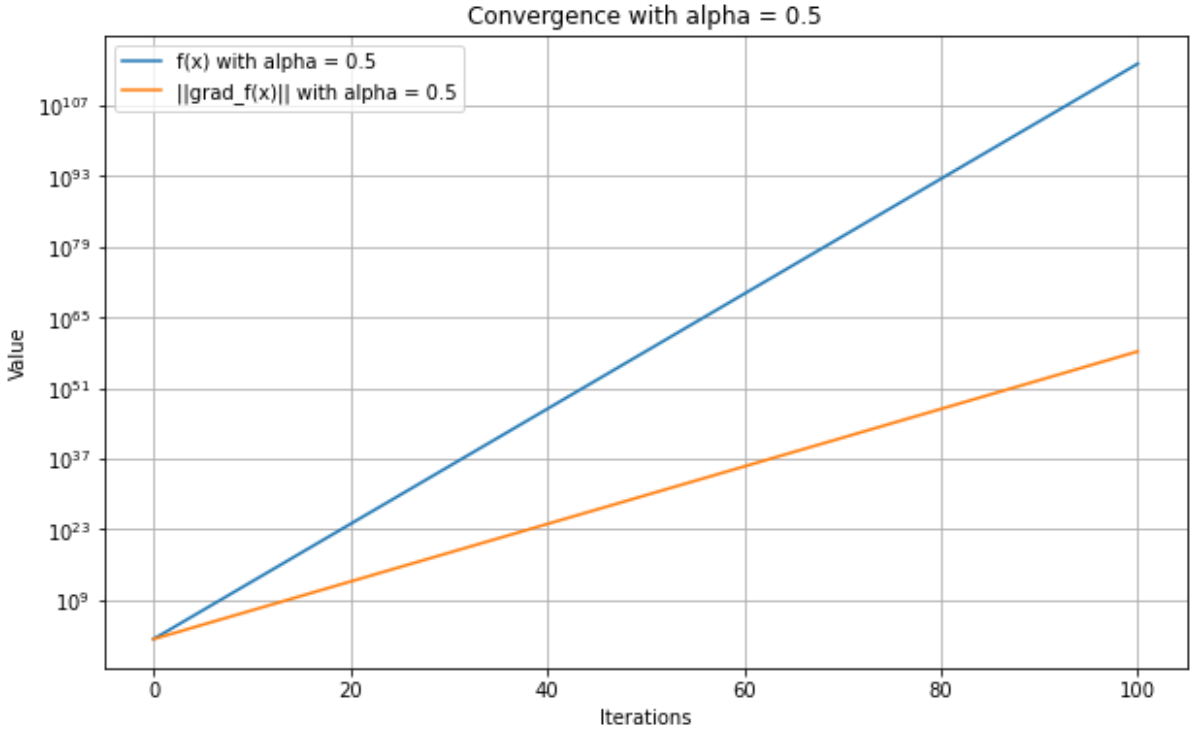

Norm of Gradient ||grad_f(x_k)||
||grad_f(x_k)||
Iteration
Solution Error ||x_k - x_true||
||x_k - x_true||
Iteration

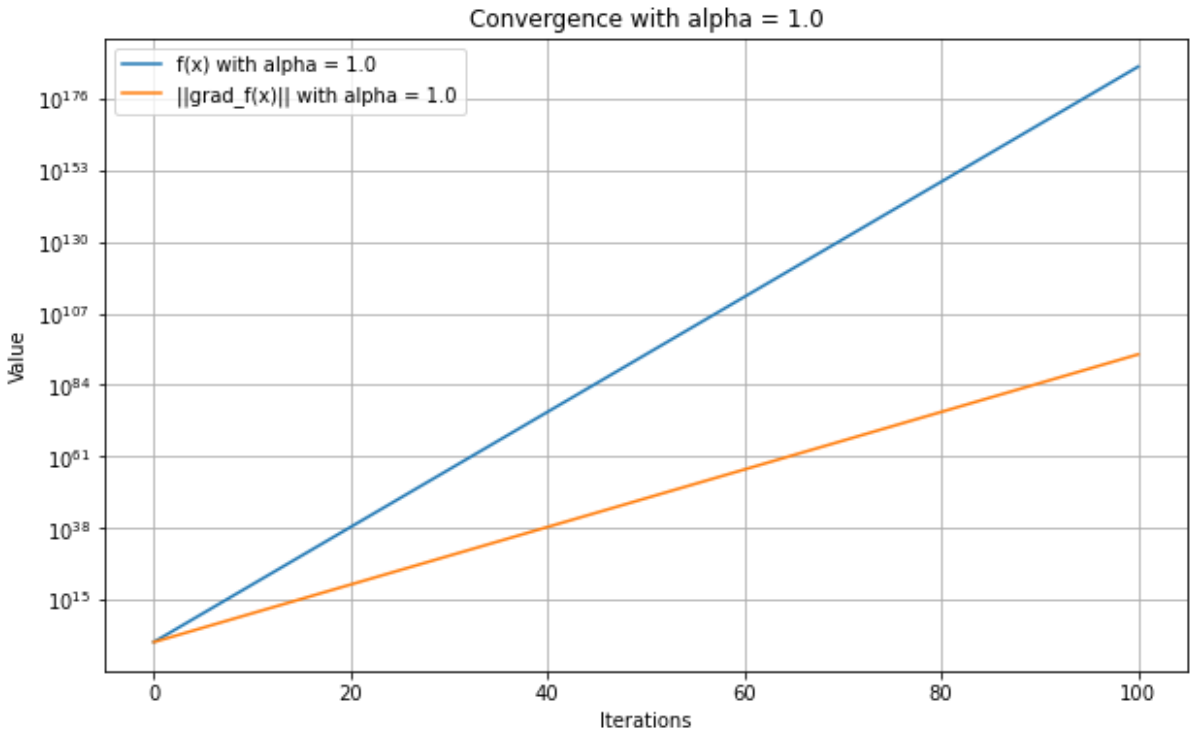

Norm of Gradient ||grad_f(x_k)||
||grad_f(x_k)||
Iteration
Solution Error ||x_k - x_true||
||x_k - x_true||
Iteration

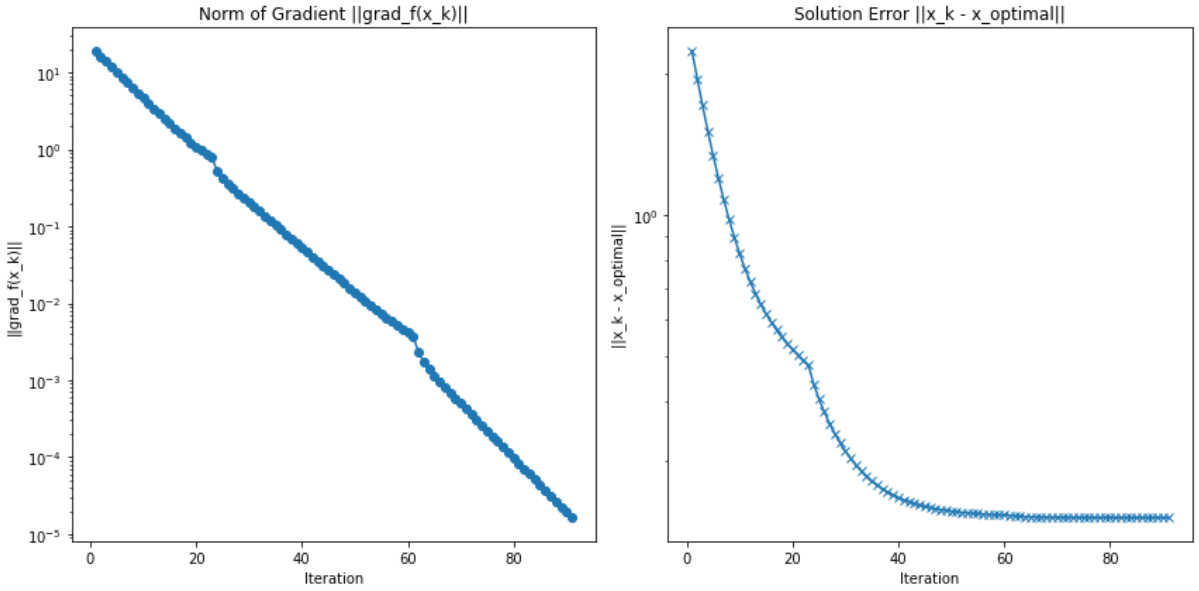

Convergence with alpha = 0.01
f(x) with alpha = 0.01
||grad_f(x)|| with alpha = 0.01
Value
Iterations


Convergence with alpha = 0.1
f(x) with alpha = 0.1
||grad_f(x)|| with alpha = 0.1
Value
Iterations


Convergence with alpha = 0.2
f(x) with alpha = 0.2
||grad_f(x)|| with alpha = 0.2
Value
Iterations

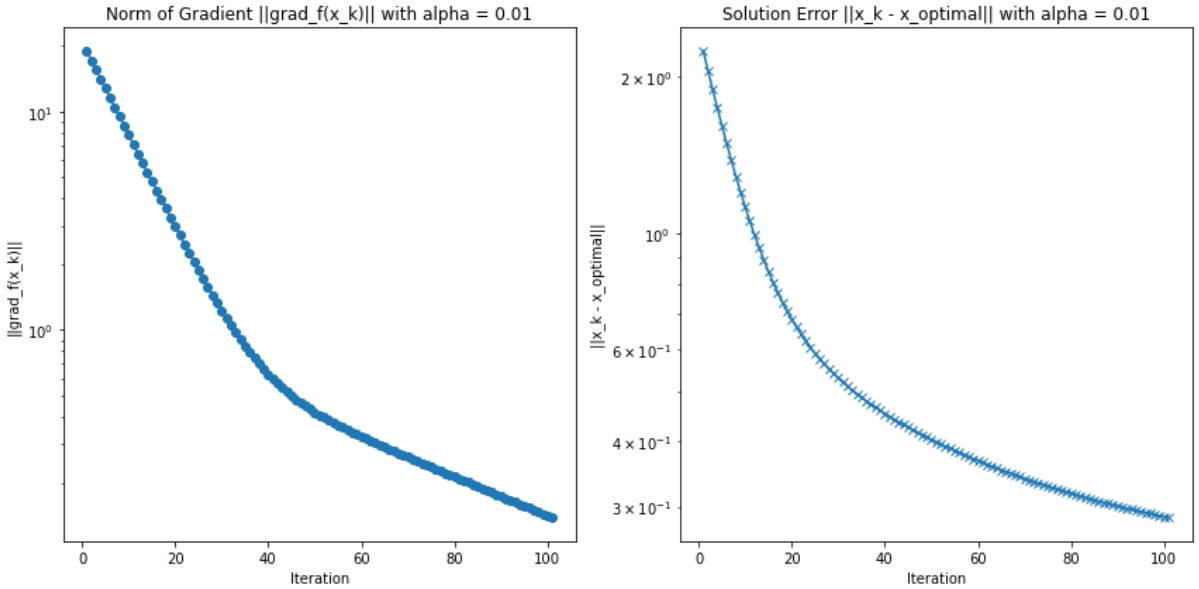

Convergence with alpha = 0.5
f(x) with alpha = 0.5
||grad_f(x)|| with alpha = 0.5
Value
Iterations

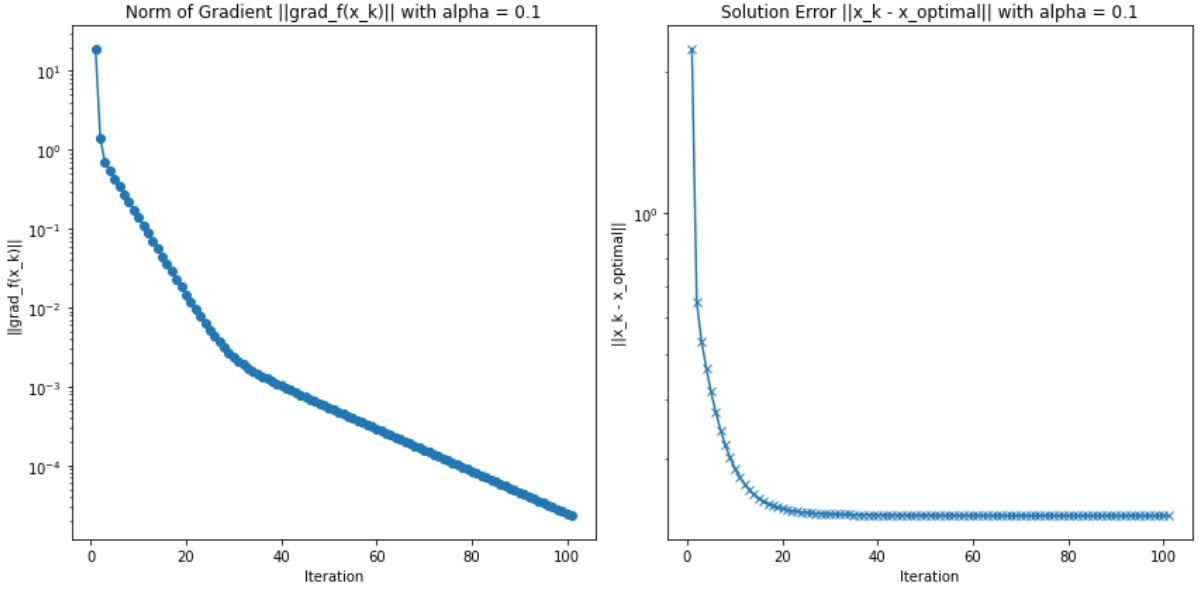


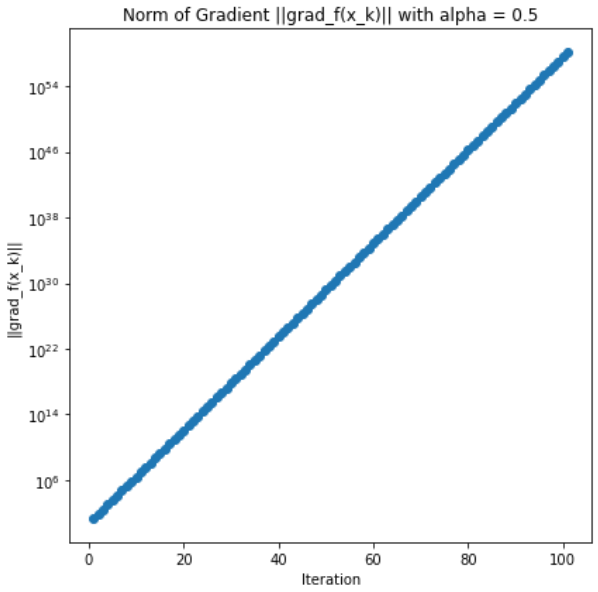


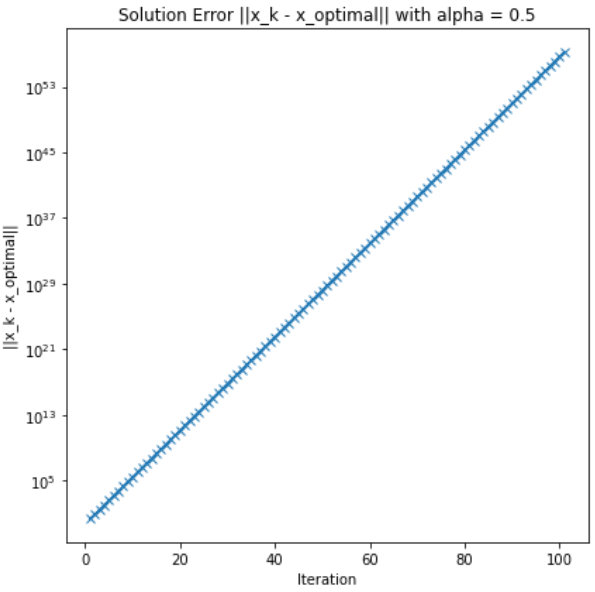
backtracking f4

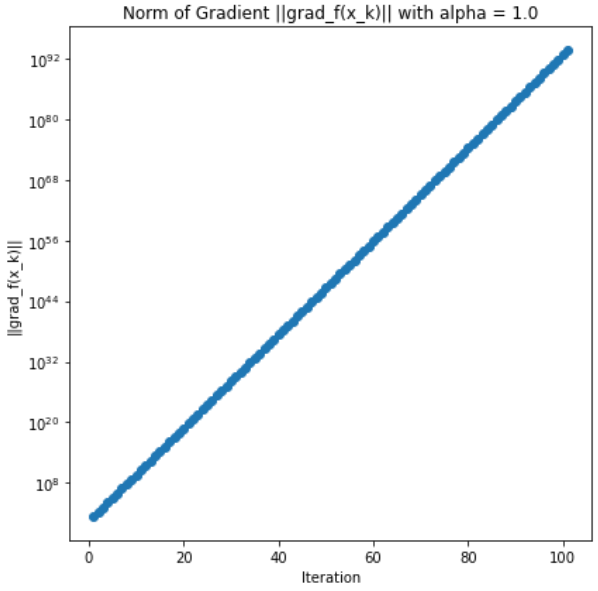
no backtracking f4

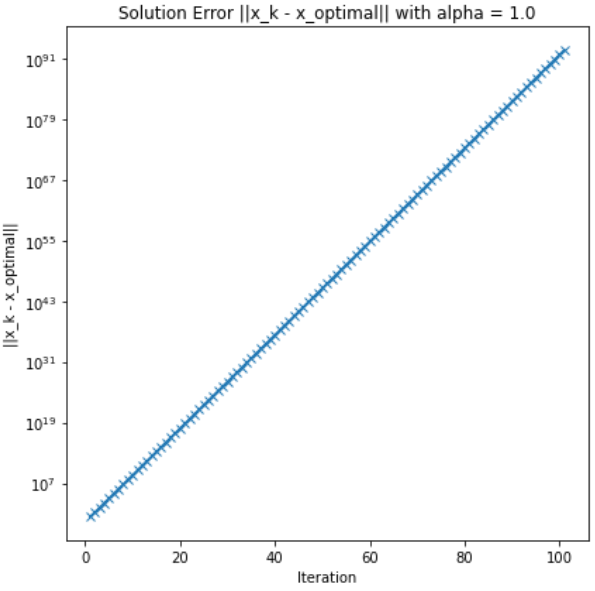


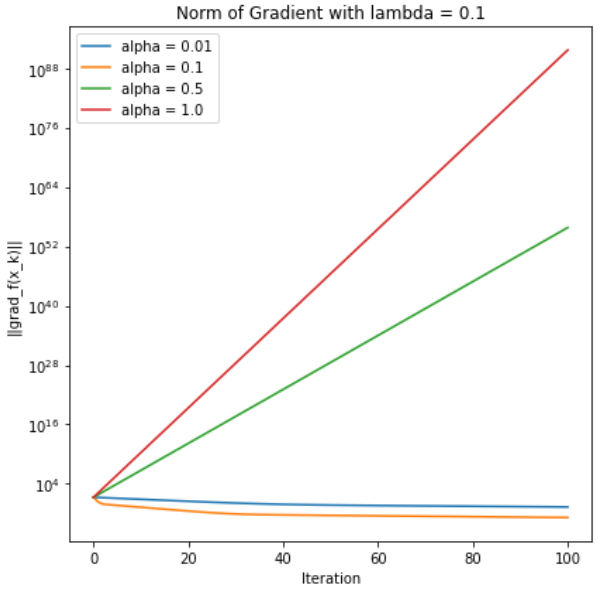


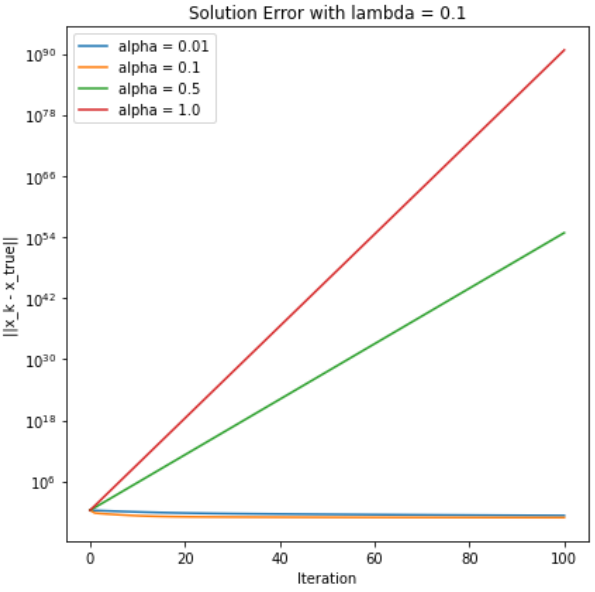

Norm of Gradient ||grad_f(x_k)|| with alpha = 0.5
||grad_f(x_k)||
Iteration

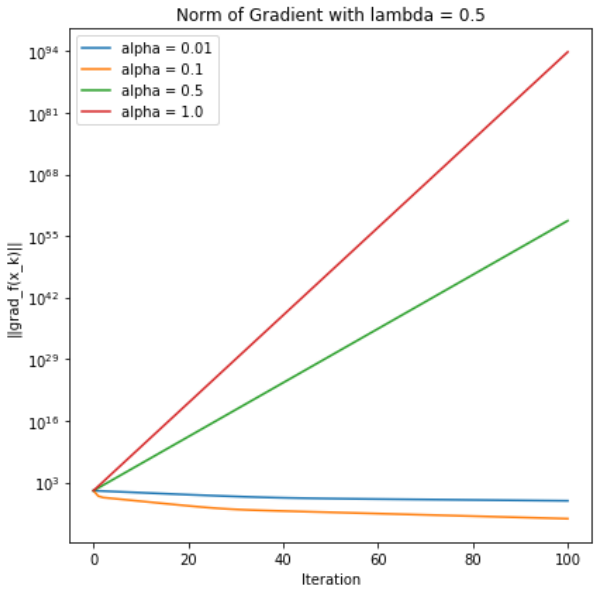

Solution Error ||x_k - x_optimal|| with alpha = 0.5
||x_k - x_optimal||
Iteration

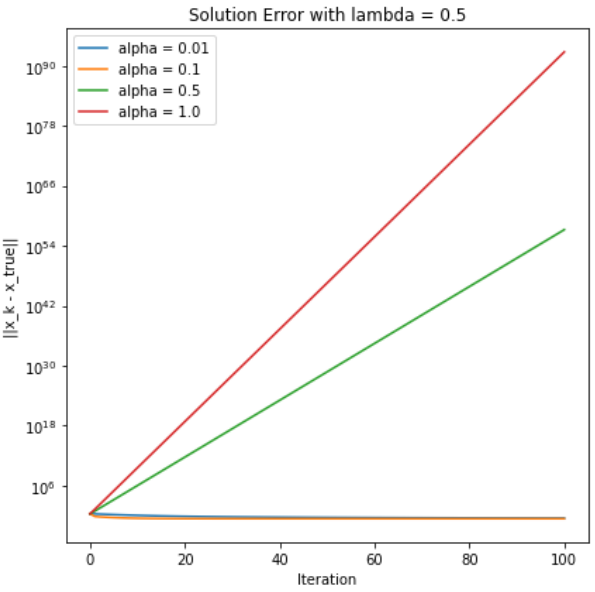

Norm of Gradient ||grad_f(x_k)|| with alpha = 1.0
||grad_f(x_k)||
Iteration

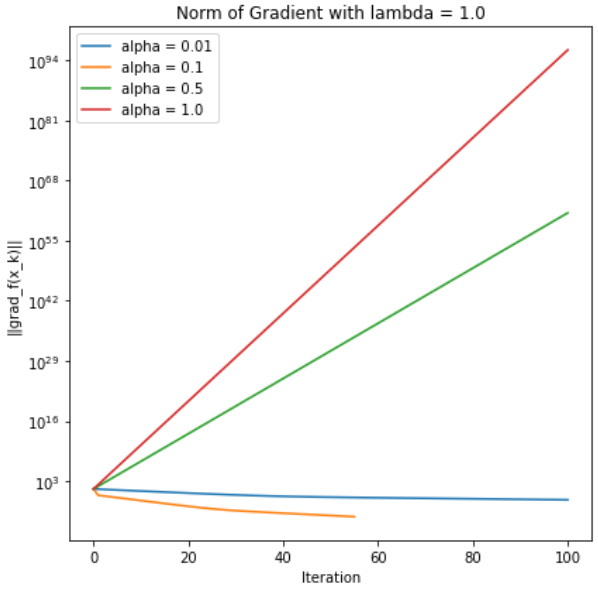

Solution Error ||x_k - x_optimal|| with alpha = 1.0
||x_k - x_optimal||
Iteration

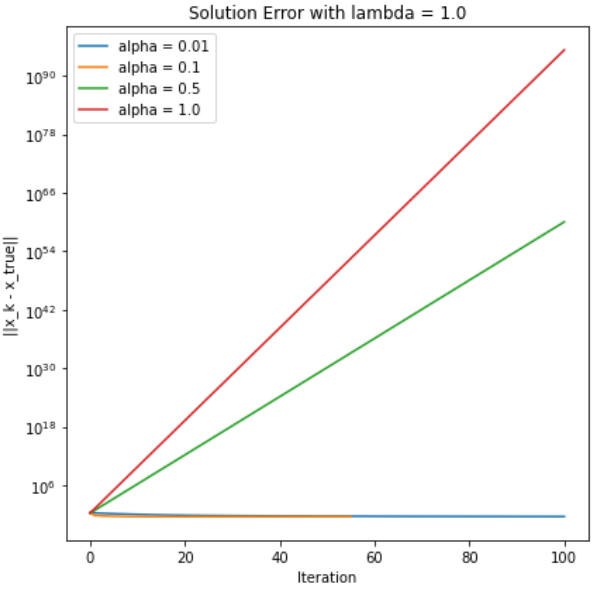

Norm of Gradient with lambda = 0.1
alpha = 0.01
alpha = 0.1
alpha = 0.5
alpha = 1.0
||grad_f(x_k)||
Iteration

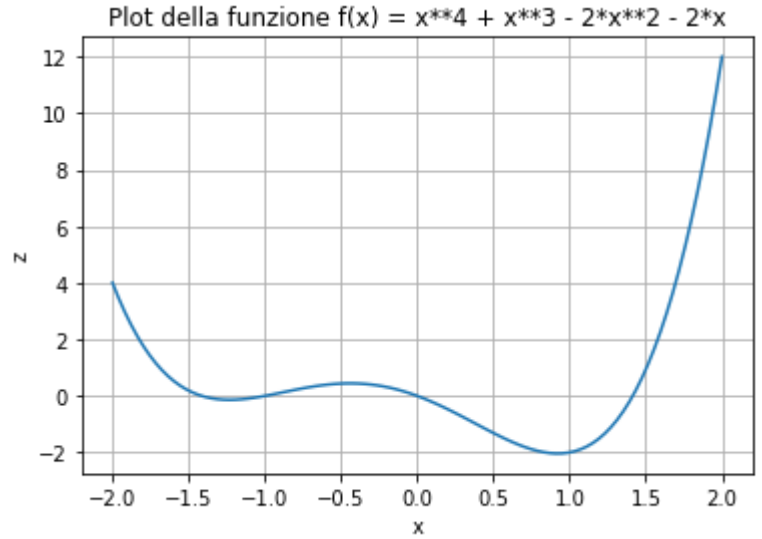

Solution Error with lambda = 0.1
alpha = 0.01
alpha = 0.1
alpha = 0.5
alpha = 1.0
||x_k - x_true||
Iteration

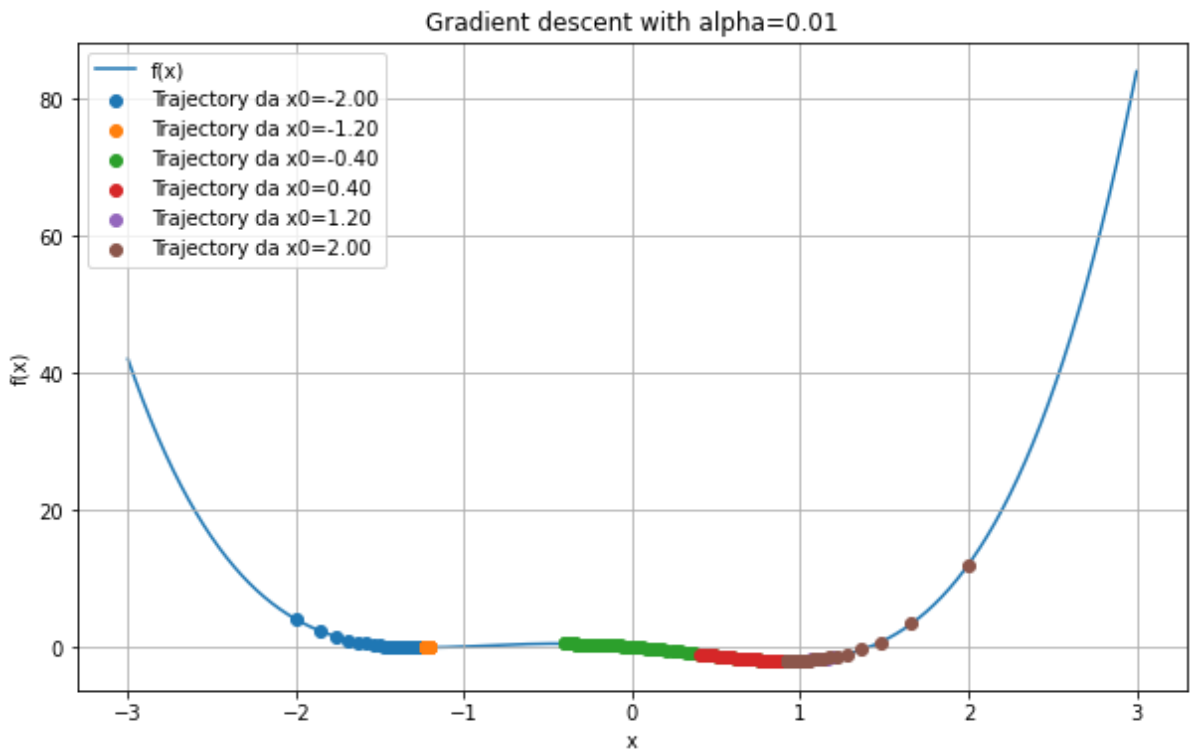

Norm of Gradient with lambda = 0.5
alpha = 0.01
alpha = 0.1
alpha = 0.5
alpha = 1.0
||grad_f(x_k)||
Iteration

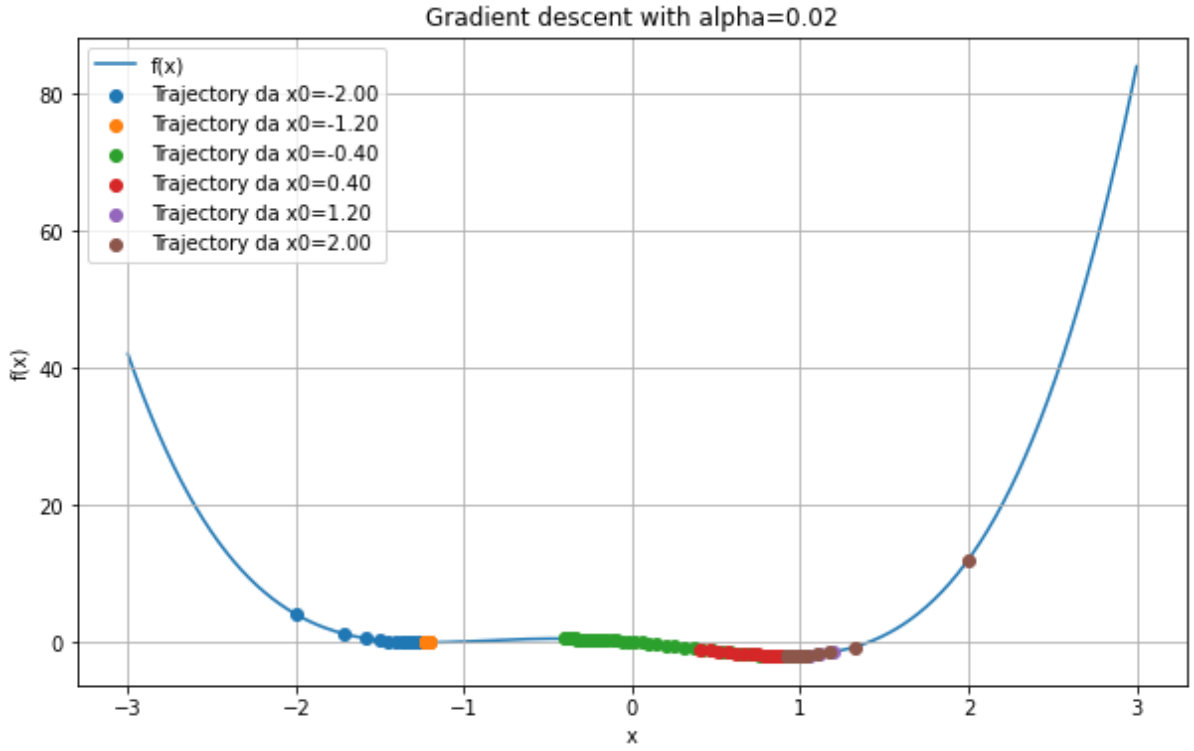

Solution Error with lambda = 0.5
alpha = 0.01
alpha = 0.1
alpha = 0.5
alpha = 1.0
||x_k - x_true||
Iteration

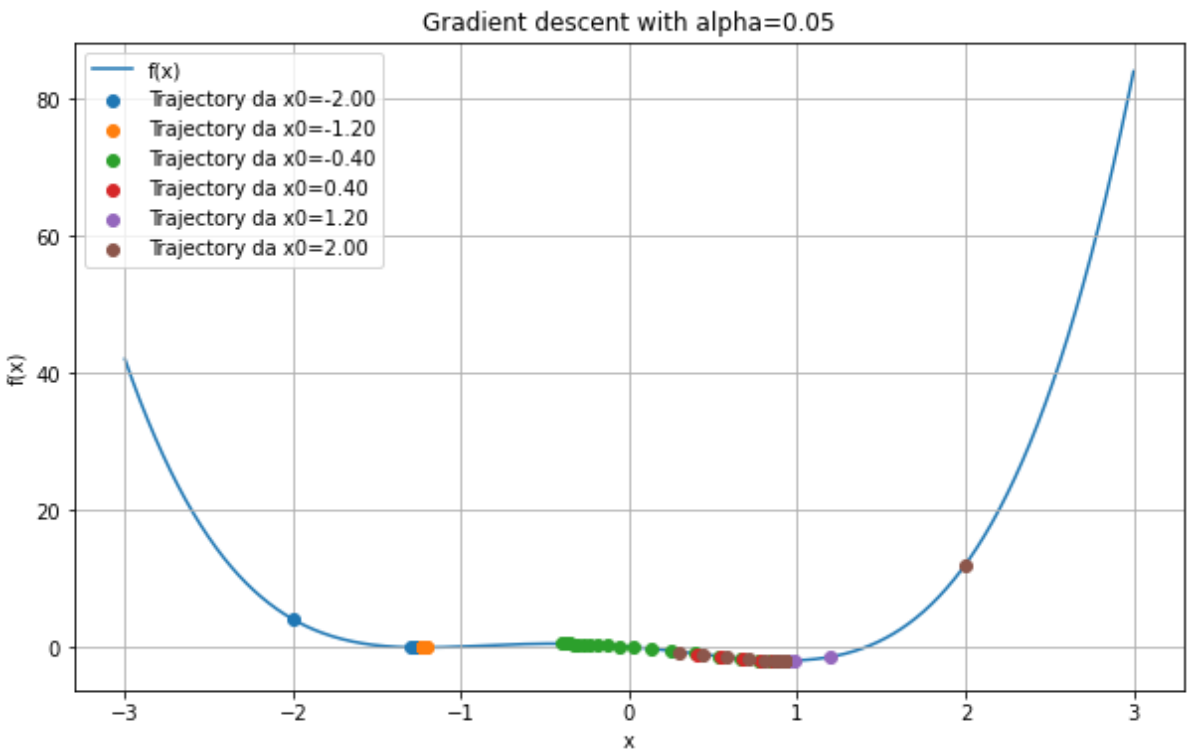

Norm of Gradient with lambda = 1.0
alpha = 0.01
alpha = 0.1
alpha = 0.5
alpha = 1.0
||grad_f(x_k)||
Iteration

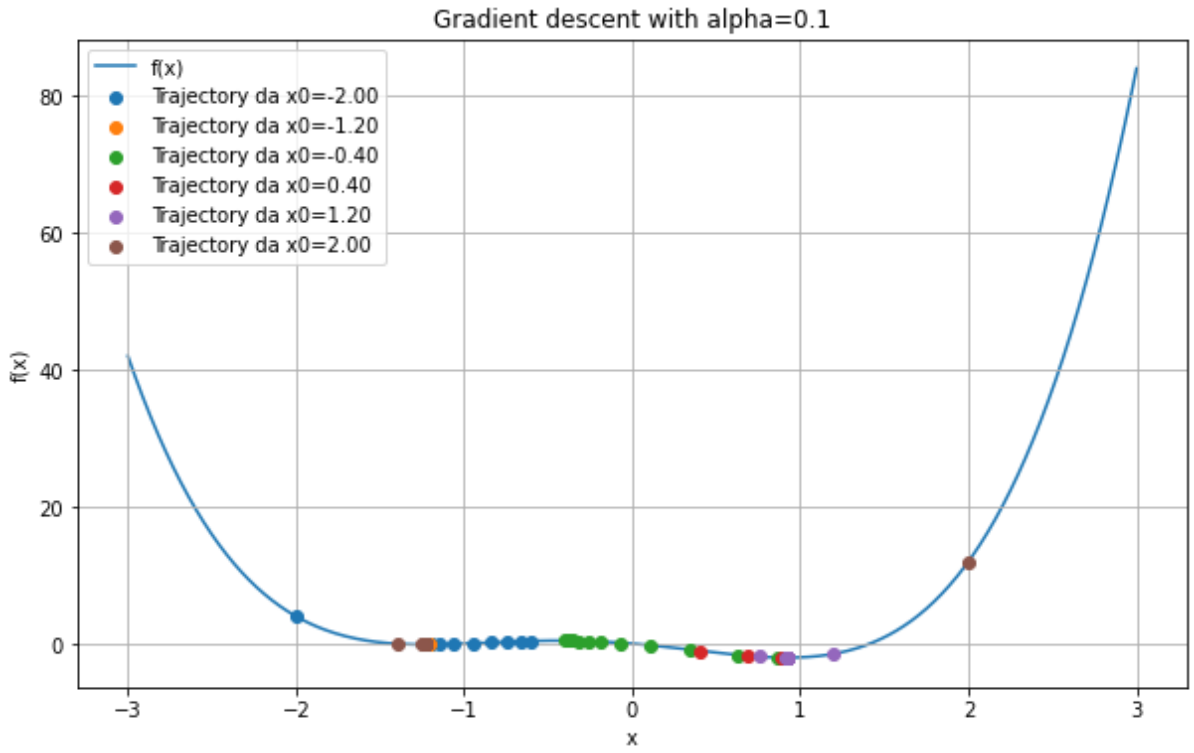

Solution Error with lambda = 1.0
alpha = 0.01
alpha = 0.1
alpha = 0.5
alpha = 1.0
||x_k - x_true||
Iteration

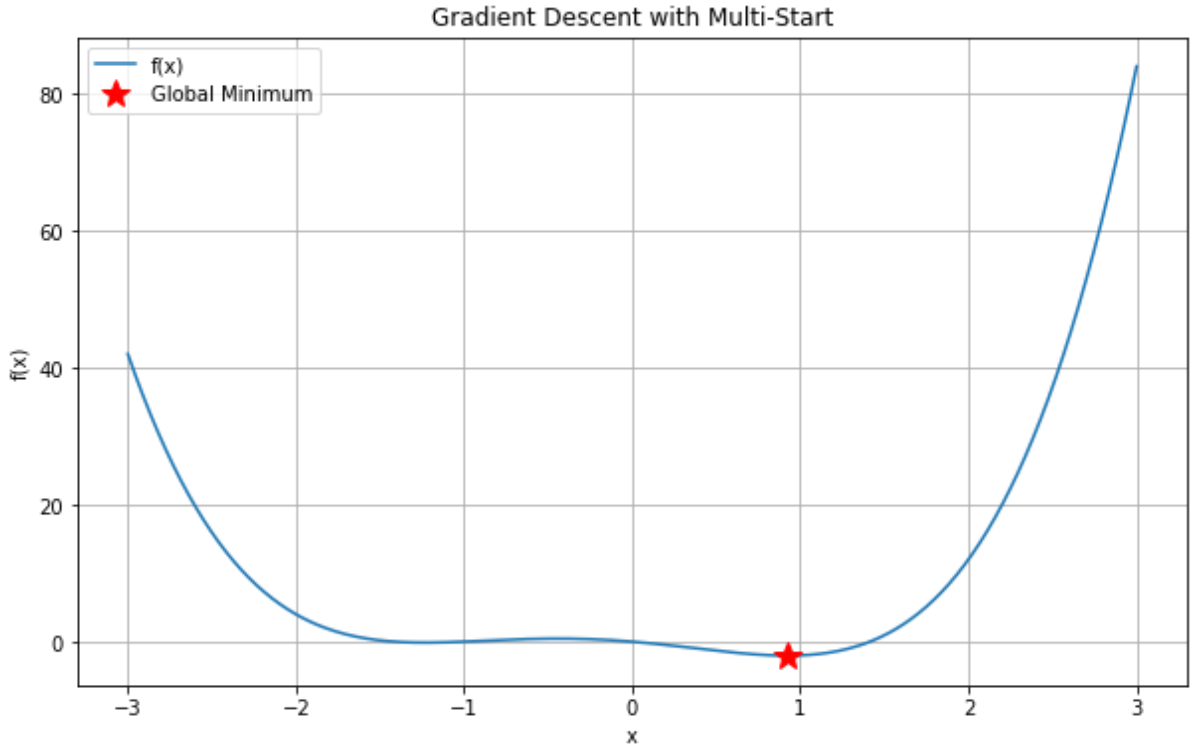

Plot della funzione f(x) = x**4 + x**3 - 2*x**2 - 2*x
z
x


Gradient descent with alpha=0.01
f(x)
Trajectory da x0=-2.00
Trajectory da x0=-1.20
Trajectory da x0=-0.40
Trajectory da x0=0.40
Trajectory da x0=1.20
Trajectory da x0=2.00
f(x)
x


Gradient descent with alpha=0.02
f(x)
Trajectory da x0=-2.00
Trajectory da x0=-1.20
Trajectory da x0=-0.40
Trajectory da x0=0.40
Trajectory da x0=1.20
Trajectory da x0=2.00
f(x)
x

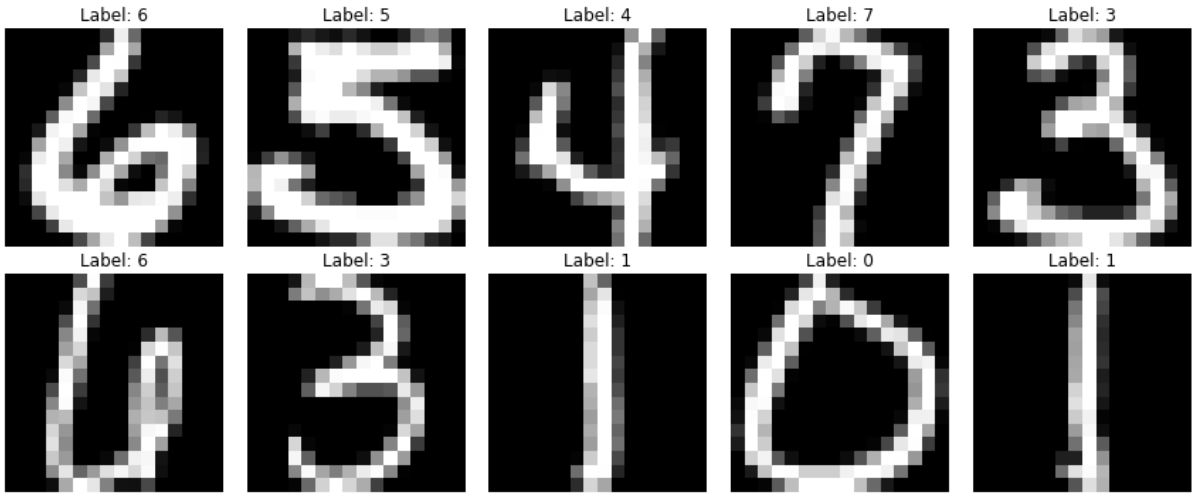

Gradient descent with alpha=0.05
f(x)
Trajectory da x0=-2.00
Trajectory da x0=-1.20
Trajectory da x0=-0.40
Trajectory da x0=0.40
Trajectory da x0=1.20
Trajectory da x0=2.00
f(x)
x
80
60
40
20
0
−3
−2
−1
0
1
2
3



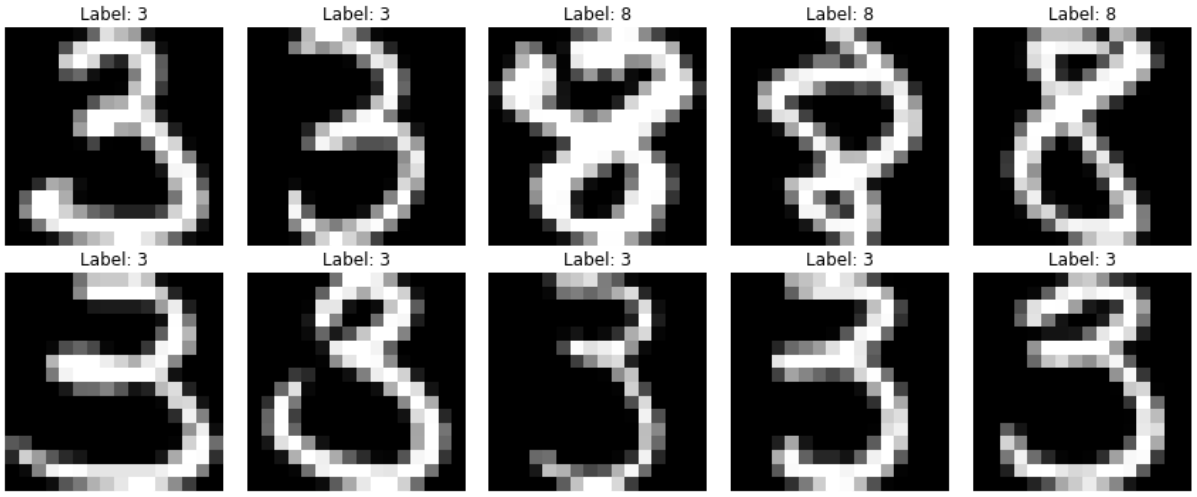


HomeWork 3_2

```
Shape of X (images): (256, 1707)
Shape of I (labels): (1, 1707)
```



(256, 275) (1, 275)



(257, 220) (1, 220) (257, 55) (1, 55)
Accuracy: 1.0

Shape of X (images): (256, 1707)
Shape of I (labels): (1, 1707)
Filtered Data for digits (0, 1): X_filtered.shape = (257, 571), Y_filtered.shape = (1, 571)
Data Split: X_train.shape = (257, 285), Y_train.shape = (1, 285), X_test.shape = (257, 286), Y_test.shape = (1, 286)
Binary Labels: [[0 1 1 1 0 1 0 1 0 0 0 1 0 0 0 1 0 1 0 0 1 1 1 1 0 0 0 1 1 1 0 1 0 1 1 0
  1 0 0 0 0 0 0 0 0 0 1 1 0 1 0 0 1 0 0 1 0 0 1 0 1 0 1 0 0 0 0 0 0 1 0 0
  0 1 1 1 0 0 1 1 1 0 1 0 0 0 1 0 0 1 0 1 1 0 1 1 1 0 0 0 0 1 0 0 0 1 0 0
  1 1 0 1 1 0 0 0 1 1 0 1 1 0 1 0 0 0 1 1 1 0 0 1 1 1 0 1 0 0 0 0 1 1 1 0
  0 0 0 0 1 0 1 0 1 0 0 1 1 1 1 1 0 1 0 1 1 0 1 0 1 1 0 1 1 0 0 0 0 1 0 0
  1 1 0 0 1 1 1 0 0 1 0 1 1 0 0 1 0 0 0 0 0 0 1 1 1 1 0 0 0 0 1 0 1 1 0 0 1
  0 1 1 1 0 1 1 1 0 1 0 0 0 1 0 0 1 0 1 1 0 0 0 1 0 1 1 0 0 1 0 0 0 0 0 0 0
  0 0 1 0 0 0 1 0 0 0 1 1 1 1 0 0 1 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 1 1]]
Accuracy for digits 0, 1 with batch size 32 and training percentage 0.5: 0.9965034965034965
Binary Labels: [[0 1 1 1 0 1 0 1 0 0 0 1 0 0 0 1 0 1 0 0 1 1 1 1 0 0 0 1 1 1 0 1 0 1 1 0
  1 0 0 0 0 0 0 0 0 0 1 1 0 1 0 0 1 0 0 1 0 0 1 0 1 0 1 0 0 0 0 0 0 1 0 0
  0 1 1 1 0 0 1 1 1 0 1 0 0 0 1 0 0 1 0 1 1 0 1 1 1 0 0 0 0 1 0 0 0 1 0 0
  1 1 0 1 1 0 0 0 1 1 0 1 1 0 1 0 0 0 1 1 1 0 0 1 1 1 0 1 0 0 0 0 1 1 1 0
  0 0 0 0 1 0 1 0 1 0 0 1 1 1 1 1 0 1 0 1 1 0 1 0 1 1 0 1 1 0 0 0 0 1 0 0
  1 1 0 0 1 1 1 0 0 1 0 1 1 0 0 1 0 0 0 0 0 0 1 1 1 1 0 0 0 0 1 0 1 1 0 0 1
  0 1 1 1 0 1 1 1 0 1 0 0 0 1 0 0 1 0 1 1 0 0 0 1 0 1 1 0 0 1 0 0 0 0 0 0 0
  0 0 1 0 0 0 1 0 0 0 1 1 1 1 0 0 1 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 1 1]]

0 0 0 0 1 0 1 0 1 0 0 1 1 1 1 1 0 1 0 1 1 0 1 0 1 1 0 1 1 0 0 0 0 1 0 0
1 1 0 0 1 1 1 0 0 1 0 1 1 0 0 1 0 0 0 0 0 1 1 1 1 0 0 0 0 1 0 1 1 0 0 1
0 1 1 1 0 1 1 1 0 1 0 0 0 1 0 0 1 0 1 1 0 0 0 1 0 1 1 0 0 1 0 0 0 0 0 0
0 0 1 0 0 0 1 0 0 0 1 1 1 1 0 0 1 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 1 1]]
Accuracy for digits 0, 1 with batch size 32 and training percentage 0.7: 0.9965034965034965
Binary Labels: [[0 1 1 1 0 1 0 1 0 0 0 1 0 0 0 1 0 1 0 0 1 1 1 1 0 0 0 1 1 1 0 1 0 1 1 0
1 0 0 0 0 0 0 0 0 0 1 1 0 1 0 0 1 0 0 1 0 0 1 0 1 0 1 0 0 0 0 0 0 1 0 0
0 1 1 1 0 0 1 1 1 0 1 0 0 0 1 0 0 1 0 1 1 0 1 1 1 0 0 0 0 1 0 0 0 1 0 0
1 1 0 1 1 0 0 0 1 1 0 1 1 0 1 0 0 0 1 1 1 0 0 1 1 1 0 1 0 0 0 0 1 1 1 0
0 0 0 0 1 0 1 0 1 0 0 1 1 1 1 1 0 1 0 1 1 0 1 0 1 1 0 1 1 0 0 0 0 1 0 0
1 1 0 0 1 1 1 0 0 1 0 1 1 0 0 1 0 0 0 0 0 1 1 1 1 0 0 0 0 1 0 1 1 0 0 1
0 1 1 1 0 1 1 1 0 1 0 0 0 1 0 0 1 0 1 1 0 0 0 1 0 1 1 0 0 1 0 0 0 0 0 0
0 0 1 0 0 0 1 0 0 0 1 1 1 1 0 0 1 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 1 1]]
Accuracy for digits 0, 1 with batch size 32 and training percentage 0.9: 0.9965034965034965
Binary Labels: [[0 1 1 1 0 1 0 1 0 0 0 1 0 0 0 1 0 1 0 0 1 1 1 1 0 0 0 1 1 1 0 1 0 1 1 0
1 0 0 0 0 0 0 0 0 0 1 1 0 1 0 0 1 0 0 1 0 0 1 0 1 0 1 0 0 0 0 0 0 1 0 0
0 1 1 1 0 0 1 1 1 0 1 0 0 0 1 0 0 1 0 1 1 0 1 1 1 0 0 0 0 1 0 0 0 1 0 0
1 1 0 1 1 0 0 0 1 1 0 1 1 0 1 0 0 0 1 1 1 0 0 1 1 1 0 1 0 0 0 0 1 1 1 0
0 0 0 0 1 0 1 0 1 0 0 1 1 1 1 1 0 1 0 1 1 0 1 0 1 1 0 1 1 0 0 0 0 1 0 0
1 1 0 0 1 1 1 0 0 1 0 1 1 0 0 1 0 0 0 0 0 1 1 1 1 0 0 0 0 1 0 1 1 0 0 1
0 1 1 1 0 1 1 1 0 1 0 0 0 1 0 0 1 0 1 1 0 0 0 1 0 1 1 0 0 1 0 0 0 0 0 0
0 0 1 0 0 0 1 0 0 0 1 1 1 1 0 0 1 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 1 1]]
Accuracy for digits 0, 1 with batch size 64 and training percentage 0.5: 0.9965034965034965
Binary Labels: [[0 1 1 1 0 1 0 1 0 0 0 1 0 0 0 1 0 1 0 0 1 1 1 1 0 0 0 1 1 1 0 1 0 1 1 0
1 0 0 0 0 0 0 0 0 0 1 1 0 1 0 0 1 0 0 1 0 0 1 0 1 0 1 0 0 0 0 0 0 1 0 0
0 1 1 1 0 0 1 1 1 0 1 0 0 0 1 0 0 1 0 1 1 0 1 1 1 0 0 0 0 1 0 0 0 1 0 0
1 1 0 1 1 0 0 0 1 1 0 1 1 0 1 0 0 0 1 1 1 0 0 1 1 1 0 1 0 0 0 0 1 1 1 0
0 0 0 0 1 0 1 0 1 0 0 1 1 1 1 1 0 1 0 1 1 0 1 0 1 1 0 1 1 0 0 0 0 1 0 0
1 1 0 0 1 1 1 0 0 1 0 1 1 0 0 1 0 0 0 0 0 1 1 1 1 0 0 0 0 1 0 1 1 0 0 1
0 1 1 1 0 1 1 1 0 1 0 0 0 1 0 0 1 0 1 1 0 0 0 1 0 1 1 0 0 1 0 0 0 0 0 0
0 0 1 0 0 0 1 0 0 0 1 1 1 1 0 0 1 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 1 1]]
Accuracy for digits 0, 1 with batch size 64 and training percentage 0.7: 0.9965034965034965
Binary Labels: [[0 1 1 1 0 1 0 1 0 0 0 1 0 0 0 1 0 1 0 0 1 1 1 1 0 0 0 1 1 1 0 1 0 1 1 0
1 0 0 0 0 0 0 0 0 0 1 1 0 1 0 0 1 0 0 1 0 0 1 0 1 0 1 0 0 0 0 0 0 1 0 0
0 1 1 1 0 0 1 1 1 0 1 0 0 0 1 0 0 1 0 1 1 0 1 1 1 0 0 0 0 1 0 0 0 1 0 0
1 1 0 1 1 0 0 0 1 1 0 1 1 0 1 0 0 0 1 1 1 0 0 1 1 1 0 1 0 0 0 0 1 1 1 0
0 0 0 0 1 0 1 0 1 0 0 1 1 1 1 1 0 1 0 1 1 0 1 0 1 1 0 1 1 0 0 0 0 1 0 0
1 1 0 0 1 1 1 0 0 1 0 1 1 0 0 1 0 0 0 0 0 1 1 1 1 0 0 0 0 1 0 1 1 0 0 1
0 1 1 1 0 1 1 1 0 1 0 0 0 1 0 0 1 0 1 1 0 0 0 1 0 1 1 0 0 1 0 0 0 0 0 0
0 0 1 0 0 0 1 0 0 0 1 1 1 1 0 0 1 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 1 1]]
Accuracy for digits 0, 1 with batch size 64 and training percentage 0.9: 0.9965034965034965
Binary Labels: [[0 1 1 1 0 1 0 1 0 0 0 1 0 0 0 1 0 1 0 0 1 1 1 1 0 0 0 1 1 1 0 1 0 1 1 0
1 0 0 0 0 0 0 0 0 0 1 1 0 1 0 0 1 0 0 1 0 0 1 0 1 0 1 0 0 0 0 0 0 1 0 0
0 1 1 1 0 0 1 1 1 0 1 0 0 0 1 0 0 1 0 1 1 0 1 1 1 0 0 0 0 1 0 0 0 1 0 0
1 1 0 1 1 0 0 0 1 1 0 1 1 0 1 0 0 0 1 1 1 0 0 1 1 1 0 1 0 0 0 0 1 1 1 0
0 0 0 0 1 0 1 0 1 0 0 1 1 1 1 1 0 1 0 1 1 0 1 0 1 1 0 1 1 0 0 0 0 1 0 0
1 1 0 0 1 1 1 0 0 1 0 1 1 0 0 1 0 0 0 0 0 1 1 1 1 0 0 0 0 1 0 1 1 0 0 1
0 1 1 1 0 1 1 1 0 1 0 0 0 1 0 0 1 0 1 1 0 0 0 1 0 1 1 0 0 1 0 0 0 0 0 0
0 0 1 0 0 0 1 0 0 0 1 1 1 1 0 0 1 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 1 1]]
Accuracy for digits 0, 1 with batch size 128 and training percentage 0.5: 0.9965034965034965
Binary Labels: [[0 1 1 1 0 1 0 1 0 0 0 1 0 0 0 1 0 1 0 0 1 1 1 1 0 0 0 1 1 1 0 1 0 1 1 0
1 0 0 0 0 0 0 0 0 0 1 1 0 1 0 0 1 0 0 1 0 0 1 0 1 0 1 0 0 0 0 0 0 1 0 0

```
 0 1 1 1 0 0 1 1 1 0 1 0 0 0 1 0 0 1 0 1 1 0 1 1 1 0 0 0 0 1 0 0 0 1 0 0
 1 1 0 1 1 0 0 0 1 1 0 1 1 0 1 0 0 0 1 1 1 0 0 1 1 1 0 1 0 0 0 0 1 1 1 0
 0 0 0 0 1 0 1 0 1 0 0 1 1 1 1 1 0 1 0 1 1 0 1 0 1 1 0 1 1 0 0 0 0 1 0 0
 1 1 0 0 1 1 1 0 0 1 0 1 1 0 0 1 0 0 0 0 0 1 1 1 1 0 0 0 0 1 0 1 1 0 0 1
 0 1 1 1 0 1 1 1 0 1 0 0 0 1 0 0 1 0 1 1 0 0 0 1 0 1 1 0 0 1 0 0 0 0 0 0
 0 0 1 0 0 0 1 0 0 0 1 1 1 1 0 0 1 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 1 1]]
Accuracy for digits 0, 1 with batch size 128 and training percentage 0.7: 0.9965034965034965
Binary Labels: [[0 1 1 1 0 1 0 1 0 0 0 1 0 0 0 1 0 1 0 0 1 1 1 1 0 0 0 1 1 1 0 1 0 1 1 0
 1 0 0 0 0 0 0 0 0 0 1 1 0 1 0 0 1 0 0 1 0 0 1 0 1 0 1 0 0 0 0 0 0 1 0 0
 0 1 1 1 0 0 1 1 1 0 1 0 0 0 1 0 0 1 0 1 1 0 1 1 1 0 0 0 0 1 0 0 0 1 0 0
 1 1 0 1 1 0 0 0 1 1 0 1 1 0 1 0 0 0 1 1 1 0 0 1 1 1 0 1 0 0 0 0 1 1 1 0
 0 0 0 0 1 0 1 0 1 0 0 1 1 1 1 1 0 1 0 1 1 0 1 0 1 1 0 1 1 0 0 0 0 1 0 0
 1 1 0 0 1 1 1 0 0 1 0 1 1 0 0 1 0 0 0 0 0 1 1 1 1 0 0 0 0 1 0 1 1 0 0 1
 0 1 1 1 0 1 1 1 0 1 0 0 0 1 0 0 1 0 1 1 0 0 0 1 0 1 1 0 0 1 0 0 0 0 0 0
 0 0 1 0 0 0 1 0 0 0 1 1 1 1 0 0 1 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 1 1]]
Accuracy for digits 0, 1 with batch size 128 and training percentage 0.9: 0.9965034965034965
Data Split: X_train.shape = (257, 399), Y_train.shape = (1, 399), X_test.shape = (257, 172),
Y_test.shape = (1, 172)
Binary Labels: [[1 0 0 0 0 1 0 0 1 0 1 0 1 0 1 0 0 1 0 1 0 1 1 0 1 1 0 1 1 0 1 0 1 1 1 0
 0 1 0 0 1 1 1 0 1 1 1 0 1 0 1 1 0 0 0 1 0 0 0 1 1 1 0 0 1 1 1 0 0 0 0 0
 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 1 1 1 0 0 1 1 1 1 1 0 1 1 0 0 0 1
 1 0 1 0 0 0 0 1 1 0 1 1 0 1 0 0 1 0 1 0 0 0 0 0 0 0 0 1 0 0 1 0 0 1 1 0
 0 1 0 0 0 1 0 0 0 0 0 1 1 1 0 1 1 1 0 0 0 0 0 0 1 1 0 0]]
Accuracy for digits 0, 1 with batch size 32 and training percentage 0.5: 0.9941860465116279
Binary Labels: [[1 0 0 0 0 1 0 0 1 0 1 0 1 0 1 0 0 1 0 1 0 1 1 0 1 1 0 1 1 0 1 0 1 1 1 0
 0 1 0 0 1 1 1 0 1 1 1 0 1 0 1 1 0 0 0 1 0 0 0 1 1 1 0 0 1 1 1 0 0 0 0 0
 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 1 1 1 0 0 1 1 1 1 1 0 1 1 0 0 0 1
 1 0 1 0 0 0 0 1 1 0 1 1 0 1 0 0 1 0 1 0 0 0 0 0 0 0 0 1 0 0 1 0 0 1 1 0
 0 1 0 0 0 1 0 0 0 0 0 1 1 1 0 1 1 1 0 0 0 0 0 0 1 1 0 0]]
Accuracy for digits 0, 1 with batch size 32 and training percentage 0.7: 0.9941860465116279
Binary Labels: [[1 0 0 0 0 1 0 0 1 0 1 0 1 0 1 0 0 1 0 1 0 1 1 0 1 1 0 1 1 0 1 0 1 1 1 0
 0 1 0 0 1 1 1 0 1 1 1 0 1 0 1 1 0 0 0 1 0 0 0 1 1 1 0 0 1 1 1 0 0 0 0 0
 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 1 1 1 0 0 1 1 1 1 1 0 1 1 0 0 0 1
 1 0 1 0 0 0 0 1 1 0 1 1 0 1 0 0 1 0 1 0 0 0 0 0 0 0 0 1 0 0 1 0 0 1 1 0
 0 1 0 0 0 1 0 0 0 0 0 1 1 1 0 1 1 1 0 0 0 0 0 0 1 1 0 0]]
Accuracy for digits 0, 1 with batch size 32 and training percentage 0.9: 0.9941860465116279
Binary Labels: [[1 0 0 0 0 1 0 0 1 0 1 0 1 0 1 0 0 1 0 1 0 1 1 0 1 1 0 1 1 0 1 0 1 1 1 0
 0 1 0 0 1 1 1 0 1 1 1 0 1 0 1 1 0 0 0 1 0 0 0 1 1 1 0 0 1 1 1 0 0 0 0 0
 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 1 1 1 0 0 1 1 1 1 1 0 1 1 0 0 0 1
 1 0 1 0 0 0 0 1 1 0 1 1 0 1 0 0 1 0 1 0 0 0 0 0 0 0 0 1 0 0 1 0 0 1 1 0
 0 1 0 0 0 1 0 0 0 0 0 1 1 1 0 1 1 1 0 0 0 0 0 0 1 1 0 0]]
Accuracy for digits 0, 1 with batch size 64 and training percentage 0.5: 0.9941860465116279
Binary Labels: [[1 0 0 0 0 1 0 0 1 0 1 0 1 0 1 0 0 1 0 1 0 1 1 0 1 1 0 1 1 0 1 0 1 1 1 0
 0 1 0 0 1 1 1 0 1 1 1 0 1 0 1 1 0 0 0 1 0 0 0 1 1 1 0 0 1 1 1 0 0 0 0 0
 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 1 1 1 0 0 1 1 1 1 1 0 1 1 0 0 0 1
 1 0 1 0 0 0 0 1 1 0 1 1 0 1 0 0 1 0 1 0 0 0 0 0 0 0 0 1 0 0 1 0 0 1 1 0
 0 1 0 0 0 1 0 0 0 0 0 1 1 1 0 1 1 1 0 0 0 0 0 0 1 1 0 0]]
Accuracy for digits 0, 1 with batch size 64 and training percentage 0.7: 0.9941860465116279
Binary Labels: [[1 0 0 0 0 1 0 0 1 0 1 0 1 0 1 0 0 1 0 1 0 1 1 0 1 1 0 1 1 0 1 0 1 1 1 0
 0 1 0 0 1 1 1 0 1 1 1 0 1 0 1 1 0 0 0 1 0 0 0 1 1 1 0 0 1 1 1 0 0 0 0 0
 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 1 1 1 0 0 1 1 1 1 1 0 1 1 0 0 0 1
 1 0 1 0 0 0 0 1 1 0 1 1 0 1 0 0 1 0 1 0 0 0 0 0 0 0 0 1 0 0 1 0 0 1 1 0
```

```
  0 1 0 0 0 1 0 0 0 0 0 1 1 1 0 1 1 1 0 0 0 0 0 0 1 1 0 0]]
Accuracy for digits 0, 1 with batch size 64 and training percentage 0.9: 0.9941860465116279
Binary Labels: [[1 0 0 0 0 1 0 0 1 0 1 0 1 0 1 0 0 1 0 1 0 1 1 0 1 1 0 1 1 0 1 0 1 1 1 0
  0 1 0 0 1 1 1 0 1 1 1 0 1 0 1 1 0 0 0 1 0 0 0 1 1 1 0 0 1 1 1 0 0 0 0 0
  0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 1 1 1 0 0 1 1 1 1 1 0 1 1 0 0 0 1
  1 0 1 0 0 0 0 1 1 0 1 1 0 1 0 0 1 0 1 0 0 0 0 0 0 0 0 1 0 0 1 0 0 1 1 0
  0 1 0 0 0 1 0 0 0 0 0 1 1 1 0 1 1 1 0 0 0 0 0 0 1 1 0 0]]
Accuracy for digits 0, 1 with batch size 128 and training percentage 0.5: 0.9941860465116279
Binary Labels: [[1 0 0 0 0 1 0 0 1 0 1 0 1 0 1 0 0 1 0 1 0 1 1 0 1 1 0 1 1 0 1 0 1 1 1 0
  0 1 0 0 1 1 1 0 1 1 1 0 1 0 1 1 0 0 0 1 0 0 0 1 1 1 0 0 1 1 1 0 0 0 0 0
  0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 1 1 1 0 0 1 1 1 1 1 0 1 1 0 0 0 1
  1 0 1 0 0 0 0 1 1 0 1 1 0 1 0 0 1 0 1 0 0 0 0 0 0 0 0 1 0 0 1 0 0 1 1 0
  0 1 0 0 0 1 0 0 0 0 0 1 1 1 0 1 1 1 0 0 0 0 0 0 1 1 0 0]]
Accuracy for digits 0, 1 with batch size 128 and training percentage 0.7: 0.9941860465116279
Binary Labels: [[1 0 0 0 0 1 0 0 1 0 1 0 1 0 1 0 0 1 0 1 0 1 1 0 1 1 0 1 1 0 1 0 1 1 1 0
  0 1 0 0 1 1 1 0 1 1 1 0 1 0 1 1 0 0 0 1 0 0 0 1 1 1 0 0 1 1 1 0 0 0 0 0
  0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 1 1 1 0 0 1 1 1 1 1 0 1 1 0 0 0 1
  1 0 1 0 0 0 0 1 1 0 1 1 0 1 0 0 1 0 1 0 0 0 0 0 0 0 0 1 0 0 1 0 0 1 1 0
  0 1 0 0 0 1 0 0 0 0 0 1 1 1 0 1 1 1 0 0 0 0 0 0 1 1 0 0]]
Accuracy for digits 0, 1 with batch size 128 and training percentage 0.9: 0.9941860465116279
Data Split: X_train.shape = (257, 513), Y_train.shape = (1, 513), X_test.shape = (257, 58),
Y_test.shape = (1, 58)
Binary Labels: [[0 0 0 0 0 0 0 0 1 0 0 1 0 0 1 1 0 0 0 0 1 0 0 1 1 0 0 1 0 1 1 1 0 0 0 1
  0 1 0 0 1 0 1 0 1 0 0 0 1 0 0 0 0 0 1 0 0 1]]
Accuracy for digits 0, 1 with batch size 32 and training percentage 0.5: 1.0
Binary Labels: [[0 0 0 0 0 0 0 0 1 0 0 1 0 0 1 1 0 0 0 0 1 0 0 1 1 0 0 1 0 1 1 1 0 0 0 1
  0 1 0 0 1 0 1 0 1 0 0 0 1 0 0 0 0 0 1 0 0 1]]
Accuracy for digits 0, 1 with batch size 32 and training percentage 0.7: 1.0
Binary Labels: [[0 0 0 0 0 0 0 0 1 0 0 1 0 0 1 1 0 0 0 0 1 0 0 1 1 0 0 1 0 1 1 1 0 0 0 1
  0 1 0 0 1 0 1 0 1 0 0 0 1 0 0 0 0 0 1 0 0 1]]
Accuracy for digits 0, 1 with batch size 32 and training percentage 0.9: 1.0
Binary Labels: [[0 0 0 0 0 0 0 0 1 0 0 1 0 0 1 1 0 0 0 0 1 0 0 1 1 0 0 1 0 1 1 1 0 0 0 1
  0 1 0 0 1 0 1 0 1 0 0 0 1 0 0 0 0 0 1 0 0 1]]
Accuracy for digits 0, 1 with batch size 64 and training percentage 0.5: 1.0
Binary Labels: [[0 0 0 0 0 0 0 0 1 0 0 1 0 0 1 1 0 0 0 0 1 0 0 1 1 0 0 1 0 1 1 1 0 0 0 1
  0 1 0 0 1 0 1 0 1 0 0 0 1 0 0 0 0 0 1 0 0 1]]
Accuracy for digits 0, 1 with batch size 64 and training percentage 0.7: 1.0
Binary Labels: [[0 0 0 0 0 0 0 0 1 0 0 1 0 0 1 1 0 0 0 0 1 0 0 1 1 0 0 1 0 1 1 1 0 0 0 1
  0 1 0 0 1 0 1 0 1 0 0 0 1 0 0 0 0 0 1 0 0 1]]
Accuracy for digits 0, 1 with batch size 64 and training percentage 0.9: 1.0
Binary Labels: [[0 0 0 0 0 0 0 0 1 0 0 1 0 0 1 1 0 0 0 0 1 0 0 1 1 0 0 1 0 1 1 1 0 0 0 1
  0 1 0 0 1 0 1 0 1 0 0 0 1 0 0 0 0 0 1 0 0 1]]
Accuracy for digits 0, 1 with batch size 128 and training percentage 0.5: 1.0
Binary Labels: [[0 0 0 0 0 0 0 0 1 0 0 1 0 0 1 1 0 0 0 0 1 0 0 1 1 0 0 1 0 1 1 1 0 0 0 1
  0 1 0 0 1 0 1 0 1 0 0 0 1 0 0 0 0 0 1 0 0 1]]
Accuracy for digits 0, 1 with batch size 128 and training percentage 0.7: 1.0
Binary Labels: [[0 0 0 0 0 0 0 0 1 0 0 1 0 0 1 1 0 0 0 0 1 0 0 1 1 0 0 1 0 1 1 1 0 0 0 1
  0 1 0 0 1 0 1 0 1 0 0 0 1 0 0 0 0 0 1 0 0 1]]
Accuracy for digits 0, 1 with batch size 128 and training percentage 0.9: 1.0
Filtered Data for digits (2, 3): X_filtered.shape = (257, 333), Y_filtered.shape = (1, 333)
Data Split: X_train.shape = (257, 166), Y_train.shape = (1, 166), X_test.shape = (257, 167),
Y_test.shape = (1, 167)
```

Binary Labels: [[1 0 0 0 1 1 0 1 0 1 0 0 1 1 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 1 1 1 0
0 0 1 1 0 0 1 0 0 0 0 0 0 1 0 1 0 0 1 0 0 1 1 0 0 0 0 0 0 0 1 0 0 0 1 1
1 1 1 0 1 1 1 0 1 0 0 1 0 0 1 0 0 0 0 0 1 1 1 0 1 0 1 0 1 1 1 1 1 1 0 1 0
0 1 1 0 1 0 0 0 0 0 1 0 0 0 0 1 0 1 1 0 1 1 1 0 1 1 0 1 0 0 0 0 1 0 0 1
0 0 0 1 1 0 0 0 0 0 0 0 1 0 1 1 1 1 0 1 0 0 0]]
Accuracy for digits 2, 3 with batch size 32 and training percentage 0.5: 0.39520958083832336
Binary Labels: [[1 0 0 0 1 1 0 1 0 1 0 0 1 1 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 1 1 1 0
0 0 1 1 0 0 1 0 0 0 0 0 0 1 0 1 0 0 1 0 0 1 1 0 0 0 0 0 0 0 1 0 0 0 1 1
1 1 1 0 1 1 1 0 1 0 0 1 0 0 1 0 0 0 0 0 1 1 1 0 1 0 1 0 1 1 1 1 1 1 0 1 0
0 1 1 0 1 0 0 0 0 0 1 0 0 0 0 1 0 1 1 0 1 1 1 0 1 1 0 1 0 0 0 0 1 0 0 1
0 0 0 1 1 0 0 0 0 0 0 0 1 0 1 1 1 1 0 1 0 0 0]]
Accuracy for digits 2, 3 with batch size 32 and training percentage 0.7: 0.39520958083832336
Binary Labels: [[1 0 0 0 1 1 0 1 0 1 0 0 1 1 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 1 1 1 0
0 0 1 1 0 0 1 0 0 0 0 0 0 1 0 1 0 0 1 0 0 1 1 0 0 0 0 0 0 0 1 0 0 0 1 1
1 1 1 0 1 1 1 0 1 0 0 1 0 0 1 0 0 0 0 0 1 1 1 0 1 0 1 0 1 1 1 1 1 1 0 1 0
0 1 1 0 1 0 0 0 0 0 1 0 0 0 0 1 0 1 1 0 1 1 1 0 1 1 0 1 0 0 0 0 1 0 0 1
0 0 0 1 1 0 0 0 0 0 0 0 1 0 1 1 1 1 0 1 0 0 0]]
Accuracy for digits 2, 3 with batch size 32 and training percentage 0.9: 0.39520958083832336
Binary Labels: [[1 0 0 0 1 1 0 1 0 1 0 0 1 1 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 1 1 1 0
0 0 1 1 0 0 1 0 0 0 0 0 0 1 0 1 0 0 1 0 0 1 1 0 0 0 0 0 0 0 1 0 0 0 1 1
1 1 1 0 1 1 1 0 1 0 0 1 0 0 1 0 0 0 0 0 1 1 1 0 1 0 1 0 1 1 1 1 1 1 0 1 0
0 1 1 0 1 0 0 0 0 0 1 0 0 0 0 1 0 1 1 0 1 1 1 0 1 1 0 1 0 0 0 0 1 0 0 1
0 0 0 1 1 0 0 0 0 0 0 0 1 0 1 1 1 1 0 1 0 0 0]]
Accuracy for digits 2, 3 with batch size 64 and training percentage 0.5: 0.39520958083832336
Binary Labels: [[1 0 0 0 1 1 0 1 0 1 0 0 1 1 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 1 1 1 0
0 0 1 1 0 0 1 0 0 0 0 0 0 1 0 1 0 0 1 0 0 1 1 0 0 0 0 0 0 0 1 0 0 0 1 1
1 1 1 0 1 1 1 0 1 0 0 1 0 0 1 0 0 0 0 0 1 1 1 0 1 0 1 0 1 1 1 1 1 1 0 1 0
0 1 1 0 1 0 0 0 0 0 1 0 0 0 0 1 0 1 1 0 1 1 1 0 1 1 0 1 0 0 0 0 1 0 0 1
0 0 0 1 1 0 0 0 0 0 0 0 1 0 1 1 1 1 0 1 0 0 0]]
Accuracy for digits 2, 3 with batch size 64 and training percentage 0.7: 0.39520958083832336
Binary Labels: [[1 0 0 0 1 1 0 1 0 1 0 0 1 1 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 1 1 1 0
0 0 1 1 0 0 1 0 0 0 0 0 0 1 0 1 0 0 1 0 0 1 1 0 0 0 0 0 0 0 1 0 0 0 1 1
1 1 1 0 1 1 1 0 1 0 0 1 0 0 1 0 0 0 0 0 1 1 1 0 1 0 1 0 1 1 1 1 1 1 0 1 0
0 1 1 0 1 0 0 0 0 0 1 0 0 0 0 1 0 1 1 0 1 1 1 0 1 1 0 1 0 0 0 0 1 0 0 1
0 0 0 1 1 0 0 0 0 0 0 0 1 0 1 1 1 1 0 1 0 0 0]]
Accuracy for digits 2, 3 with batch size 64 and training percentage 0.9: 0.39520958083832336
Binary Labels: [[1 0 0 0 1 1 0 1 0 1 0 0 1 1 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 1 1 1 0
0 0 1 1 0 0 1 0 0 0 0 0 0 1 0 1 0 0 1 0 0 1 1 0 0 0 0 0 0 0 1 0 0 0 1 1
1 1 1 0 1 1 1 0 1 0 0 1 0 0 1 0 0 0 0 0 1 1 1 0 1 0 1 0 1 1 1 1 1 1 0 1 0
0 1 1 0 1 0 0 0 0 0 1 0 0 0 0 1 0 1 1 0 1 1 1 0 1 1 0 1 0 0 0 0 1 0 0 1
0 0 0 1 1 0 0 0 0 0 0 0 1 0 1 1 1 1 0 1 0 0 0]]
Accuracy for digits 2, 3 with batch size 128 and training percentage 0.5: 0.39520958083832336
Binary Labels: [[1 0 0 0 1 1 0 1 0 1 0 0 1 1 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 1 1 1 0
0 0 1 1 0 0 1 0 0 0 0 0 0 1 0 1 0 0 1 0 0 1 1 0 0 0 0 0 0 0 1 0 0 0 1 1
1 1 1 0 1 1 1 0 1 0 0 1 0 0 1 0 0 0 0 0 1 1 1 0 1 0 1 0 1 1 1 1 1 1 0 1 0
0 1 1 0 1 0 0 0 0 0 1 0 0 0 0 1 0 1 1 0 1 1 1 0 1 1 0 1 0 0 0 0 1 0 0 1
0 0 0 1 1 0 0 0 0 0 0 0 1 0 1 1 1 1 0 1 0 0 0]]
Accuracy for digits 2, 3 with batch size 128 and training percentage 0.7: 0.39520958083832336
Binary Labels: [[1 0 0 0 1 1 0 1 0 1 0 0 1 1 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 1 1 1 0
0 0 1 1 0 0 1 0 0 0 0 0 0 1 0 1 0 0 1 0 0 1 1 0 0 0 0 0 0 0 1 0 0 0 1 1
1 1 1 0 1 1 1 0 1 0 0 1 0 0 1 0 0 0 0 0 1 1 1 0 1 0 1 0 1 1 1 1 1 1 0 1 0
0 1 1 0 1 0 0 0 0 0 1 0 0 0 0 1 0 1 1 0 1 1 1 0 1 1 0 1 0 0 0 0 1 0 0 1

0 0 0 1 1 0 0 0 0 0 0 0 1 0 1 1 1 1 0 1 0 0 0]]
Accuracy for digits 2, 3 with batch size 128 and training percentage 0.9: 0.39520958083832336
Data Split: X_train.shape = (257, 233), Y_train.shape = (1, 233), X_test.shape = (257, 100), Y_test.shape = (1, 100)
Binary Labels: [[0 0 0 0 0 1 0 0 0 0 0 0 0 1 1 0 1 0 0 1 0 0 0 0 0 0 0 1 0 0 1 0 0 1 1 0
0 0 1 1 1 1 0 0 1 0 1 0 0 0 0 0 1 1 1 0 1 0 1 1 1 1 1 1 1 0 1 0 1 1 1 1
0 1 0 0 0 1 1 1 0 1 1 0 1 0 0 0 0 1 0 1 1 1 1 1 0 0 1 1]]
Accuracy for digits 2, 3 with batch size 32 and training percentage 0.5: 0.46
Binary Labels: [[0 0 0 0 0 1 0 0 0 0 0 0 0 1 1 0 1 0 0 1 0 0 0 0 0 0 0 1 0 0 1 0 0 1 1 0
0 0 1 1 1 1 0 0 1 0 1 0 0 0 0 0 1 1 1 0 1 0 1 1 1 1 1 1 1 0 1 0 1 1 1 1
0 1 0 0 0 1 1 1 0 1 1 0 1 0 0 0 0 1 0 1 1 1 1 1 0 0 1 1]]
Accuracy for digits 2, 3 with batch size 32 and training percentage 0.7: 0.46
Binary Labels: [[0 0 0 0 0 1 0 0 0 0 0 0 0 1 1 0 1 0 0 1 0 0 0 0 0 0 0 1 0 0 1 0 0 1 1 0
0 0 1 1 1 1 0 0 1 0 1 0 0 0 0 0 1 1 1 0 1 0 1 1 1 1 1 1 1 0 1 0 1 1 1 1
0 1 0 0 0 1 1 1 0 1 1 0 1 0 0 0 0 1 0 1 1 1 1 1 0 0 1 1]]
Accuracy for digits 2, 3 with batch size 32 and training percentage 0.9: 0.46
Binary Labels: [[0 0 0 0 0 1 0 0 0 0 0 0 0 1 1 0 1 0 0 1 0 0 0 0 0 0 0 1 0 0 1 0 0 1 1 0
0 0 1 1 1 1 0 0 1 0 1 0 0 0 0 0 1 1 1 0 1 0 1 1 1 1 1 1 1 0 1 0 1 1 1 1
0 1 0 0 0 1 1 1 0 1 1 0 1 0 0 0 0 1 0 1 1 1 1 1 0 0 1 1]]
Accuracy for digits 2, 3 with batch size 64 and training percentage 0.5: 0.46
Binary Labels: [[0 0 0 0 0 1 0 0 0 0 0 0 0 1 1 0 1 0 0 1 0 0 0 0 0 0 0 1 0 0 1 0 0 1 1 0
0 0 1 1 1 1 0 0 1 0 1 0 0 0 0 0 1 1 1 0 1 0 1 1 1 1 1 1 1 0 1 0 1 1 1 1
0 1 0 0 0 1 1 1 0 1 1 0 1 0 0 0 0 1 0 1 1 1 1 1 0 0 1 1]]
Accuracy for digits 2, 3 with batch size 64 and training percentage 0.7: 0.46
Binary Labels: [[0 0 0 0 0 1 0 0 0 0 0 0 0 1 1 0 1 0 0 1 0 0 0 0 0 0 0 1 0 0 1 0 0 1 1 0
0 0 1 1 1 1 0 0 1 0 1 0 0 0 0 0 1 1 1 0 1 0 1 1 1 1 1 1 1 0 1 0 1 1 1 1
0 1 0 0 0 1 1 1 0 1 1 0 1 0 0 0 0 1 0 1 1 1 1 1 0 0 1 1]]
Accuracy for digits 2, 3 with batch size 64 and training percentage 0.9: 0.46
Binary Labels: [[0 0 0 0 0 1 0 0 0 0 0 0 0 1 1 0 1 0 0 1 0 0 0 0 0 0 0 1 0 0 1 0 0 1 1 0
0 0 1 1 1 1 0 0 1 0 1 0 0 0 0 0 1 1 1 0 1 0 1 1 1 1 1 1 1 0 1 0 1 1 1 1
0 1 0 0 0 1 1 1 0 1 1 0 1 0 0 0 0 1 0 1 1 1 1 1 0 0 1 1]]
Accuracy for digits 2, 3 with batch size 128 and training percentage 0.5: 0.46
Binary Labels: [[0 0 0 0 0 1 0 0 0 0 0 0 0 1 1 0 1 0 0 1 0 0 0 0 0 0 0 1 0 0 1 0 0 1 1 0
0 0 1 1 1 1 0 0 1 0 1 0 0 0 0 0 1 1 1 0 1 0 1 1 1 1 1 1 1 0 1 0 1 1 1 1
0 1 0 0 0 1 1 1 0 1 1 0 1 0 0 0 0 1 0 1 1 1 1 1 0 0 1 1]]
Accuracy for digits 2, 3 with batch size 128 and training percentage 0.7: 0.46
Binary Labels: [[0 0 0 0 0 1 0 0 0 0 0 0 0 1 1 0 1 0 0 1 0 0 0 0 0 0 0 1 0 0 1 0 0 1 1 0
0 0 1 1 1 1 0 0 1 0 1 0 0 0 0 0 1 1 1 0 1 0 1 1 1 1 1 1 1 0 1 0 1 1 1 1
0 1 0 0 0 1 1 1 0 1 1 0 1 0 0 0 0 1 0 1 1 1 1 1 0 0 1 1]]
Accuracy for digits 2, 3 with batch size 128 and training percentage 0.9: 0.46
Data Split: X_train.shape = (257, 299), Y_train.shape = (1, 299), X_test.shape = (257, 34), Y_test.shape = (1, 34)
Binary Labels: [[0 0 1 1 1 1 1 1 1 0 0 0 1 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 1 0 1 0 0 1]]
Accuracy for digits 2, 3 with batch size 32 and training percentage 0.5: 0.4117647058823529
Binary Labels: [[0 0 1 1 1 1 1 1 1 0 0 0 1 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 1 0 1 0 0 1]]
Accuracy for digits 2, 3 with batch size 32 and training percentage 0.7: 0.4117647058823529
Binary Labels: [[0 0 1 1 1 1 1 1 1 0 0 0 1 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 1 0 1 0 0 1]]
Accuracy for digits 2, 3 with batch size 32 and training percentage 0.9: 0.4117647058823529
Binary Labels: [[0 0 1 1 1 1 1 1 1 0 0 0 1 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 1 0 1 0 0 1]]
Accuracy for digits 2, 3 with batch size 64 and training percentage 0.5: 0.4117647058823529
Binary Labels: [[0 0 1 1 1 1 1 1 1 0 0 0 1 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 1 0 1 0 0 1]]
Accuracy for digits 2, 3 with batch size 64 and training percentage 0.7: 0.4117647058823529

Binary Labels: [[0 0 1 1 1 1 1 1 1 0 0 0 1 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 1 0 1 0 0 1]]
Accuracy for digits 2, 3 with batch size 64 and training percentage 0.9: 0.4117647058823529
Binary Labels: [[0 0 1 1 1 1 1 1 1 0 0 0 1 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 1 0 1 0 0 1]]
Accuracy for digits 2, 3 with batch size 128 and training percentage 0.5: 0.4117647058823529
Binary Labels: [[0 0 1 1 1 1 1 1 1 0 0 0 1 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 1 0 1 0 0 1]]
Accuracy for digits 2, 3 with batch size 128 and training percentage 0.7: 0.4117647058823529
Binary Labels: [[0 0 1 1 1 1 1 1 1 0 0 0 1 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 1 0 1 0 0 1]]
Accuracy for digits 2, 3 with batch size 128 and training percentage 0.9: 0.4117647058823529
Filtered Data for digits (4, 5): X_filtered.shape = (257, 210), Y_filtered.shape = (1, 210)
Data Split: X_train.shape = (257, 105), Y_train.shape = (1, 105), X_test.shape = (257, 105), Y_test.shape = (1, 105)
Binary Labels: [[0 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 1 0 1 0 1 0 1 0 0 0 0 0 1 1 1 0 1 0 1
  0 0 0 0 0 0 1 0 0 0 0 1 1 0 0 1 0 0 0 0 0 1 1 1 0 1 0 1 1 1 0 1 0 0 1 1
  1 0 1 0 0 0 0 0 0 0 1 1 1 0 1 1 1 0 0 0 0 0 1 0 1 1 0 0 0 1 1 0]]
Accuracy for digits 4, 5 with batch size 32 and training percentage 0.5: 0.3904761904761905
Binary Labels: [[0 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 1 0 1 0 1 0 1 0 0 0 0 0 1 1 1 0 1 0 1
  0 0 0 0 0 0 1 0 0 0 0 1 1 0 0 1 0 0 0 0 0 1 1 1 0 1 0 1 1 1 0 1 0 0 1 1
  1 0 1 0 0 0 0 0 0 0 1 1 1 0 1 1 1 0 0 0 0 0 1 0 1 1 0 0 0 1 1 0]]
Accuracy for digits 4, 5 with batch size 32 and training percentage 0.7: 0.3904761904761905
Binary Labels: [[0 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 1 0 1 0 1 0 1 0 0 0 0 0 1 1 1 0 1 0 1
  0 0 0 0 0 0 1 0 0 0 0 1 1 0 0 1 0 0 0 0 0 1 1 1 0 1 0 1 1 1 0 1 0 0 1 1
  1 0 1 0 0 0 0 0 0 0 1 1 1 0 1 1 1 0 0 0 0 0 1 0 1 1 0 0 0 1 1 0]]
Accuracy for digits 4, 5 with batch size 32 and training percentage 0.9: 0.3904761904761905
Binary Labels: [[0 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 1 0 1 0 1 0 1 0 0 0 0 0 1 1 1 0 1 0 1
  0 0 0 0 0 0 1 0 0 0 0 1 1 0 0 1 0 0 0 0 0 1 1 1 0 1 0 1 1 1 0 1 0 0 1 1
  1 0 1 0 0 0 0 0 0 0 1 1 1 0 1 1 1 0 0 0 0 0 1 0 1 1 0 0 0 1 1 0]]
Accuracy for digits 4, 5 with batch size 64 and training percentage 0.5: 0.3904761904761905
Binary Labels: [[0 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 1 0 1 0 1 0 1 0 0 0 0 0 1 1 1 0 1 0 1
  0 0 0 0 0 0 1 0 0 0 0 1 1 0 0 1 0 0 0 0 0 1 1 1 0 1 0 1 1 1 0 1 0 0 1 1
  1 0 1 0 0 0 0 0 0 0 1 1 1 0 1 1 1 0 0 0 0 0 1 0 1 1 0 0 0 1 1 0]]
Accuracy for digits 4, 5 with batch size 64 and training percentage 0.7: 0.3904761904761905
Binary Labels: [[0 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 1 0 1 0 1 0 1 0 0 0 0 0 1 1 1 0 1 0 1
  0 0 0 0 0 0 1 0 0 0 0 1 1 0 0 1 0 0 0 0 0 1 1 1 0 1 0 1 1 1 0 1 0 0 1 1
  1 0 1 0 0 0 0 0 0 0 1 1 1 0 1 1 1 0 0 0 0 0 1 0 1 1 0 0 0 1 1 0]]
Accuracy for digits 4, 5 with batch size 64 and training percentage 0.9: 0.3904761904761905
Binary Labels: [[0 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 1 0 1 0 1 0 1 0 0 0 0 0 1 1 1 0 1 0 1
  0 0 0 0 0 0 1 0 0 0 0 1 1 0 0 1 0 0 0 0 0 1 1 1 0 1 0 1 1 1 0 1 0 0 1 1
  1 0 1 0 0 0 0 0 0 0 1 1 1 0 1 1 1 0 0 0 0 0 1 0 1 1 0 0 0 1 1 0]]
Accuracy for digits 4, 5 with batch size 128 and training percentage 0.5: 0.3904761904761905
Binary Labels: [[0 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 1 0 1 0 1 0 1 0 0 0 0 0 1 1 1 0 1 0 1
  0 0 0 0 0 0 1 0 0 0 0 1 1 0 0 1 0 0 0 0 0 1 1 1 0 1 0 1 1 1 0 1 0 0 1 1
  1 0 1 0 0 0 0 0 0 0 1 1 1 0 1 1 1 0 0 0 0 0 1 0 1 1 0 0 0 1 1 0]]
Accuracy for digits 4, 5 with batch size 128 and training percentage 0.7: 0.3904761904761905
Binary Labels: [[0 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 1 0 1 0 1 0 1 0 0 0 0 0 1 1 1 0 1 0 1
  0 0 0 0 0 0 1 0 0 0 0 1 1 0 0 1 0 0 0 0 0 1 1 1 0 1 0 1 1 1 0 1 0 0 1 1
  1 0 1 0 0 0 0 0 0 0 1 1 1 0 1 1 1 0 0 0 0 0 1 0 1 1 0 0 0 1 1 0]]
Accuracy for digits 4, 5 with batch size 128 and training percentage 0.9: 0.3904761904761905
Data Split: X_train.shape = (257, 147), Y_train.shape = (1, 147), X_test.shape = (257, 63), Y_test.shape = (1, 63)
Binary Labels: [[1 0 0 1 1 0 0 0 1 1 0 0 0 0 0 0 1 0 1 1 0 0 1 1 0 1 0 0 0 0 0 0 0 1 1 1
  1 1 1 0 0 1 1 1 1 0 1 1 0 1 1 1 0 0 0 1 0 1 0 1 1 1 0 0]]
Accuracy for digits 4, 5 with batch size 32 and training percentage 0.5: 0.47619047619047616

Binary Labels: [[1 0 0 1 1 0 0 0 1 1 0 0 0 0 0 0 1 0 1 1 0 0 1 1 0 1 0 0 0 0 0 0 0 1 1 1
  1 1 1 0 0 1 1 1 1 0 1 1 0 1 1 1 0 0 0 1 0 1 0 1 1 0 0]]
Accuracy for digits 4, 5 with batch size 32 and training percentage 0.7: 0.47619047619047616
Binary Labels: [[1 0 0 1 1 0 0 0 1 1 0 0 0 0 0 0 1 0 1 1 0 0 1 1 0 1 0 0 0 0 0 0 0 1 1 1
  1 1 1 0 0 1 1 1 1 0 1 1 0 1 1 1 0 0 0 1 0 1 0 1 1 0 0]]
Accuracy for digits 4, 5 with batch size 32 and training percentage 0.9: 0.47619047619047616
Binary Labels: [[1 0 0 1 1 0 0 0 1 1 0 0 0 0 0 0 1 0 1 1 0 0 1 1 0 1 0 0 0 0 0 0 0 1 1 1
  1 1 1 0 0 1 1 1 1 0 1 1 0 1 1 1 0 0 0 1 0 1 0 1 1 0 0]]
Accuracy for digits 4, 5 with batch size 64 and training percentage 0.5: 0.47619047619047616
Binary Labels: [[1 0 0 1 1 0 0 0 1 1 0 0 0 0 0 0 1 0 1 1 0 0 1 1 0 1 0 0 0 0 0 0 0 1 1 1
  1 1 1 0 0 1 1 1 1 0 1 1 0 1 1 1 0 0 0 1 0 1 0 1 1 0 0]]
Accuracy for digits 4, 5 with batch size 64 and training percentage 0.7: 0.47619047619047616
Binary Labels: [[1 0 0 1 1 0 0 0 1 1 0 0 0 0 0 0 1 0 1 1 0 0 1 1 0 1 0 0 0 0 0 0 0 1 1 1
  1 1 1 0 0 1 1 1 1 0 1 1 0 1 1 1 0 0 0 1 0 1 0 1 1 0 0]]
Accuracy for digits 4, 5 with batch size 64 and training percentage 0.9: 0.47619047619047616
Binary Labels: [[1 0 0 1 1 0 0 0 1 1 0 0 0 0 0 0 1 0 1 1 0 0 1 1 0 1 0 0 0 0 0 0 0 1 1 1
  1 1 1 0 0 1 1 1 1 0 1 1 0 1 1 1 0 0 0 1 0 1 0 1 1 0 0]]
Accuracy for digits 4, 5 with batch size 128 and training percentage 0.5: 0.47619047619047616
Binary Labels: [[1 0 0 1 1 0 0 0 1 1 0 0 0 0 0 0 1 0 1 1 0 0 1 1 0 1 0 0 0 0 0 0 0 1 1 1
  1 1 1 0 0 1 1 1 1 0 1 1 0 1 1 1 0 0 0 1 0 1 0 1 1 0 0]]
Accuracy for digits 4, 5 with batch size 128 and training percentage 0.7: 0.47619047619047616
Binary Labels: [[1 0 0 1 1 0 0 0 1 1 0 0 0 0 0 0 1 0 1 1 0 0 1 1 0 1 0 0 0 0 0 0 0 1 1 1
  1 1 1 0 0 1 1 1 1 0 1 1 0 1 1 1 0 0 0 1 0 1 0 1 1 0 0]]
Accuracy for digits 4, 5 with batch size 128 and training percentage 0.9: 0.47619047619047616
Data Split: X_train.shape = (257, 189), Y_train.shape = (1, 189), X_test.shape = (257, 21),
Y_test.shape = (1, 21)
Binary Labels: [[1 1 0 1 0 0 0 0 1 0 1 0 0 0 0 1 0 0 1 0 1]]
Accuracy for digits 4, 5 with batch size 32 and training percentage 0.5: 0.38095238095238093
Binary Labels: [[1 1 0 1 0 0 0 0 1 0 1 0 0 0 0 1 0 0 1 0 1]]
Accuracy for digits 4, 5 with batch size 32 and training percentage 0.7: 0.38095238095238093
Binary Labels: [[1 1 0 1 0 0 0 0 1 0 1 0 0 0 0 1 0 0 1 0 1]]
Accuracy for digits 4, 5 with batch size 32 and training percentage 0.9: 0.38095238095238093
Binary Labels: [[1 1 0 1 0 0 0 0 1 0 1 0 0 0 0 1 0 0 1 0 1]]
Accuracy for digits 4, 5 with batch size 64 and training percentage 0.5: 0.38095238095238093
Binary Labels: [[1 1 0 1 0 0 0 0 1 0 1 0 0 0 0 1 0 0 1 0 1]]
Accuracy for digits 4, 5 with batch size 64 and training percentage 0.7: 0.38095238095238093
Binary Labels: [[1 1 0 1 0 0 0 0 1 0 1 0 0 0 0 1 0 0 1 0 1]]
Accuracy for digits 4, 5 with batch size 64 and training percentage 0.9: 0.38095238095238093
Binary Labels: [[1 1 0 1 0 0 0 0 1 0 1 0 0 0 0 1 0 0 1 0 1]]
Accuracy for digits 4, 5 with batch size 128 and training percentage 0.5: 0.38095238095238093
Binary Labels: [[1 1 0 1 0 0 0 0 1 0 1 0 0 0 0 1 0 0 1 0 1]]
Accuracy for digits 4, 5 with batch size 128 and training percentage 0.7: 0.38095238095238093
Binary Labels: [[1 1 0 1 0 0 0 0 1 0 1 0 0 0 0 1 0 0 1 0 1]]
Accuracy for digits 4, 5 with batch size 128 and training percentage 0.9: 0.38095238095238093

ACCURACY

{(0, 1): {32: {0.5: 1.0, 0.7: 1.0, 0.9: 1.0}, 64: {0.5: 1.0, 0.7: 1.0, 0.9: 1.0}, 128: {0.5: 1.0, 0.7: 1.0, 0.9: 1.0}},
(2, 3): {32: {0.5: 0.4117647058823529, 0.7: 0.4117647058823529, 0.9: 0.4117647058823529}, 64: {0.5: 0.4117647058823529, 0.7: 0.4117647058823529, 0.9: 0.4117647058823529}, 128: {0.5: 0.4117647058823529, 0.7: 0.4117647058823529, 0.9: 0.4117647058823529}},
(4, 5): {32: {0.5: 0.38095238095238093, 0.7: 0.38095238095238093, 0.9: 0.38095238095238093}, 64: {0.5: 0.38095238095238093, 0.7: 0.38095238095238093, 0.9: 0.38095238095238093}, 128: {0.5: 0.38095238095238093, 0.7: 0.38095238095238093, 0.9: 0.38095238095238093}}}