CME Homework Assignment 3 Computational Work

Jacob Perricone
Stanford University
jacobp2@stanford.edu
,

Ian Shaw
Stanford University
ieshaw@stanford.edu
,

Weronika Ŝwięchowicz
Stanford University
wswiecho@stanford.edu

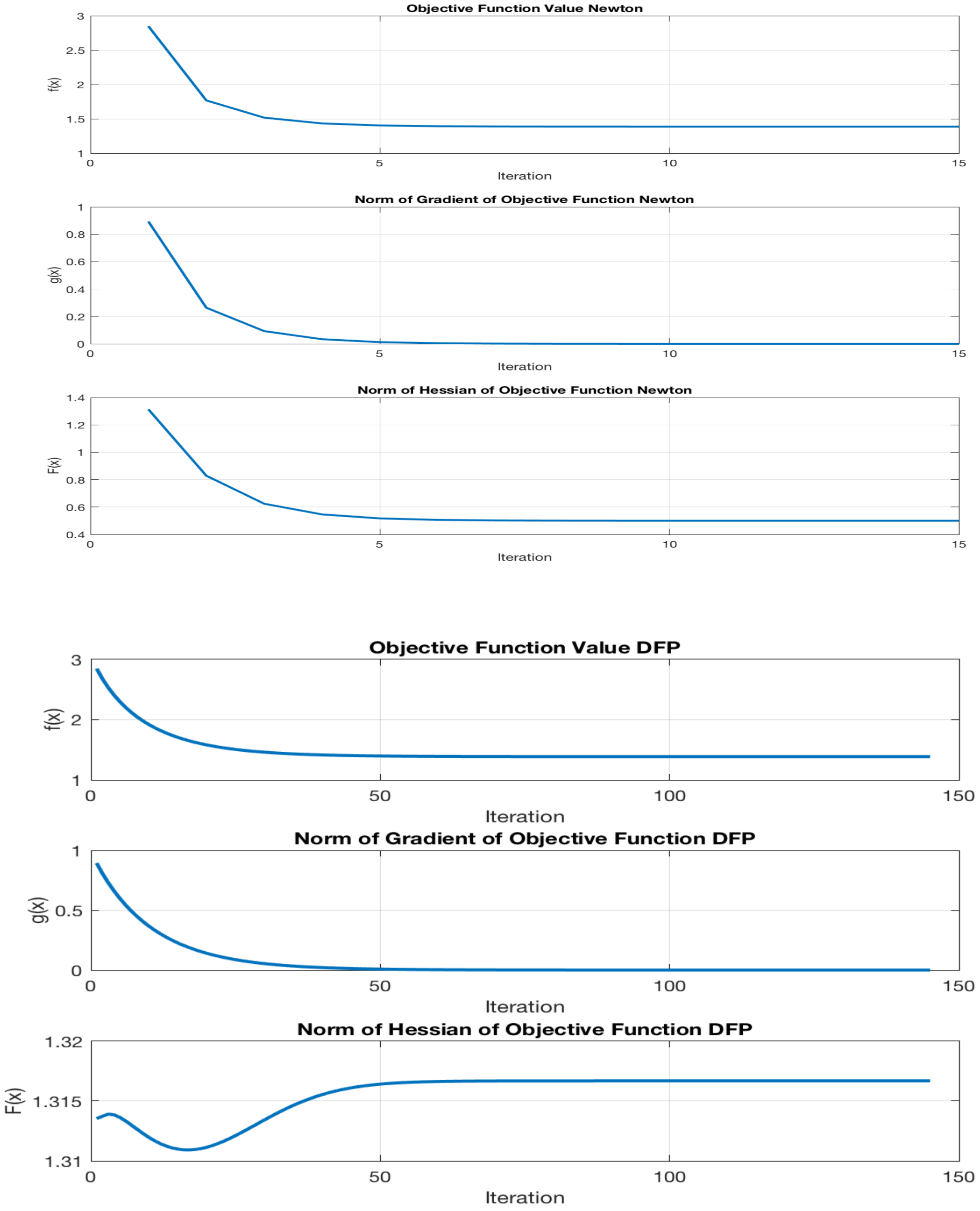
Problem 2.d

(Computation Team Work) Apply any Quasi-Newton (e.g., slide 18 of Lecture #13 or L&Y Chapter 10) and Newton methods to solve the problem using the data in HW2 for SVM (may or may not with regulation), randomly generate data sets, and/or benchmark data sets you can find. Compare the two methods with each other and with the previous methods used in HW3.

Using the dataset from last week and scrapping the regularization component, let us compare the different methods:

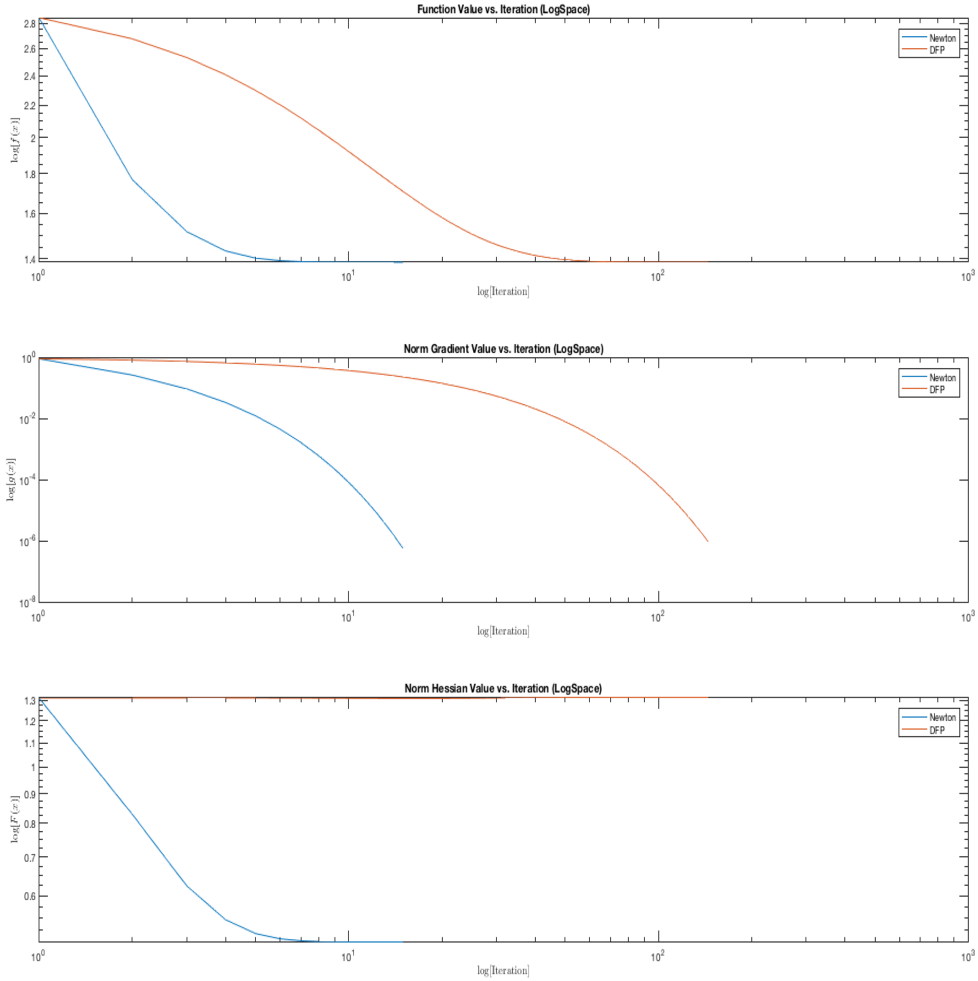
Both the Newton Method an the DFP method appear to enjoy fast convergence rates. An α of one was chosen for Newton Method, whereas for the DFP method, α_k was chosen to minimize the objective function in the direction of $d = -H_k g_k$.

Figure 1: Convergence Of Second Order Method



Newton's Method Appears to converge more quickly than the DFP method, converging after only 15 iterations. The DFP method, on the other hand, converges after 145 iterations, which relative to the methods last week, is very quick. I believe that it is possible to speed up the DFP method if one changes the initial α passed into the line search routine. Essentially, the line search just halves α if the $f(x + \alpha_k H_k d_k) \geq f(x)$. Indeed, if I begin the search with an alpha of 1 the DFP converges after only 30 iterations, whereas if I start with an alpha of $.1$ the DFP method takes 145 iterations. THe plot below overlays the convergence of the two methods in logspace.

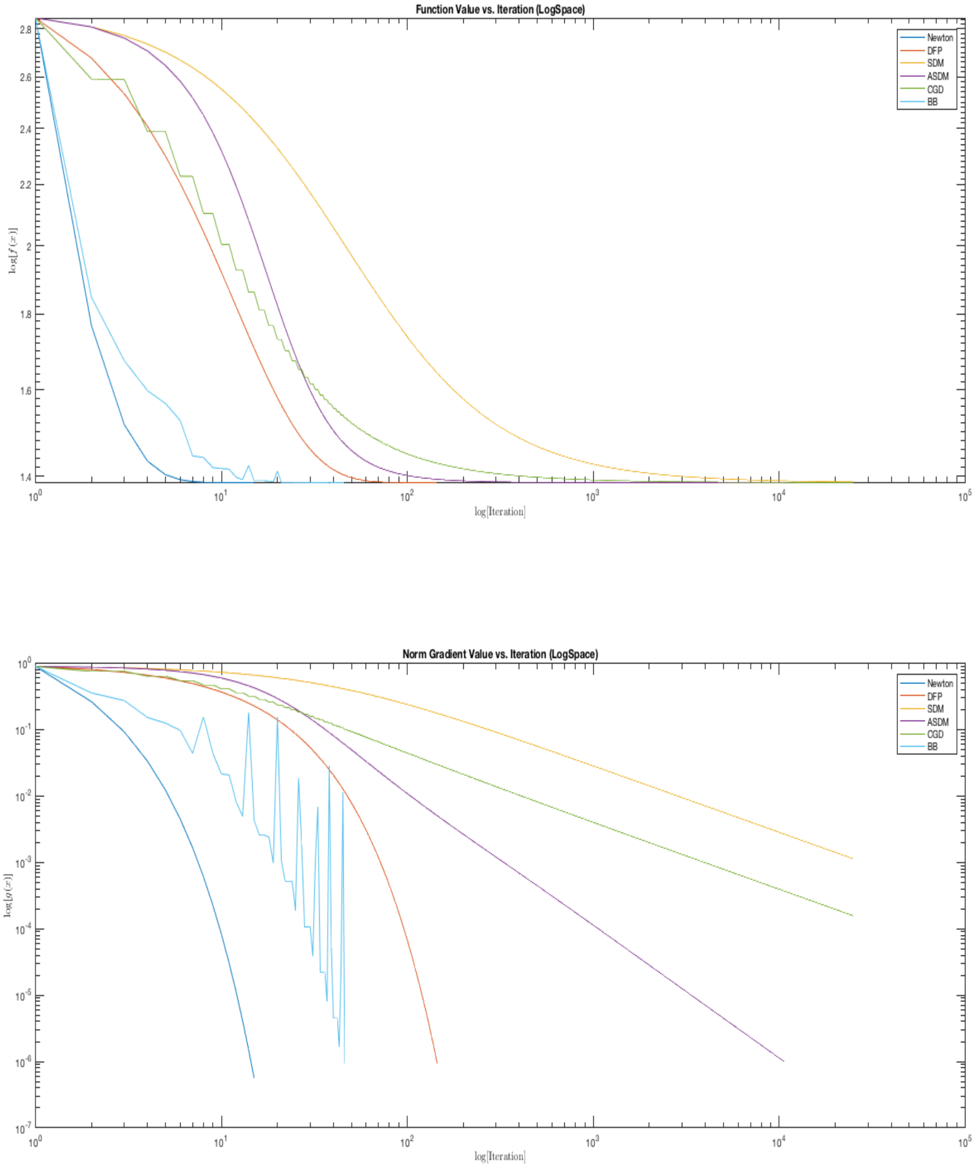
Figure 2: Convergence Comparison



Notice how the norm of the hessian for the DFP method does not decrease with iteration. This is attributed to the fact that it locally approximates the Hessian instead of actually computing it.

Now let us examine the convergence of the second order methods with respect to the first order methods of last homework.

Figure 3: Convergence Of All Methods



Newton's Method converges the most quickly followed by the Barzilai-Borwein Method, the DFP Method, the CGDM, and the stepest decent methods. The BB method works well with $\alpha_k = \frac{(\nabla_x^k)^\top \nabla_g^k}{(\nabla_g^k)^\top \nabla_g^k}$, but is less appealing than the Newton or DFP methods due to the volatility of its gradient values. This is due to the fact that the stepsize is based on an approximation of the second derivative which can change largely per iteration. Overall, the second order method's have far superior performance than the first order methods.

Problem 3.d

(Computation Team Work) Draw $\mathbf{x}$ part of the primal-dual potential function level sets:

$$\psi_6(\mathbf{x},\mathbf{s}) \le 0 \quad \text{and} \quad \psi_6(\mathbf{x},\mathbf{s}) \le -10,$$

and

$$\psi_{12}(\mathbf{x},\mathbf{s}) \le 0 \quad \text{and} \quad \psi_{12}(\mathbf{x},\mathbf{s}) \le -10;$$

respectively in the primal feasible region (on a plane).

Hint: Sample interior points in the primal and dual feasible regions.To plot the $\mathbf{x}$ part of the level set of potential function, say $\psi_6(\mathbf{x},\mathbf{s}) \le 0$, in primal feasible region F_p, you plot

$$\{\mathbf{x} \in F_p : \min_{\mathbf{s}\in F_d} \psi_6(\mathbf{x},\mathbf{s}) \le 0\}$$

where F_d represents the dual feasible region. This can be approximately done by sampling as follows.

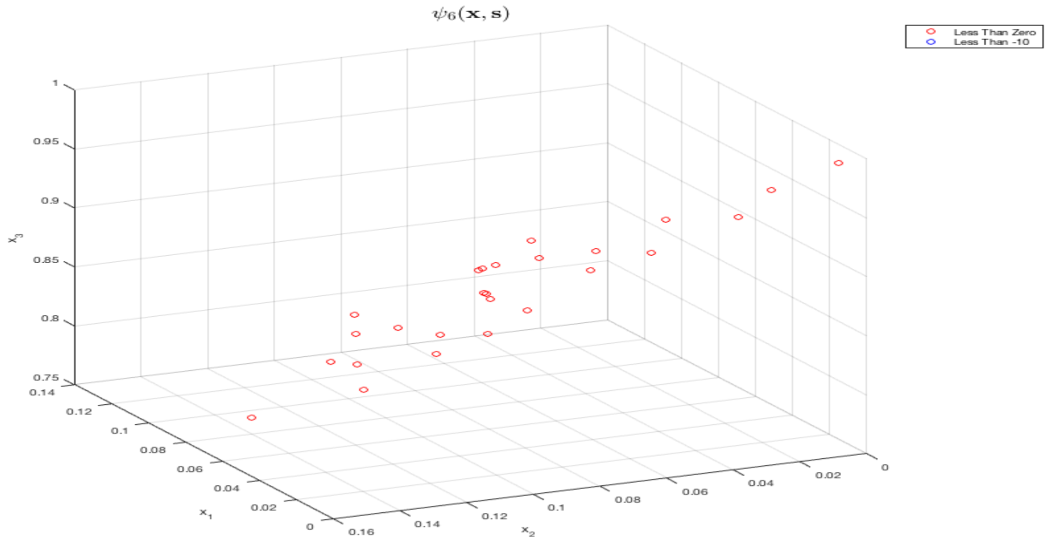
You randomly generate N interior feasible points of the primal $\mathbf{x}^p$ and the dual $(y^q, \mathbf{s}^q)$, respectively. For each primal point $\mathbf{x}^p$, you find if it is true that

$$\min_{q=1,\ldots,N} \psi_6(\mathbf{x}^p, \mathbf{s}^q) \le 0.$$

Then, you plot those $\mathbf{x}^p$ who give an ”yes” answer.

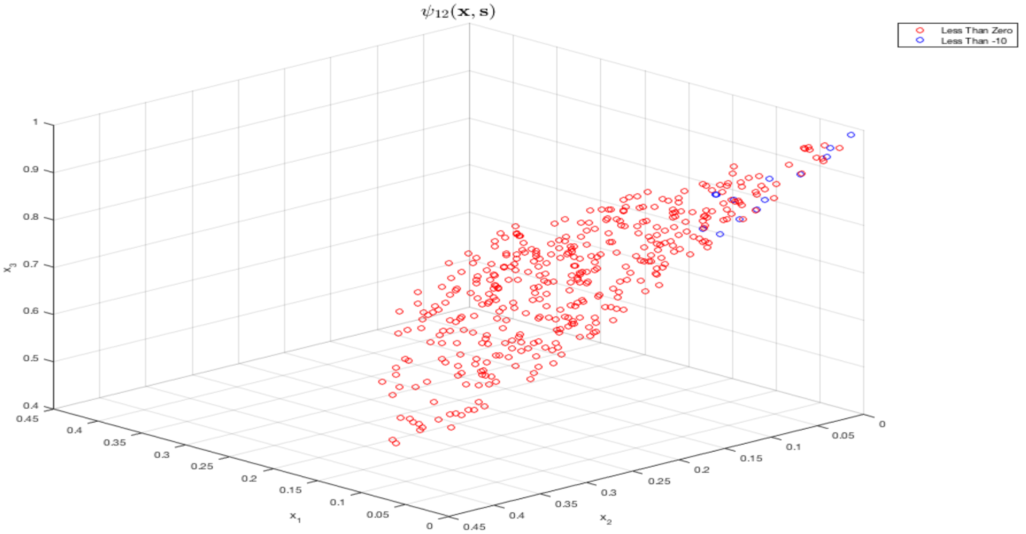
For the plot of the level set $\phi_6(\mathbf{x},\mathbf{s})$, even with 10,000 points generated, not points fall within the level set paramaterized by $\phi(\mathbf{x},\mathbf{s}) \le -10$.

Figure 4: $\phi_6(\mathbf{x},\mathbf{s})$, $f(x) = x_1 + x_2$



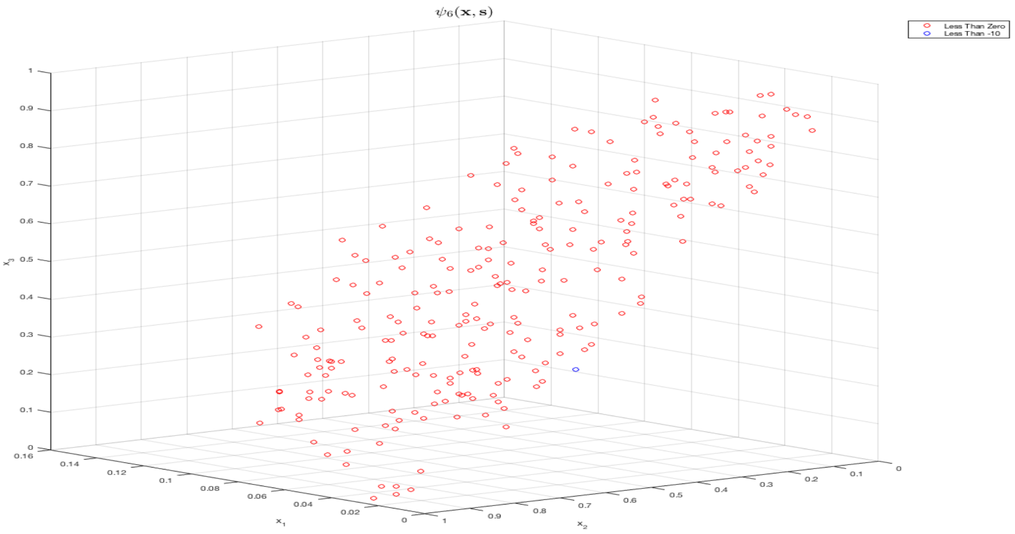
With $\phi_{12}(\mathbf{x}, \mathbf{s})$ a smaller number of points near $(0, 0, 1)$– which is the none optimal solution, have a potential of less than -10. This is as expected, since as you decrease the level set of the potential, the duality gap decreases, indicating an optimal solution.

Figure 5: $\phi_{12}(\mathbf{x}, \mathbf{s}), f(x) = x_1 + x_2$



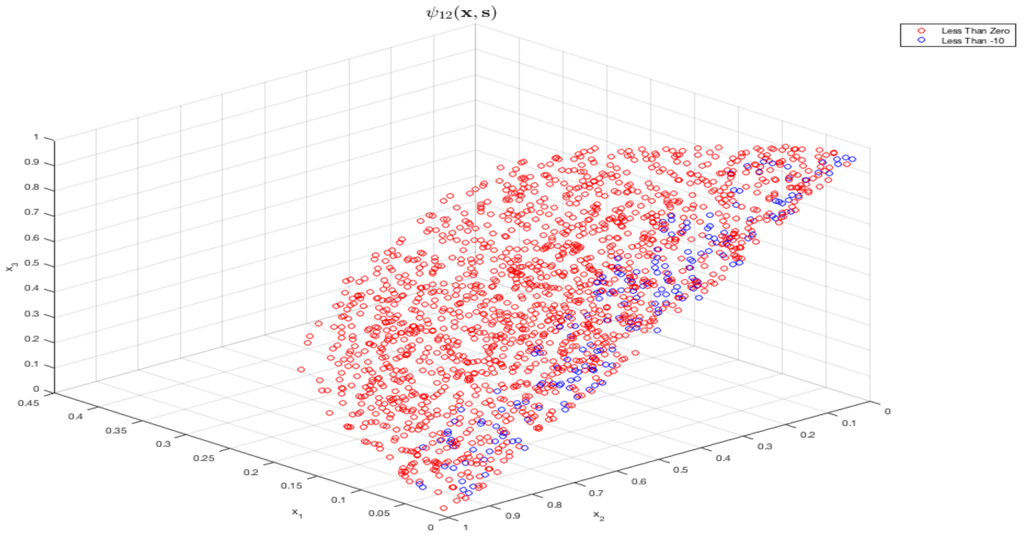
In the case where the objective $f(x) = x_1$, the optimal solution is given by $(0, \frac{1}{2}, \frac{1}{2})$. Similar to above, the level set of $\phi_6(\mathbf{x}, \mathbf{s})$ has very few points with potential less than -10, with the only blue point appearing around $(0, .4, .6)$.

Figure 6: $\phi_6(\mathbf{x}, \mathbf{s}), f(x) = x_1$



In the case of $\phi_{12}(\mathbf{x}, \mathbf{s})$, we have many more points with potential less than -10, most of which are clustered around $x_1 = 0$. The largest concentration appears in the region $(0, [.4, .6], [.4, .6])$ indicating that decreasing the potential indeed decreases the duality gap. However, there is a large number of points spread along the x_2, x_3 plane, suggesting a lower potential set is needed to shorten the duality gap. In fact, if one parameterizes the level set $\phi_{12}(\mathbf{x}, \mathbf{s}) \leq -45$, the only blue point appears at $x_1 = 0, x_2 = .47, x_3 = .53$, which is close to the opimum

Figure 7: $\phi_{12}(\mathbf{x}, \mathbf{s}), f(x) = x_1$



Problem 7

Implement the ADMM to solve the divergence example in Lecture &16.

(a) Try $\beta = 0.1$, $\beta = 1$, and $\beta = 10$, respectively. Does the choice of β make a difference?

By studying left plots in figures 11-13, it should be clear that the ADMM fails to find a convergent solution to the given problem independently of the value of β. The only difference we can observe in these graphs is the magnitude of oscillation from the true value of x, which decreases with the increases in the value of β.

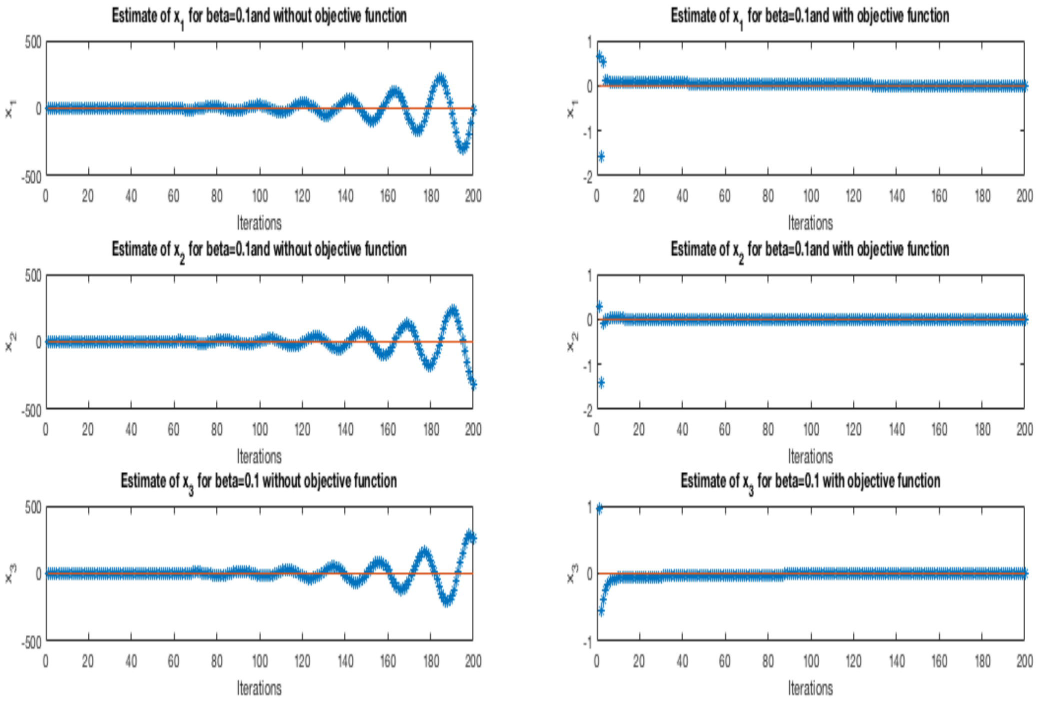
(b) Add the objective function to minimize

$$0.5(x_1^2 + x_2^2 + x_3^2)$$

to the problem, and retry $\beta = 0.1$, $\beta = 1$, and $\beta = 10$, respectively. Does the choice of β make a difference?

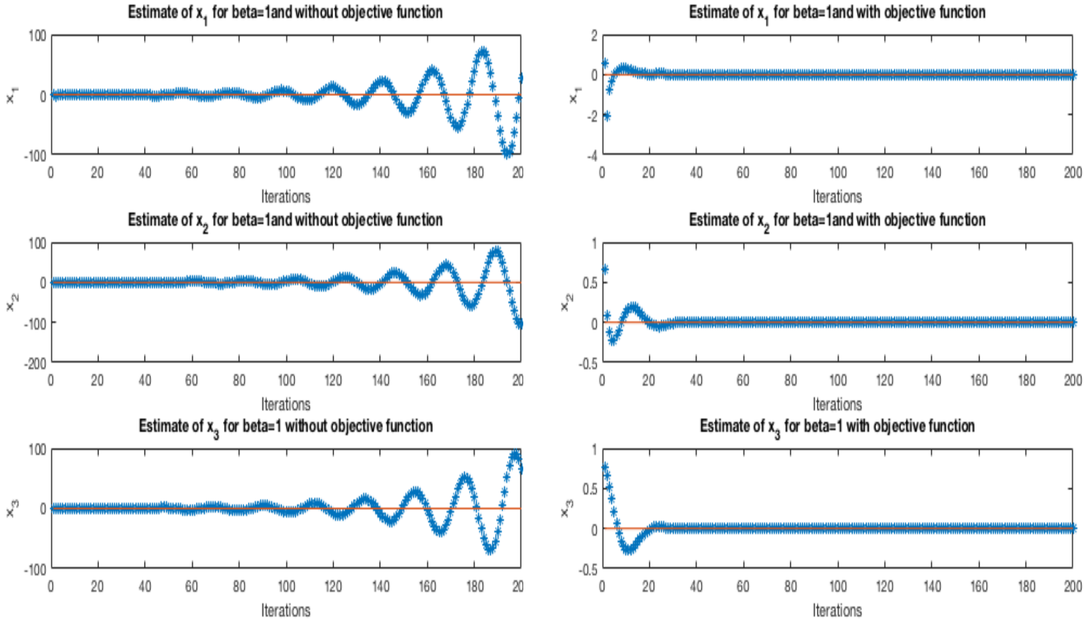
The right hand side plots in figures 11-13 depict the convergence/divergence of the estimated value of x for different values of β when objective function is introduced to the general problem. It should be clear that the ADMM now fails to find a convergent solution only for $\beta = 10$. The estimation, however, converges for $\beta = 0.1$ and $\beta = 1$.

Figure 8: Convergence of x for fixed value of $\beta = 0.1$.



From the above plots (Fig. 8), it should be clear that the ADMM fails to find a convergent solution when we optimize the divergent problem without the objective function. However, when the objective function is introduced, it follows that the ADMM converges in less than 100 iterations.

Figure 9: Convergence of x for fixed value of $\beta = 0.1$.



From the plots in Fig. 9, it should be clear that the ADMM fails to find a convergent solution when we optimize the divergent problem without the objective function. However, when the objective function is introduced, it follows that the ADMM converges in less than 40 iterations. If we observe the difference between Fig. 8 and Fig. 9, we notice that for a larger value of β the ADMM converges with a smaller number of iterations. Similarly, when the ADMM is applied to a divergent problem with no objective function, the magnitude of divergence from the real soltuion is smaller for greater value of β, in this case for $\beta = 1$.

Figure 10: Convergence of x for fixed value of $\beta = 10$.

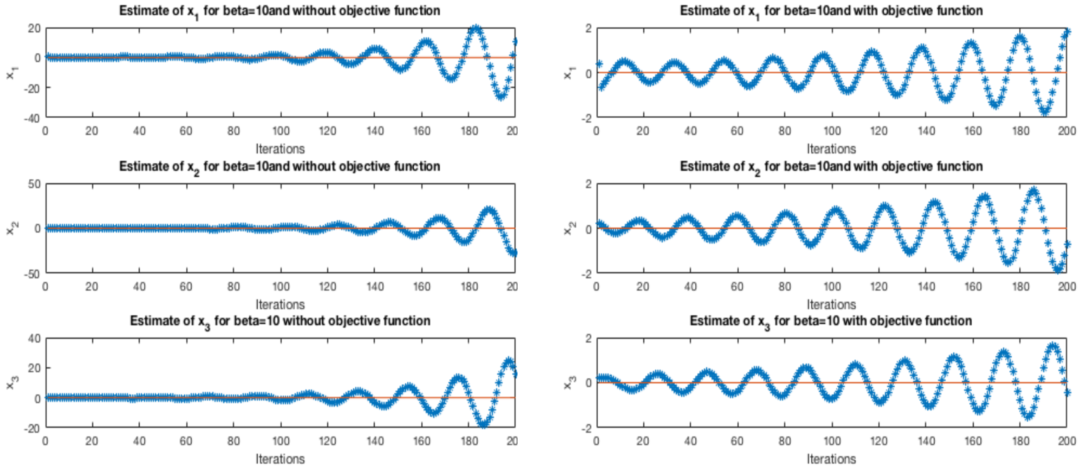


Fig. 10 shows that the ADMM method fails to find a convergent solution with and without the objective function. That may suggest that when β is too large the objective function cannot regularize the optimization problem which will continue to grow without a bound. It addition, the ADMM estimator without the objective function has a smaller magnitude of oscillation from the true value of x than the ADMM estimator with the addition of the objective $f(x)$.

Figure 11: $||Ax||$ for all Methods.

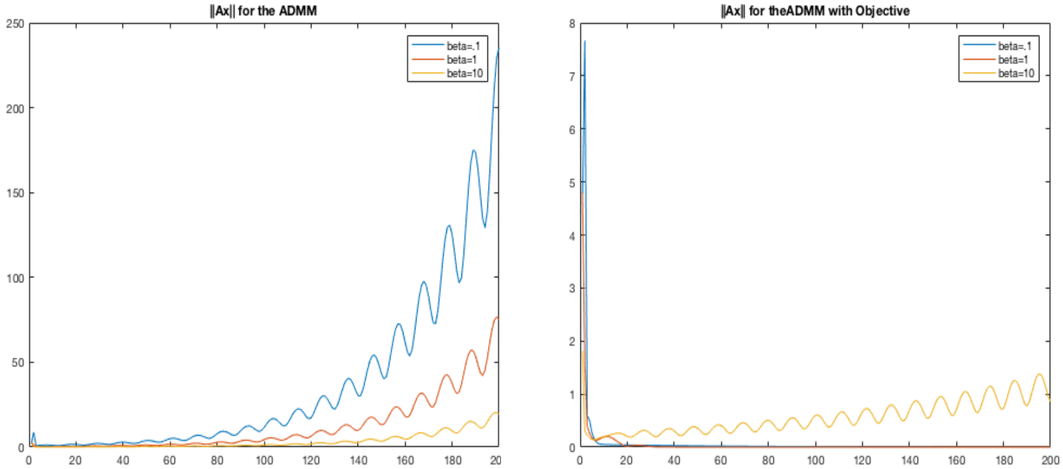


Fig. 11 shows the norm of the product of a matrix A and a solution of the problem x. As it can be easily seen from the plot the left, the ADMM fails to find the optimal solution that converges and thus the norm grows periodical without a bound. That growth is independent of the value of β, as well as the initial guess for x, as long as $x_0 \neq 0$. The plot on the right, on the other hand, shows that when $\beta \neq 10$ (norm diverges for $\beta = 10$), the ADMM supported with the objective function finds the true solution of the divergent problem. Thus, for $\beta = 0.1$ and $\beta = 1$, the norm converges to 0, as the true solution of $x = 0$.

(c) Set $\beta = 1$ and apply the random permutation updating order of $\mathbf{x}$ (discussed in class) to solving each of the two problems in (a) and (b). Does the iterate converge?

Based on Fig. 12-14, it should be clear that the ADMM with permutation finds a convergent solution that equals the true solution of for the given problem. We see that the method converges for all values of β not only $\beta = 1$. That implies that the introduction of random updating of x_i with $i = \{1, 2, 3\}$ prevents the solution from growing without a bound. In addition, the mentioned graphs show that the ADMM with permutation converges in a smaller number (approximately 20 vs. 200) of iterations when the objective function is added to the original problem.

Figure 12: Permutation Based Convergence of x for fixed value of $\beta = 0.1$.

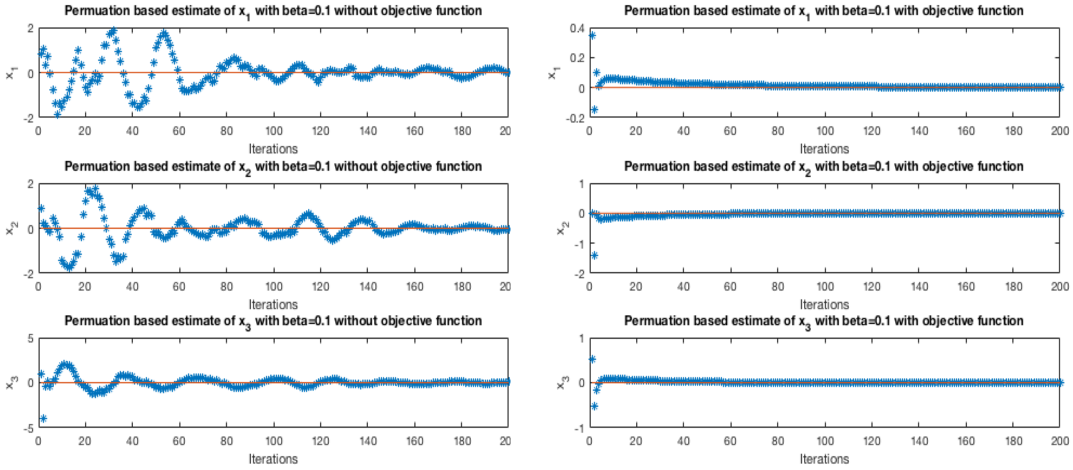


Figure 13: Permutation Based Convergence of x for fixed value of $\beta = 1$.

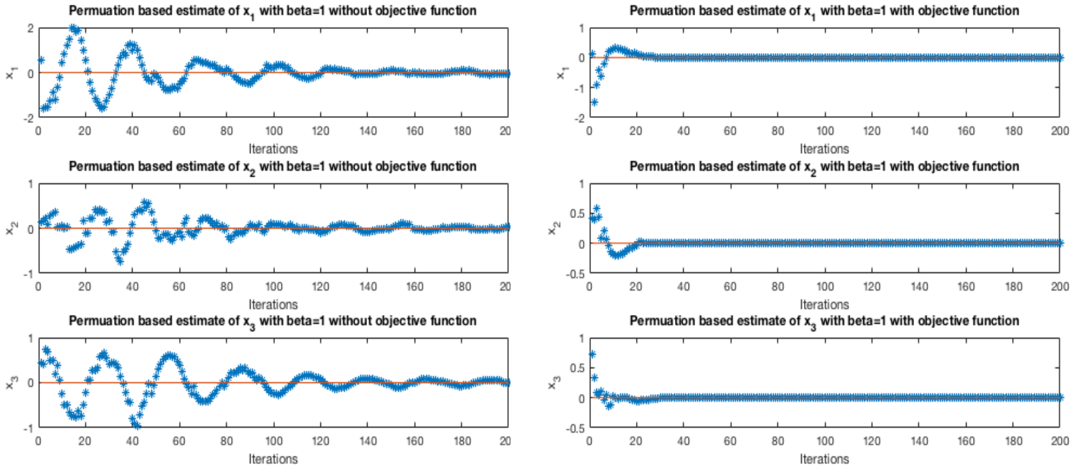


Figure 14: Permutation Based Convergence of x for fixed value of $\beta = 10$.

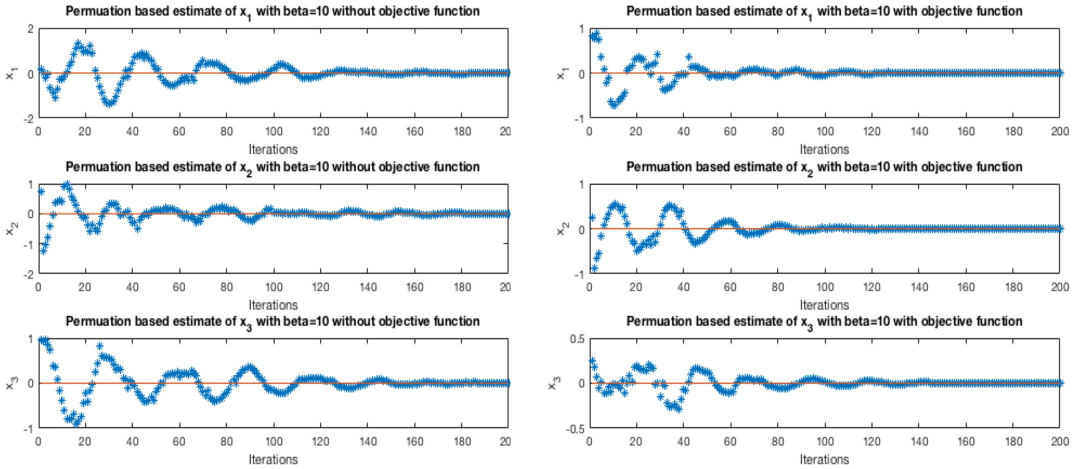


Figure 15: Permutation Based $||Ax||$ for all Methods.

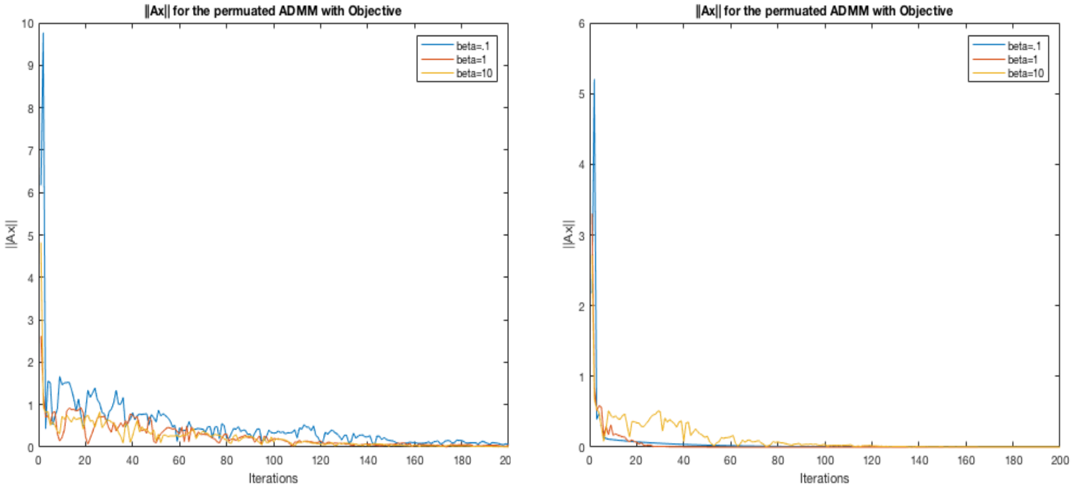


Fig. 15 shows the norm of the product of a matrix A and a solution of the problem x. As it can be easily seen from the plot, the ADMM with permutation successfully finds the optimal solution and thus the norm converges to 0 (as the solution of the problem is $x = 0$). The only difference in the we notice is the periodic behavior of the convergence and the speed with which the method attains the lower bound for different value of β. As previously discussed, the fastest convergence is associated with the largest value of β when the objective function is present in the original problem and the smallest β when the objective is absent from the formulation.

Problem 8

All pieces of Computation Team Work stated above, and one more described below.

You understand the SDP relaxation for SNL very well now, e.g., the two sensors and three anchors formulation: Find

$$Z = \begin{pmatrix} I & X \\ X^T & Y \end{pmatrix} \in S^4$$

to meet the constraints in the standard form:

$$\begin{aligned}
(1;0;0;0)(1;0;0;0)^T \bullet Z &= 1, \\
(0;1;0;0)(0;1;0;0)^T \bullet Z &= 1, \\
(1;1;0;0)(1;1;0;0)^T \bullet Z &= 2, \\
(a_i;-1;0)(a_i;-1;0)^T \bullet Z &= d_{1i}^2,\ i = 1,2, \\
(a_i;0;-1)(a_i;0;-1)^T \bullet Z &= d_{2i}^2,\ i = 2,3, \\
(0;0;1;-1)(0;0;1;-1)^T \bullet Z &= \hat{d}_{12}^2, \\
Z &\succeq 0 \in S^4.
\end{aligned}$$

where no objective function is present.

The question is: could we add an suitable (or regulative) objective function to improve localization quality:

$$\begin{array}{lrl}
\min & C \bullet Z & \\
\text{s.t.} & (1;0;0;0)(1;0;0;0)^T \bullet Z & = 1, \\
& (0;1;0;0)(0;1;0;0)^T \bullet Z & = 1, \\
& (1;1;0;0)(1;1;0;0)^T \bullet Z & = 2, \\
& (a_i;-1;0)(a_i;-1;0)^T \bullet Z & = d_{1i}^2,\ i = 1,2, \\
& (a_i;0;-1)(a_i;0;-1)^T \bullet Z & = d_{2i}^2,\ i = 2,3, \\
& (0;0;1;-1)(0;0;1;-1)^T \bullet Z & = \hat{d}_{12}^2, \\
& Z & \succeq 0 \in S^4.
\end{array}$$

Try the following regulative objectives and construct the corresponding objective matrix C, and see which works best:

a) Minimize the trace of Z

b) Maximize the trace of Z,

c) Minimize the sum of all the non-edge distance squares (a non-edge is a edge whose distance information is unknown)

d) Maximize the sum of the non-edge distance squares

Solution

Figure 16: Log Error of All Methods for 900 Iterations when x is Fixed at (1, 0)

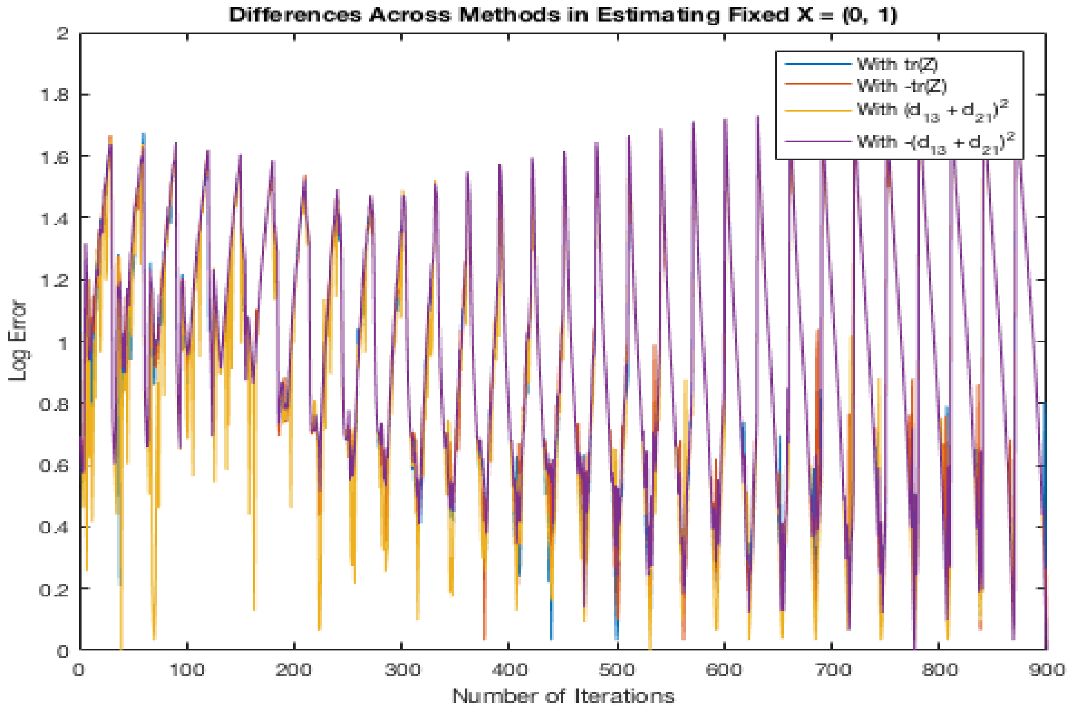


Figure 17: Log Error of All Methods for the First 49 Iterations when x is Fixed at (1, 0)

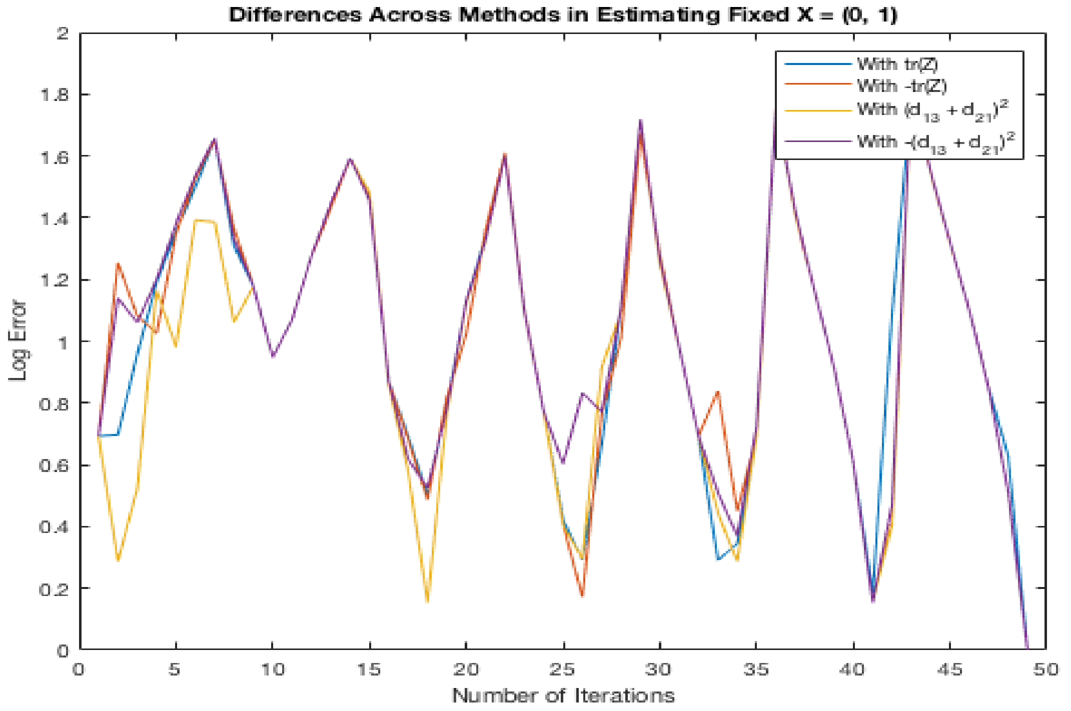


Figure 18: Log Error of All Methods for 900 Iterations when x is Fixed at (0.75, 0.35)

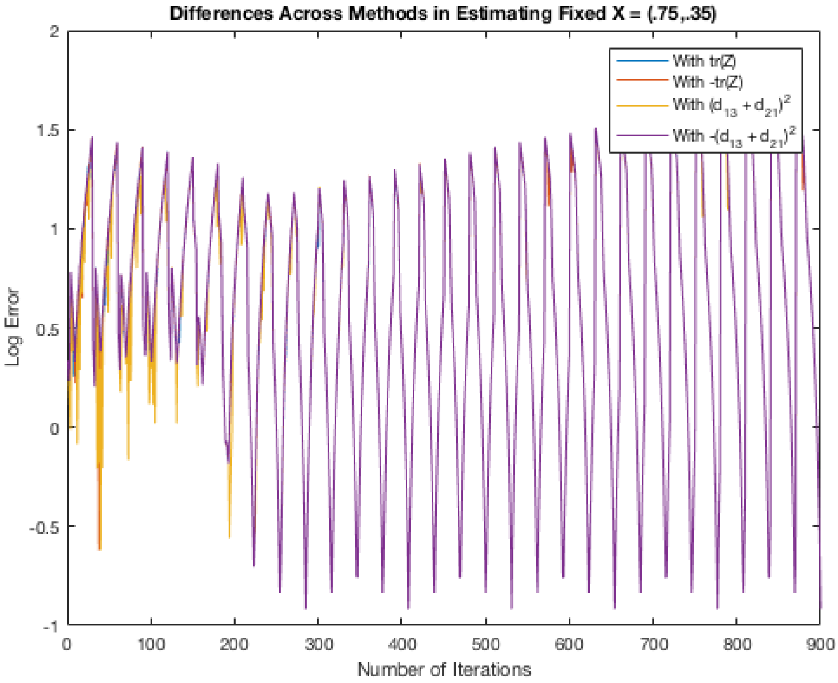
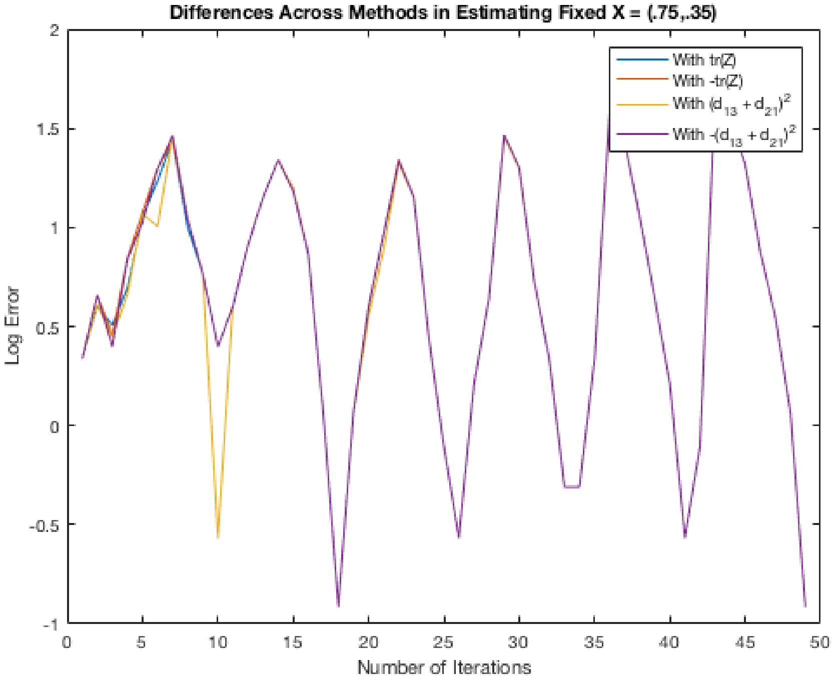


Figure 19: Log Error of All Methods for the First 49 Iterations when x is Fixed at (0.75, 0.35)



From plots depicted in Fig. 16-19, it can be noticed that the smallest magnitude of the error in estimating x is associated with method that uses a minimization of the sum of all non-edge distance squared as a regularizer. It should also be noted that a regularizer that minimizes the trace of Z performs well across all 900 iterations (Fig. 16 and Fig. 18), however it has bigger magnitude of error than the method mentioned above. Also, we notice that the magnitude of the error is mostly the same across methods when x is fixed at (0.75, 0.35). The explanation behind this phenomena is simple. When x is fixed closed to a vertex, each one of the given methods has a problem with finding an accurate value of x.

Figure 20: Histogram of the Log Error for Each Method for the first 900 Iterations when x is fixed at (0,1) and (0.75, 0.35)

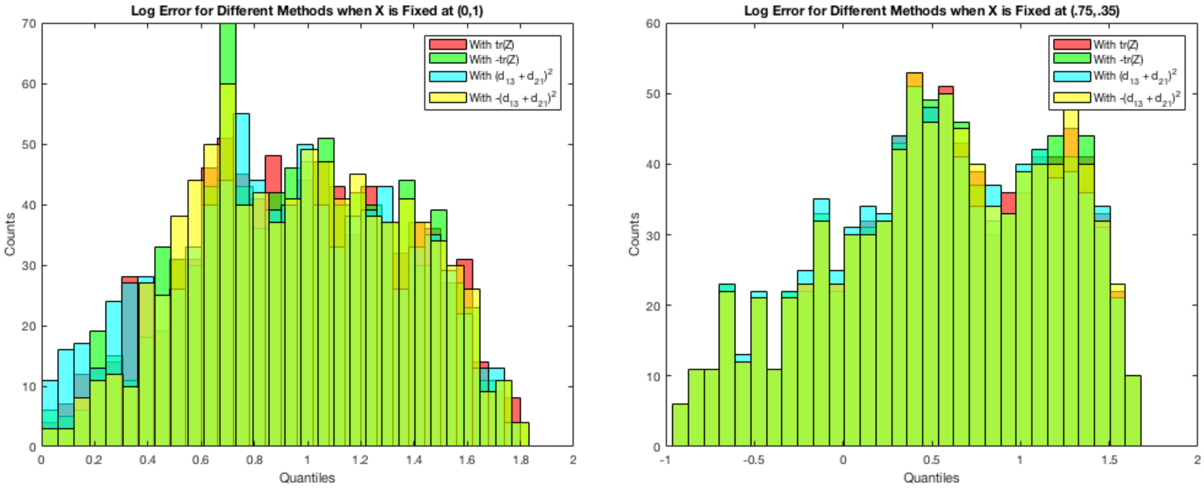
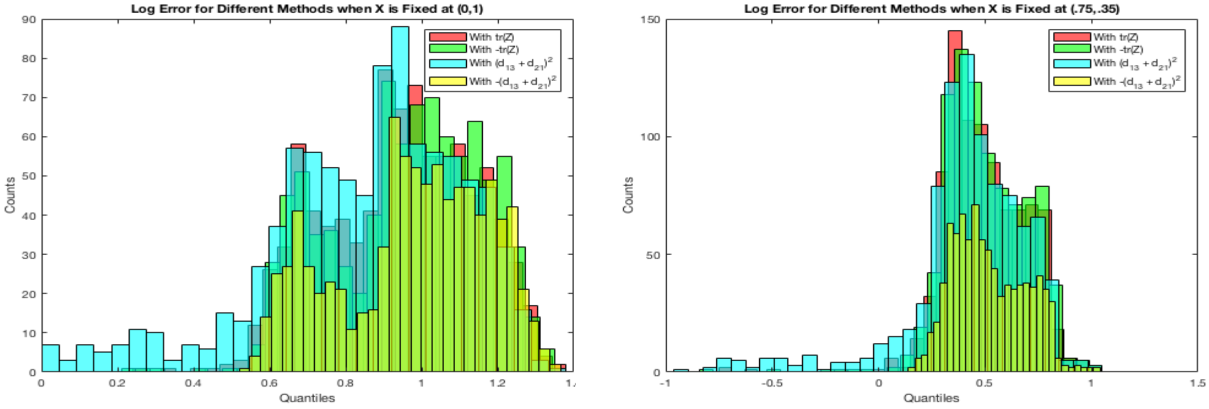


Figure 21: Histogram of the Log Error for Each Method for the first 200 Iterations



Similarly to the previously stated argument, we observe that histograms depicted by Fig. 20 and Fig. 21 show that the the method that uses the minimization of the sum of all non-edge distance squared has the smallest magnitude of error for almost all 900 different points (iterations), similarly for only first 200 estimations. It should also be noted that when x is fixed at $(0.75, 0.35)$ the overall difference in the magnitude of the error is smaller across all the methods when the number of points that we try to find is large. That is, all methods have similar error when estimating the value of x.

Figure 22: Magnitude of Error when X is Fixed at (0,1) and the Objective is to Minimize the Trace of Z.

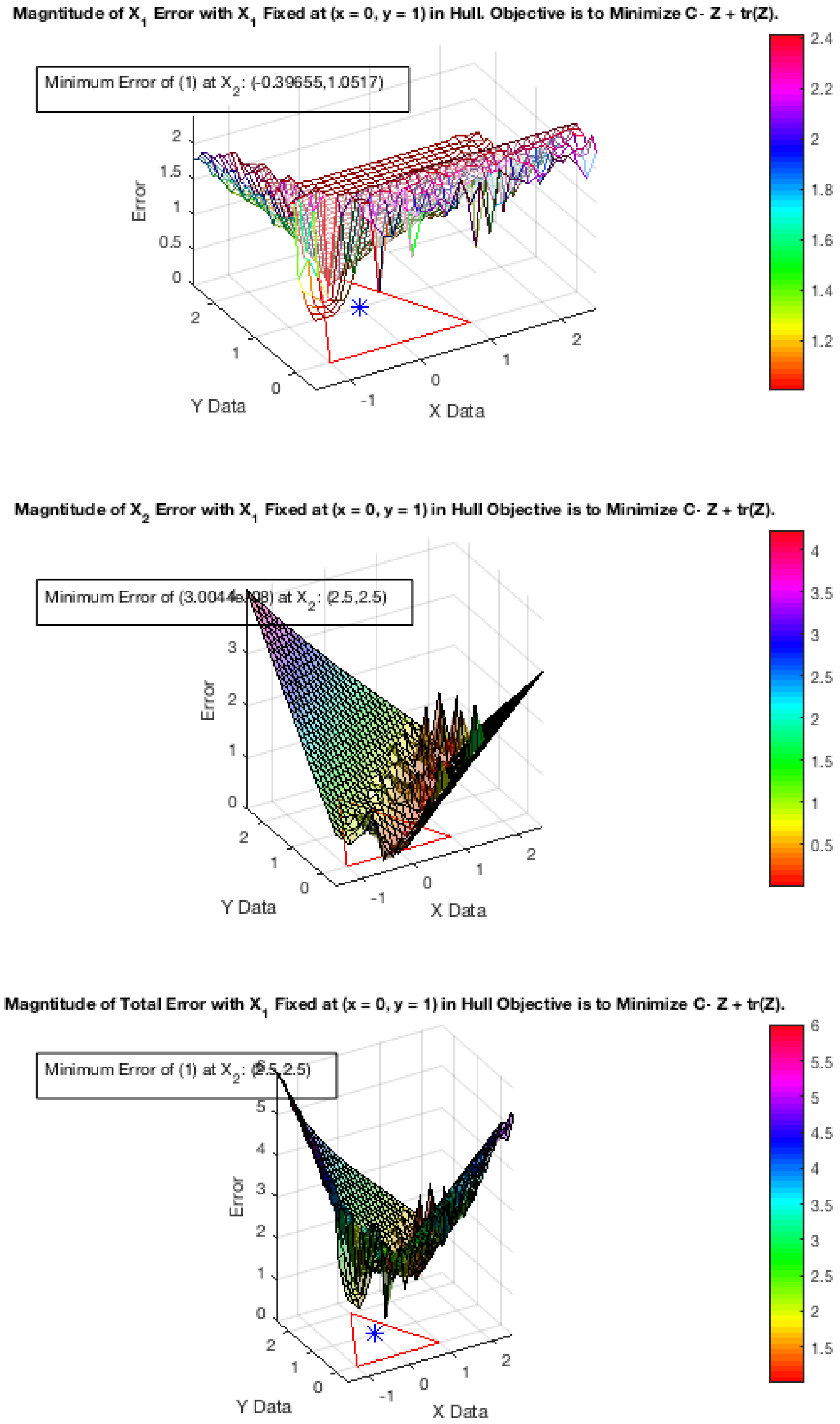
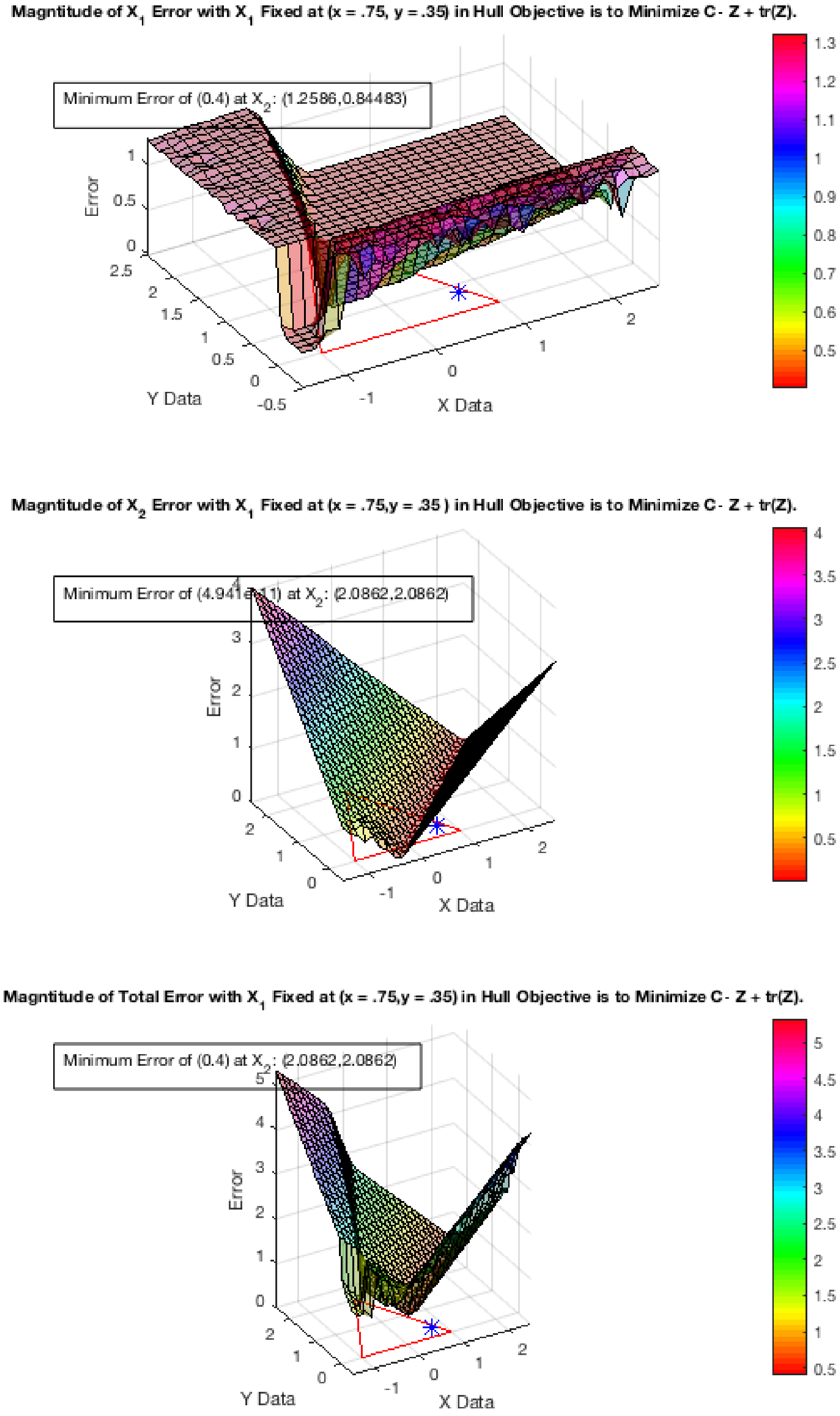


Figure 23: Magnitude of Error when X is Fixed at (0.75,0.35) and the Objective is to Minimize the Trace of Z.

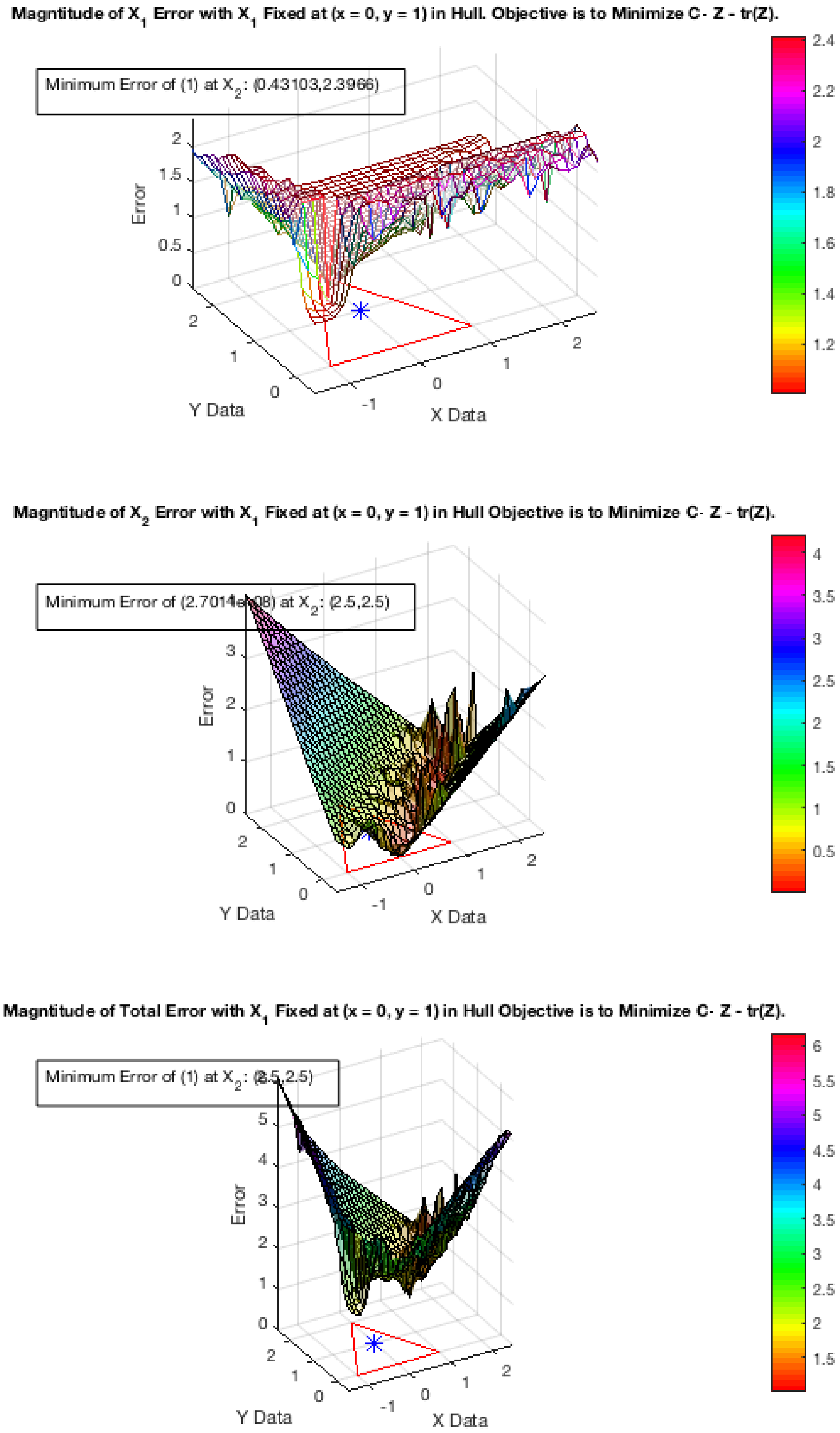


Magntitude of X_1 Error with X_1 Fixed at (x = .75, y = .35) in Hull Objective is to Minimize C - Z + tr(Z).

Magntitude of X_2 Error with X_1 Fixed at (x = .75,y = .35) in Hull Objective is to Minimize C - Z + tr(Z).

Magntitude of Total Error with X_1 Fixed at (x = .75,y = .35) in Hull Objective is to Minimize C - Z + tr(Z).

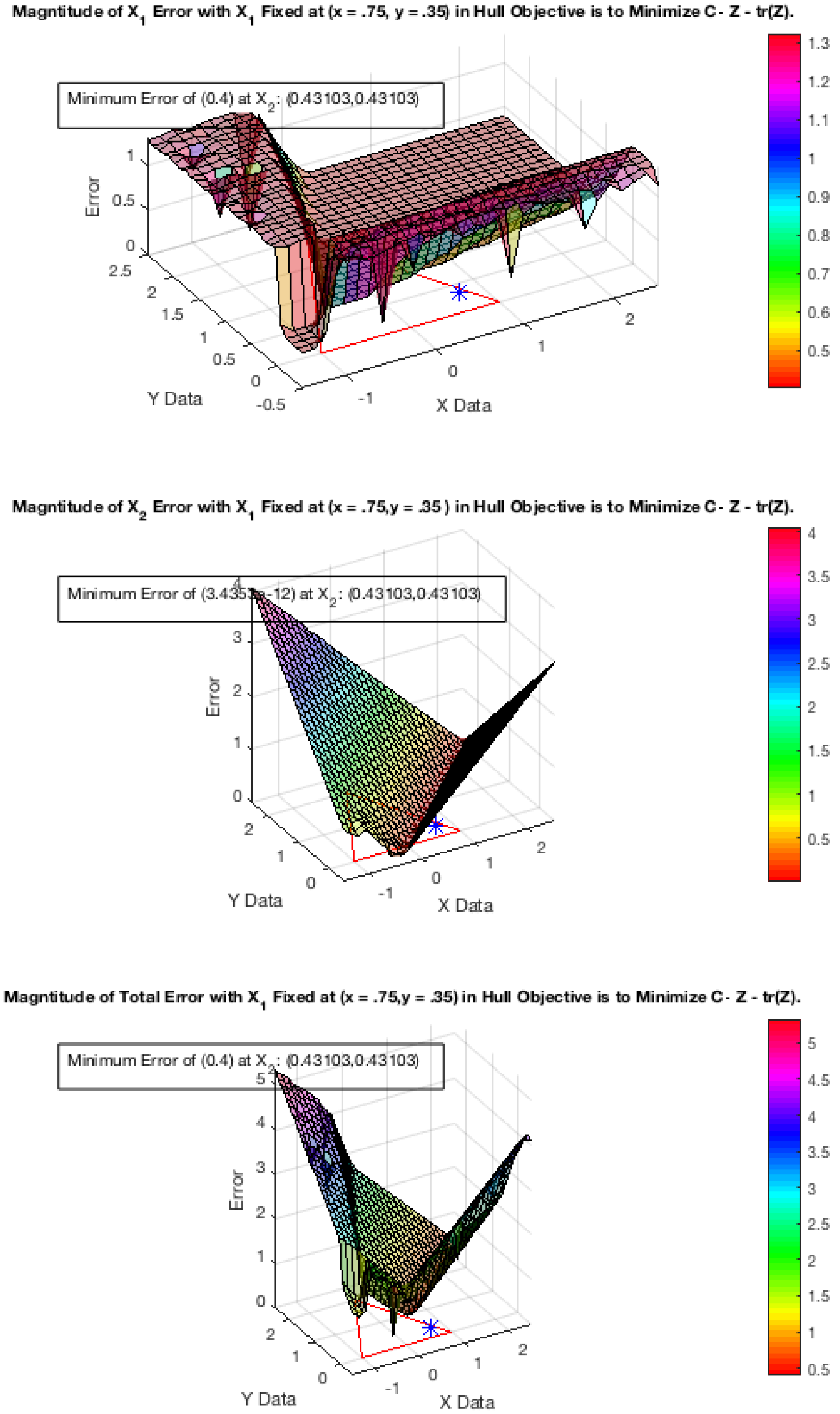
Figure 24: Magnitude of Error when X is Fixed at (1,0) and the Objective is to Maximize the Trace of Z.


Magntitude of X_1 Error with X_1 Fixed at (x = 0, y = 1) in Hull. Objective is to Minimize C- Z - tr(Z).

Magntitude of X_2 Error with X_1 Fixed at (x = 0, y = 1) in Hull Objective is to Minimize C- Z - tr(Z).

Magntitude of Total Error with X_1 Fixed at (x = 0, y = 1) in Hull Objective is to Minimize C- Z - tr(Z).

Figure 25: Magnitude of Error when X is Fixed at (.75,.35) and the Objective is to Maximize the Trace of Z.



Magntitude of X_1 Error with X_1 Fixed at (x = .75, y = .35) in Hull Objective is to Minimize C - Z - tr(Z).

Magntitude of X_2 Error with X_1 Fixed at (x = .75,y = .35) in Hull Objective is to Minimize C - Z - tr(Z).

Magntitude of Total Error with X_1 Fixed at (x = .75,y = .35) in Hull Objective is to Minimize C - Z - tr(Z).

Figure 26: Magnitude of Error when X is Fixed at (1,0) and the Objective is to Minimize the Distance.

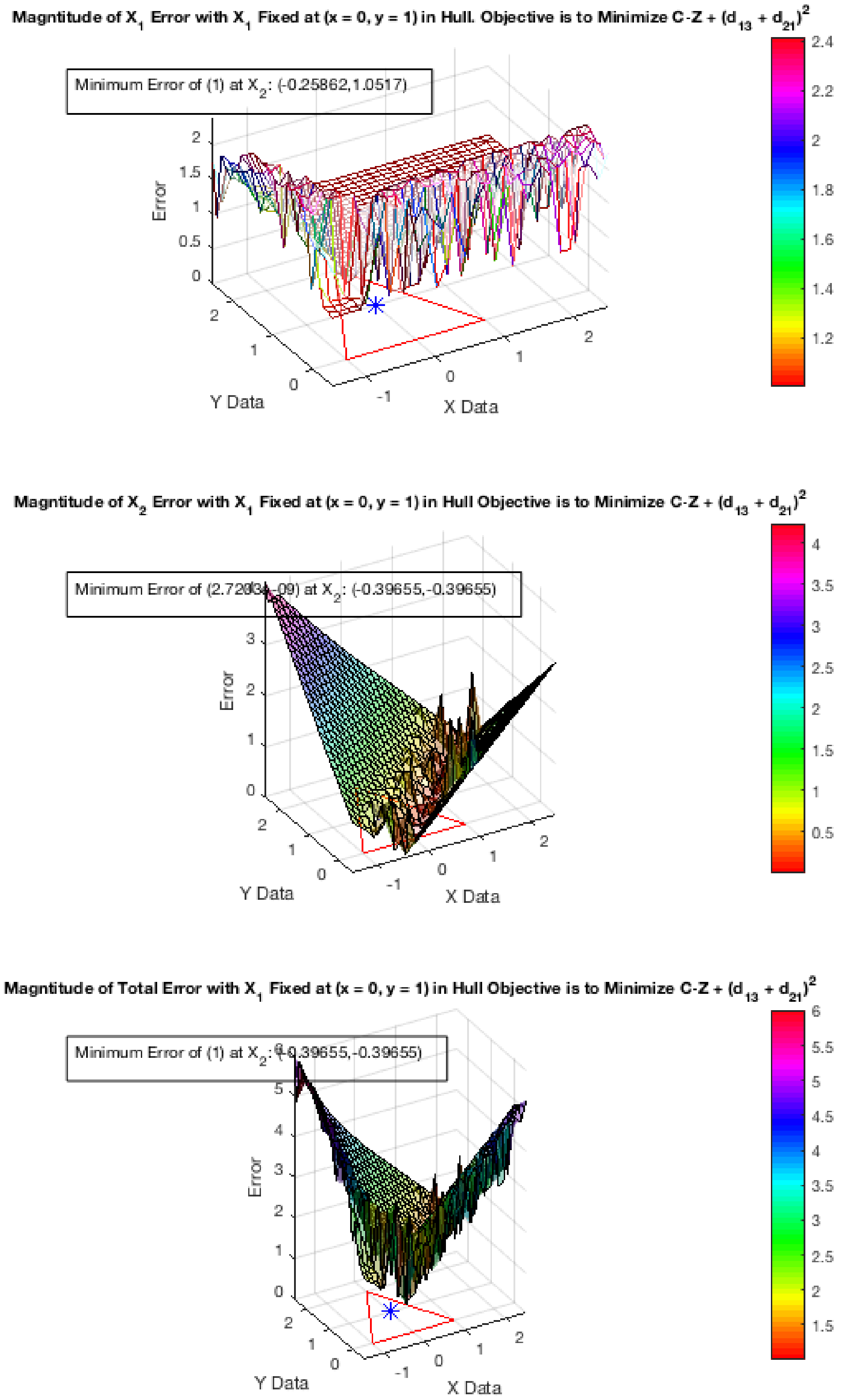


Figure 27: Magnitude of Error when X is Fixed at (.75,.35) and the Objective is to Minimize the Distance.

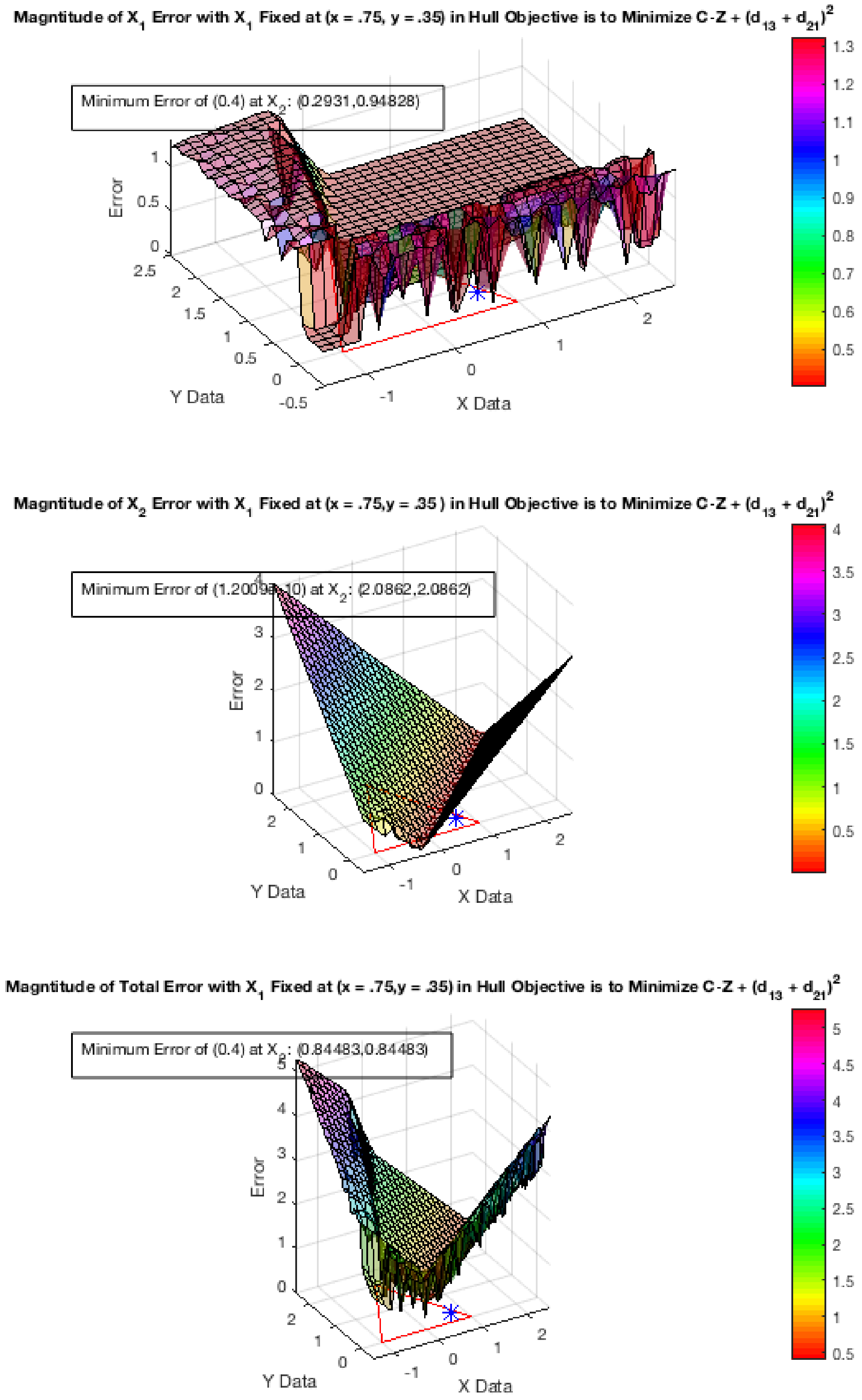


Figure 28: Magnitude of Error when X is Fixed at (0,1) and the Objective is to Maximize the Distance.

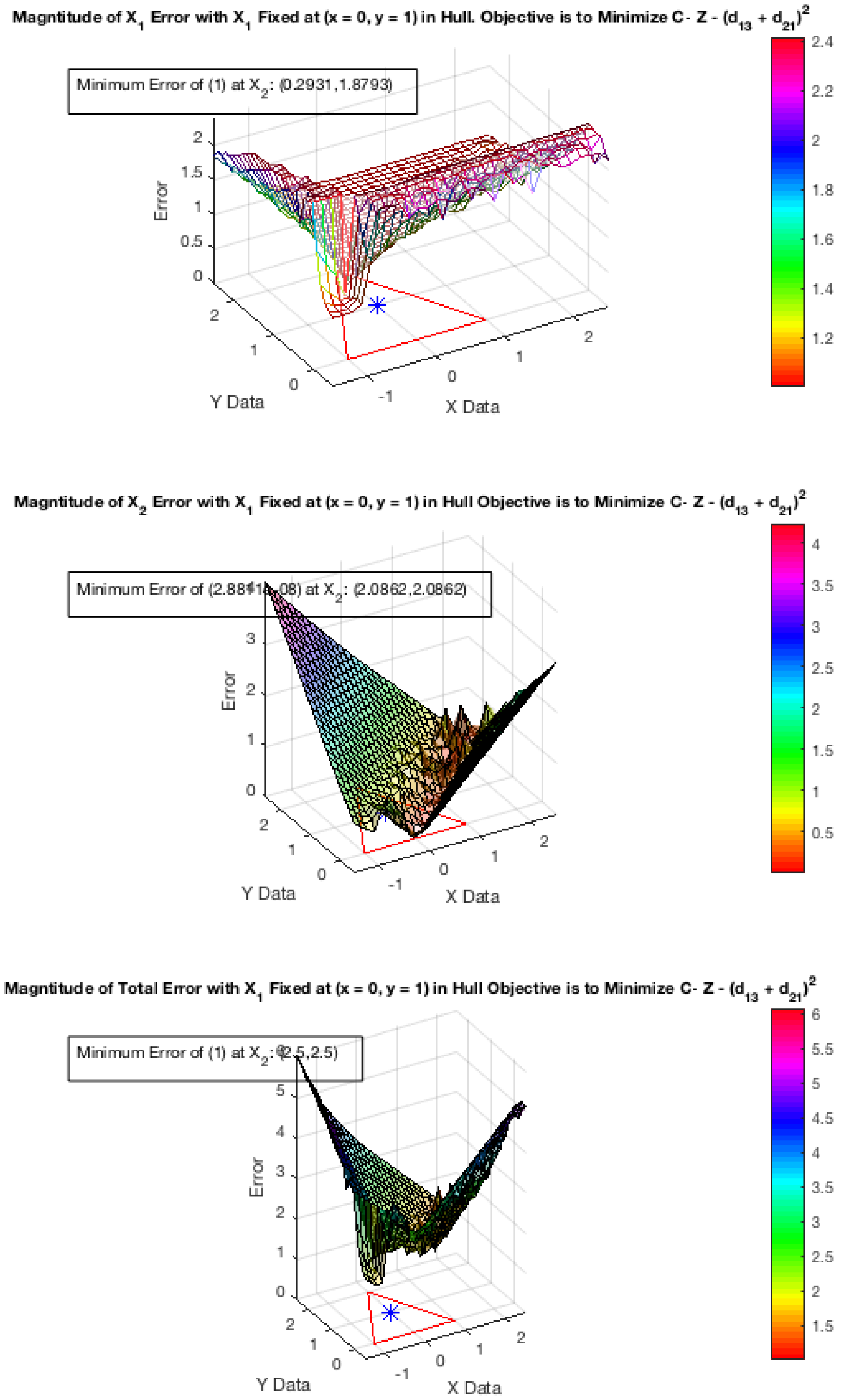
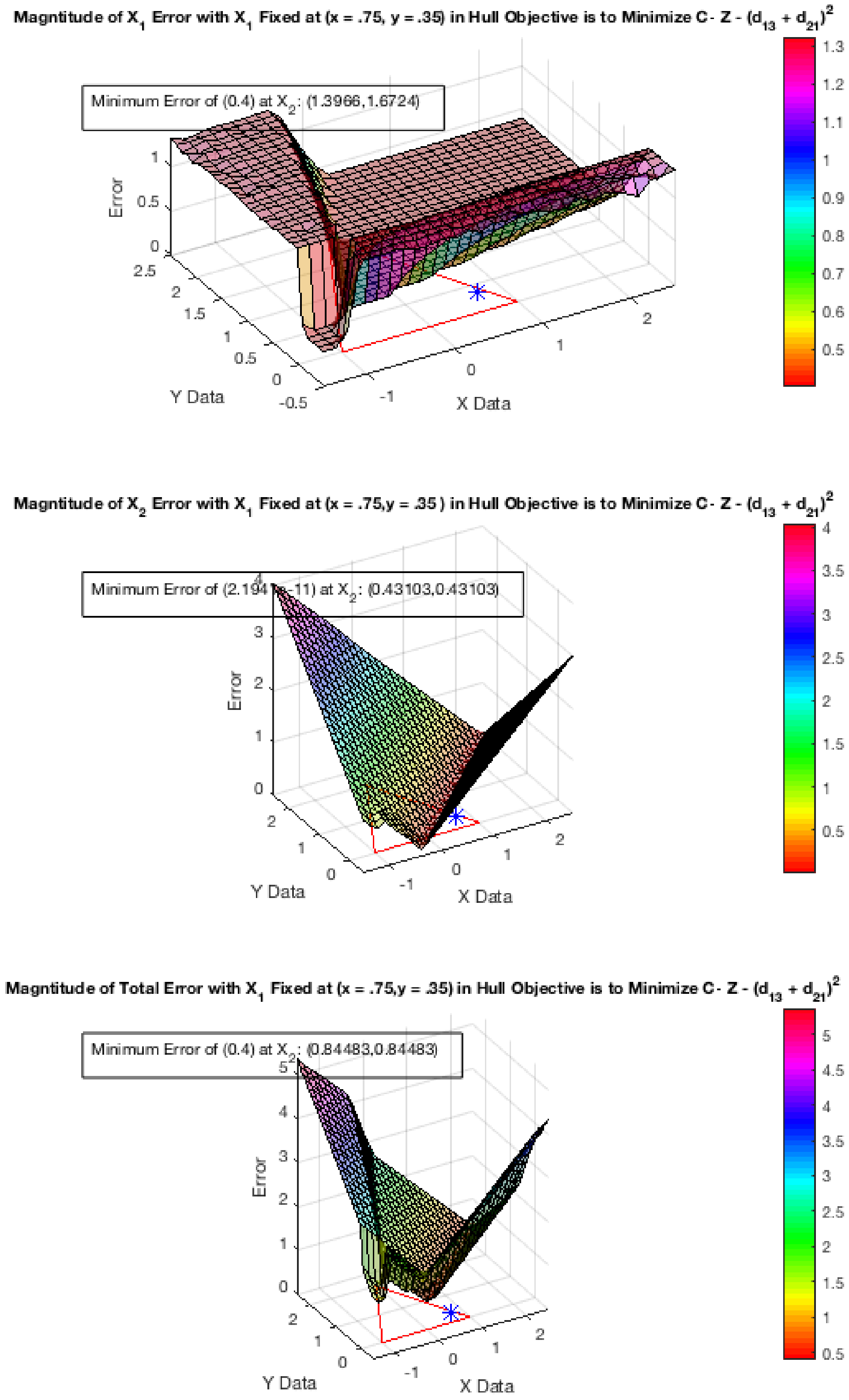


Figure 29: Magnitude of Error when X is Fixed at (0.75,0.35) and the Objective is to Maximize the Distance.



Magntitude of X_1 Error with X_1 Fixed at (x = .75, y = .35) in Hull Objective is to Minimize C - Z - $(d_{13} + d_{21})^2$

Minimum Error of (0.4) at X_2: (1.3966,1.6724)

Magntitude of X_2 Error with X_1 Fixed at (x = .75,y = .35) in Hull Objective is to Minimize C - Z - $(d_{13} + d_{21})^2$

Minimum Error of (2.1941e-11) at X_2: (0.43103,0.43103)

Magntitude of Total Error with X_1 Fixed at (x = .75,y = .35) in Hull Objective is to Minimize C - Z - $(d_{13} + d_{21})^2$

Minimum Error of (0.4) at X_2: (0.84483,0.84483)