OMML HOMEWORK 01 REPORT FOR GROUP 23

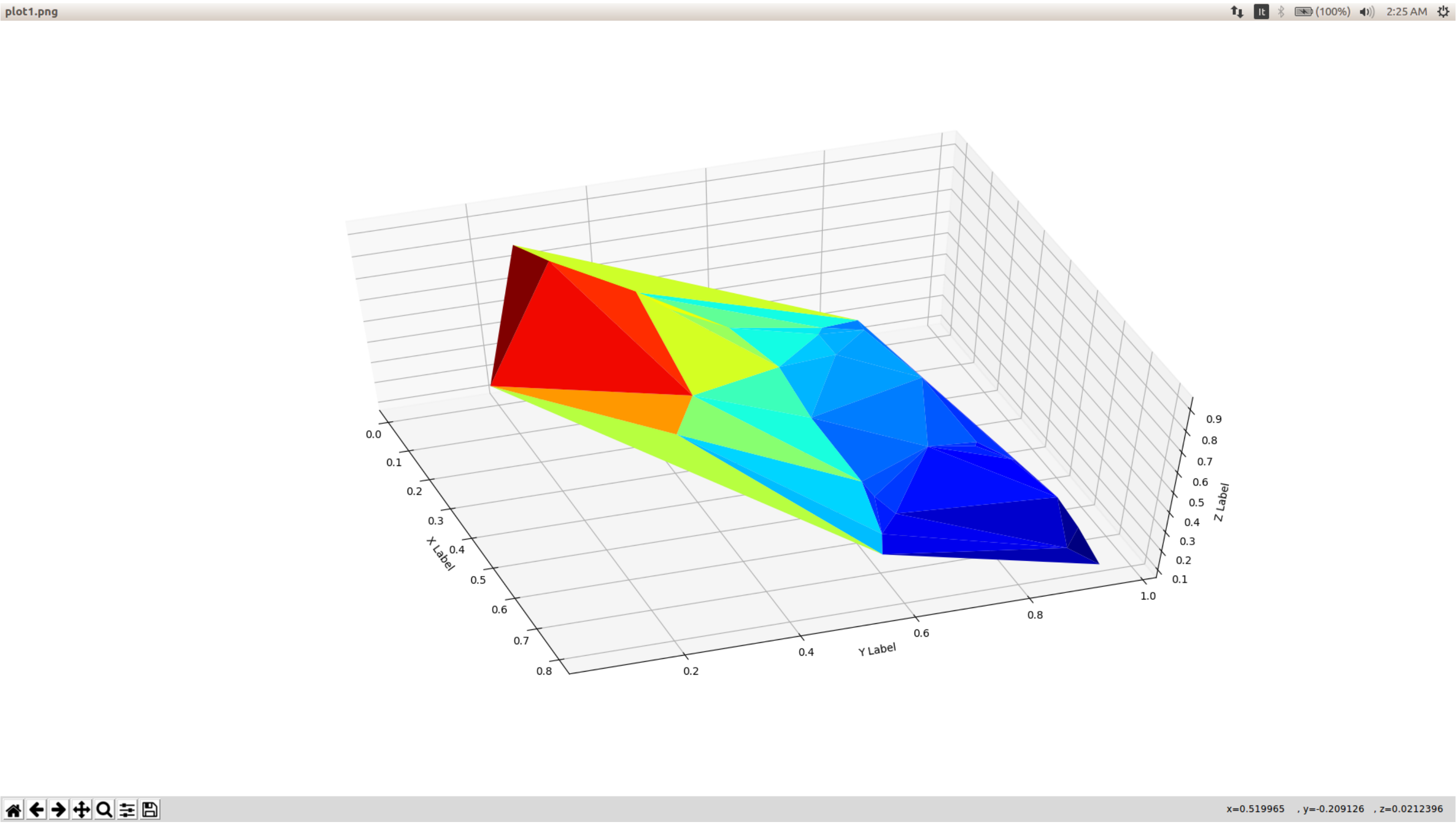
Question 01:
part 01 MLP Full minimization
parameters:
N = 11, p = 10^-10 and sigma = 1

On optimization routine we used is scipy.optimize and the returned message was as below:

```
  message: 'Optimization terminated successfully.'
     nfev: 80592
      nit: 1648
     njev: 1752
   status: 0
  success: True
```

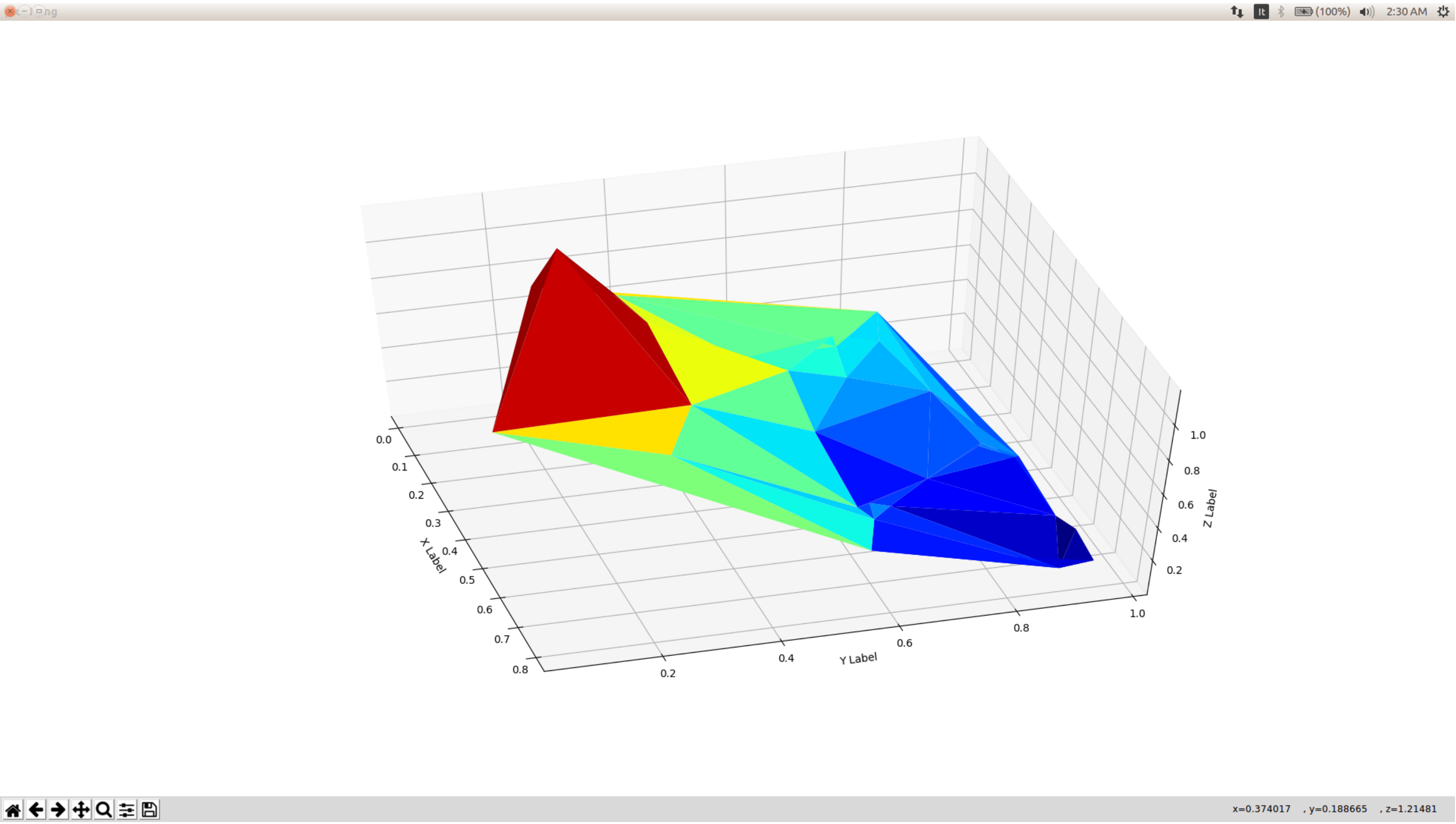
MLP Computing time is: 6.21475887298584 seconds
MSE on training = 0.0014
MSE on test = 0.2459

plots for the prediction and actual values:
- Prediction Graph



x=0.519965 , y=-0.209126 , z=0.0212396

- True data Graph



x=0.374017 , y=0.188665 , z=1.21481

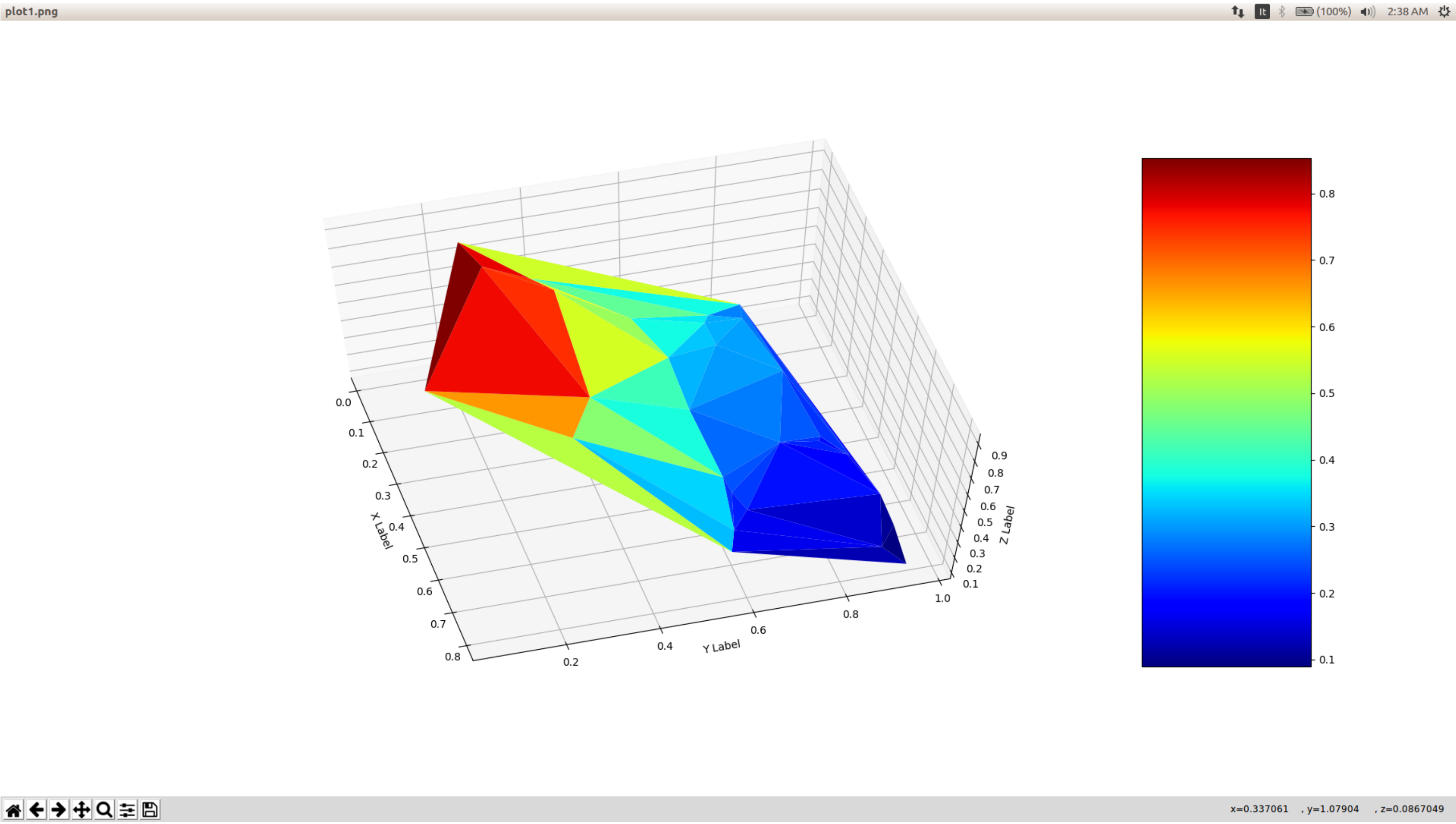
part 02 RBF network

- values of p are between [10^-6, 10^-3] and 10^-5 was used in our setting
- Number neurons are 6 and sigma was set to 0.3

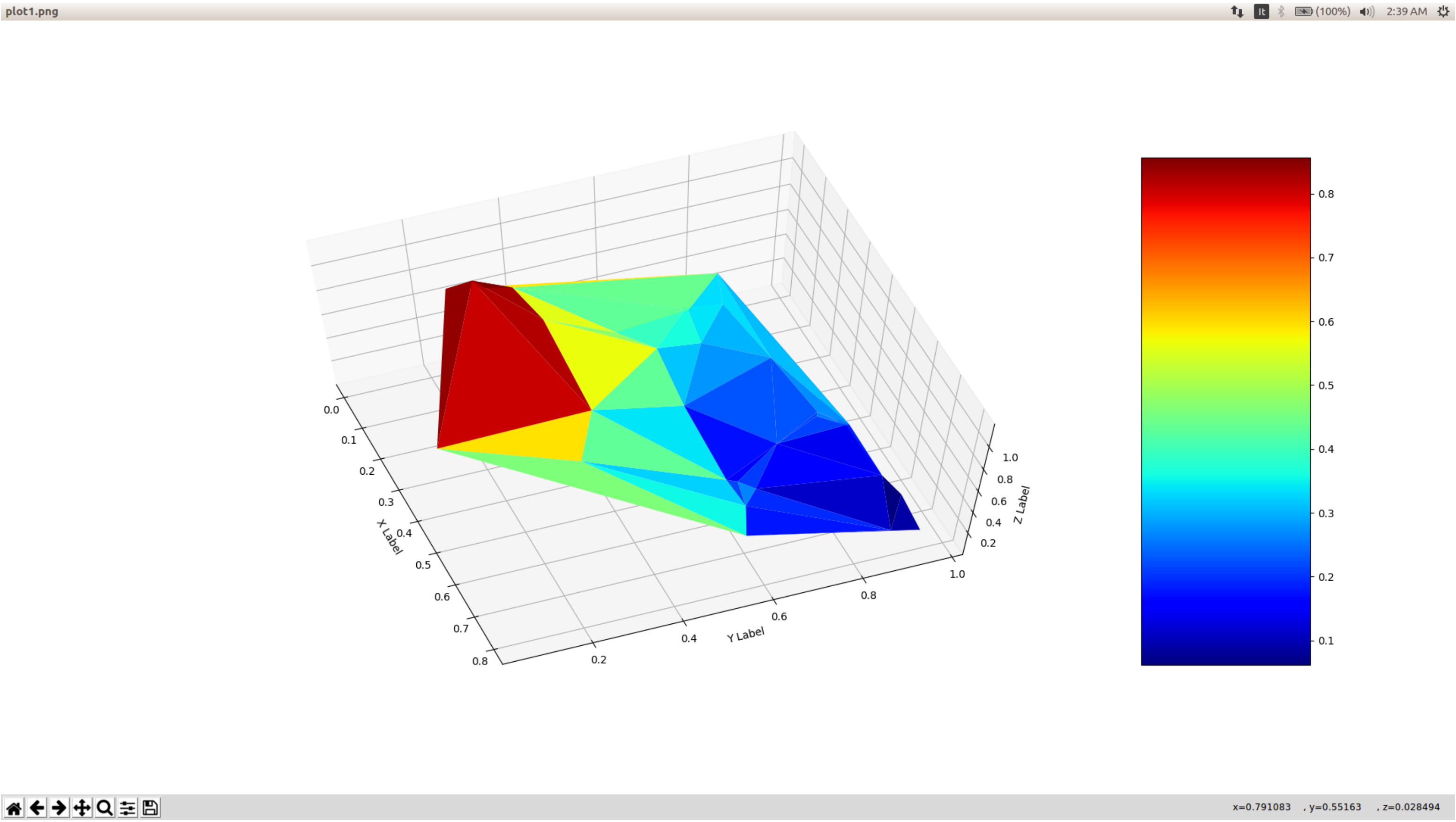
-On optimization routine we used scipy.optimize and the returned message was as below:

```
 message: 'Optimization terminated successfully.'
    nfev: 3720
     nit: 177
    njev: 186
  status: 0
 success: True
RBF Computing time is: 14.436548233032227 seconds
MSE on training = 0.00136
MSE on test = 0.1282
```

-the plot of the function representing the approximating function
plots of prediction graph:



plot of actual True data:



x=0.791083 , y=0.55163 , z=0.028494

Comparing the two networks in terms of quality of approximation (training and test error) and in efficiency of the optimization (number of function/gradient evaluation and computational time needed to get the solution) RBF networks performs better than MLP. These results are depicted in the table below;

Network	Training Error	Test Error	Function Evaluations	Gradient Evaluations	Computation time in seconds
MLP	0.0015	0.2265	48990	1065	3.84
RBF	0.0014	0.1466	3720	186	13.42

From the results in the table it is seen that MLP takes little time to achieve the results, however, it is not better than RBF as it make more function and gradient evaluations which would take more time for large data samples. RBF also achieves good results over small number of function and gradient evaluations than the MLP which shows that there is more likelihood of over fitting data in MLP than in RBF as seen from training and test errors in both schemes.

Question 02 (Two blocks Methods):

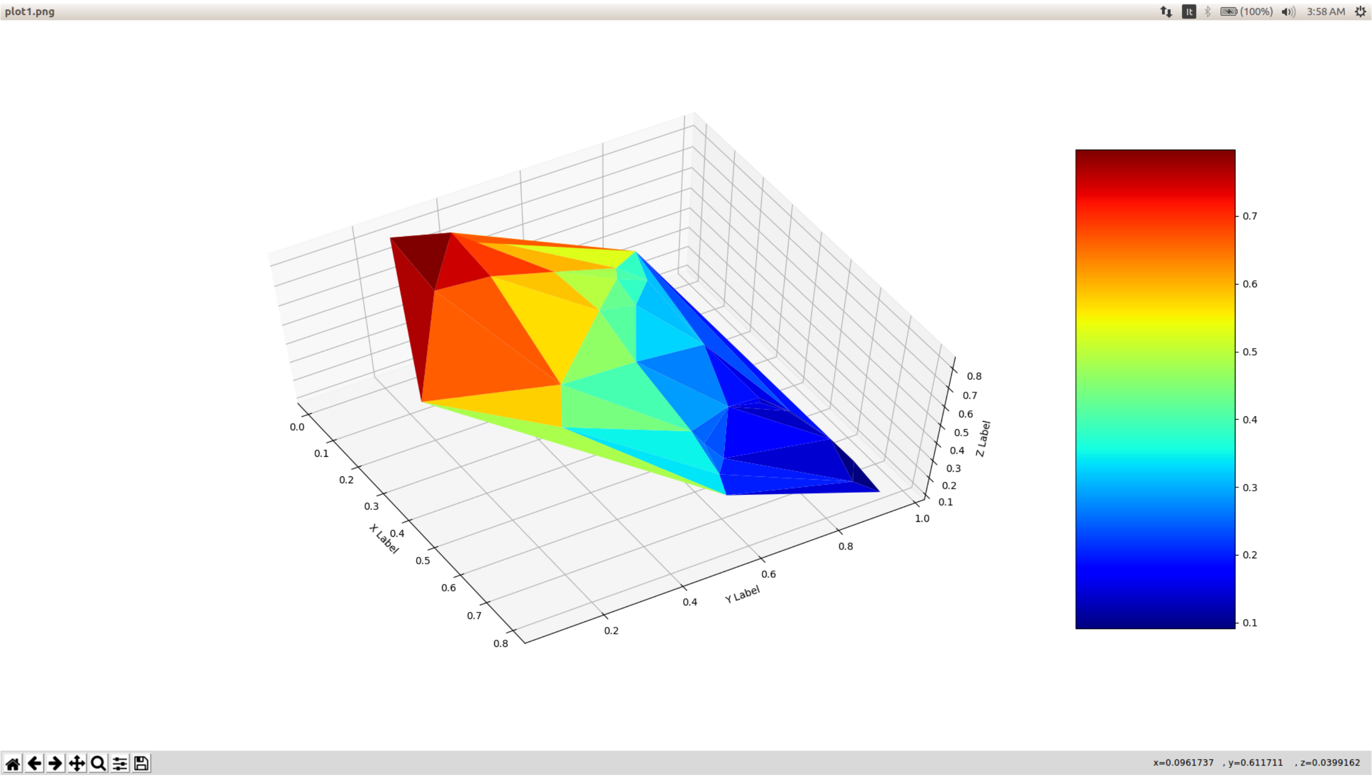
On optimization routine we used scipy.optimize and the returned message was as below:

Training MSE: 0.006931928024042126
Test MSE: 0.429045638531
function evaluations: 923
gradient evaluations: 71
Number of Iteration : 70

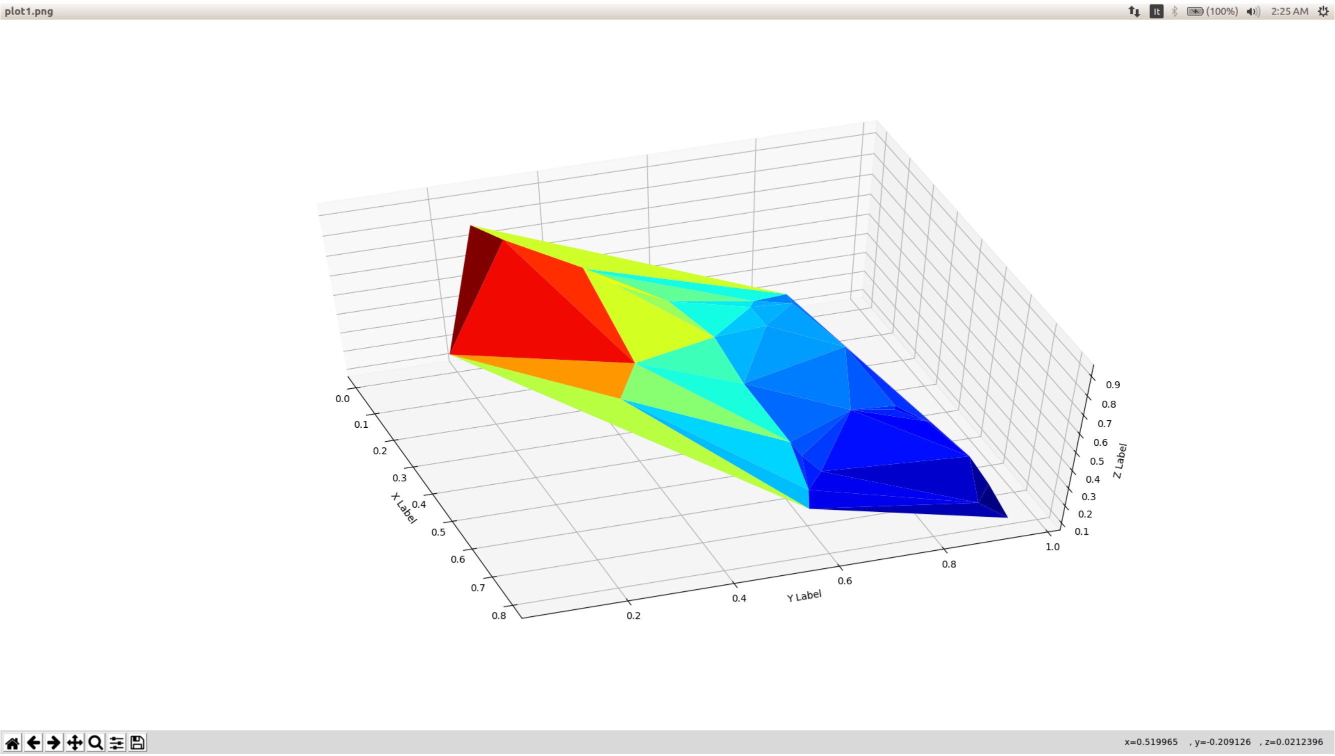
Network	Training Error	Test Error	Function Evaluations	Gradient Evaluations	Computation time in seconds
MLP	0.0015	0.2265	48990	1065	3.84
EL MLP	0.0069	0.4290	923	71	25.67

Looking at the table above in comparison of the two networks, we can easily conclude according to the results that MLP has shorter execution time that EL MLP but on the contrary EL MLP requires less function and gradient evaluations and furthermore, shows that there is less of over-fitting and more of generalization. Considering function and gradient evaluations for large data sets MLP would indeed perform poorly as it would need more evaluations of both function and gradient .

-the plots of the function representing the approximating function:
prediction EL_MLP :



graph obtained by full minimization prediction graph:



x=0.519965 , y=-0.209126 , z=0.0212396

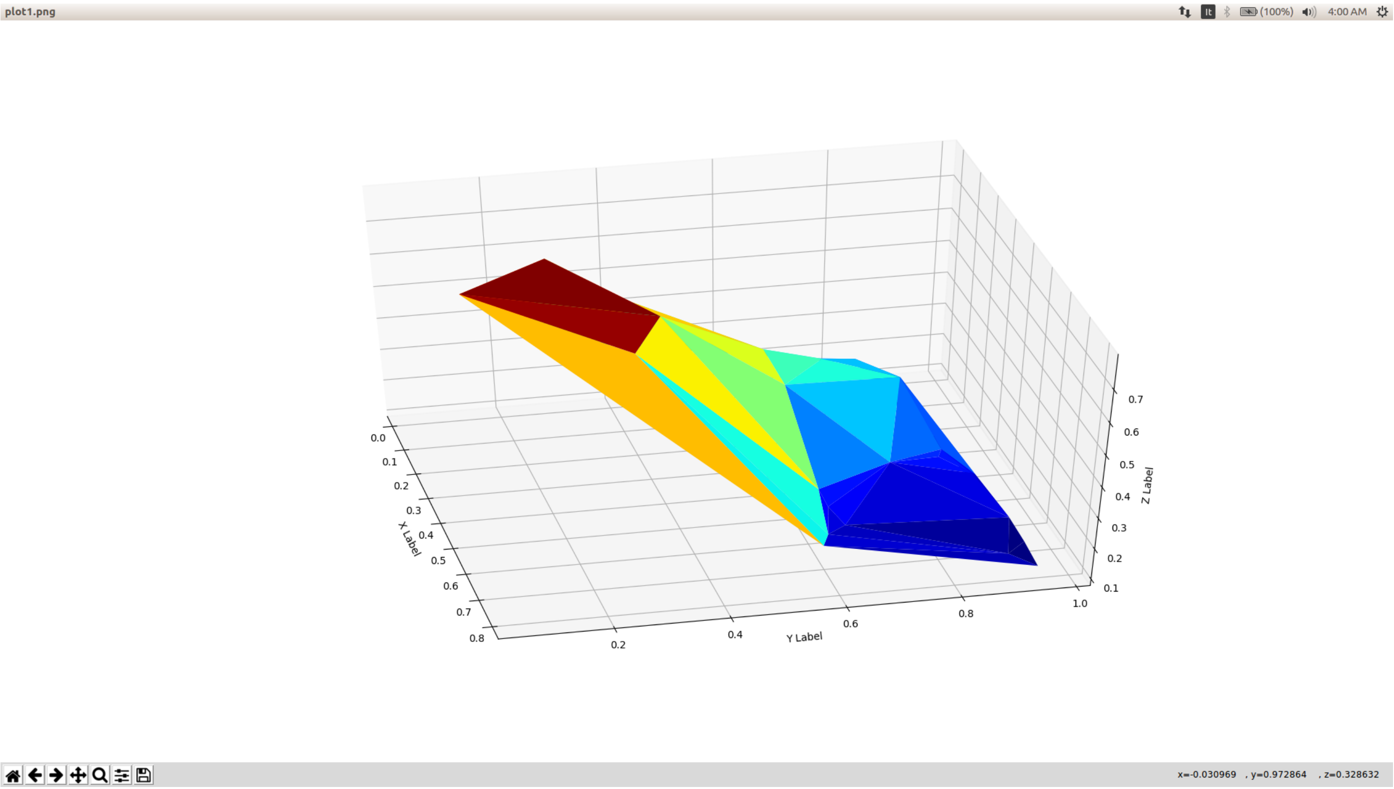
part 02 of Question 02:

On optimization routine we used scipy.optimize and the returned message was as below:

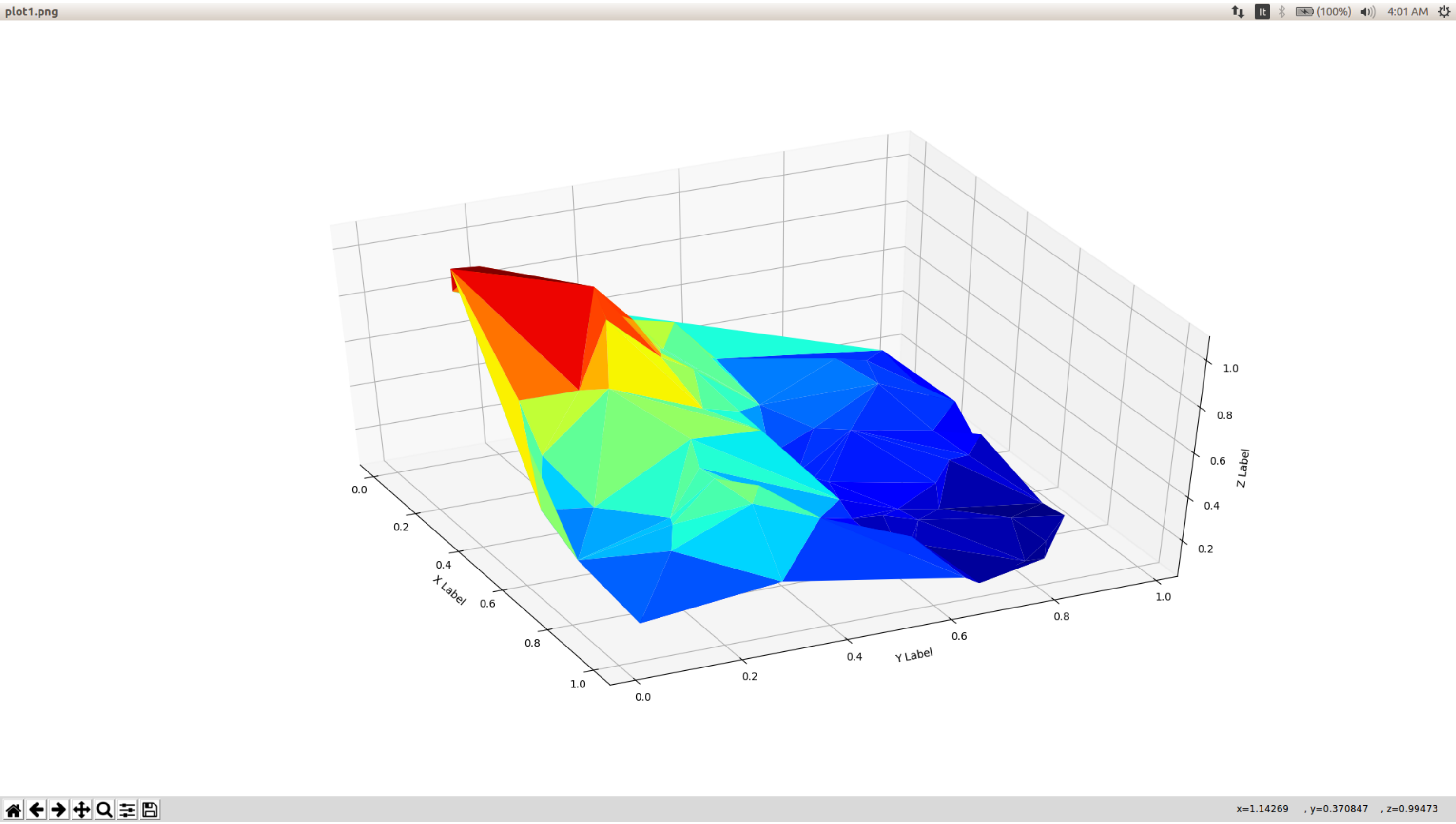
```
Optimization terminated successfully.
         Current function value: 0.008745
         Iterations: 21
         Function evaluations: 176
         Gradient evaluations: 22
values for Eta = 1e-05 , num units = 6 and sigma = 0.3
Training MSE: 0.008744920621156711
Test MSE: 0.749911721392
function evaluations: 176
gradient evaluations: 22
Number of Iteration : 21


MLP Computing time is: 1.003429889678955 seconds
```

Graphs of EL RBF of True values and the predicted

predicted graph:



EL RBF True graph:



Question 03 (Decomposition method):

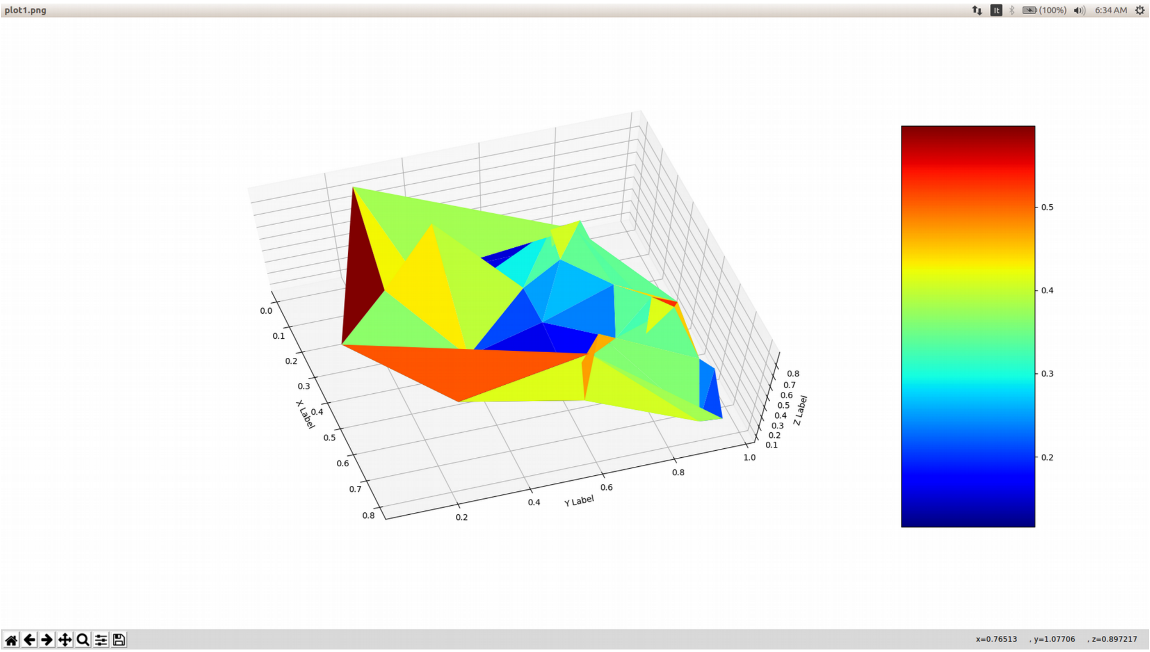
On optimization routine we used scipy.optimize and the returned message was as below:

```
values for Eta = 0.001, num units = 2 and sigma = 0.7
Training MSE: 0.014089507929
Test MSE: 0.739574466578
function evaluations: 4
gradient evaluations: 1
  message: 'Optimization terminated successfully.'
     nfev: 4
      nit: 0
     njev: 1
   status: 0
  success: True
```

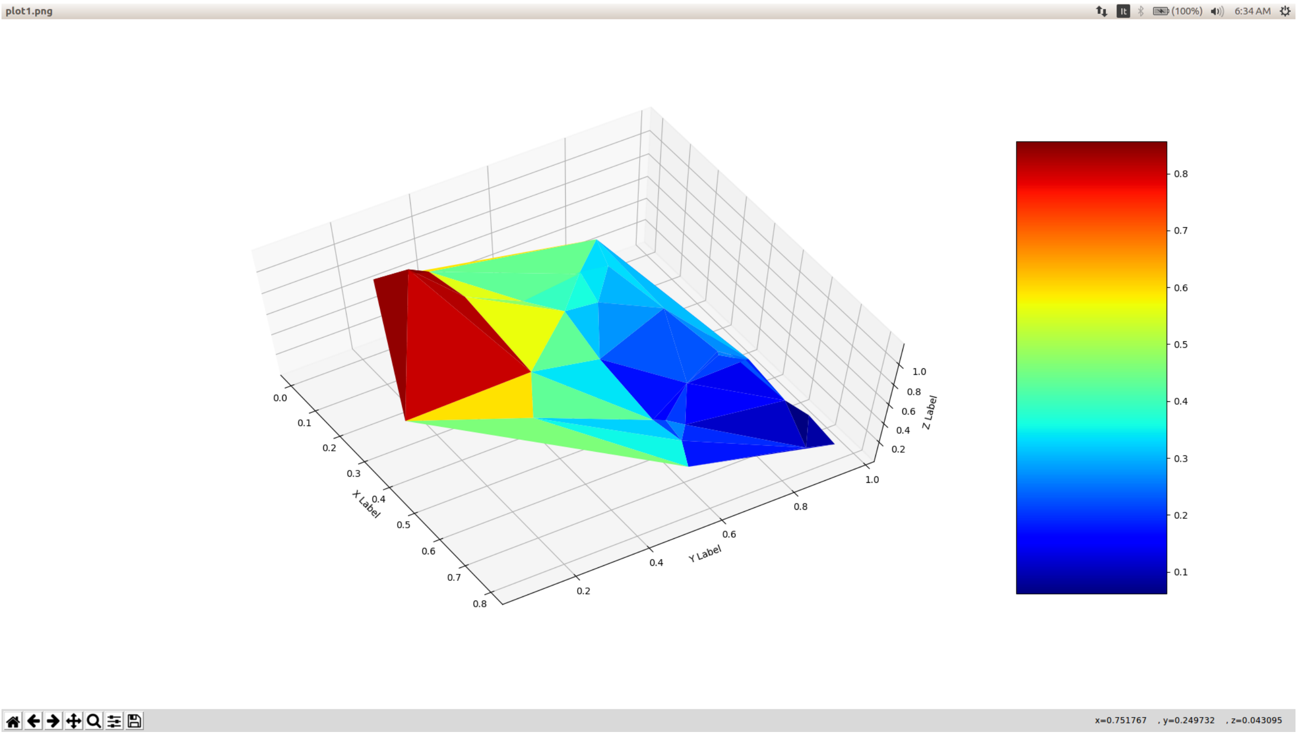
For calculation of gradient we used the toolbox Numdifftools which gives the values of the calculated gradients.

- the stopping criteria of the decomposition procedure was defined as the norm of the gradients being in the interval [10^-6, 10^-3]

Graph for the predicted.



True Graph



Comparison of the network from question 2 for unsupervised learning and the RBF for block decomposition :

Network	Training Error	Test Error	Function Evaluations	Gradient Evaluations	Computation time in seconds
RBF_EL	0.0087	0.7499	176	22	1.00342
Block RBF	0.0140	0.7395	1	4	0.90374

Question 04:

Ex	FFN	Num units (N)	Sigma	eta(p)	Trainining error	Test error	Optimization time
Q1.1	Full MLP	11		10^-6	0.00277	0.2509	39.84 seconds
Q1.2	Full RBF	6	0.3	10^5	0.00136	0.1282	15 minutes
Q2.1	E MLP	11		10^-6	0.0069	0.4290	25.6768 seconds
Q2.2	Unsupervised c RBF	6	0.3	10^-5	0.0087	0.7499	7 minutes
Q2.3	RBF Decomposition	2	0.7	10^-3	0.0140	0.7395	25 minutes