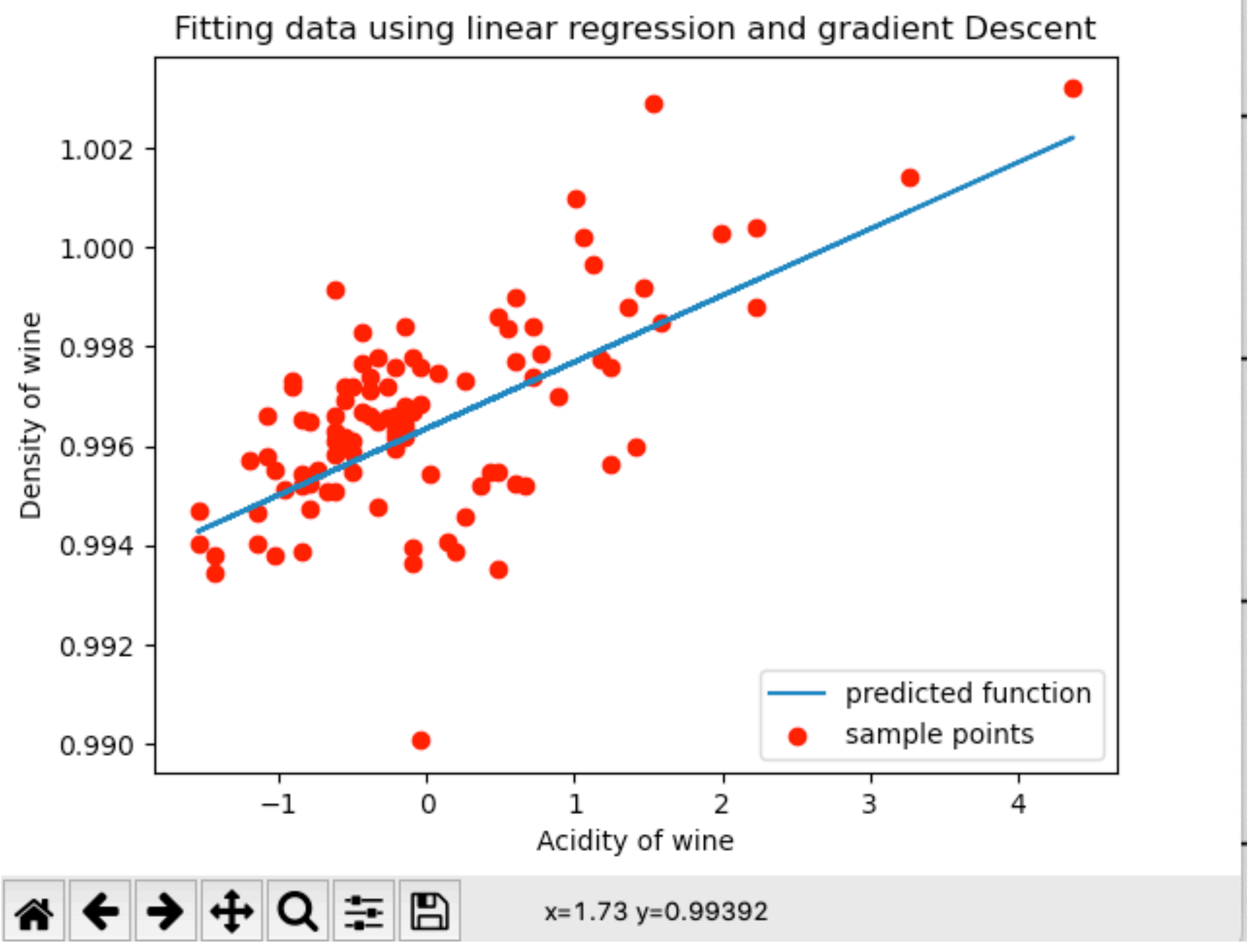
1) The table below gives a clear idea how convergence varies wrt the learning criterion and eps.

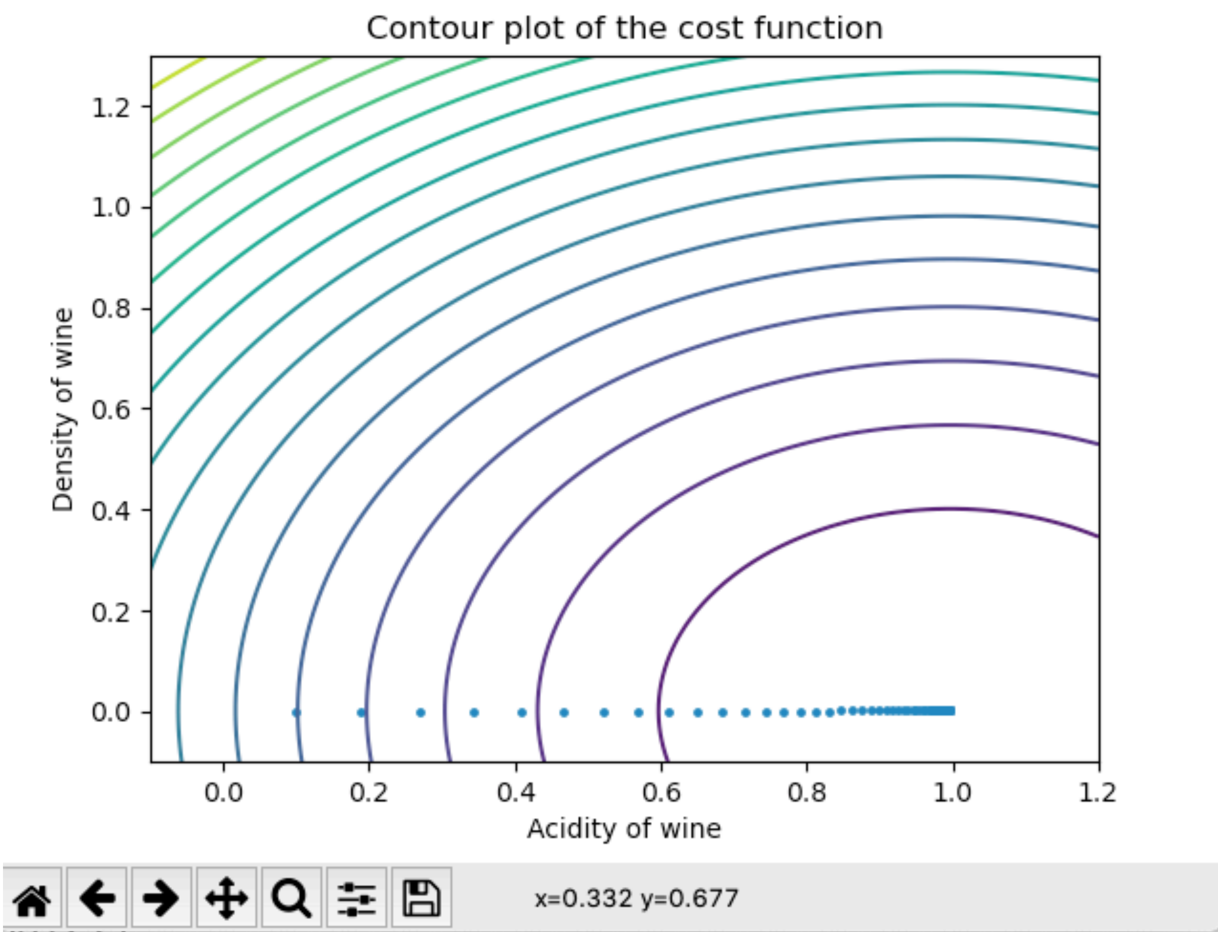
Learning_rate	Eps	epochs	Parameters	Error	Error difference
0.001	1E-09	6902	[0.995621180.00133885]	1.69370E-06	
0.01	1E-09	803	[0.9963085,0.00133978]	1.24333E-06	9.85678E-10
0.1	1E-08	78	[0.996351290.00133983]	1.230919E-06	8.47481E-09
1	1E-08	2	[0.9966201 0.0013402]	1.1947898E-06	2.9646E-21

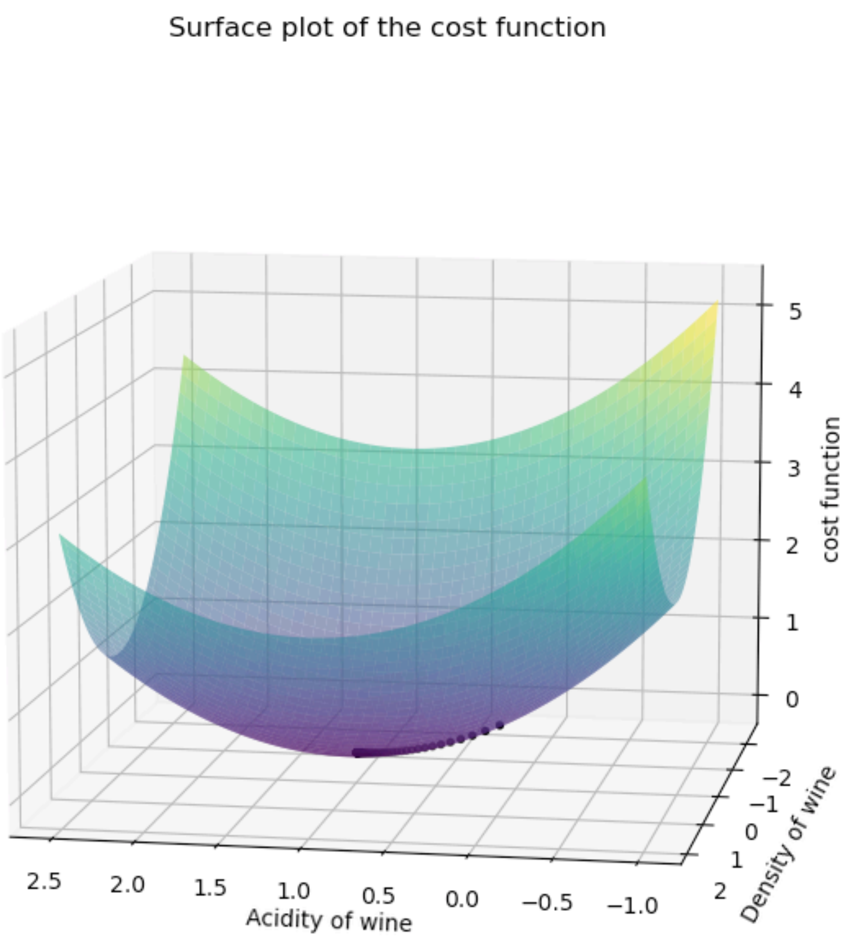
So, I could have chosen the learning rate to be 1 as it converges faster with eps=1e-8. But for visualization and plotting, I chose **learning_rate= 0.1 with eps=1e-9.**

Command to run: **python q1.py --eps 1e-9 --lr 0.1**

```
Parameter vector is:
[[0.99653574]
 [0.00134008]]
No of iterations is: 89
```


Fitting data using linear regression and gradient Descent
1.002
1.000
0.998
0.996
0.994
0.992
0.990
Density of wine
−1
0
1
2
3
4
Acidity of wine
predicted function
sample points
x=1.73 y=0.99392

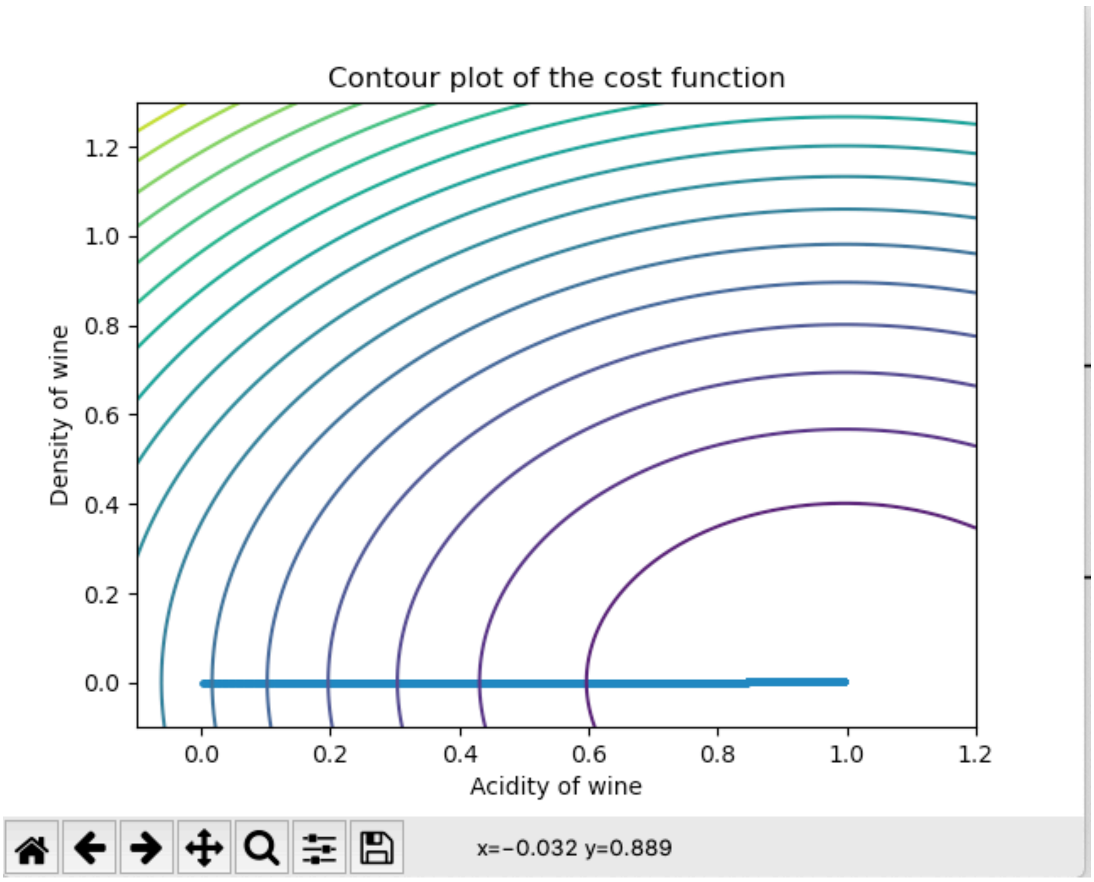

Contour plot of the cost function
1.2
1.0
0.8
0.6
0.4
0.2
0.0
Density of wine
0.0
0.2
0.4
0.6
0.8
1.0
1.2
Acidity of wine
x=0.332 y=0.677



Learning_rate=0.001

Command to run: python q1.py --eps 1e-9 --lr 0.001

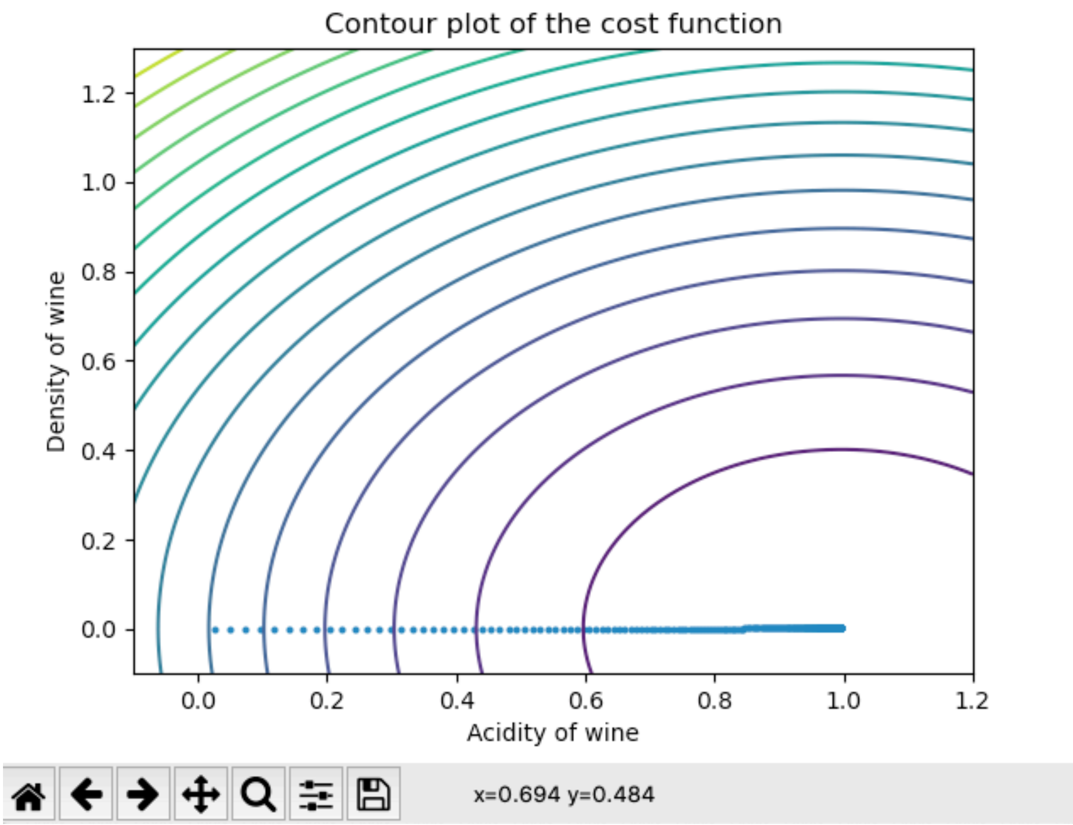
```
No of iterations is: 6902
```



Learning_rate=0.025

Command to run: python q1.py --eps 1e-9 --lr 0.025

```
No of iterations is: 338
```



As we can see, the number of iterations increases manifold as we keep decreasing the learning_rate. Also, if the learning _rate is small, then threshold/eps value has to be kept lower to get the same precision in the final parameters.

Q2)python q2.py --eps 1e-8 --bs 1

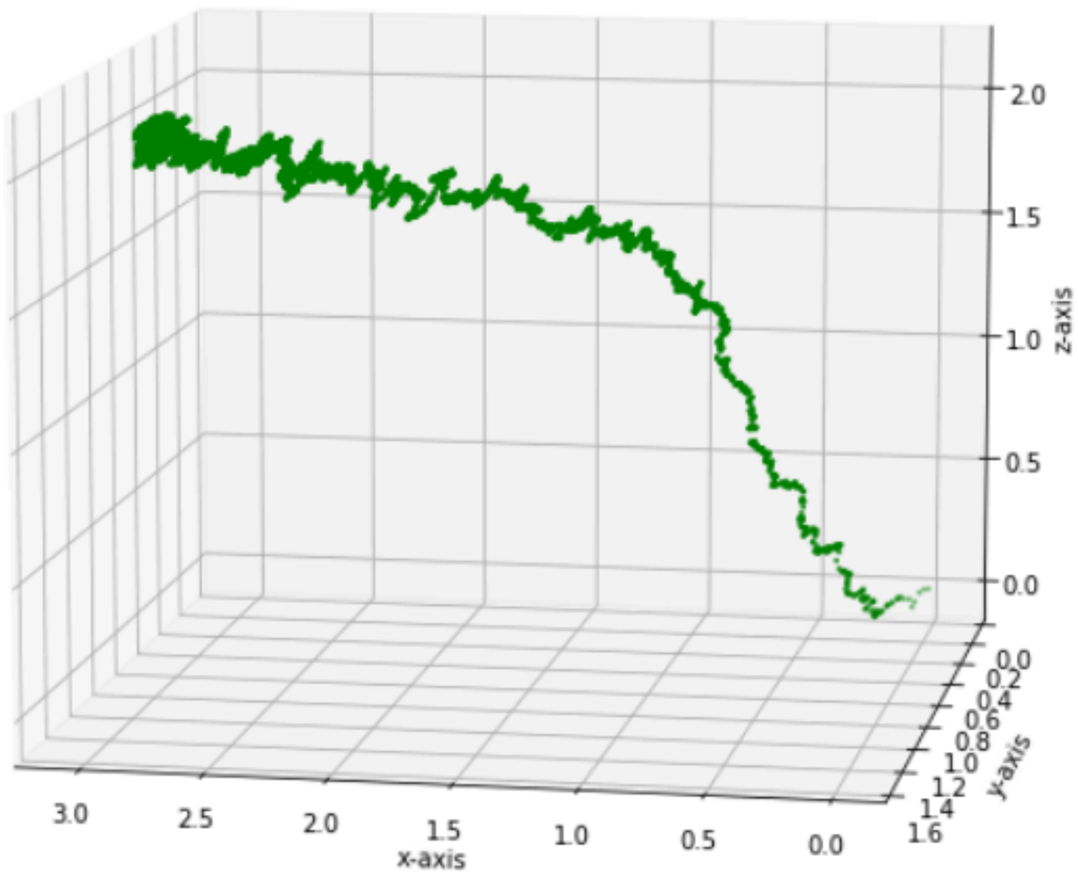
Note: K- it is the no of iterations after which we take the average of the cost so remove the spurious oscillations.

```
Error wrt the original hypothesis is: [[1.96589384]]
```

Batch_size=1

```
Error wrt the learned hypothesis is: [[1.97123999]]
```

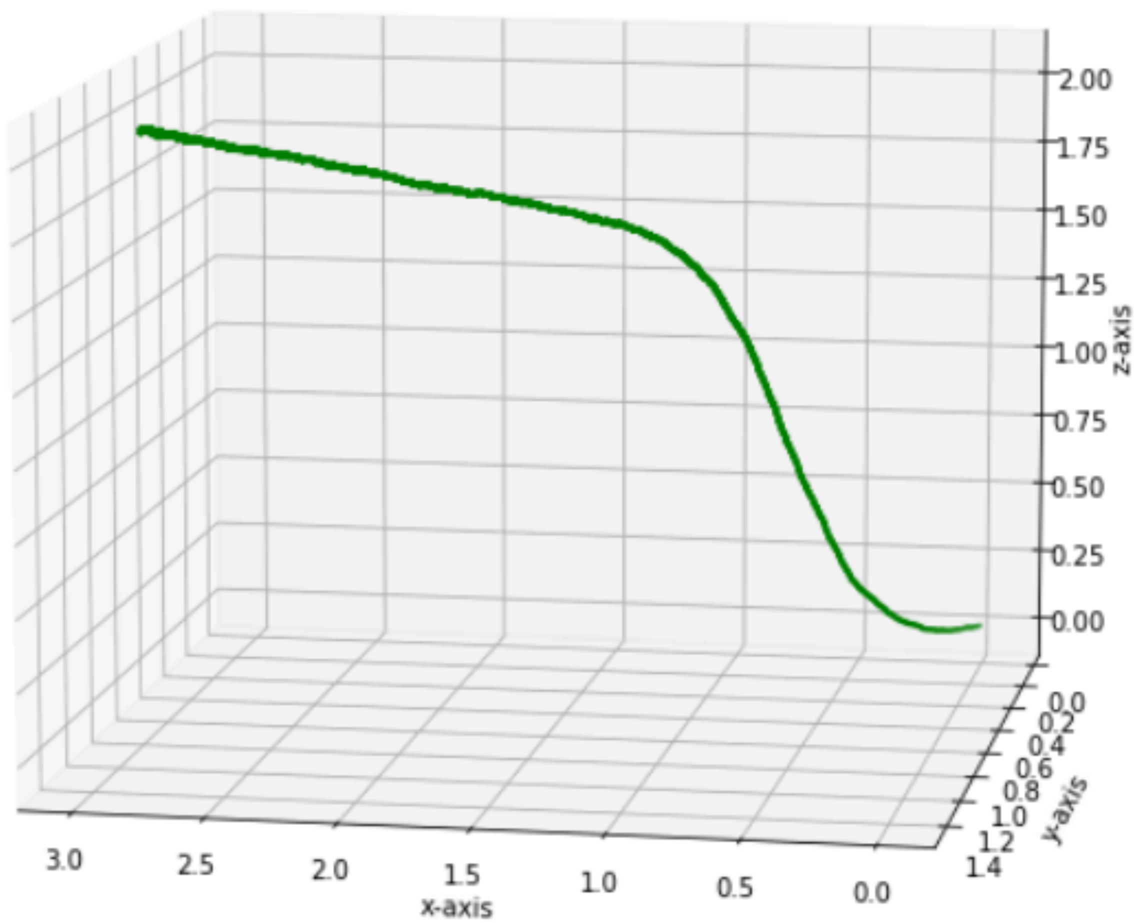
K	Time taken	Epochs	Parameters	Error difference
100	4.1007	30000	[3.01699166] [1.00603733] [2.00643416]	0.0157



Batch_size=100

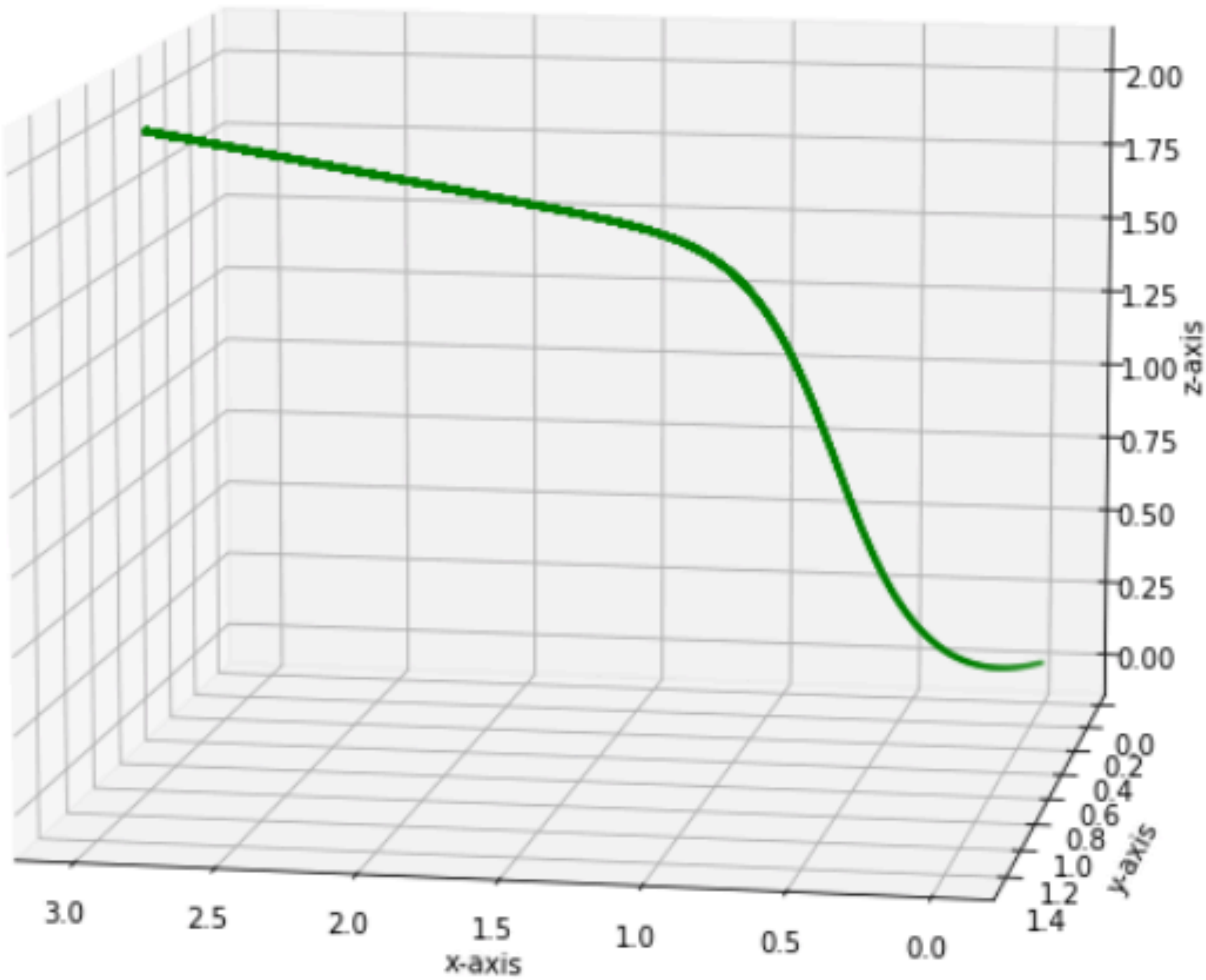
K	Time	Epochs	Parameters	Error difference	
	5	3.85	20000	[2.98572244] [1.001279] [1.99553391]	0.0666

Error wrt the learned hypothesis is: 1.9693803717273972

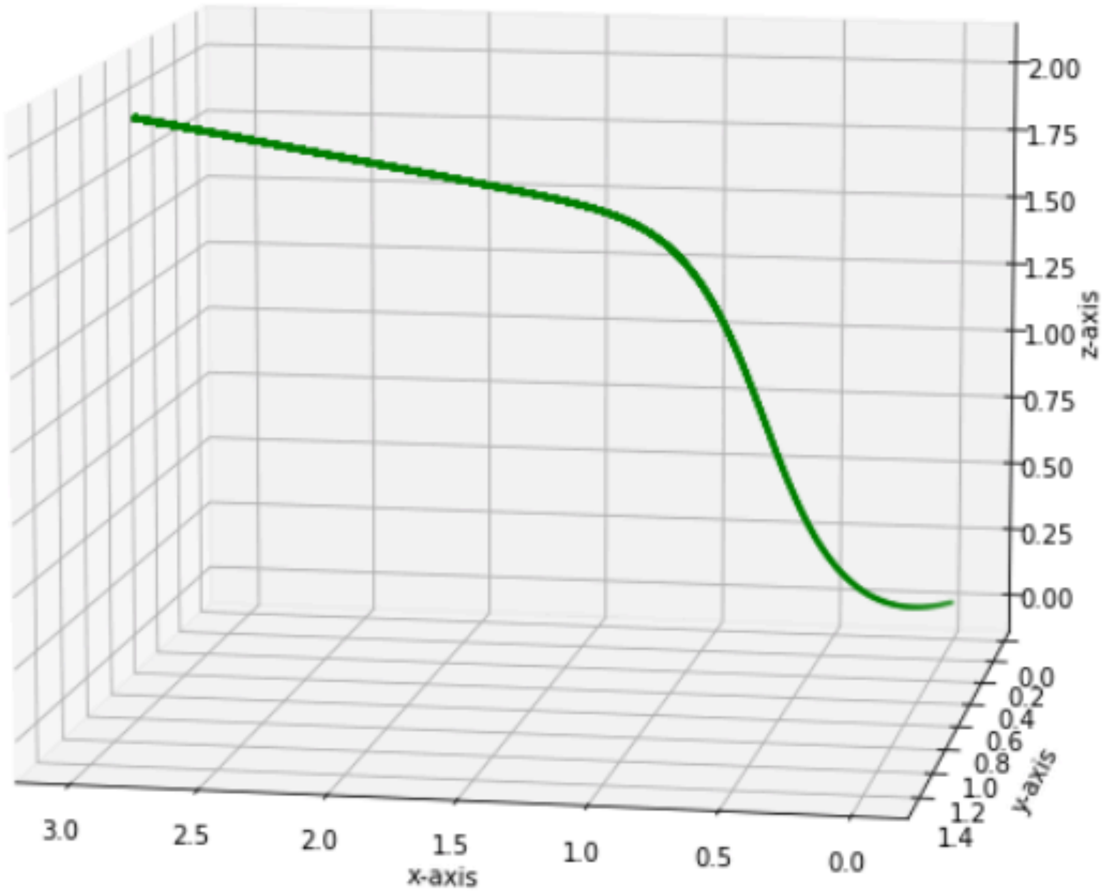


```
Batch_size=10000
Error wrt the learned hypothesis is: 1.96670090
```

Eps	K	Time	Epochs	Parameters	Error difference
1E-05	2	22.95	22000	[2.99227157] [1.00116254] [1.99839991]	0.00869



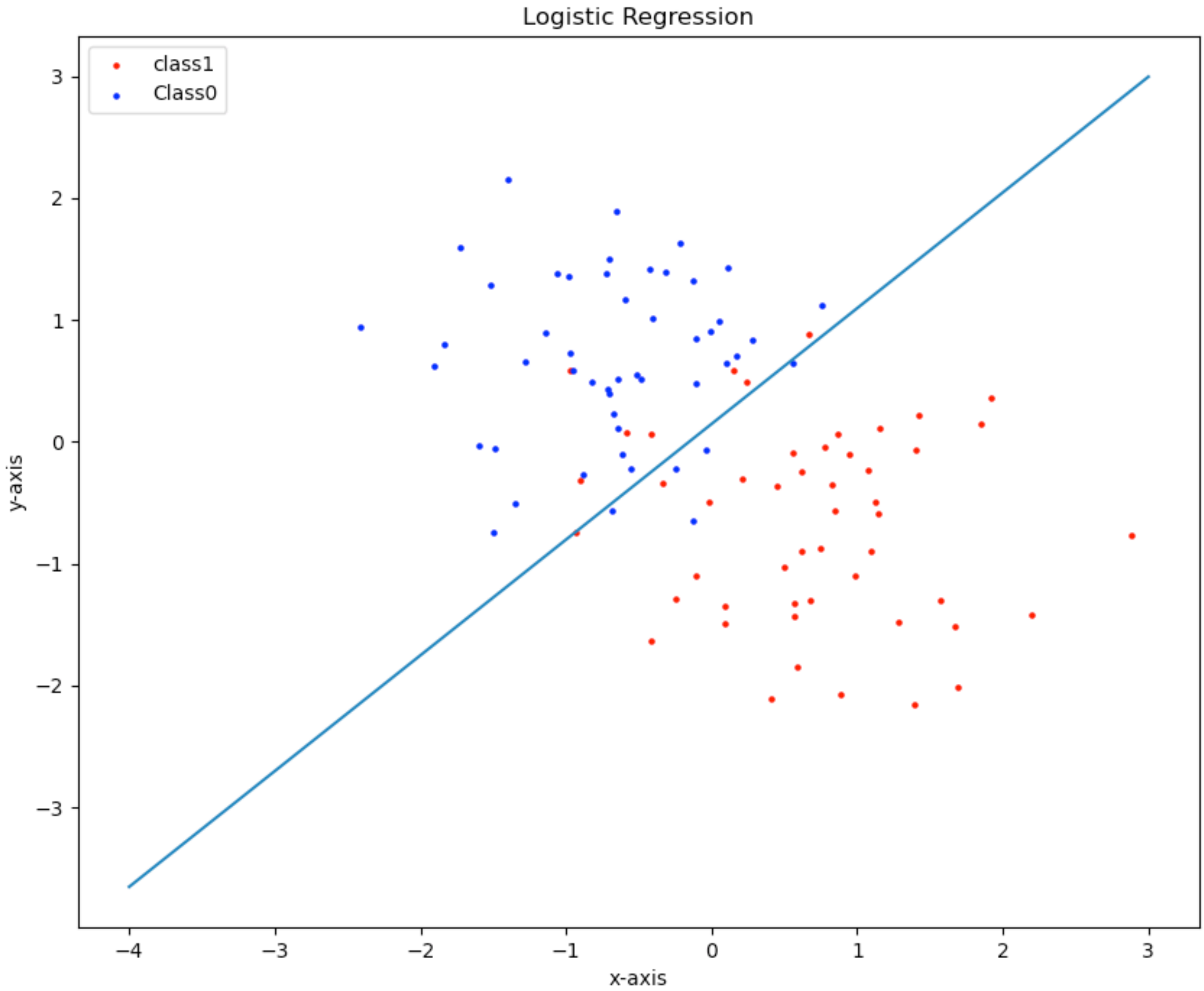
```
Batch_size=1000000, epochs=25000
Error wrt the learned hypothesis is: 1.96650090
```



```
Q3)python q3.py --lr 1 --eps 1e-8
lr=learning_rate
eps= threshold for convergence

Parameter vector is:
 [[ 0.40114656]
 [ 2.58823623]
 [-2.72525064]]

Equation of the decision boundry is:
0.4011+2.5882*x-2.7252*y=0
```



Qs4)Command to run: python q4.py

Gaussian Discriminant Analysis:

case 1 : $\Sigma_0 \neq \Sigma_1$

The decision boundary is defined where

$$P(x|y=0;\theta)\,P(y=0;\theta) = P(x|y=1;\theta)\,P(y=1;\theta)$$

$$\Rightarrow \frac{P(x|y=0;\theta)\,P(y=0;\theta)}{P(x|y=1;\theta)\,P(y=1;\theta)} = 1$$

$$\Rightarrow \log\left[\frac{P(x|y=0;\theta)\,P(y=0;\theta)}{P(x|y=1;\theta)\,P(y=1;\theta)}\right] = 0 \quad \text{—— (1)}$$

& we know

$$P(x|y=0;\theta) = \frac{1}{(2\pi)^{d/2}|\Sigma_0|^{1/2}} e^{-\frac{(x-\mu_0)^T\Sigma_0^{-1}(x-\mu_0)}{2}}$$

$$P(x|y=1;\theta) = \frac{1}{(2\pi)^{d/2}|\Sigma_1|^{1/2}} e^{-\frac{(x-\mu_1)^T\Sigma_1^{-1}(x-\mu_1)}{2}}$$

$$P(y=0;\theta) = 1-\phi \quad > \text{sample proportions}$$

$$P(y=1;\theta) = \phi$$

Eqn (1) becomes

$$\underbrace{\log\left[\frac{1-\phi}{\phi}\left(\frac{|\Sigma_1|}{|\Sigma_0|}\right)^{1/2}\right]}_{C} - \frac{1}{2}\left[(x-\mu_0)^T\Sigma_0^{-1}(x-\mu_0) - (x-\mu_1)^T\Sigma_1^{-1}(x-\mu_1)^T\right] = 0$$

Case 2 : $\Sigma_0 = \Sigma_1$, Equation 1 becomes

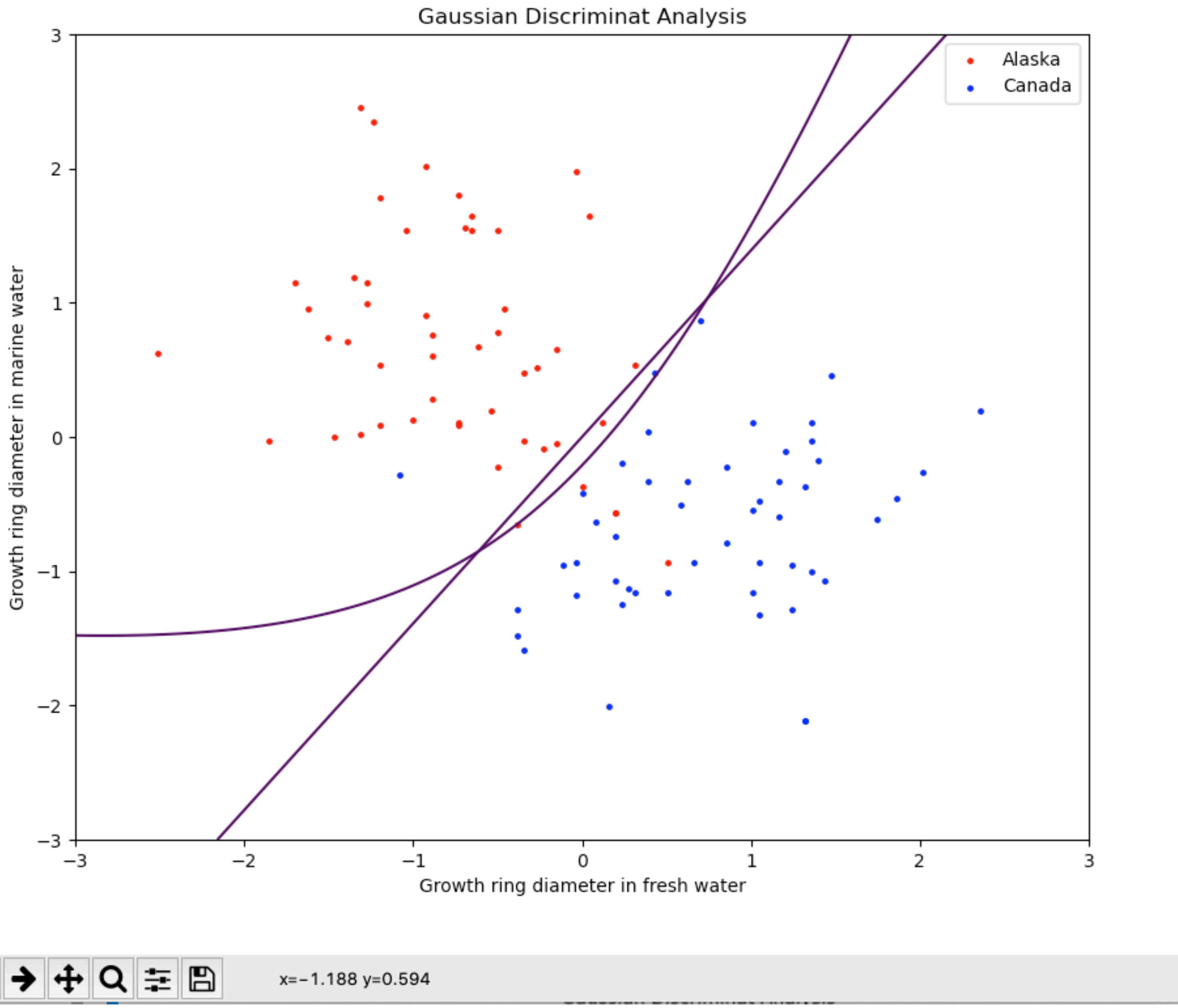
$$C + \frac{1}{2}\left[x^T\cancel{(\Sigma_1^{-1}-\Sigma_0^{-1})}^{\,0}x - 2(\mu_1^T\Sigma_1^{-1} - \mu_0^T\Sigma_0^{-1})x + \mu_1^T\Sigma_1^{-1}\mu_1 - \mu_0\Sigma_0^{-1}\mu_0\right] = 0$$

$$\Rightarrow C + \frac{1}{2}(\mu_1^T\Sigma^{-1}\mu_1 - \mu_0\Sigma^{-1}\mu_0) - (\mu_1^T - \mu_0^T)\Sigma^{-1}x = 0.$$

Mean of class Canada:

```
[[ 0.75529433]
 [-0.68509431]]
```

```
Mean of class Alaska:
 [[-0.75529433]
 [ 0.68509431]]
Covariance matrix of class Canada is:
 [[0.47747117 0.1099206 ]
 [0.1099206  0.41355441]]
Covariance matrix of class Alaska is:
 [[ 0.38158978 -0.15486516]
 [-0.15486516  0.64773717]]
Covariance matrix for in the linear case is:
 [[ 0.42953048 -0.02247228]
 [-0.02247228  0.53064579]]
```



The quadratic boundry gives more space of the x-y plane to class Canada which might cause underfitting as it is not evident from the training data.