Efficient thin plate spline interpolation and its application to adaptive optics

Ghosh, A.

Award date:
2010

JOHANNES KEPLER
UNIVERSITÄT LINZ | JKU

TNF

Technisch-Naturwissenschaftliche
Fakultät

Efficient Thin Plate Spline Interpolation and its Application to Adaptive Optics

MASTERARBEIT

zur Erlangung des akademischen Grades

DIPLOMINGENIEUR

im Masterstudium

INDUSTRIEMATHEMATIK

Eingereicht von:
Arpan Ghosh

Angefertigt am:
Institut für Industriemathematik

Beurteilung:
DI Dr. Stefan Kindermann (Betreuung)
Univ. Prof. Dr. Ronny Ramlau

Linz, August, 2010

.

Abstract

Thin plate splines provide smooth interpolation of the given data in two or more dimensions. These are analogous to cubic splines in one dimension. The main objectives of this report is to provide an implementation of the thin plate spine interpolation of data using various efficient methods and to investigate the possibility of using thin plate splines in adaptive optics. In this report, we consider the inverse problem derived from a minimization problem for thin plate spline interpolation. We solve it in Matlab using various methods and compare results. We also consider solving the problem by using QR decomposition as given by Wahba in Spline Models for Observation Data (SIAM 1990). Then, we look for possible applications of the thin plate splines in problems arising from adaptive optics.

Acknowledgements

It is my distinct pleasure to express my heartfelt gratitude to my master thesis supervisor Dr. Stefan Kindermann for providing me with invaluable encouragement and patient guidance to complete this thesis. I also thank my second supervisor Dr. Ronny Ramlau and my supervisor at TU/e Dr M Hochstenbach for their support which made this thesis possible. I would also like to thank my friends for their support throughout my master studies. I would specially like to thank Chhitiz Buchasia and Mikalai Zhudro for their priceless support and help throughout the year. The two year Erasmus Mundus scholarship awarded to me by the European Union has enabled me to complete my master studies. The experience I gained during these two years is really priceless. I would also like to thank Dr Martijn Anthonissen and Dr. Ewald Lindner for guiding me through the various stages of this master degree course. And a special thanks to my family for their eternal support and encouragement throughout my education.

Arpan Ghosh

August 18, 2010

Contents

Chapter 1

Introduction

Very often we come across the problem of interpolation of data by a function. For the interpolation of data in one dimension, one of the most popular tools is the cubic spline. The usefulness of the cubic splines takes us towards the next level, i.e. interpolation of data in more than one dimension. This is where the thin plate splines are used. Essentially, the thin plate splines are the generalizations of cubic splines in more than one dimension. The first chapter of this report is concerned with the theory of the thin plate splines.

Though our main goal is to investigate the implementation of the thin plate spline in two dimensions, we start off with a section on the theory of the general case of the thin plate spline. This section presents the minimization problem whose solution gives the thin plate spline. A brief discussion on the penalty functional follows. Then there is a look at the general form of the solution with the suitable basis functions. Then a section with a theorem on the two dimension case follows. This theorem gives us the existence and uniqueness of the solution of the minimization problem under certain conditions in the two dimension case which is the case of our primary interest. This is followed by an outline of its proof as given in [7].

The next chapter focusses on investigating the inverse problem to be solved in order to get the thin plate splines. The first section in this chapter describes the linear form of this problem while defining the involved matrices and vectors. The next section is dedicated to the derivation of the QR decomposition method as described in [6]. The following section is about condition numbers. Under this section the condition numbers of the matrices involved with our problem, are discussed comparatively.

The next two chapters concern the numerics applied and the results obtained for our problem respectively. First, some methods are discussed which we think could be useful to solve the problem on hand. This includes the GMRES method, the conjugate gradient method and Uzawa's iterative method for saddle point problems. And then some results are presented which are generated by using these methods to solve the linear problems.

The motivation for this report comes from adaptive optics for astronomy. In adaptive optics, the problem is to find a function whose other forms, for instance the derivative, will approximate the given data. The next chapter starts with an introduction on adaptive optics. Then a mathematical model for the approximation problem is described. The next two sections describe methods built to solve the adaptive optics problem while incorporating the knowledge from the theory of thin plate splines. The first of these uses normal equations and the other uses the usual thin plate spline techniques employed multiple times. The report is concluded with a chapter where some results are presented by implementing these methods.

Chapter 2

Thin Plate Splines

Thin Plate Splines are used to produce approximations to given data in more than one dimension. These are analogous to the cubic splines in one dimension. Duchon [1] [2] [3] and Meinguet [4] built the foundations for the thin plate splines. Further results and applications to meteorological problems were given by Wahba and Wendelberger[5]. The name Thin Plate Spline comes from the physical situation of bending of a thin surface. The Thin Plate Splines minimize the bending energy of a thin plate clamped at the data sites.

2.1 General case[6]

A Thin plate smoothing spline is produced by minimizing the following optimization problem

$$\frac{1}{n}\sum_{i=1}^{n}(z_i - f(t_i))^2 + \lambda J_m^d(f) \tag{2.1}$$

with $i \in \{1, \ldots, n\}$ where, z_i represent the i-th data and $t_i = (x_1(i), \ldots, x_d(i))$ represent the i-th data site given in d-dimension. n is the total number of data and J_m^d is a smoothness penalty functional with m-derivatives in d-

dimensions. λ is the smoothness parameter.

By choosing the value of λ, one can get the desired level of smoothness in the approximation at the cost of accuracy at the data sites. When we take this parameter to be zero, then we get the problem of just interpolating the data points without any smoothing. On the other hand taking the parameter towards infinity, we get the problem of finding a plane which is the least square fit of the data.

The smoothness penalty method can be chosen by some criteria. Generalized cross validation is one of the methods used for this.

The penalty functional in general, is given by

$$J_m^d(f) = \sum_{\alpha_1+\ldots+\alpha_d=m} \frac{m!}{\alpha_1!\ldots\alpha_d!} \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} \left(\frac{\partial^m f}{\partial x_1^{\alpha_1} \ldots \partial x_d^{\alpha_d}} \right)^2 \prod_j dx_j \quad (2.2)$$

The thin plate penalty functional for $d = 3$ and $m = 2$ for example is given by

$$\begin{aligned} J_2(f) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} (f_{x_1x_1}^2 &+ f_{x_2x_2}^2 + f_{x_3x_3}^2 \\ &+ 2[f_{x_1x_2}^2 + f_{x_2x_3}^2 + f_{x_3x_1}^2])dx_1dx_2dx_3 \end{aligned} \quad (2.3)$$

Let

$$\langle f, g \rangle = \sum_{\alpha_1+\ldots+\alpha_d=m} \frac{m!}{\alpha_1!\ldots\alpha_d!}\mathbf{I} \quad (2.4)$$

where

$$\mathbf{I} = \int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} \left(\frac{\partial^m f}{\partial x_1^{\alpha_1} \dots \partial x_d^{\alpha_d}} \right) \left(\frac{\partial^m g}{\partial x_1^{\alpha_1} \dots \partial x_d^{\alpha_d}} \right) \prod_j dx_j$$

Formal integration by parts yields

$$\langle f, g \rangle = (-1)^m \int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} f \cdot \Delta^m g + c_\infty \tag{2.5}$$

where c_∞ is due to the boundary values.

We let f and g to be in a function space χ which consists of elements having partial derivatives of total order m in $\mathcal{L}_2(E^d)$.
This space of functions is a reproducing kernel Hilbert space (with the above mentioned inner product), if and only if $2m - d > 0$.

The null space of the penalty functional J_m^d is the $M = \binom{d+m-1}{d}$ dimensional space spanned by polynomials in d variables of degree $\leq m - 1$. As an example, for the case $d = 2$ and $m = 2$, the 3-dimensional null space is spanned by the monomials $1, x_1, x_2$.

If $t_1, \dots, t_n$ are chosen such that the least square regression on the M monomials of degree less than m, denoted by $\phi_1, \dots, \phi_M$ is unique then the minimization problem has a unique solution f_λ, which is given by

$$f_\lambda(t) = \sum_{i=1}^{M} d_i \phi_i(t) + \sum_{j=1}^{n} c_j E_m(t, t_j) \tag{2.6}$$

where t is a d-dimensional variable, $\{d_i\}_{i=1}^M$ and $\{c_j\}_{j=1}^n$ are constants and

E_m is a Green's function for the m-iterated Laplacian given by

$$E_m(t,s) = \begin{cases} \alpha_{m,d}|t-s|^{2m-d}\ln|t-s|^2, & \text{if } 2m-d \text{ is even} \\ \\ \beta_{m,d}|t-s|^{2m-d}, & \text{otherwise} \end{cases} \tag{2.7}$$

where

$$\alpha_{m,d} = \frac{(-1)^{d/2+1+m}}{2^{2m}\pi^{d/2}(m-1)!(m-d/2)!}$$

and

$$\beta_{m,d} = \frac{\Gamma(d/2-m)}{2^{2m}\pi^{d/2}(m-1)!}$$

E_m has the following property

$$\Delta^m E_m(\cdot, s) = \delta_s \tag{2.8}$$

where Δ_s is the Dirac delta function.

Taking $m = 2$ for example, E_m for $d = 1, 2, 3$ are as follows

$$E_m(t,s) = \begin{cases} \dfrac{1}{12}|t-s|^3, & \text{for } d=1 \\ \\ \dfrac{1}{16\pi}\|t-s\|^2\ln\|t-s\|^2, & \text{for } d=2 \\ \\ \dfrac{-1}{8\pi}\|t-s\|, & \text{for } d=3 \end{cases} \tag{2.9}$$

The interesting thing to note here is that as the dimension increases, the basis function looses its smoothness making it less useful for smooth interpolation of data in higher dimensions. This can be resolved by increasing

m for higher dimensions. Which means for smooth interpolation in higher dimension, the smoothness penalty functional must be with higher order derivatives.

Now let T be the $n \times M$ matrix with ij-th entry given by $\phi_j(t_i)$. And, let K be the $n \times n$ matrix with ij-th entry given by $E_m(t_i, t_j)$.

If $t_1, \ldots, t_n$ are such that T is of full rank then

$$c'Kc > 0 \tag{2.10}$$

for any $c = (c_1, \ldots, c_n)$ satisfying $T'c = 0$.

In one dimension, taking distinct t_i's ensures full rank for T. In two dimensions, T is of full rank unless t_i's are collinear.

By taking f as in 2.6 and taking inner product as defined in 2.4, we have that taking c to satisfy $T'c = 0$, we get

$$\begin{aligned} \langle f, f \rangle &= \langle \sum_{i=1}^{n} c_i E_m(\cdot, t_i), \sum_{j=1}^{n} c_j E_m(\cdot, t_j) \rangle \\ &= \sum_{i,j=1}^{n} c_i c_j E_m(t_i, t_j) \\ &= c'Kc > 0 \end{aligned} \tag{2.11}$$

where we have used 2.5, 2.8 and 2.10.

Then by using 2.1, 2.6 and 2.11 we get that c, d minimize

$$\frac{1}{n}\|y - Td - Kc\|^2 + \lambda c'Kc \tag{2.12}$$

subject to $T'c = 0$.

2.2 The 2-dimension case

The following theorem about the solution in 2-dimension case.

Theorem 2.2.1. [7] *Let* $t_i = (x_i, y_i), t = (x, y)$ *and* $|t - t_i| = ((x - x_i)^2 + (y - y_i)^2)^{1/2}$. *Let* $m \geq 2$ *and* $n \geq M = \binom{m+1}{2}$. *The solution* $u_{n,m,\lambda}$ *to the problem: Find* $u \in H$ *to minimize*

$$\frac{1}{n}\sum_{i=1}^{n}(u(t_i) - z_i)^2 + \lambda \int\int \sum_{j=0}^{m} \binom{m}{j} \left(\frac{\partial^m u}{\partial x^j \partial y^{m-j}}\right)^2 dxdy \tag{2.13}$$

is given by

$$u_{n,m,\lambda}(t) = \sum_{j=1}^{n} c_j E_m(t, t_j) + \sum_{i=1}^{n} d_i \phi_i(t), \tag{2.14}$$

where

$$E_m(s, t) = \theta_m |s - t|^{2m-2} \log |s - t|,$$

$$\theta_m = (2^{2m-1}\pi[(m-1)!]^2)^{-1}$$

$$\phi_i(t) = x^\alpha y^\beta \text{ for } i = 1, \ldots, M$$

where α, β *run over all the* M *combinations of non-negative integers with* $\alpha + \beta \leq m - 1$, *provided the* $n \times M$ *matrix* T *with* ij*-th entry* $\phi_j(t_i)$ *is of*

rank M*.*

The coefficients $\underline{c} = (c_1, \ldots, c_n)'$ *and* $\underline{d} = (d_1, \ldots, d_M)'$ *is determined by*

$$(K + \alpha I)\underline{c} + T\underline{d} = \underline{z} \tag{2.15}$$

$$T'\underline{c} = 0 \tag{2.16}$$

where K *is the* $n \times n$ *matrix with* jk*-th entry* $E_m(t_j, t_k)$ *and* $\alpha = n\lambda$

Proof. The outline of the proof of this theorem goes like this as in [7]
Let $r_1, \ldots, r_M$ be a subset of M points selected from $t_1, \ldots, t_n$ with the property that the $M \times M$ matrix T with ij-th entry $\phi_j(r_i)$ is of full rank. The space $H = \{u : u \in \mathcal{D}', \frac{\partial^m u}{\partial x^j \partial y^{m-j}} \in L_2, j = 0, 1, \ldots, m-1\}$ can be decomposed into the direct sum of two spaces:

$$H = x_{m-1} \oplus \overline{X} \tag{2.17}$$

where x_{m-1} is the M dimensional space of polynomials of total degree $m-1$ or less and

$$\overline{X} = \{u : u \in H, u(r_i) = 0, i = 1, \ldots, M\}.$$

It can then be shown that

$$\langle u, v \rangle_{\overline{X}} = \int\int \sum_{j=0}^{m} \binom{m}{j} \frac{\partial^m u}{\partial x^j \partial y^{m-j}} \frac{\partial^m v}{\partial x^j \partial y^{m-j}} dxdy \tag{2.18}$$

defines an inner product on $\overline{X}$. If an inner product is defined on x_{m-1} by

$$\langle u, v \rangle_{x_{m-1}} = \sum_{i=1}^{M} u(r_i)v(r_i),$$

then x_{m-1} and $\overline{X}$ are orthogonal subspaces. $\overline{X}$ (and x_{m-1}, and hence H) are reproducing kernel spaces.
If the reproducing kernel $K(s,t)$ for $\overline{X}$ can be found, then the solution $u_{n,m,\lambda}$ to the minimization problem of 2.13 will have a representation

$$u_{n,m,\lambda}(t) = \sum_{j=1}^{n} c_j K(t,t_j) + \sum_{i=1}^{n} d_i \phi_i(t) \tag{2.19}$$

$u_{n,m,\lambda}$ will be independent of the choice of $r_1, \ldots, r_M$.

The reproducing kernel K has been found by Meinguet [8] [4] and is given by

$$\begin{aligned} K(s,t) = E_m(s,t) \; &- \; \sum_{k=1}^{M} p_k(s) E_m(t,r_k) \\ &- \; \sum_{l=1}^{M} p_l(s) E_m(t,r_l) \\ &+ \; \sum_{k,l=1}^{M} p_k(s) p_l(t) E_m(r_k,r_l) \end{aligned} \tag{2.20}$$

where $\{p_k\}_{k=1}^{M}$ span x_{m-1} and are chosen so that $p_k(r_l) = 1$ if $k = l$, $= 0$ if $k \neq l$.

Substituting, 2.20 into 2.19 , it is seen that a representation of the form 2.14 for $u_{n,m,\lambda}$ holds.
To show that K is the reproducing kernel for $\overline{X}$, it is necessary to show that

$$K(s,\cdot) \in \overline{X} \text{ and } \langle K(s,\cdot), K(t,\cdot) \rangle_{\overline{X}} = K(s,t) \tag{2.21}$$

Define

$$H_s(t) = E_m(s,t) - \sum_{k=1}^{M} p_k(s) E_m(r_k,t). \tag{2.22}$$

Then

$$K(s,t) = H_s(t) - \sum_{l=1}^{M} p_l(t) H_s(r_l) \tag{2.23}$$

Meinguet showed that $H_s \in H$, for each s. It then follows that $K(s,\cdot) \in H$ and since $\sum_{l=1}^{M} p_l(\cdot) H_s(r_l)$ is the polynomial interpolating to H_s at $r_1, \ldots, r_M$,

$$K(s, r_l) = 0, l = 0, 1, \ldots, M,$$

and so $K(s,\cdot) \in \overline{X}$. To establish 2.21 first note that

$$\frac{\partial^m u}{\partial x^j \partial y^{m-j}} K(s,\cdot) = \frac{\partial^m u}{\partial x^j \partial y^{m-j}} H_s(\cdot) \tag{2.24}$$

Consider the Green's formula

$$(-1)^m \sum_{j=0}^{m} \binom{m}{j} \int\int \frac{\partial^m u}{\partial x^j \partial y^{m-j}} \frac{\partial^m v}{\partial x^j \partial y^{m-j}} dxdy = \int\int \triangle^m u \cdot v dxdy \tag{2.25}$$

where $\triangle = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$. This formula holds provided, e.g. $v \in H \cap L_2$ and $u \in \mathcal{D}$. If $u \in \mathcal{D}$, then the potential formula

$$\int\int (\triangle^m u)(t) E_m(s,t) dt = u(s)$$

holds and in particular

$$\int\int \triangle^m u \cdot H_s = u(s) - \sum_{l=1}^{M} p_l(s) u(r_l). \tag{2.26}$$

Meinguet argues that 2.25 and 2.26 hold for $u = H_t, v = H_s$, giving

$(-1)^m \sum_{j=0}^{m} \binom{m}{j} \int\int \frac{\partial^m}{\partial x^j \partial y^{m-j}} H_t \frac{\partial^m v}{\partial x^j \partial y^{m-j}} H_s dxdy = H_t(s) - \sum_{l=1}^{M} p_l(s) H_s(r_l) \equiv K(s,t)$

which combined with 2.24 gives 2.21. Equation 2.16 can be obtained as follows: Considering $K(t, t_j$ as a function of t,

$$K(t,t_j) = E_m(t,t_j) \; - \; \sum_{l=1}^{M} p_l(t_j) E_m(t,r_l) \qquad (2.27)$$
$$+ \quad \text{a polynomial of degree } m-1 \text{ or less.}$$

Now, if ϕ is any element of x_{m-1}, we have

$$\phi(t) - \sum_{l=1}^{M} p_l(t)\phi(r_l) \equiv 0 \qquad (2.28)$$

Letting $\alpha_1(j), \ldots, \alpha_n(j)$ be the coefficients of $E_m(\cdot, t_1), \ldots, E_m(\cdot, t_n)$, in 2.27 it can be verified from 2.28 that

$$\sum_{k=1}^{n} \alpha_k(j)\phi(t_k) \equiv 0, j = 1, \ldots, n,$$

which results directly in the conditions 2.16 on the coefficient vector $\underline{c}$ in 2.14, namely, $T'\underline{c} = 0$.

Equation 2.15 is obtained as follows: one substitutes 2.19 into 2.13 and then uses 2.21 to evaluate the expression 2.13 to be minimized. By repeatedly using $T'\underline{c} = 0$, one obtains that $\underline{c}$ and $\underline{d}$ are chosen subject to $T'\underline{c} = 0$, to minimize

$$\|\underline{x} - K\underline{c} - T\underline{d}\|^2 + n\lambda \underline{c}' K \underline{c}. \qquad (2.29)$$

Differentiating this expression with respect to $\underline{c}$ and setting the result equal to zero and using $T'\underline{c} = 0$, gives 2.15 □

This theorem guarantees the existence of the solution for the minimization problem for $d = 2$ subject to the conditions $m \geq 2$, $n \geq M$ and T is of full rank.

We are mainly interested in the case $m = 2$. So the first condition is already met in this case. As for the second, it means for us that we should have at least three data points to produce a unique approximation of a surface which is a very obvious condition. And the full rank condition for T is taken care by choosing non collinear data points. In our case, the smoothness penalty functional is

$$J_2^2(f) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (f_{x_1x_1}^2 + 2f_{x_1x_2}^2 + f_{x_2x_2}^2) dx_1 dx_2 \tag{2.30}$$

In the next chapter we will look at the inverse problem which we have to program.

Chapter 3

The Inverse Problem

Equipped with all the knowledge from 2 we are now in the position to form and investigate the inverse problem to be solved with Matlab for producing the thin plate spline approximations. We recall that we are interested in solving the following primary problem

$$\min_f \frac{1}{n}\sum_{i=1}^{n}(z_i - f(t_i))^2 + \lambda J_2^2(f) \tag{3.1}$$

where n is the number of data points, t_i is the ith data point z_i is the observation at the ith point for $i \in \{1, \ldots, n\}$, λ is the smoothness parameter and J_2^2 as in 2.30.

3.1 The linear form

Let E_m be as in 2.9 ($d = 2$ case). Let K be an $n \times n$ matrix with the ij-th entry given by $E_m(t_i, t_j)$. Let T be an $n \times 3$ matrix with the ith row being $(1, x_i, y_i)$ where we have $t_i = (x_i, y_i)$.

By theorem 2.2.1 the solution to the minimization problem has the form as in 2.14. So, in order to know the solution to the minimization prob-

lem, it is required for us to find the coefficient vectors $c = (c_1, \ldots, c_n)'$ and $d = (d_1, d_2, d_3)'$.

By theorem 2.2.1 again, we know that c and d are given by the relations

$$(K + \alpha I)\underline{c} + T\underline{d} = \underline{z} \tag{3.2}$$

$$T'\underline{c} = 0 \tag{3.3}$$

where $\alpha = n\lambda$.

These relations are expressed as a single relation as follows

$$\begin{bmatrix} K + \alpha I & T \\ T' & 0 \end{bmatrix} \begin{bmatrix} c \\ d \end{bmatrix} = \begin{bmatrix} z \\ 0 \end{bmatrix} \tag{3.4}$$

where $z = (z_1, \ldots, z_n)$.

This is the main inverse problem we would try to solve for different values of α with various tools in the next chapter.

From here onwards, we use

$$M_\alpha = \begin{bmatrix} K + \alpha I & T \\ T' & 0 \end{bmatrix}$$

M_α is clearly symmetric.

For $\alpha = 0$, the problem simply becomes an interpolation problem. We

would also investigate this case along with others. One of the main computational difficulties in solving this problem using simple methods is that the matrix M_α is not at all sparse for large n. In fact, only 9 entries in case of $\alpha = 0$ and $n + 9$ entries in case of non zero α are zeros.

Another problem is that M_α is not positive definite, ruling out the possibility to use methods like conjugate gradient. But we can avoid this by using the QR method which is discussed in a later section. And moreover, the high condition number of the matrices K and M pose serious restrictions on the number of data points to be used. We shall discuss the condition numbers in a section.

3.2 Using QR decomposition[6]

A QR decomposition of a matrix A produces an orthogonal matrix Q and an upper triangular matrix R such that $A = QR$. We recall that our aim is to find c and d which minimize the expression 2.15. For this, we apply the QR decomposition to the matrix T. Due to the dimensions of T, we can have the following form of the QR decomposition of T.

$$T = (Q_1 : Q_2) \begin{pmatrix} R \\ 0 \end{pmatrix} \tag{3.5}$$

where $(Q_1 : Q_2)$ is orthogonal and R is upper triangular. Here, Q_1 is $n \times 3$, Q_2 is $n \times n - 3$ and R is 3×3.

Now this means $T = Q_1 R$. Which yields $T' = R'Q_1'$. So, $T'Q_2 = R'Q_1'Q_2 = 0$ as the columns of Q_1 and Q_2 are mutually orthogonal.

As T is $n \times 3$ with rank 3, so the null space of T' is $n-3$ dimensional. Thus, we see that the columns of Q_2 forms a basis for the null space of T'.

Since $T'c = 0$, c is in the column space of Q_2. So, $c = Q_2 b$ with b being some $n-3$ vector.

Due to the orthogonality of $(Q_1 : Q_2)$ we can write

$$\|x\|^2 = \|Q_1'x\|^2 + \|Q_2'x\|^2$$

for any n vector x.

So, now we have from 2.15

$$\begin{aligned}
&\frac{1}{n}\|z - Td - Kc\|^2 + \lambda c'Kc \\
&\qquad = \quad \frac{1}{n}\|z - Q_1Rd - KQ_2b\|^2 + \lambda b'Q_2'KQ_2b \\
&= \frac{1}{n}\|Q_1'(z - Q_1Rd - KQ_2b)\|^2 \;+\; \frac{1}{n}\|Q_2'(z - Q_1Rd - KQ_2b)\|^2 \\
&\qquad\qquad + \; \lambda b'Q_2'KQ_2b \\
&= \frac{1}{n}\|Q_1'z - Q_1'Q_1Rd - Q_1'KQ_2b\|^2 \;+\; \frac{1}{n}\|Q_2'z - Q_2'Q_1Rd - Q_2'KQ_2b\|^2 \\
&\qquad\qquad + \; \lambda b'Q_2'KQ_2b \\
&= \frac{1}{n}\|Q_1'z - Rd - Q_1'KQ_2b\|^2 + \frac{1}{n}\|Q_2'z - Q_2'KQ_2b\|^2 + \lambda b'Q_2'KQ_2b
\end{aligned}$$

From the above, we get that the minimizers d and b of the above expression have to satisfy

$$Rd = Q_1'(z - KQ_2b) \tag{3.6}$$

and

$$Q_2'z = (Q_2'KQ_2 + n\lambda I)b \tag{3.7}$$

So, our main problem becomes solving 3.7 for b. Once this has been done, we get c by using $c = Q_2b$. And we can use b in 3.6 to solve for d. As R is 3×3 upper triangular, it is very simple to numerically compute d.

3.3 Condition number of the matrices involved

For any invertible matrix A, the number

$$\kappa(A) = \|A\|\|A^{-1}\|$$

is known as the condition number of the matrix A. The definition holds for any valid norm. But we are concerned about the case of ℓ_2 norm. Moreover, if the matrix is symmetric, that is $A^T = A$, then $\kappa(A)$ is given by the ratio of the largest eigen value and the smallest eigen value in absolute value of the matrix A.

In terms of solving a linear system $Ax = b$, the condition number gives an idea about the solution x behavior with changes in data b. That is, if condition number is very high, a small noise in the data will cause a large error in the solution. But for small condition number, a small amount of noise in the data will induce a small error in the solution.

Let x^* be the solution of $Ax = b$ and x^ϵ be the solution when a noise ϵ

is added to the data b. Then

$$\begin{aligned} Ax^{\epsilon} &= b + \epsilon \\ &= Ax^{*} + \epsilon \end{aligned}$$

$$\begin{aligned}
\Leftrightarrow \quad A(x^{\epsilon} - x^{*}) &= \epsilon \\
\Leftrightarrow \quad x^{\epsilon} - x^{*} &= A^{-1}\epsilon \\
\Rightarrow \quad \|x^{\epsilon} - x^{*}\| &= \|A^{-1}\epsilon\| \\
\Rightarrow \quad \|x^{\epsilon} - x^{*}\| &\leq \|A^{-1}\|\|\epsilon\| \\
\Rightarrow \quad \tfrac{\|x^{\epsilon} - x^{*}\|}{\|x^{*}\|} &\leq \|A^{-1}\|\frac{\|\epsilon\|}{\|x^{*}\|} \\
&\leq \|A\|\|A^{-1}\|\frac{\|\epsilon\|}{\|A\|\|x^{*}\|} \\
&\leq \|A\|\|A^{-1}\|\frac{\|\epsilon\|}{\|b\|} \text{ using } \|b\| \leq \|A\|\|x^{*}\| \\
&\leq \kappa(A)\frac{\|\epsilon\|}{\|b\|}
\end{aligned}$$

In the context of the thin plate spline, Wahba (1979) suggested about 10^6 or 10^7 to be maximum acceptable condition number for computations. While Sibson and Stone (1991) suggest 10^5 to be the upper bound for IEEE single precision arithmetic and for IEEE double precision arithmetic, they say that values as high as 10^{12} should be acceptable. The condition numbers of the relevant matrices are discussed next. Only the case of uniform grid is considered here. The random grids produce different results each time and sometimes produce too large condition number due to some points being very close to each other leading to near singular matrices. From now on, we always consider the domain $[0, 1]^2$ where we get the data and approximate the functions. The results are for uniform grid.

3.3.1 The matrix M_0

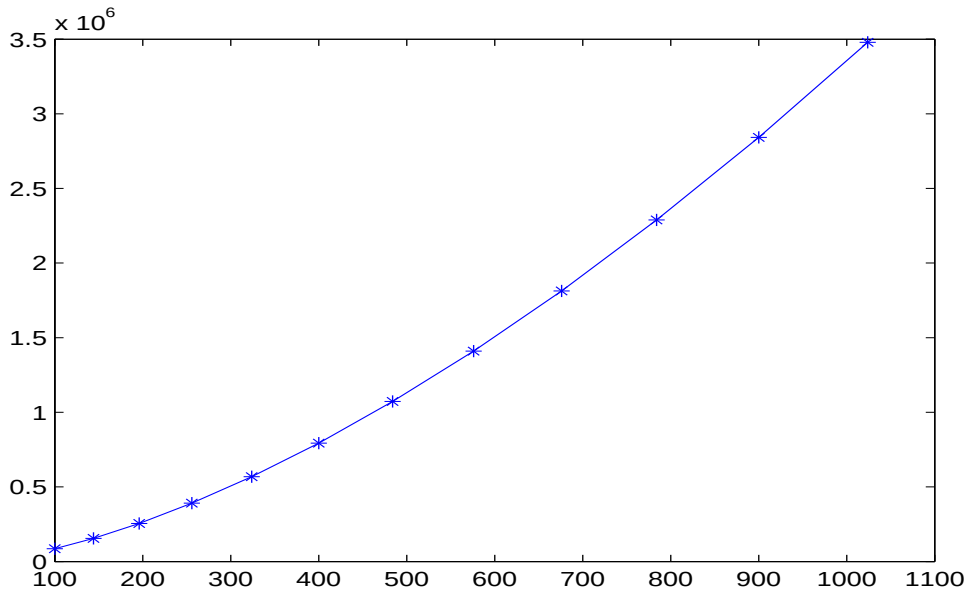

Figure 3.1: Condition number of M_0 with varying N.

The figure 3.1 shows the change in the condition number of M_0 with the increase in the number of data points. Here it is seen that the condition number increases with an increasing rate with increase in the number of data points. For $N = 100$, the condition number is 0.0853×10^6 and it increases to 3.4787×10^6 for $N = 1024$. This behavior is expected because with the increase in the number of data, the points get closer to each other and thus they approach a singular matrix. But the values we get here are quite acceptable for our computations.

3.3.2 The matrix M_α

The effect of α on the condition number of M_α is that it lowers the condition number considerably given that α is not too small. For example, as in figure 3.2, the condition number of $M_{0.001}$ is 1.0990×10^4 for $N = 100$ which is

about 8 times smaller than that of M_0 for the same N. And the condition number of $M_{0.001}$ for $N = 1024$ is 4.1629×10^4 which is about 84 times smaller than that of M_0 for the same N. For another example, for $M_{0.01}$ (see figure 3.3), the condition number grows from 1.2426×10^3 for $N = 100$ to 4.2078×10^3 for $N = 1024$. Thus we see that the smoothing parameter not only gives a smooth approximation of the function, but also makes computations to have less error.

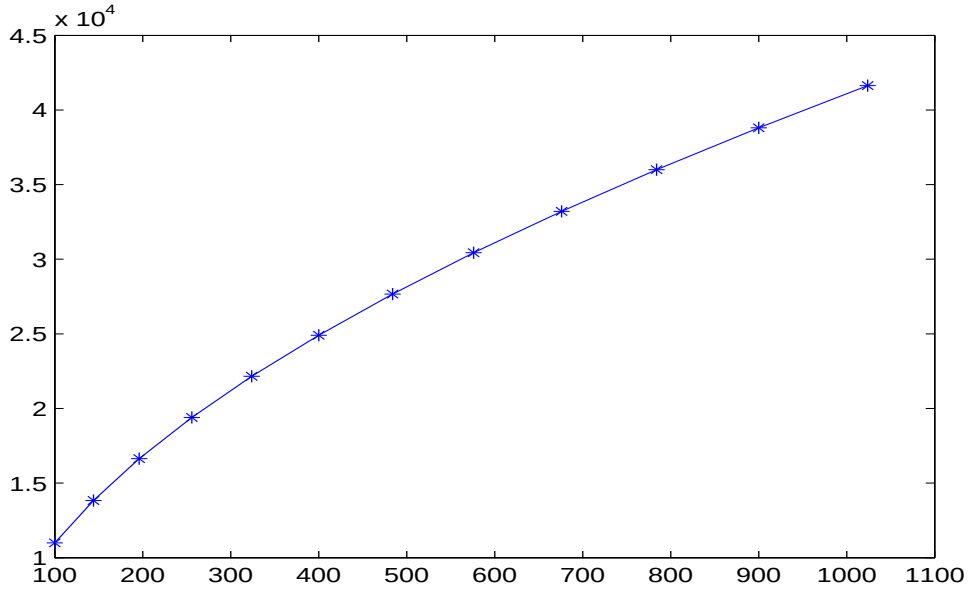

Figure 3.2: Condition number of $M_{0.001}$ with varying N.

3.3.3 The matrix $Q_2'KQ_2$

Now we look at the condition number of the matrix $Q_2'KQ_2$ obtained in the QR decomposition method. As is clear by the figure 3.4, there is an improvement in the condition number compared to that of M_0. The increasing trend of the condition numbers is maintained here as expected. For $N = 1024$, the condition number is 1.1694×10^5 which is a good im-

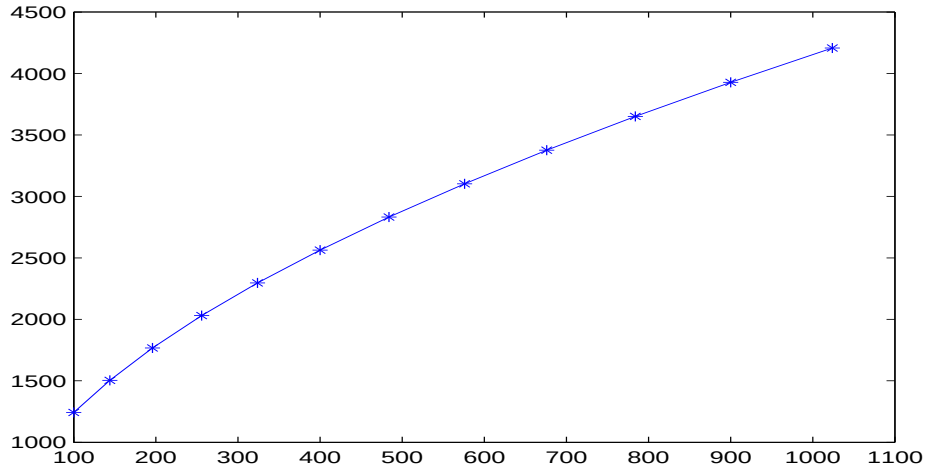

Figure 3.3: Condition number of $M_{0.01}$ with varying N.

provement over that of M_0. Thus we already see an advantage of using the QR decomposition method.

3.3.4 The matrix $Q_2'KQ_2 + \alpha I$

Taking a smoothing parameter affects the condition number by lowering it. Figure 3.5 shows the marked improvement in the condition number even with a small smoothness parameter as 0.001. It brings down the condition number to 1.4004×10^3 for $N = 1024$. And for $\alpha = 0.01$ the condition number becomes 1.4247×10^2 for the same N. See figure 3.6. Here again, the advantage of using the smoothness parameter is clear.

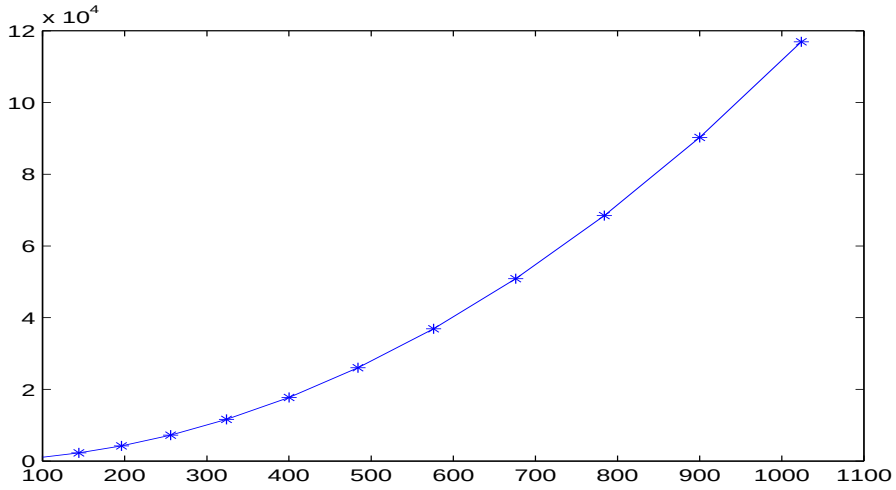

Figure 3.4: Condition number of $Q_2'KQ_2$ with varying N.

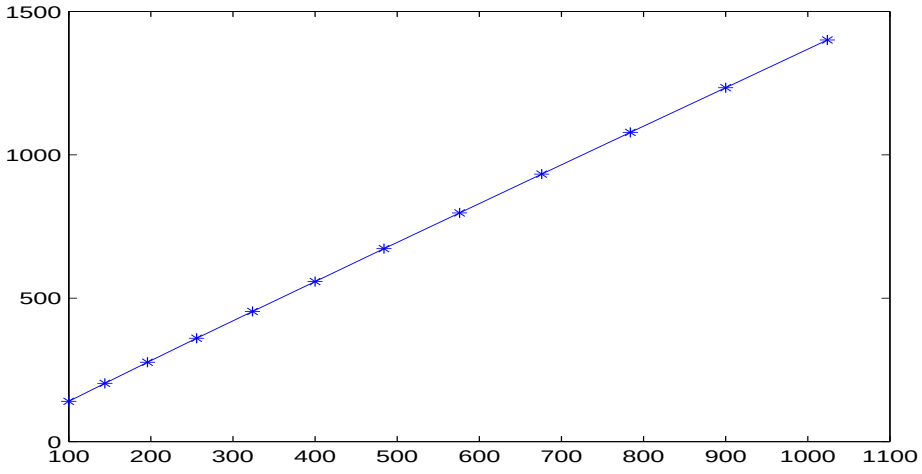

Figure 3.5: Condition number of $Q_2'KQ_2 + 0.001 \cdot I$ with varying N.

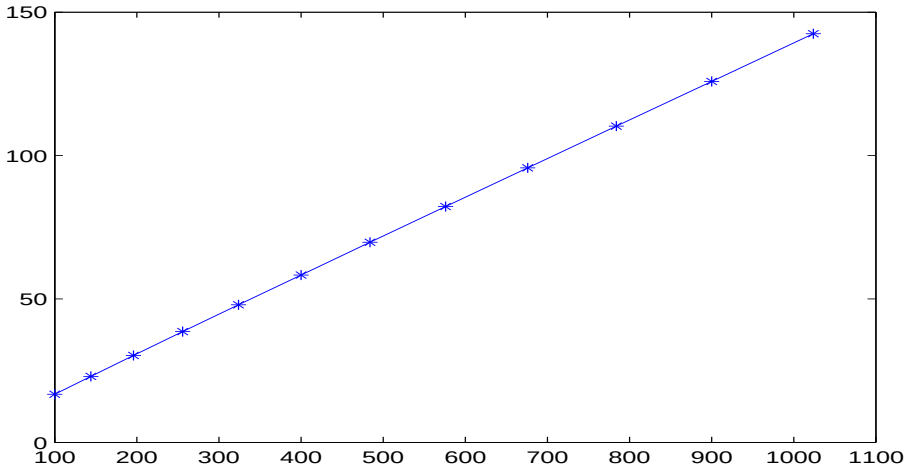

Figure 3.6: Condition number of $Q_2' K Q_2 + 0.01 \cdot I$ with varying N.

Chapter 4

Solution methods

In this chapter, we take a look at the methods employed to solve the problems 3.4 and 3.7. First we review the generalized minimum residual method which is used for solving both the problems. Then we go on to get an idea of the conjugate gradient method which we use only for the second problem. And after that we discuss the Uzawa's iterative method used for solving the first problem due to its similarity to saddle point problems.

4.1 Generalized minimum residual method

Supposing a model problem of solving for $x \in \mathbb{R}^n$ with the linear system

$$Ax = b \tag{4.1}$$

A projection method for solving this model problem works by finding an approximate solution x_m in an affine space $x_0 + \mathcal{K}_m$ such that this approximate solution satisfies the following condition

$$b - Ax_m \perp \mathcal{L}_m$$

Here x_0 serves as the initial guess for the solution. $\mathcal{K}_m$ and $\mathcal{L}_m$ are m dimensional subspaces of $\mathbb{R}^n$.

A Krylov subspace method is a projection method which takes the subspace $\mathcal{K}_m$ as the Krylov subspace form

$$\mathcal{K}_m = span\{r_0, Ar_0, A^2r_0, \ldots, A^{m-1}r_0\}$$

where $r_0 = b - Ax_0$. In other words, $\mathcal{K}_m$ is the m dimensional subspace of $\mathbb{R}^n$ which contain all the vectors of the form

$$v = p(A)r_0,$$

where p is any polynomial of degree less than m. The dimension of these Krylov subspace increases by 1 with each step of approximation process.

The Generalized minimum residual method is a projection method where we take

$$\mathcal{L}_m = A\mathcal{K}_m$$

while taking

$$v_1 = r_0/\|r_0\|.$$

Here v_1 is the first column of the matrix V_m which has columns making up a basis of $\mathcal{K}_m$. This method minimizes the residual norm over all the elements of the affine subspace $x_0 + \mathcal{K}_m$.

We discuss here an algorithm which uses Arnoldi's algorithm.
For any vector $x \in x_0 + \mathcal{K}_m$, the following holds

$$x = x_0 + V_m y \tag{4.2}$$

where y is an m vector. Now we let

$$\begin{aligned} \mathcal{J}(y) &= \|b - Ax\| \\ &= \|b - A(x_0 + V_m y)\| \end{aligned} \tag{4.3}$$

Now we use a relation

$$AV_m = V_{m+1} H_m$$

where H_m is an $(m+1) \times m$ Hessenberg matrix constructed by using Arnoldi's algorithm.

So, this gives us the following

$$\begin{aligned} b - Ax &= b - A(x_0 + V_m y) \\ &= r_0 - AV_m y \\ &= \alpha v_1 - V_{m+1} H_m y \\ &= V_{m+1}(\alpha e_1 - H_m y) \end{aligned}$$

where $\alpha = r_0/v_1$ and e_1 is the usual standard basis vector.
Then using the orthonormality of the columns of V_{m+1}, we get

$$\begin{aligned} \mathcal{J}(y) &= \|b - A(x_0 + V_m y)\| \\ &= (\alpha e_1 - H_m y) \end{aligned} \tag{4.4}$$

The GMRES approximation is a unique minimizer of 4.3 in $x_0 + \mathcal{K}_m$. This can be computed by first minimizing 4.4 and then using 4.2. So we have the minimizer x_m given by

$$x_m = x_0 + V_m y_m$$

where

$$y_m = \arg\min_y \mathcal{J}(y)$$

y_m is inexpensive to compute as it involves solving an $(m+1) \times m$ least square problem.

The following algorithm for GMRES is given by Saad(2000)

ALGORITHM 1: GMRES

1. For an initial guess x_0, $r_0 = b - Ax_0$, $\alpha = \|r_0\|$ and $v_1 = r_0/\alpha$
2. Define $H_m = \{h_{ij}\}_{1 \le i \le m+1, 1 \le j \le m}$. Set $H_m = 0$
3. For $j = 1, 2, \ldots, m$ Do:
4. Compute $w_j := Av_j$
5. For $i = 1, ..., j$ Do:
6. $h_{ij} := (w_j, v_i)$
7. $w_j := w_j - h_{ij} v_i$
8. EndDo
9. $h_{j+1,j} = \|w_j\|_2$. If $h_{j+1,j} = 0$ set $m := j$ and go to 12
10. $v_{j+1} = w_j / h_{j+1,j}$
11. EndDo
12. Compute y_m the minimizer of $\|\alpha e_1 - \bar{H}_m y\|_2$ and $x_m = x_0 + V_m y_m$.

Next we look at some convergence results for GMRES. Matrix A is called positive definite if for all real vectors $x \neq 0$ we have that $x'Ax > 0$ or in other words we have the property that $(Ax, x) > 0$. We begin by giving a global convergence result.

Theorem 4.1.1. *If A is a positive definite matrix, then GMRES(m) converges for any $m \geq 1$.*

Lemma 4.1.1. *Let x_m be the approximate solution obtained from the m-th step of the GMRES algorithm, and let $r_m = b - Ax_m$. Then, x_m is of the form*

$$x_m = x_0 + p_m(A)r_0$$

where p_m is a polynomial of degree $m-1$ such that

$$\|r_m\|_2 = \|(I - Ap_m(A)r_0)\|_2 = \min_{p \in P_{m-1}} \|(I - Ap(A)r_0)\|_2.$$

Proposition 4.1.1. *Assume that A is a diagonalizable matrix and let $A = X\Lambda X^{-1}$ where $\Lambda = diag\{\lambda_1, \lambda_2, \ldots \lambda_n\}$ is the diagonal matrix of eigenvalues. Define,*

$$\epsilon^{(m)} = \min_{p \in P_m, p(0)=1} \max_{i=1,\ldots,n} |p(\lambda_i)|.$$

Then, the residual norm achieved by the m-th step of GMRES satisfies the inequality

$$\|r_m\|_2 \leq \kappa_2(X)\epsilon^{(m)}\|r_0\|_2.$$

where $\kappa_2(X) \equiv \|X\|_2\|X^{-1}\|_2$.

Corollary 4.1.1. *Let A be a diagonalizable matrix, i.e., let $A = X\Lambda X^{-1}$ where $\Lambda = diag\{\lambda_1, \lambda_2, \ldots, \lambda_n\}$ is the diagonal matrix of eigenvalues. Assume that all the eigenvalues of A are located in the ellipse $E(c, d, a)$ which excludes the origin. Then, the residual norm achieved at the m-th step of GMRES satisfies the inequality,*

$$\|r_m\|_2 \leq \kappa_2(X)\frac{C_m(\frac{a}{d})}{|C_m(\frac{c}{d})|}\|r_0\|_2$$

4.2 Conjugate gradient method

The conjugate gradient method is one of the best known methods for solving a symmetric positive definite linear system. It is also a kind of Krylov subspace method which is a type of projection method.

Using the CG method for the problem 4.1, the approximate solution y_{j+1} after $j+1$ steps is given by

$$y_{j+1} = y_j + \alpha_j p_j$$

where α_j is a scalar and p_j is the jth search direction for the next better approximate solution. This yields the following relation between successive residuals

$$r_{j+1} = r_j - \alpha_j A p_j. \tag{4.5}$$

Now, for the CG method, the residual vectors are orthogonal. For this property to be satisfied, we should have $(r_j - \alpha_j A p_j, r_j) = 0$. And this gives us a way to compute α_j

$$\alpha_j = \frac{(r_j, r_j)}{(Ap_j, r_j)} \tag{4.6}$$

Next we use a fact for the search directions that p_{j+1} is a linear combination of r_{j+1} and p_j. So, with some scaling we can write

$$p_{j+1} = r_{j+1} + \beta_j p_j \tag{4.7}$$

with some scalar β_j. Using all these we get

$$(Ap_j, r_j) = (Ap_j, p_j - \beta_{j-1} p_{j-1}) = (Ap_j, p_j)$$

due to another property of the search directions namely orthogonality with respect to the A-norm, i.e.

$$(Ap_j, p_{j-1}) = 0. \tag{4.8}$$

Then 4.6 becomes

$$\alpha_j = \frac{(r_j, r_j)}{(Ap_j, p_j)}.$$

Using 4.7 and 4.8

$$(r_{j+1} + \beta_j p_j, Ap_j) = 0$$

$$\beta_j = -\frac{(r_{j+1}, Ap_j)}{(p_j, Ap_j)}.$$

4.5 gives

$$Ap_j = -\frac{1}{\alpha_j}(r_{j+1} - r_j).$$

Hence,

$$\begin{aligned} \beta_j &= \frac{1}{\alpha_j}\frac{(r_{j+1}, (r_{j+1} - r_j))}{(p_j, Ap_j)} \\ &= \frac{(r_{j+1}, r_{j+1})}{(r_j, r_j)} \end{aligned}$$

ALGORITHM 2: Conjugate Gradient

1. Compute $r_0 := b - Ay_0, p_0 := r_0$.
2. For j = 0, 1, . . . , until convergence DO:
3. $\alpha_j := \dfrac{(r_j, r_j)}{(Ap_j, p_j)}$
4. $y_{j+1} := y_j + \alpha_j p_j$

5. $r_{j+1} := r_j - \alpha_j A p_j$

6. $\beta_j := \dfrac{(r_{j+1}, r_{j+1})}{(r_j, r_j)}$

7. $p_{j+1} := r_{j+1} + \beta_j p_j$

8. EndDo

We now present some convergence results for CG method.

Lemma 4.2.1. *Let y_n be the approximation solution obtained by the n-th step of the CG algorithm, and let $d_n = y^* - y_m$ where y^* is the exact solution. Then, y_n is of the form*

$$y_m = y_0 + p_n(A) r_0$$

where p_m is a polynomial of degree $n-1$ such that

$$\|(I - A p_m(A)) d_0\|_A = \min_{p \in P_{m-1}} \|(I - A q(A)) d_0)\|_A$$

Theorem 4.2.1. *Let y_n be the approximate solution obtained at the n-th step of the Conjugate Gradient algorithm, and y^* the exact solution and define*

$$\eta = \frac{\lambda_{min}}{\lambda_{max} - \lambda_{min}}.$$

Then,

$$\|y^* - y_n\|_A \leq \frac{\|y^* - y_0\|_A}{C_n(1 + 2\eta)}$$

in which C_n is the Chebyshev polynomial of degree n of the first kind given as:

$$C_n(z) = \frac{1}{2}\left[(z - \sqrt{z^2 - 1})^n + (z + \sqrt{z^2 - 1})^n\right]$$

4.3 Uzawa's iterative method

Consider the following constrained quadratic optimization problem

$$\text{minimize } f(x) \equiv \frac{1}{2}(Ax, x) - (x, b)$$

subject to

$$B'x = c$$

with the assumption than the number of columns of B is not more than its number of rows. Considering the necessary optimality conditions, the following linear system is obtained

$$\begin{bmatrix} A & B \\ B' & 0 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} b \\ c \end{bmatrix} \tag{4.9}$$

The Lagrangian for the optimization problem is

$$L(x, y) = \frac{1}{2}(Ax, x) - (x, b) + (y, (B'x - c)).$$

The saddle point of this Lagrangian is the solution of 4.9. Uzawa's method is a well known iterative method for solving linear systems of the form 4.9.

ALGORITHM 3: Uzawa's Method

1. Choose x_0, y_0
2. For k = 0, 1, . . . , until convergence DO:
3. $x_{k+1} = A^{-1}(b - By_k)$
4. $y_{k+1} := y_k + \omega(B^T x_{k+1} - c)$

5. EndDo

The following theorem can be useful to choose a value for the parameter ω used in the algorithm

Theorem 4.3.1. *Let A be a symmetric positive definite matrix and B a matrix of full rank. Then $S = B'A^{-1}B$ is also symmetric positive definite and Uzawa's algorithm converges if and only if*

$$0 < \omega < \frac{2}{\lambda_{max}(S)}.$$

In addition, the optimal convergence parameter ω is given by

$$\omega_{opt} = \frac{2}{\lambda_{min}(S) + \lambda_{max}(S)}.$$

For details on GMRES, CG and Uzawa's methods refer to [10].

Chapter 5

Numerical results

In this chapter, we will see some implementations of thin plate splines using the methods described in the previous chapter. The domain of our interest is $[0, 1] \times [0, 1]$. The data is taken on 1024 random points in this domain using the sample function $x^2 + y^2$. A small amount of random noise is also added to the data using the function $0.005 \times (Normal)(0.1)$. Irrespective of the number of data points, the final plot of the approximation function obtained is given over a uniform grid of 121 points in the specified domain. Before starting with the sections, the exact plot of the sample function is presented below for comparisons with the approximations.

5.1 GMRES

GMRES method has been used for both the problems 3.4 and 2.29. From now on, we refer to 3.4 by main problem.

5.1.1 GMRES in main problem

Figures 5.2, 5.3 and 5.4 show the approximations produced by solving the main problem by the GMRES method for $\alpha = 0, 0.001, 0.01$ respectively.

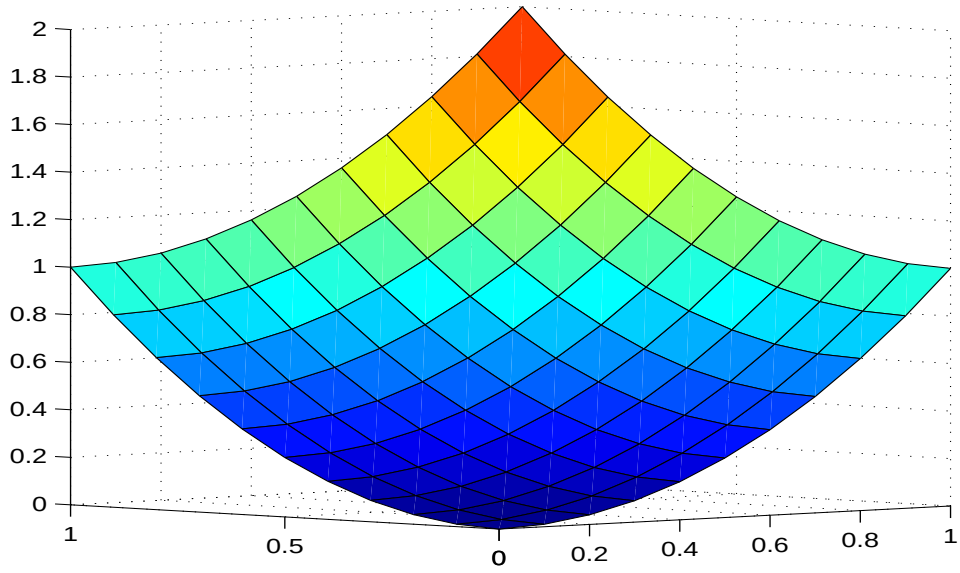

Figure 5.1: Sample function

For the figure 5.2, the condition number of the corresponding matrix was 1.9159×10^8, which is quite high compared to the condition number associated with the uniform grid. GMRES converged at iteration 260. While the condition number for the matrix corresponding to the figure 5.3 was 4.2121×10^4 which is comparable to the uniform grid case and GMRES converged at iteration 41. And for the figure 5.4, the corresponding matrix has condition number 4.2197×10^3 which is again comparable and GMRES converged at iteration 25.

5.1.2 GMRES in QR method

Figures 5.5, 5.6 and 5.7 show the approximations produced by solving the QR method problem by the GMRES method for $\alpha = 0, 0.001, 0.01$ respectively. For the figure 5.5, the condition number of the corresponding matrix was 9.8710×10^6 which is higher than the uniform grid case. GMRES converged at iteration 317. While the condition number for the matrix cor-

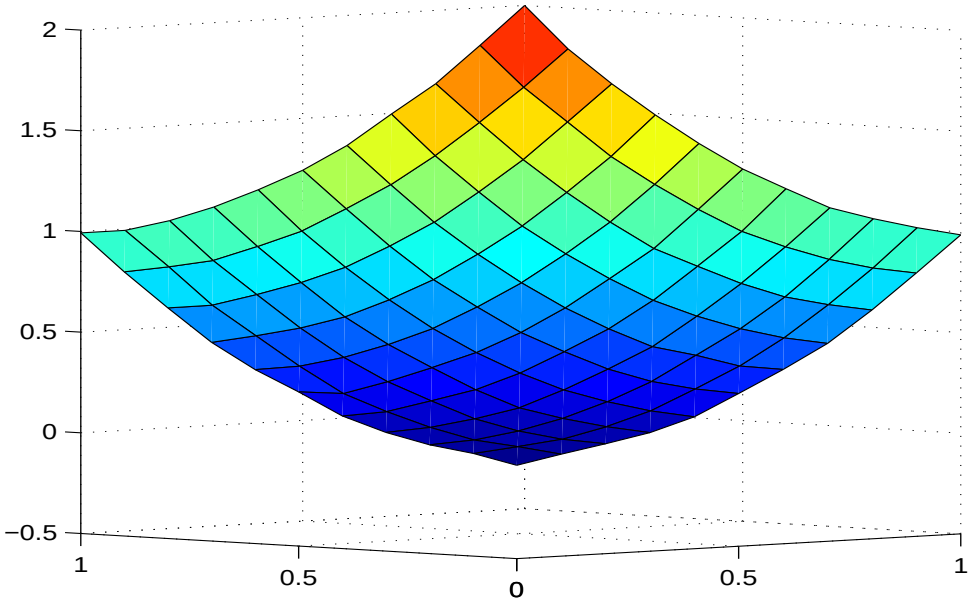

Figure 5.2: Approximation of $x^2 + y^2$ using GMRES with $\alpha = 0$

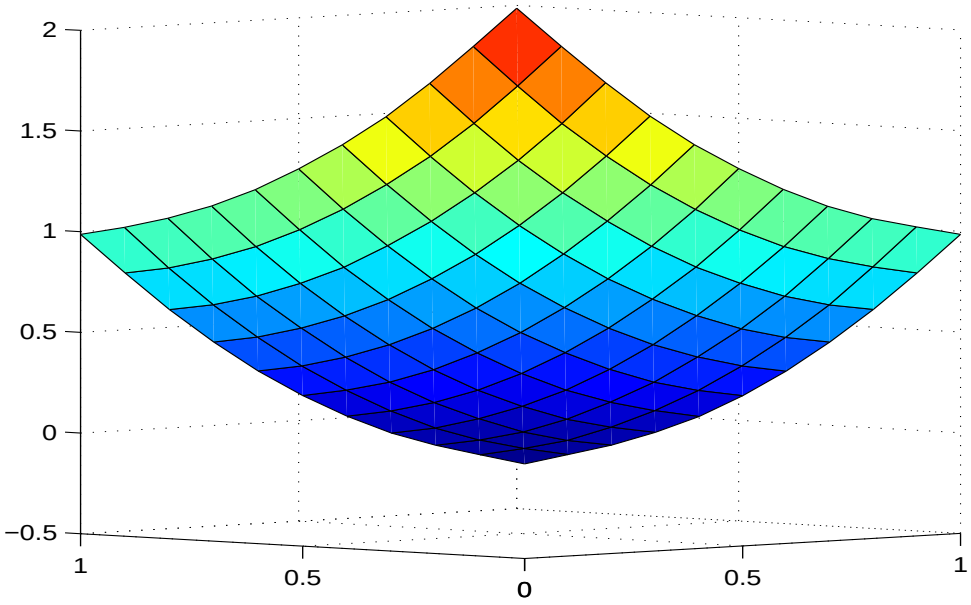

Figure 5.3: Approximation of $x^2 + y^2$ using GMRES with $\alpha = 0.001$

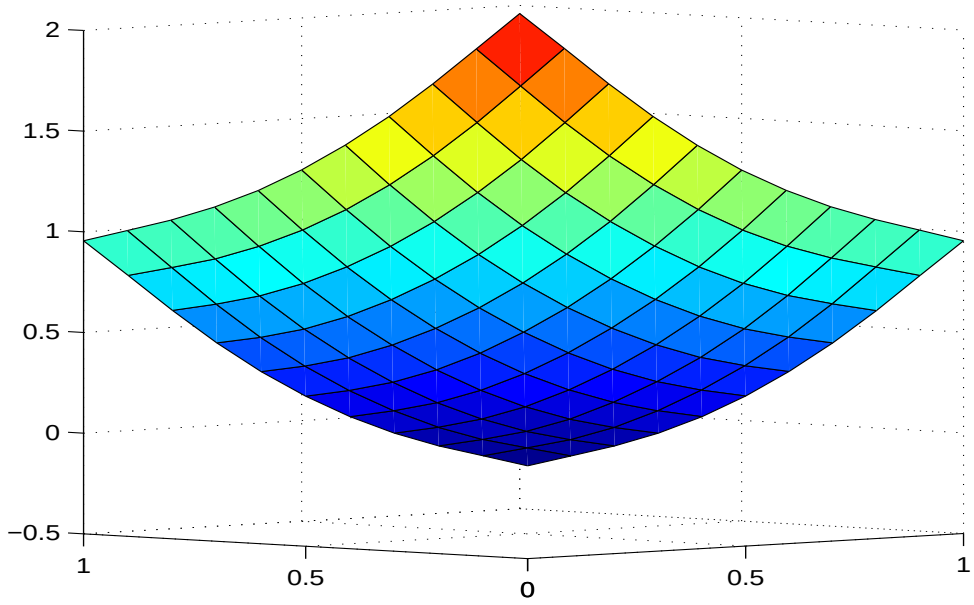

Figure 5.4: Approximation of $x^2 + y^2$ using GMRES with $\alpha = 0.01$

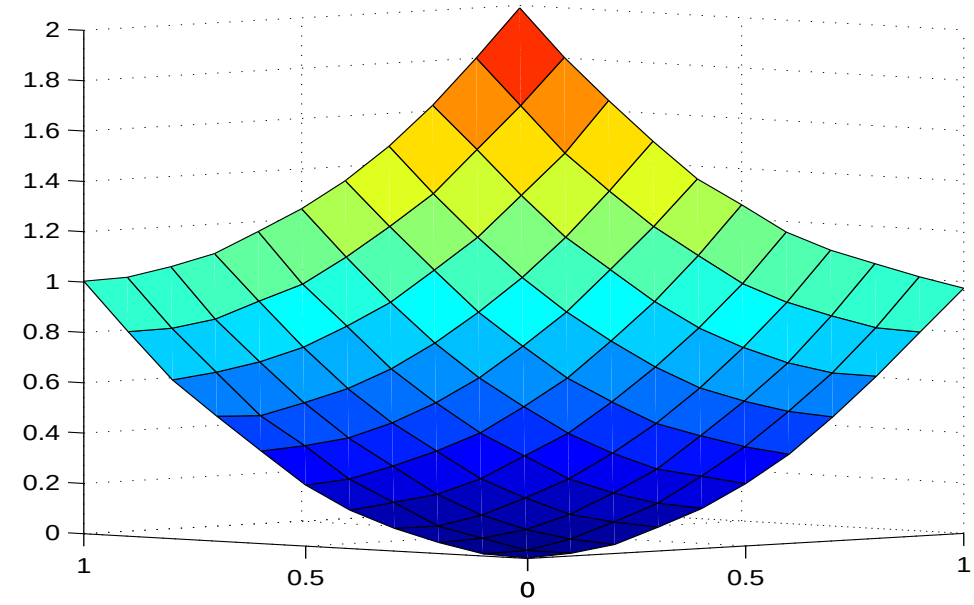

Figure 5.5: Approximation of $x^2 + y^2$ using GMRES in QR method with $\alpha = 0$

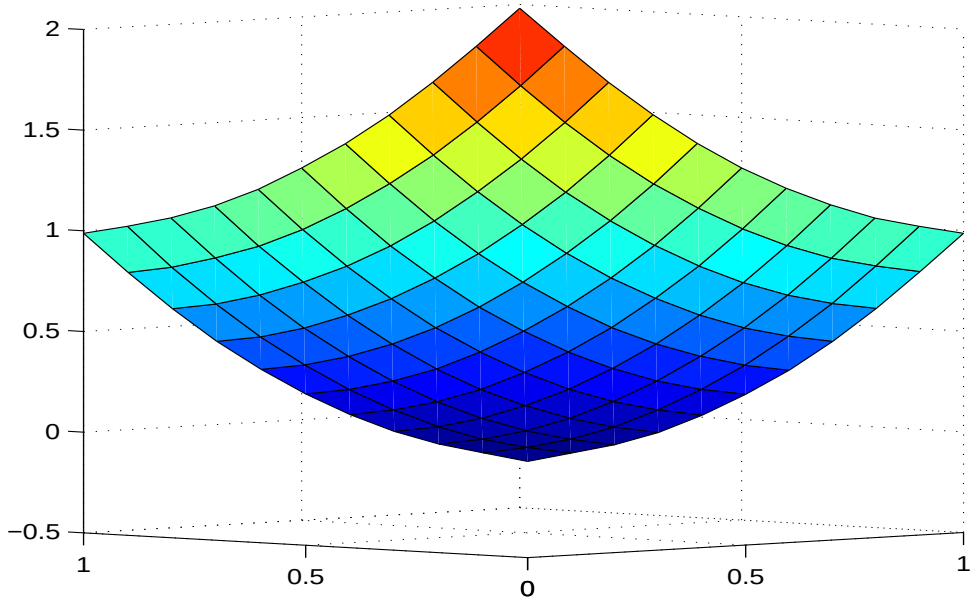

Figure 5.6: Approximation of $x^2 + y^2$ using GMRES in QR method with $\alpha = 0.001$

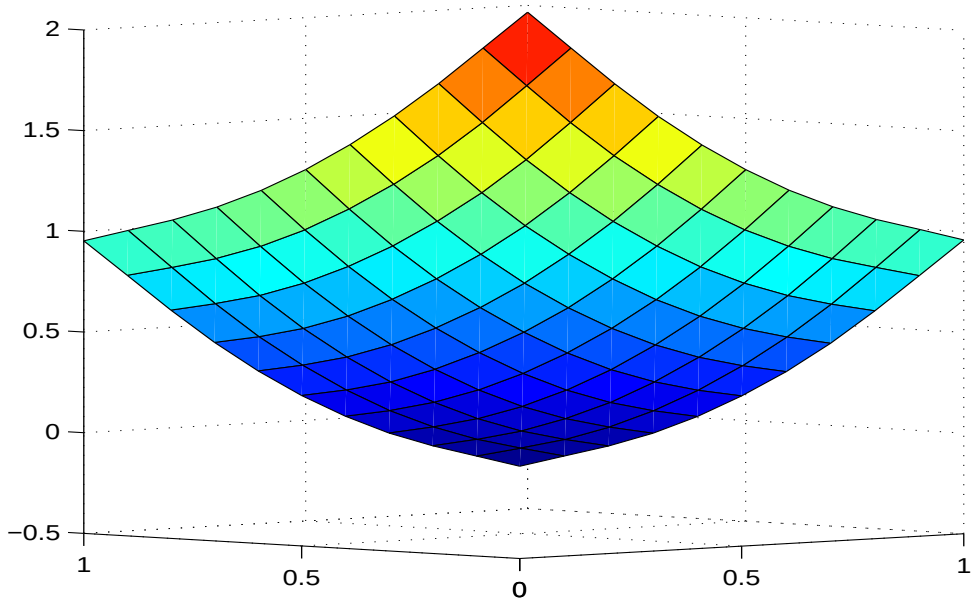

Figure 5.7: Approximation of $x^2 + y^2$ using GMRES in QR method with $\alpha = 0.01$

responding to the figure 5.6 was 1.3392×10^3 which is comparable to the uniform grid case. GMRES converged at iteration 39. And for the figure 5.7, the corresponding matrix has condition number 138.0042 which is very close to the uniform grid case. GMRES converged at iteration 22.

5.2 CG

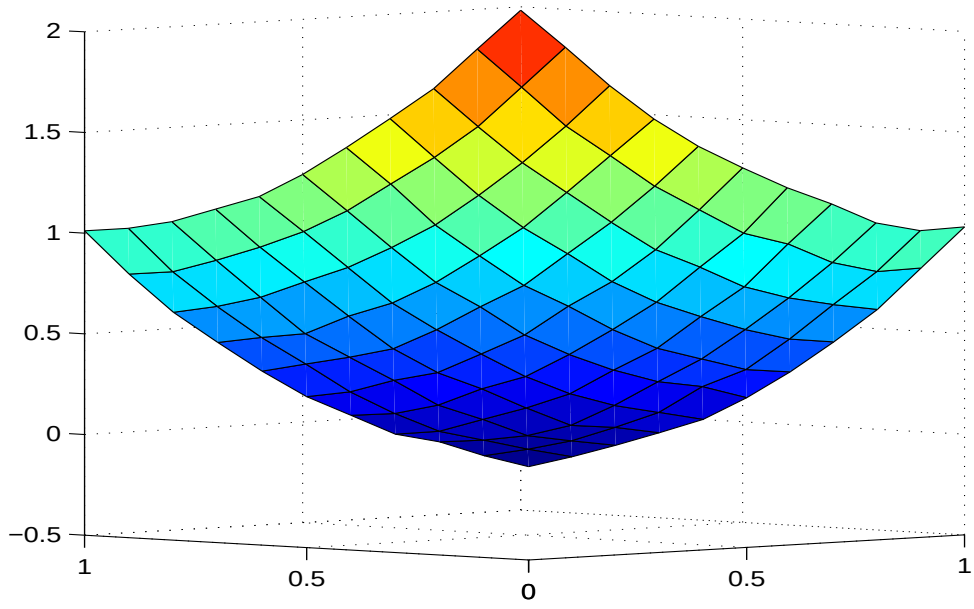

Figure 5.8: Approximation of $x^2 + y^2$ using CG in QR method with $\alpha = 0$

Figures 5.8, 5.9 and 5.10 show the approximations produced by solving the QR method problem by the CG method for $\alpha = 0, 0.001, 0.01$ respectively. For the figure 5.8, the condition number of the corresponding matrix was 1.1869×10^7, which is quite high compared to the condition number associated with the uniform grid. CG does not converge even at iteration 1000. While the condition number for the matrix corresponding to the figure 5.9 was 1.3162×10^3 which is comparable to the uniform grid case

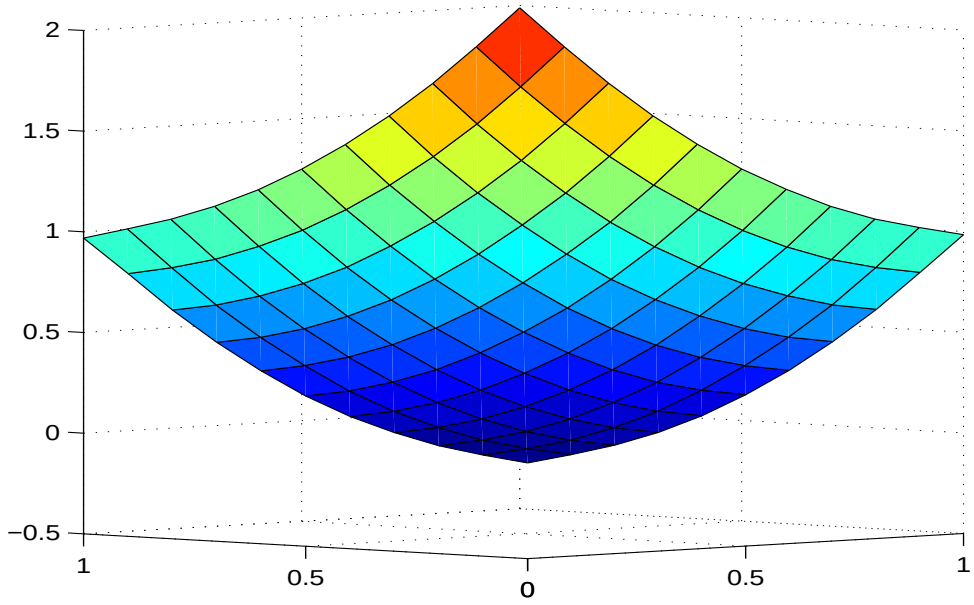

Figure 5.9: Approximation of $x^2 + y^2$ using CG in QR method with $\alpha = 0.001$

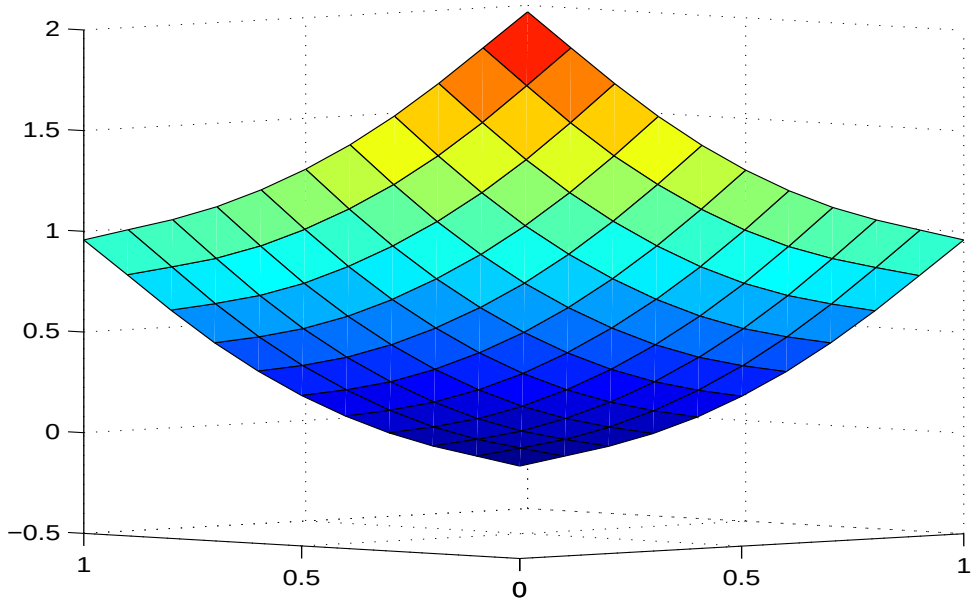

Figure 5.10: Approximation of $x^2 + y^2$ using CG in QR method with $\alpha = 0.01$

and CG converged at iteration 60. And for the figure 5.10, the corresponding matrix has condition number 133.3477 which is again comparable and GMRES converged at iteration 25.

5.3 Uzaawa's iterative method

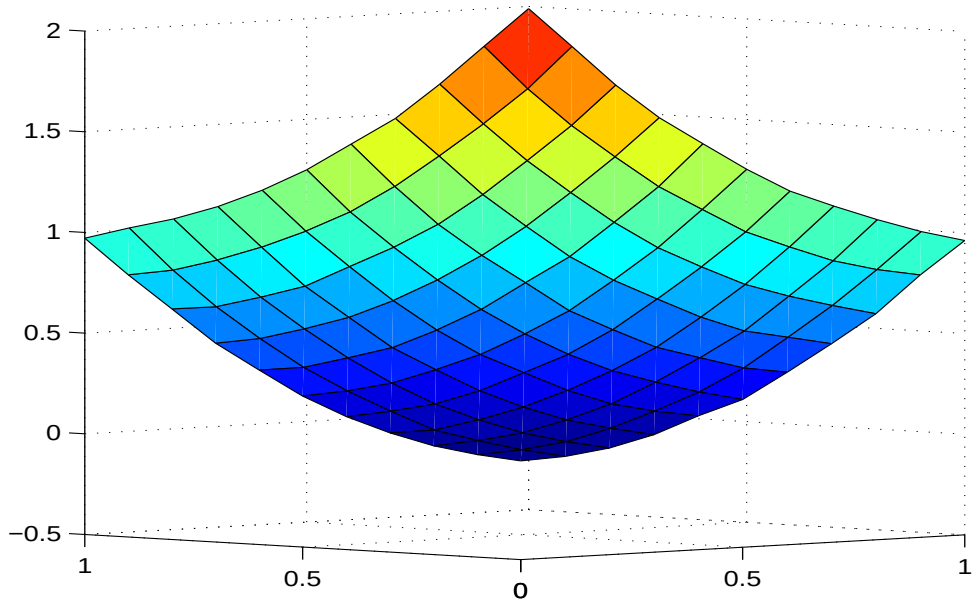

Figure 5.11: Approximation of $x^2 + y^2$ using Uzawa's iterative method with $\alpha = 0$

Figures 5.11, 5.12 and 5.13 show the approximations produced by solving the main problem by Uzawa's iterative method for $\alpha = 0, 0.001, 0.01$ respectively. For the figure 5.11, the condition number of the corresponding matrix was 9.4839×10^8, which is quite high compared to the condition number associated with the uniform grid. The method converged at iteration 119. While the condition number for the matrix corresponding to the figure 5.12 was 4.1909×10^4 which is comparable to the uniform grid case and the method converged at iteration 114. And for the figure 5.13,

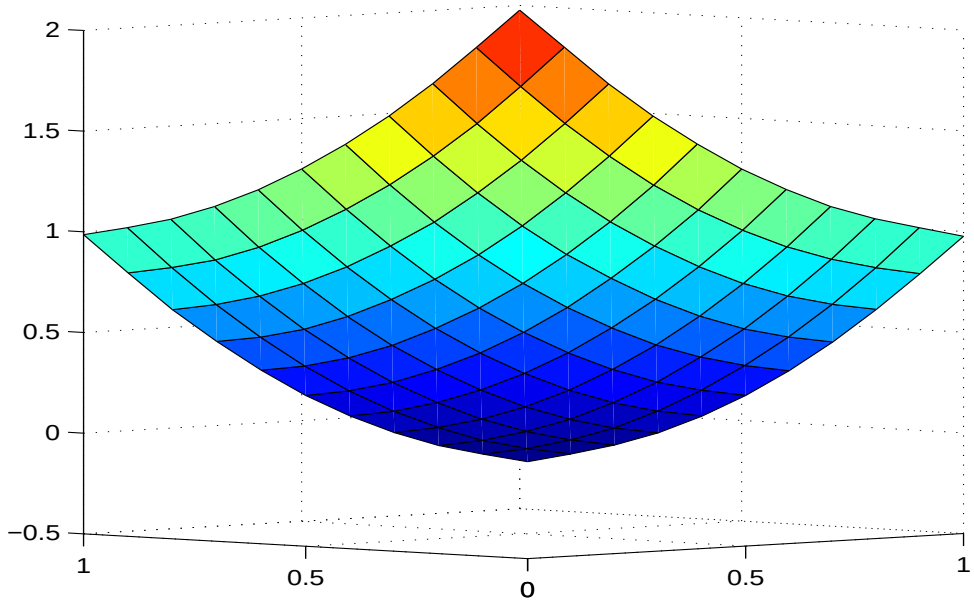

Figure 5.12: Approximation of $x^2 + y^2$ using Uzawa's iterative method with $\alpha = 0.001$

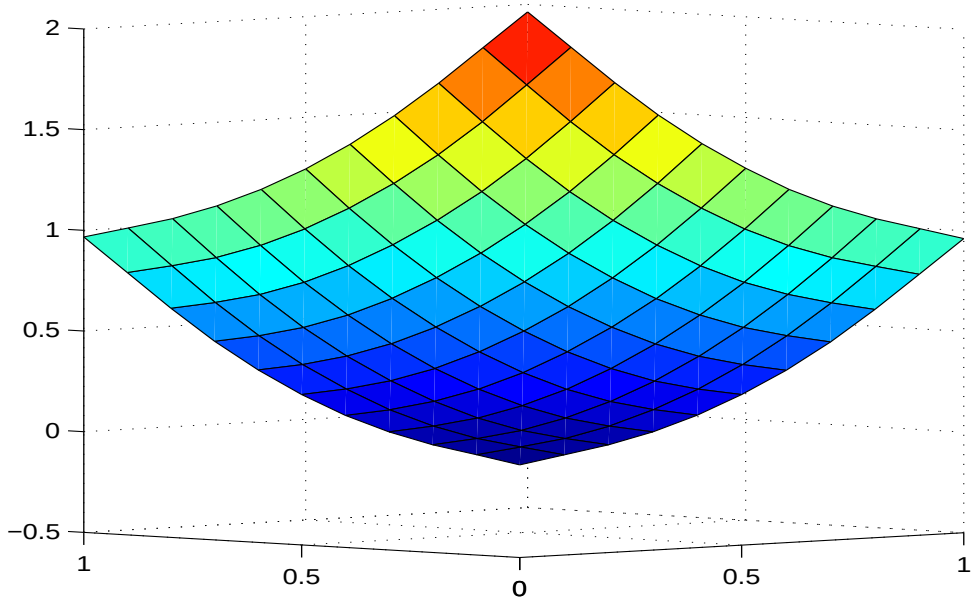

Figure 5.13: Approximation of $x^2 + y^2$ using Uzawa's iterative method with $\alpha = 0.01$

the corresponding matrix has condition number 4.2066×10^3 which is again comparable and the method converged at iteration 108.

While all the methods tried give very satisfactory pictures, the effect of the smoothness parameter can be seen here. The curves are flattened more as the smoothness parameter increases and the corners of the surface can be seen to be falling short of the exact value of the sample plot. And we see that even for high condition number, Uzawa's iterative method outperforms the others in the case of $\alpha = 0$ with respect to the number of iterations. But it does not improve with increase in the value of the smoothness parameter. In case of non zero smoothness parameter, QR method with GMRES converges the fastest. The condition number is also lowered in the QR method.

Chapter 6

TPS and adaptive optics

In this chapter, we will explore possibilities of using Thin Plate Splines in the context of adaptive optics. We start with the section giving a short introduction to adaptive optics. Then the mathematical formulation in the context of adaptive optics is introduced. We proceed with the descriptions of two methods to use Thin Plate Splines in the adaptive optics problem.

6.1 Adaptive optics

[11]

The term adaptive optics is essentially used for a technology that many optical systems use to counter the distortions in the wavefronts by adapting themselves. These distortions are introduced in the wavefronts by the medium between the source and the receiver. For example, in the case of ground based astronomy, the light from distant astronomical objects undergo distortion while crossing the atmosphere of earth due to variations in atmospheric conditions at different levels. Due to this, the final image obtained with a telescope is not of as high quality as we would like. Adaptive optics is used in improvement of the image quality by physical adjustments.

The problems caused by the atmosphere on astronomical images has been noticed for a long time now. The idea of removing the astronomical disturbances from the images was formed as early as 1908. But it was Horace W.Babcock who in 1953, came up with the idea of adaptive optics. But at that time, due to the limitations in technology, it did not have popular usage. But with the advancements in the fields of computers and optical systems, it has gained popularity.

Adaptive optics is now widely used in the modern large astronomical telescopes. It has been successfully used in imaging Galilean moons of Jupiter, precisely measuring the orbits of stars located near the center of the Milky Way galaxy to measure the mass of the black hole at the center and in imaging of extra solar planets of nearby stars. It is being considered very important for the large telescopes being built.

The conventional adaptive optics setup has the following components (See figure 6.1)

1. corrective element to make the wavefronts free of distortions.
2. wavefront sensor to measure the distortion
3. light source to drive the wavefront sensor.
4. wavefront reconstructor that directs the corrective elements using the sensor measurements.

An example of a corrective element is a deformable mirror. It is a thin, flexible facesheet coated with highly reflective material. The controlled transverse displacement of the sheet is caused by actuators attached to

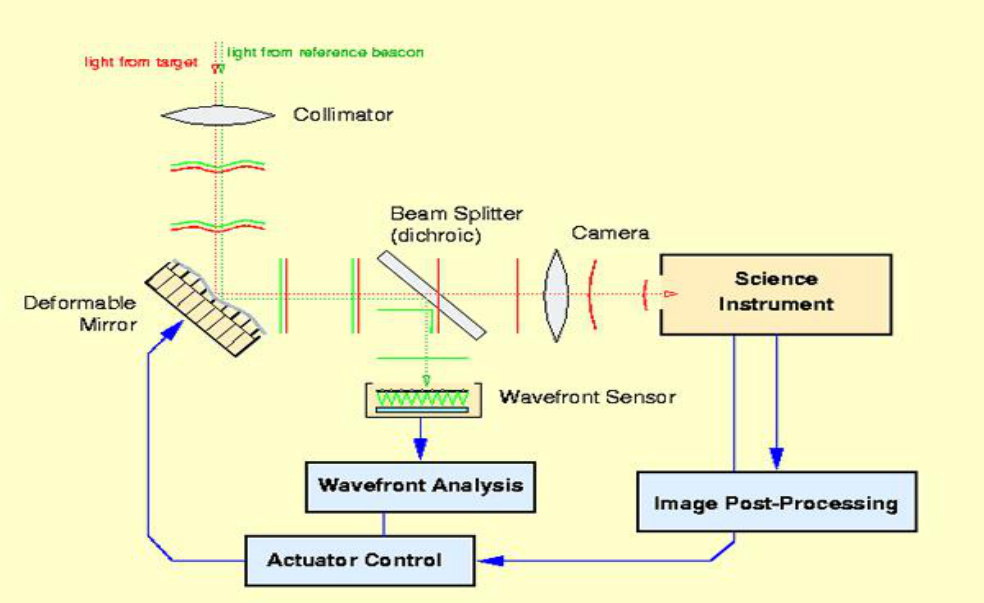

Figure 6.1: Schematic diagram of a simple adaptive optics system.[11].

the back. Some common deformable mirror technologies for astronomical adaptive optics are

1. piezo-stack deformable mirrors.
2. adaptive secondary mirrors.
3. Micro Electro Mechanical System (MEMS).

6.2 Mathematical model

[11]

Let the wavefront aberration be given by the function f. This is the unknown that we want to estimate using the readings from a wavefront sensor. A very commonly used wavefront sensor is the Shack-Hartmann wavefront sensor, see figure 6.2. It consists of an array of lenslets distributed on a rectangular grid covering the field of view of the sensor. From each lenslet light is focussed on a photo detector behind the lenslet. For an

incoming signal with locally planar wavefront, the lenslet focusses it to a location with x and y components on the photo detector proportional to the x and y components of the slope of the wavefront. Thus, the x-component read out from the sensor is

$$x_i = c \int \int_{\Omega_i} \frac{\partial f}{\partial x} dxdy + \eta_i$$

where c is a known constant, Ω_i is the field of view of the ith lenslet and η_i is sensor noise. η_i is generally taken as a random variable with zero mean value. Similarly the y component read out from the sensor is

$$y_i = c \int \int_{\Omega_i} \frac{\partial f}{\partial y} dxdy + \eta_i.$$

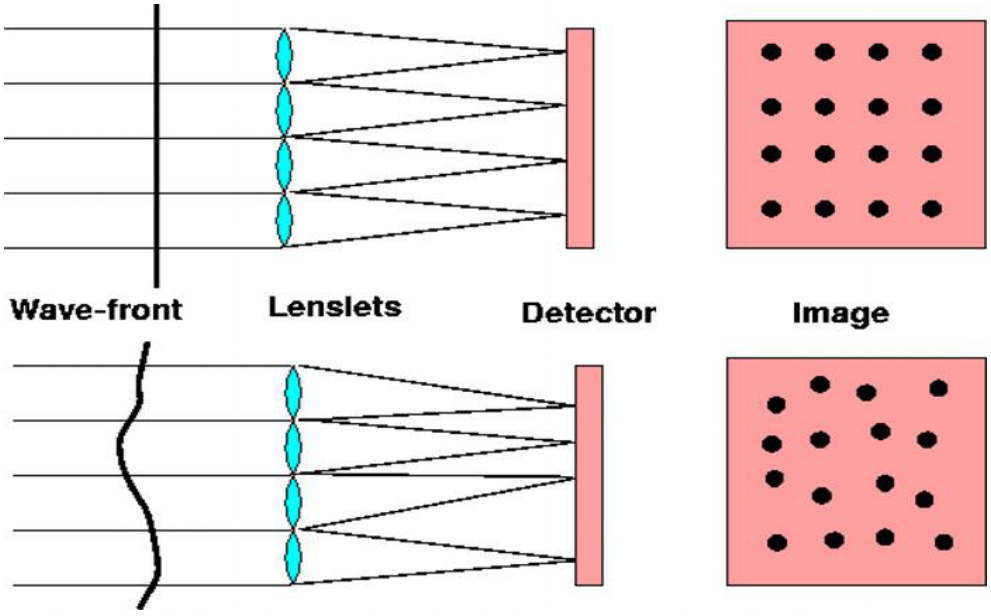

Figure 6.2: Schematic diagram of a ShackHartmann wavefront sensor.[11].

For the sake of simplicity we take our data with appropriate scaling as

$$x_i = \frac{\partial f}{\partial x}$$

and

$$y_i = \frac{\partial f}{\partial y}$$

assuming Ω_i is small enough so that the partial derivatives can be assumed to be constant over Ω_i.

Now our goal is to determine f using these data. Once f is known, the deformable mirror can be adjusted to counter the abberation of the wavefront. So the interpolation problem we wish to solve is as follows:

$$\min_f \left(\|\underline{x} - \frac{\partial f}{\partial x}\|^2 + \|\underline{y} - \frac{\partial f}{\partial y}\|^2 \right) \tag{6.1}$$

6.3 First proposed method

From the previous chapters, we already know that the thin plate spline functions of the form 2.6 are a good way to approximate functions. With this inspiration, we assume that the solution of 6.1 is approximated by a thin plate spline function of the form 2.6. So, assuming f to be of this form, we have

$$\begin{aligned} \frac{\partial f}{\partial x}(x,y) &= \frac{\partial}{\partial x}[d_0 + d_1 x + d_2 y \\ &+ \frac{1}{16\pi} \sum_{i=1}^{n} c_i \|(x,y) - (x_i,y_i)\|^2 \ln \|(x,y) - (x_i,y_i)\|^2] \\ &= d_1 + \frac{1}{8\pi} \sum_{i=1}^{n} c_i (x - x_i)(1 + \ln \|(x,y) - (x_i,y_i)\|^2) \end{aligned}$$

And

$$\begin{aligned}\frac{\partial f}{\partial y}(x,y) &= \frac{\partial}{\partial y}[d_0 + d_1 x + d_2 y \\ &+ \frac{1}{16\pi}\sum_{i=1}^{n} c_i \|(x,y)-(x_i,y_i)\|^2 \ln \|(x,y)-(x_i,y_i)\|^2] \\ &= d_2 + \frac{1}{8\pi}\sum_{i=1}^{n} c_i (y-y_i)(1+\ln \|(x,y)-(x_i,y_i)\|^2)\end{aligned}$$

Let the data be given as z_{xi} and z_{yi} where the terms stand for the ith data for the x derivative and the ith data for the y derivative respectively. Here $i \in \{1,\ldots,n\}$ where n is the number of data points. Now let us assume K_x is an $n \times n$ matrix with ij-th entry given by $\frac{1}{8\pi}(x_i - x_j)(1 + \ln \|(x_i,y_i)-(x_j,y_j)\|^2)$ and K_y is an $n \times n$ matrix with ij-th entry given by $\frac{1}{8\pi}(y_i - y_j)(1 + \ln \|(x_i,y_i)-(x_j,y_j)\|^2)$. So, we have the interpolation problem as to find c, d such that

$$K_x c + Td = z_x$$

$$K_y c + Td = z_y$$

where $c = (c_1,\ldots,c_n)'$, $d = (d_1,\ldots,d_n)'$ and T is as before. Also, $z_x = (z_{x1},\ldots,z_{xn})$ and $z_y = (z_{y1},\ldots,z_{yn})$. We also keep the constrains of the thin plate splines namely

$$T'c = 0$$

Putting together the three sets of equation in a single matrix equation, we get

$$\begin{bmatrix} K_x & 1 & 0 \\ K_y & 0 & 1 \\ T' & 0 & 0 \end{bmatrix} \begin{bmatrix} c \\ d_1 \\ d_2 \end{bmatrix} = \begin{bmatrix} z_x \\ z_y \\ 0 \end{bmatrix} \tag{6.2}$$

We let

$$M_1 = \begin{bmatrix} K_x & 1 & 0 \\ K_y & 0 & 1 \\ T' & 0 & 0 \end{bmatrix} \tag{6.3}$$

We see that this system is clearly over determined. So we try to solve instead of this, the normal equation

$$M_1' M_1 \begin{bmatrix} c \\ d_1 \\ d_2 \end{bmatrix} = M_1' \begin{bmatrix} z_x \\ z_y \\ 0 \end{bmatrix} \tag{6.4}$$

This method produces a function upto a constant.

6.3.1 Condition Number of $M_1'M_1$

The eigenvalues of $M_1'M_1$, have the expected behaviour for uniform points. The maximum eigenvalue increases linearly with N being in the order of 10^2 for $N = 100$ to $N = 625$. The minimum eigenvalue decreases inversely with increasing N, being in the order of 10^{-7} for $N = 100$ to $N = 625$. As a result, starting from 6.5×10^8 for $N = 100$, the condition number grows quadratically to 1.7569×10^{11} for $N = 625$. See figure 6.3. The random points configuration produce values close to the uniform case.

We can also try $(M_1'M_1 + \alpha I)$ in place of $M_1'M_1$, which has lower condition number. The effect of adding αI to $M_1'M_1$ is that it makes the condition

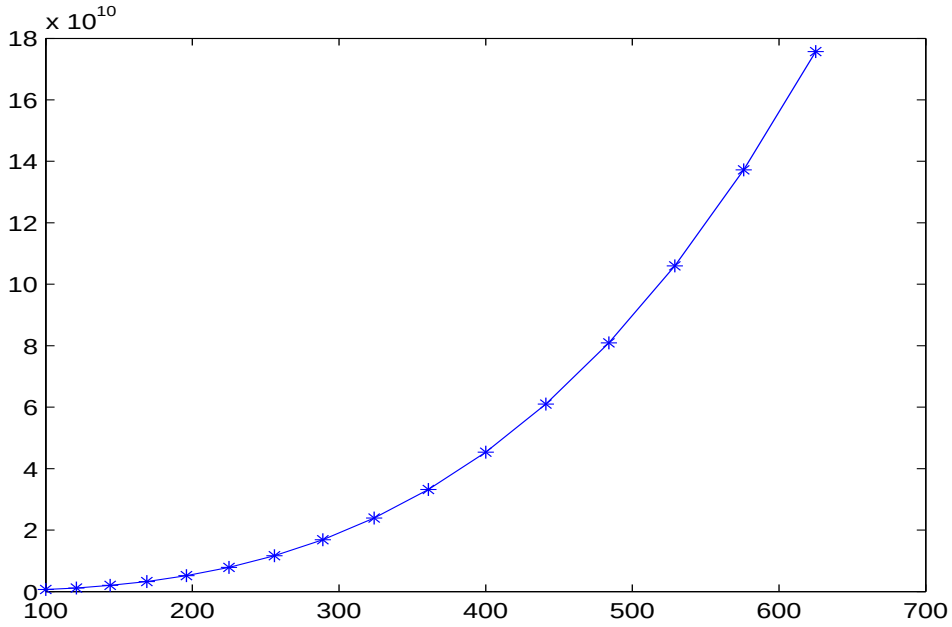

Figure 6.3: Condition numbers of $M_1'M_1$ with varying N.

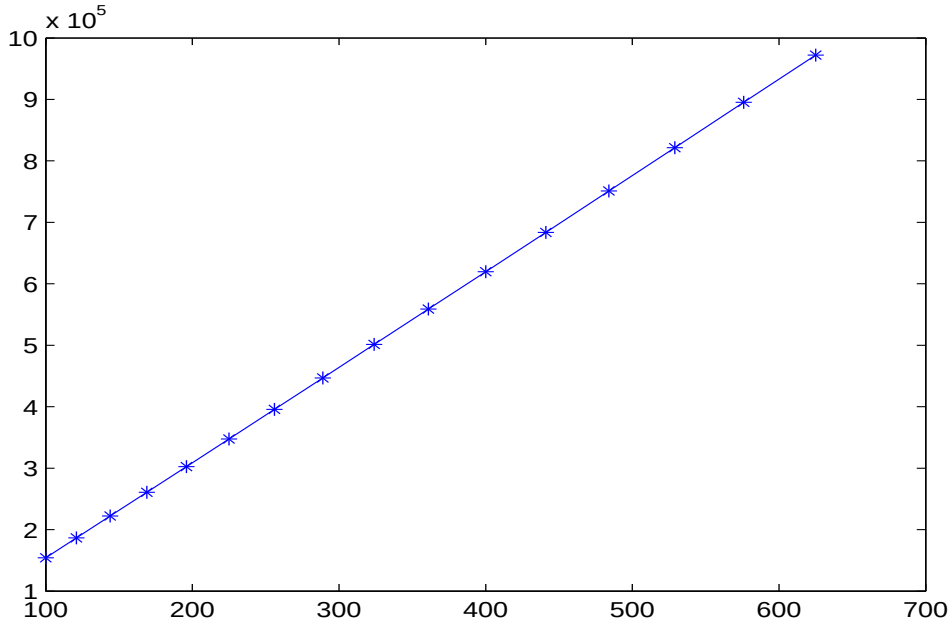

Figure 6.4: Condition numbers of $M_1'M_1 + 0.001I$ with varying N.

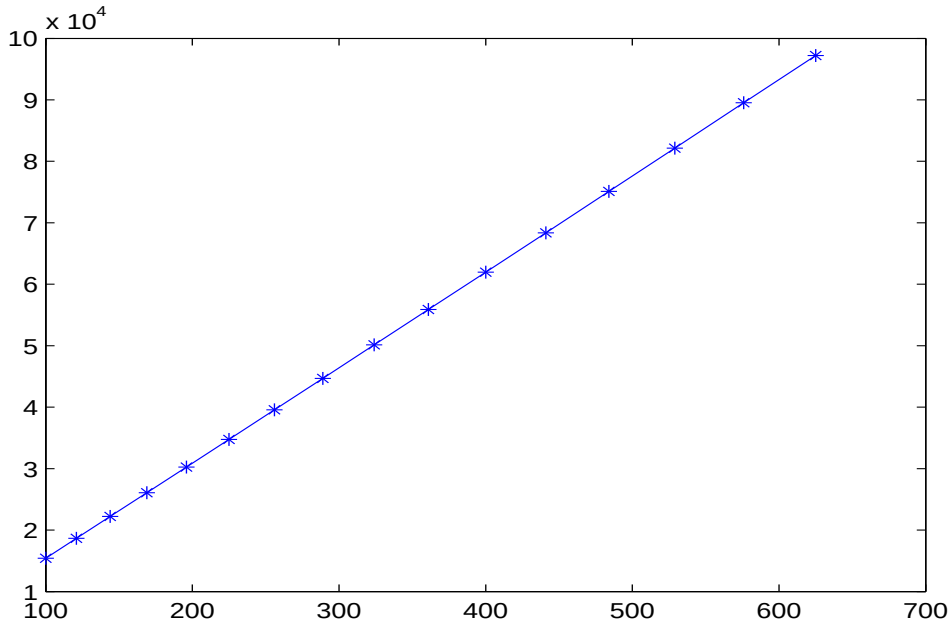

Figure 6.5: Condition numbers of $M_1'M_1 + 0.01I$ with varying N.

number growing linearly with N while also keeping the values lower than that for $\alpha = 0$. For example, for $\alpha = 10^{-2}$, it is 1.5429×10^4 for $N = 100$ and grows linearly to 9.7222×10^4 for $N = 625$. See figure 6.5. And for $\alpha = 10^{-3}$, it is 1.5425×10^5 for $N = 100$ and grows linearly to 9.7220×10^5 for $N = 625$. See figure 6.4.

6.4 Second proposed method

With this method, we try to use the TPS approximation thrice.

First, we produce a TPS approximation for the x-derivative o f the unknown function i.e. $\dfrac{\partial f}{\partial x}$ using the data z_x. Then, we produce a TPS approximation for the y-derivative of the unknown function i.e. $\dfrac{\partial f}{\partial y}$ using the data z_y. Thus we have the x & y derivative of a function, so theoretically, we can

get the function back using these values and the value of the function at any one point. We can take a uniform square grid of 121 points. We follow the following procedure to build data for approximating the function using TPS approximations of its derivatives.

1. Number the points such that the point $(i, j)^{th}$ is numbered as $10i+1+110j$
2. Let $z_1 = 0$. Denote by z_i the data for the function at the i^{th} point.
3. If $m > 1, m - 1 \equiv 0(\mod 12)$ then,

$$z_m = z_{m-12} + \frac{h}{2}(f_x(z_{m-12}) + f_x(z_m) + f_y(z_{m-12}) + f_y(z_m)) \tag{6.5}$$

4. If $m = 110j + 10i + 1$ is such that $i > j$ then,

$$z_m = z_{m-1} + \frac{h}{2}(f_x(z_{m-1}) + f_x(z_m)) \tag{6.6}$$

5. If $m = 110j + 10i + 1$ is such that $i < j$ then,

$$z_m = z_{m-11} + \frac{h}{2}(f_y(z_{m-11}) + f_y(z_m)) \tag{6.7}$$

Here $h = 0.1$, the grid size.

We have used the average value of the two points involved as the increment function. This can be derived using first order Taylor series expansion about both the points for the midpoint value.

Having built up the data, again the TPS approximation is used to approximate the unknown function. With this method, we can start with a random grid for the data for the derivatives and build data on a regular

grid for the function. Thus we can ensure a fixed condition number for the matrix to be used in the last TPS approximation.

Chapter 7

Numerical results on adaptive optics

In this chapter, some results are presented which are obtained by implementing the methods described in the previous chapter. We again consider the domain $[0,1] \times [0,1]$ where we take our data and approximate the function x^2+y^2. Again we take 625 random data points in the domain. We take the data to be from the partial derivatives of the sample function namely $2x$ and $2y$ with some noise in both as $0.005 \times \text{Normal}(0,1)$.

7.1 First method

The first method is implemented with GMRES and CG methods. The results are presented in this section.

7.1.1 GMRES

Figures 7.1, 7.2 and 7.3 show the approximations produced by solving the adaptive optics problem by the first method by GMRES method for $\alpha = 0, 0.001, 0.01$ respectively. For the figure 7.1, the condition number of the corresponding matrix was 6.0064×10^{10}, which is comparable to the condition number associated with the uniform grid. GMRES converged at

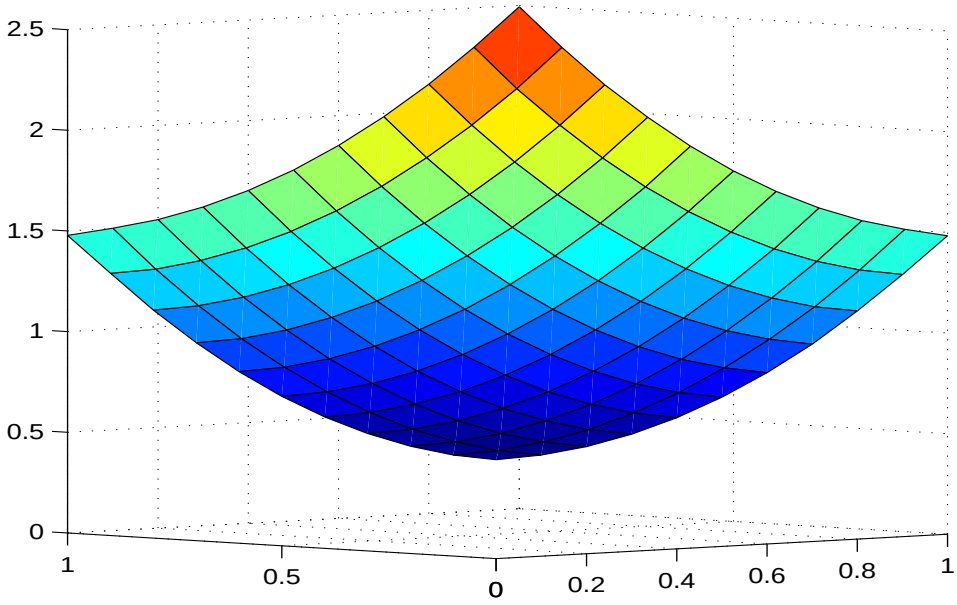

Figure 7.1: Approximation of $x^2 + y^2$ with the first method using GMRES with $\alpha = 0$

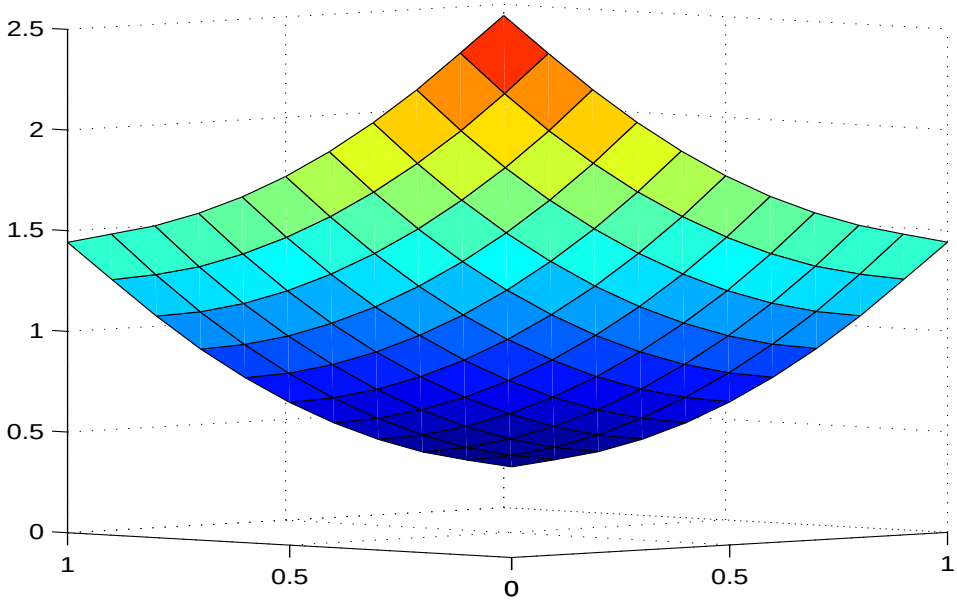

Figure 7.2: Approximation of $x^2 + y^2$ with the first method using GMRES with $\alpha = 0.001$

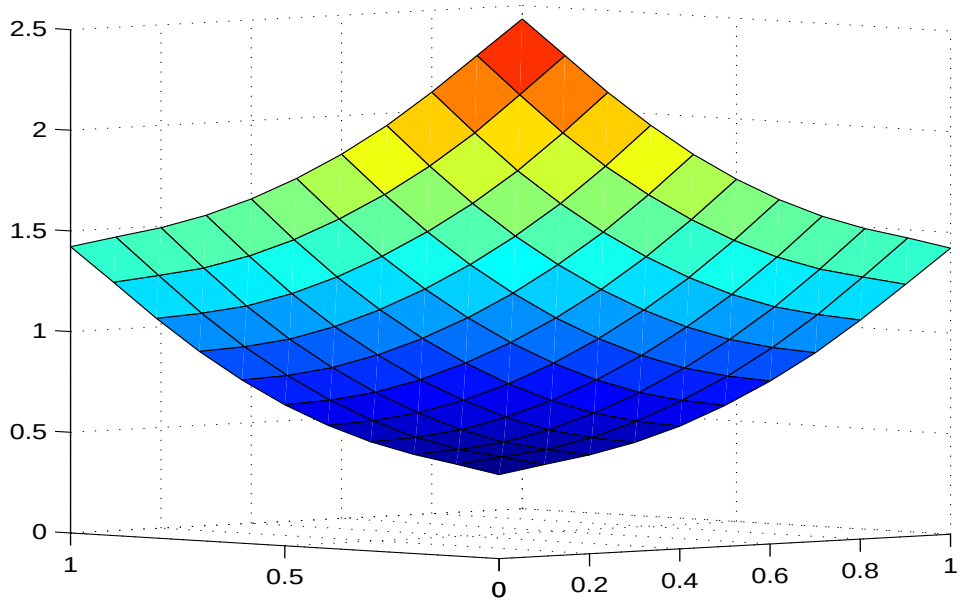

Figure 7.3: Approximation of $x^2 + y^2$ with the first method using GMRES with $\alpha = 0.01$

iteration 55. While the condition number for the matrix corresponding to the figure 7.2 was 9.7402×10^5 which is comparable to the uniform grid case and GMRES converged at iteration 33. And for the figure 7.3, the corresponding matrix has condition number 9.8335×10^4 which is again comparable and GMRES converged at iteration 25.

7.1.2 CG

Figures 7.4, 7.5 and 7.6 show the approximations produced by solving the adaptive optics problem by the first method by CG method for $\alpha = 0, 0.001, 0.01$ respectively. For the figure 7.4, the condition number of the corresponding matrix was 6.0064×10^{10}, which is the same as in GMRES subsection. CG did not converge even at iteration 500. While the condition number for the matrix corresponding to the figure 7.5 was 9.7402×10^5 which is again the same as in GMRES. CG converged at iteration 97. And for

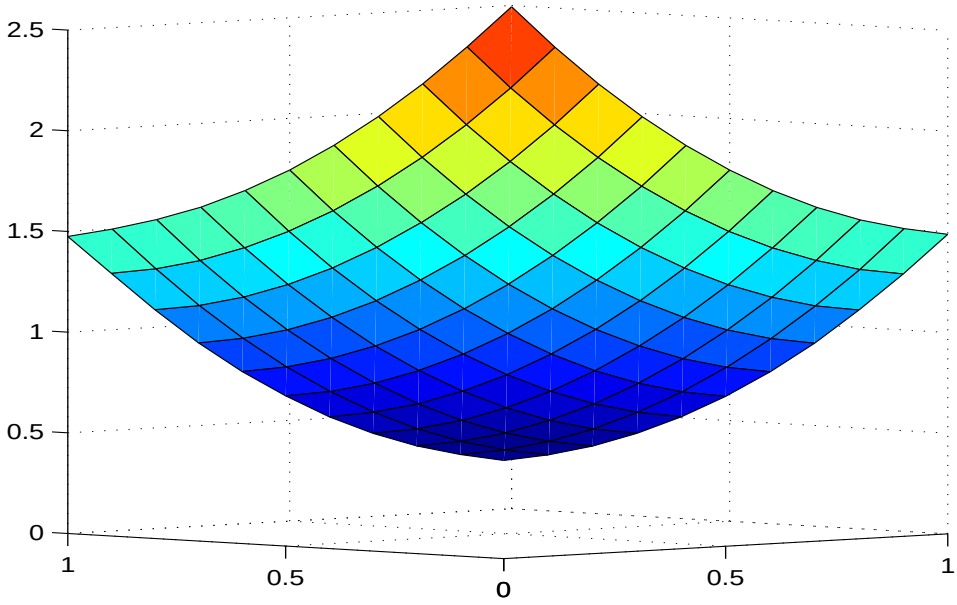

Figure 7.4: Approximation of $x^2 + y^2$ with the first method using CG with $\alpha = 0$

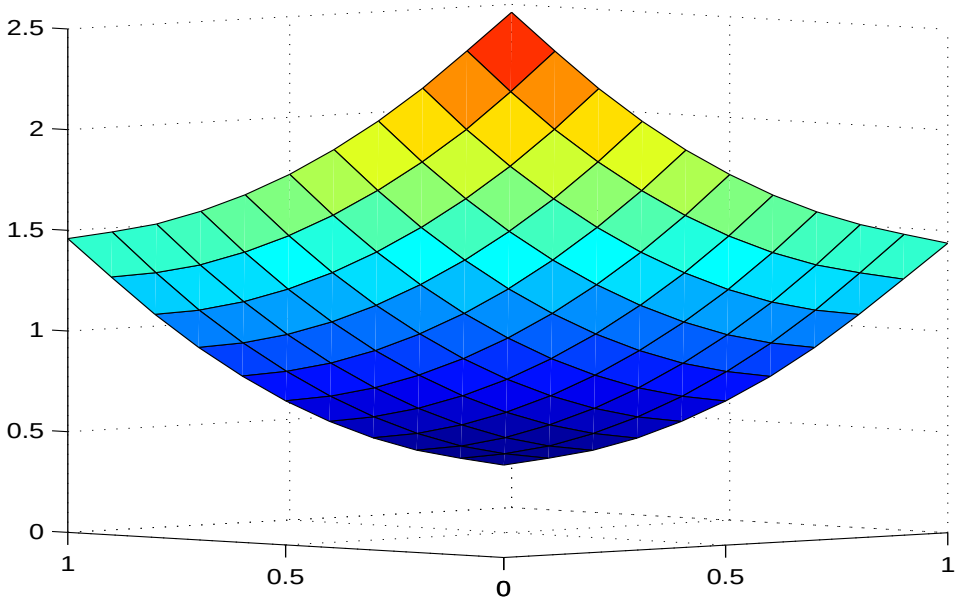

Figure 7.5: Approximation of $x^2 + y^2$ with the first method using CG with $\alpha = 0.001$

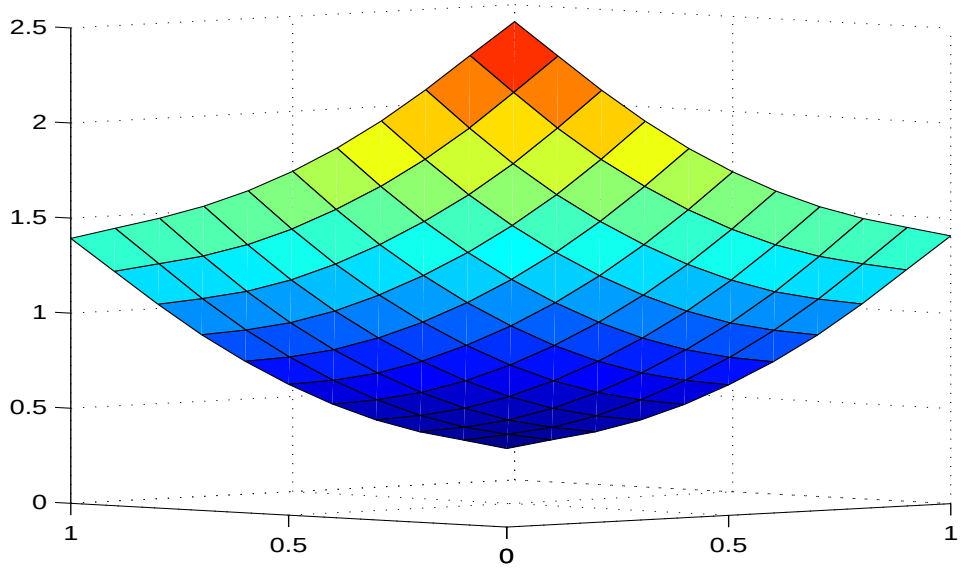

Figure 7.6: Approximation of $x^2 + y^2$ with the first method using CG with $\alpha = 0.01$

the figure 7.6, the corresponding matrix has condition number 9.8335×10^4 which is again the same as in GMRES and CG converged at iteration 52.

Both GMRES and CG produce good approximations for this method. The smoothing effect of the regularization parameter α can be seen in the pictures as for higher α, the curves are flatter and the corners clearly fall below the expected value. GMRES clearly outperforms CG with respect to the number of iterations.

7.2 Second method

Some results on the implementation of the second method for the adaptive optics problem is presented here. As this uses the same TPS program thrice, just one result for each of the three values of α are presented. Taking hint from chapter 5, the case with $\alpha = 0$ is solved by Uzawa's iterative method

while the othe two cases of non zero α is solved by QR method with GMRES.

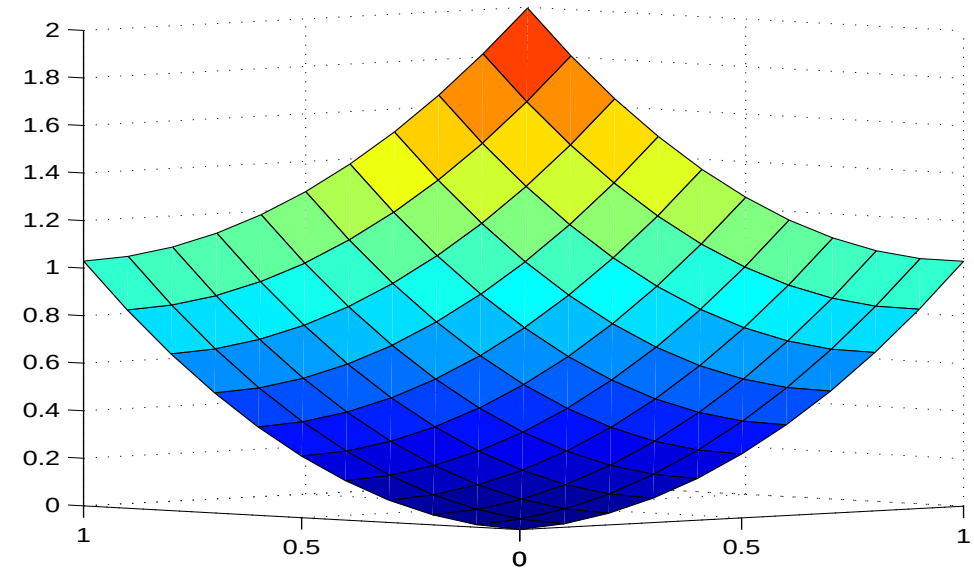

Figure 7.7: Approximation of $x^2 + y^2$ with the second method using Uzawa's iterative method with $\alpha = 0$

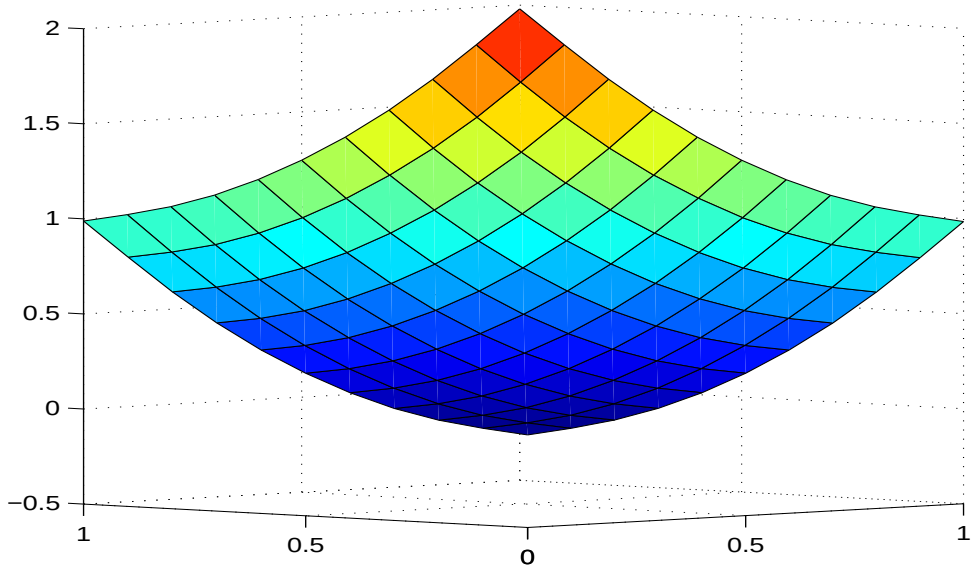

Figure 7.8: Approximation of $x^2 + y^2$ with the second method using Uzawa's iterative method with $\alpha = 0.001$

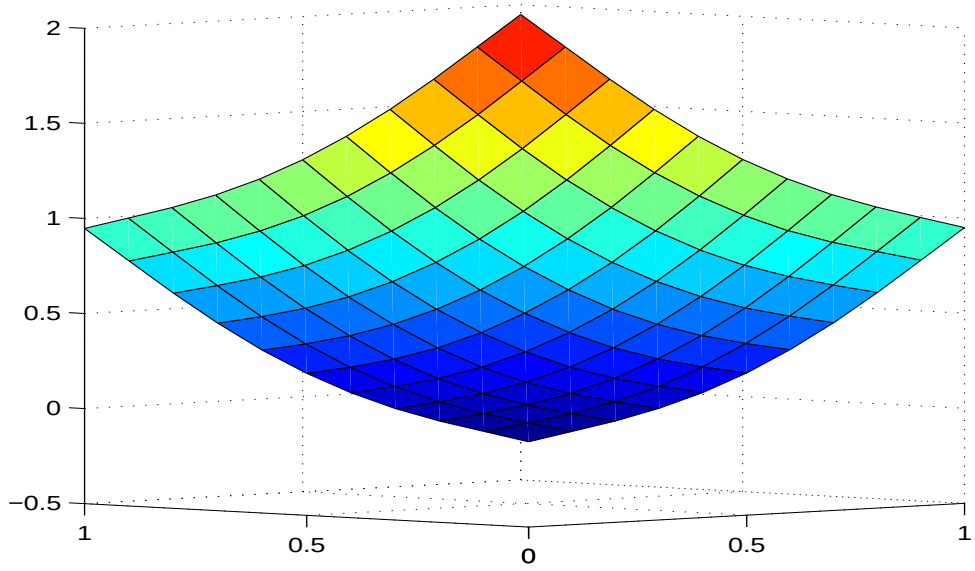

Figure 7.9: Approximation of $x^2 + y^2$ with the second method using Uzawa's iterative method with $\alpha = 0.01$

For the figure 7.7, the initial random grid gives the associated condition number as 2.4489×10^8. The method converges at iteration 120 for the approximation of both derivatives. The final uniform grid gives the condition number 1.1672×10^5. And for making the final approximation, the method converges at iteration 115. For the figure 7.8, the two condition numbers involved are 838.4815 and 170.6167. GMRES converges at iterations 39 and 17. For the figure 7.9, the two condition numbers involved are 80.8231 and 19.7584. GMRES converges at iterations 21 and 10.

We can ignore the shift in the curve and just consider the shape for now as these methods produce the functions upto a constant. The methods produce nice approximation of the sample function. In terms of number of iterations, the first method performs better while the condition numbers involved in the second method are better.

Chapter 8

Conclusion and future work

We have seen that the thin plate splines are very effective tools for approximating given data in two or more dimensions with given degree of smoothness. The minimization problem to produce the thin plate spline function was considered. Some trivial conditions ensure the existence and uniqueness of the solution in two dimension. The problem is transformed into a linear problem. We have also considered transforming the main linear problem into another linear form using QR decomposition method. After studying the condition numbers of the matrices involved in these situations, it can be concluded that the QR decomposition method results in a problem with a matrix having lower condition number. In this regard, this method is efficient. Then we have considered solving the problems with methods like GMRES, CG and Uzawa's iterative scheme. GMRES has been applied to both forms of the problem. CG has been applied only with the QR decomposition method while Uzawa's method has been applied only to the main problem. Considering the number of iterations taken by each method to solve the problems, it turns out that Uzawa's iterative method is the most efficient in the case of absence of any smoothing. But this method fails to improve considerably on the number of iterations in the presence of smoothing. In this case, QR method with GMRES turns out to be superior

both in terms of condition number involved and the number of iterations. To conclude, QR decomposition method with GMRES presents a very efficient way to produce thin plate spline approximation of a function.

Next we looked at the motivation of this report, i.e. adaptive optics. We considered a mathematical model for the adaptive optics problem where we have to produce an approximation of a function based on the data from its derivatives. Two methods have been devised in order to apply the thin plate splines to this situation. The methods give good results and hence it can be concluded that the thin plate splines can be used to efficiently solve the adaptive optics problem.

Further work on thin plate spline can be on producing specific preconditioners for the involved matrices to further lower down the condition numbers leading to the possibility of efficiently working with larger number of points.

Bibliography

[1] J. Duchon, *Fonctions splines et vecteurs aleatoires*, Tech. Report 213, Seminaire d'Analyse Numerique, Universite Scientifique et Medicale, Grenoble, 1975.

[2] J. Duchon, *Fonctions-spline et esperances conditionnelles de champs gaussiens*, Ann. Sci. Univ. Clermont Ferrand II Math, 14, pp. 19-27, 1976.

[3] J. Duchon, *Splines minimizing rotation-invariant semi-norms in Sobolev spaces*, in Constructive Theory of Functions of Several Variables, Springer-Verlag, Berlin, pp 85-100, 1977.

[4] J. Meinguet, *Multivariate interpolation at arbitarary points made simple*, J. Appl. Math. Phys.(ZAMP), 30, 1979.

[5] G. Wahba and J. Wendelberger,*Some new mathematical methods for variational objective analysis using splines and cross-validation*, Monthly Weather Rev., 108, pp. 1122-1145, 1980.

[6] G. Wahba, *Spline Models for Observational Data*, SIAM, 1990.

[7] G. Wahba, *How to smooth curves and surfaces with splines and cross-validation*, in Proc. 24th Conference on the Design of Experiements, U.S. Army Research Office, No. 79-2, Research Triangle Park, NC, pp 167-192, 1979.

[8] J. Meinguet, *An intrinsic approach to multivariate spline interpolation at arbitrary points.*, 1978.

[9] R. Sibson and G. Stone, *Computation of thin-plate splines*, SIAM, 1991.

[10] Y. Saad, *Iterative Methods for Sparse Linear Systems*, 2000.

[11] B. L. Ellerbroek and C. R. Vogel, *Inverse Problems in Astronomical Adaptive Optics*, Topical Review Article, Inverse Problems, 063001, Vol. 25, No. 6, June 2009

Eidesstattliche Erklärung

Ich, Arpan Ghosh, erkläre an Eides statt, dass ich die vorliegende Masterarbeit selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt bzw. die wörtlich oder sinngemä entnommenen Stellen als solche kenntlich gemacht habe.

Author: _____

Arpan Ghosh