UNIVERSITY OF OKLAHOMA

GRADUATE COLLEGE

Partial Differential Equations in Data Analysis

A THESIS
SUBMITTED TO THE GRADUATE FACULTY

in partial fulfilment of the requirements for the Degree of

MASTER OF SCIENCE

BY

Yilin Jiang
Norman, Oklahoma
2019

Partial Differential Equations in Data Analysis

A THESIS APPROVED FOR THE
DEPARTMENT OF MATHEMATICS

BY

Dr. Meijun Zhu, Chair

Dr. John Albert

Dr. Pengfei Zhang

Contents

Partial Differential Equations in Data Analysis

Yilin Jiang

Abstract

In this thesis, we introduce new interpolation methods for two dimensional data via constructing harmonic functions passing through the given data. Two ways to construct the harmonic function are introduced: (1) constructing the harmonic function via the heat equation, and (2) constructing the harmonic function via the boundary element method

1 Introduction

Weather forecasting is not an easy job, and sometimes its accuracy can be affected by many factors. In middle of 2014, with the sharp fall of oil prices, the Russian rouble had declined drastically. However, at beginning of 2015, China's National Weather Center said in its own microblog that China's weather forecasts have become less accurate due to the Russian financial crisis [6]. Because of the funding cut, Russian's weather center reduced its upper air data collecting through radiosonde observations from twice a day to once a day. Because of geographical reason, accurate upper air weather data from Russia are crucial to China's weather forecast especially for predicting cold air from Siberia. Missing data (or in this case, simply less data) problems are not rare in real life data analysis. Interpolation methods are some of the most useful tools to deal with such a problem.The main idea of interpolation methods is to find an approximating function that passes through all the given data points.

Definition 1.1. *A function* $P : \mathbb{R}^n \longrightarrow \mathbb{R}$ *satisfying*

$$P(x_i) = y_i \quad i = 1, 2, 3, ..., n, \tag{1}$$

is called an interpolating function passing through distinct $(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)$*, where* $x_1, x_2, ...x_n \in \mathbb{R}^2, y_1, y_2, ...y_n \in \mathbb{R}$. $(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)$ *are called interpolating nodes.*

Using interpolation methods, one can fill in missing data on a given interval by calculating the values of the interpolating function. Hence, a more accurate weather forecast can be expected. Another scenario is that there are many tough places to forecast weather around the world. One may build a personal weather station at home easily, but building weather station in some places is a time and money consuming task. The first weather station on Mountain Everest was not built untill 2004, 50 years after the first climbers stood at the top of the mountain, in 1953. [2]. In these cases, data are only given on the boundary, and the interpolation methods are needed to estimate the data inside the given area.

Before some well known interpolation methods like Lagrange method and Newton's method were invented during the Age of Scientific Revolution, many interpolation methods were used in ancient China. Second order interpolation was pioneered by the Chinese early astronomer Liu Zhuo who used this technique to calculate the relative distance between the sun and the moon [9]. However, the mathematician, astronomer, and monk Yixing created a better interpolation method based on Liu Zhuo's method, called Da Yan Calendar, using unequal interval second order interpolation [5]. In fact, many mathematicians tried to find a reasonable way to find an approximation function interpolating the given randomly distributed data on a two dimensional plane. Subdividing the domain into small regions like triangles or rectangles and connecting the given data points is a common interpolation method [1] [7] [8]. In [7], Maude introduced a way that divided the domain into several regions formed by intersections of circles. In [8], however, the author revealed that Maude's result will be awkward when some special number of data points are given. In [13], Shepard came up with an algorithm called the Inverse Distance Weight interpolation. In his algorithm, the interpolation function had the form $f(x) = \Sigma_{i=1}^{n} w_i(x) f_i$, where $w_i(x)$ is the weight coefficient which depends on the reciprocal of distance between interpolate point x and all other n known points x_i. A group of scientists studied Shepard's method and improved the interpolation function through modifying the weight function w_i. However, in [4], Franke pointed out that Shepard's method is very dependent on weight function. Franke then listed several unacceptable weight functions and time consuming functions. Both the subdividing method and the Inverse Distance Weight method require much work to be done before interpolating. Our method focus on providing another way to interpolate the given data points.

2 Interpolation

There are many ways to find P. Common interpolation methods include Lagrange Interpolation, Newton Interpolation, Hermite Interpolation, Linear Interpolation, and Cubic Spline Interpolation. Five interpolation methods are shown in Figure 1, where they are all used to find the approximation function of $f(x) = \frac{1}{x^2+1}, x \in [-5, 5]$. Notice that the node points in Figure 1 are equidistant nodes.

2.1 One Dimensional Interpolation

1. Polynomial Interpolation

   In 1885, a very famous theorem proved by Karl Weierstrass indicated the use of polynomial to approximate a continuous function:

   Theorem 2.1. (Weierstrass Approximation Theorem)
   Suppose f is a continuous real-valued function on $[a, b]$. For any given $\epsilon > 0$, there exists a polynomial P on $[a, b]$ such that $|f(x) - P(x)| < \epsilon$ for all $x \in [a, b]$.

   In words, any continuous function on an interval can be uniformly approximated by the polynomial to any degree of accuracy.
   If $P(x)$ satisfies (1) and is a polynomial degree less than n, i.e.

$$P_n(x) = a_0 + a_1 x + a_2 x^2 + ... + a_n x^n,$$

where $a_i \in \mathbb{R}$, then $P(x)$ is the interpolating polynomial, the corresponding method is called the polynomial interpolation. The following theorem shows that such a polynomial exists and it is unique.

Theorem 2.2. (Main Theorem of Polynomial Interpolation)
Let $(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)$ be n points in the plane with distinct x_i. There exists one and only one polynomial $P(x)$ of degree less than or equal to $n-1$ satisfying $P(x_i) = y_i \, for \, i = 1, ..., n$.

Because of its special properties, like smoothness, polynomials are often used in interpolation. The Lagrange interpolating polynomial, which has the form

$$P(x) = \sum_{k=1}^{n} \left(\prod_{j \neq k} \frac{x - x_j}{x_k - x_j} \right) y_k, \tag{2}$$

is one of the representations of the interpolating polynomial. However, (2) must computed repeatedly each time when adding node points. An alternative way, which has less computational cost, is called Newton interpolation. The Newton interpolating polynomial has the following form:

$$\begin{aligned} P(x) = & y_1 + f[x_1, x_2](x - x_1) + f[x_1, x_2, x_3](x - x_1)(x - x_2) \\ & + ... + f[x_1, x_2, ..., x_n](x - x_1)(x - x_2)(x - x_3)...(x - x_n - 1). \end{aligned}$$

where $f[x_i, x_j] = \frac{y_i - y_j}{x_i - x_j}$, and $f[x_1, x_2, ..., x_k] = \frac{f[x_1, x_2, ..., x_k - 1] - f[x_2, x_3, ..., x_k]}{x_1 - x_k}$. Both Lagrange interpolation and Newton interpolation require the polynomial to interpolate the given data points, i.e. $P(x_i) = y_i$. Notice that given the same data points (x_i, y_i) for $i = 1, 2, ...n$, the polynomials derived from Interpolation methods like Newton and Lagrange are the same. Hermite Interpolation requires the polynomial to satisfy the interpolation conditions on derivatives, i.e. to find the polynomial that satisfies

$$P(x_i) = y_i, \quad P'(x_i) = y'_i, \quad for \, i = 1, 2, ..., n$$

where y'_i are the derivatives of original functions at the node point (x_i, y_i).

2. Piecewise Interpolation
Five interpolation results are shown in figure 1. Notice that the original function $f(x) = \frac{1}{x^2+1}$ is shown in black, and the Lagrange interpolating polynomial is shown in red. Figure 1 shows that a high order interpolating polynomial may have unstable result at the edge of an interval. Piecewise Interpolation is often used to avoid increasing the degree of polynomial when adding more node points.

Nearest Neighbour Interpolation is the simplest interpolation method in one dimension. The result of this interpolation is a piecewise constant function constructed by selecting the value of nearest given data point and ignoring the neighbour points in each interval. In figure 1, the result of nearest neighbour interpolation for approximating $f(x) = \frac{1}{x^2+1}, x \in [-5, 5]$ are shown in pink. Compared to the original function, this method does not guarantee good accuracy, but it is a fast interpolation method on two dimensional problems. Another simple piecewise interpolation is based on Linear Interpolation. Suppose we have two node points, $(x_1, y_1), (x_2, y_2)$. Then the linear interpolating function is of the form

$$P(x) = y_1 + (y_2 - y_1)\frac{x - x_1}{x_2 - x_1} \text{ at the point } (x, y).$$

In Piecewise Linear Interpolation, one finds a piecewise function $P(x)$ satisfying:

(a) $P(x) \in C[a, b]$;

(b) $P(x_i) = y_i (i = 1, 2, ..., n)$;

(c) $P(x)$ is a linear function in each subinterval $[x_i, x_i + 1]$.

The result of Piecewise Linear Interpolation is shown in the figure 1. Apparently, it is not the most accurate method especially when the number of node points is not large enough. Piecewise Cubic Hermite interpolation is shown in the figure 1 in blue. The main idea of this method, based on Hermite interpolation, is to construct a continuously differentiable piecewise polynomial $P(x)$. Although Piecewise Linear Interpolation and Piecewise Hermite Interpolation can avoid the problem of oscillation, the functions constructed by using these two methods usually are not smooth curves. The curve constructed by Cubic Spline Interpolation is connected by piecewise cubic curves which are twice continuously differentiable at the connection points.

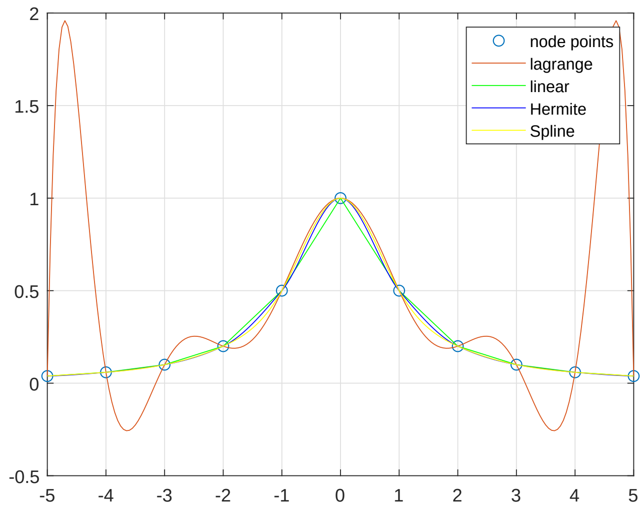


Figure 1: Four One Dimension Interpolation Methods

2.2 Multivariate Interpolation

If the interpolated function has more than one variable, Multivariate Interpolation such as, Nearest Neighbour Interpolation, Bilinear Interpolation, and Bicubic Interpolation are usually used to approximate it. In Figure 2, these three interpolation results for given data sets are drawn on a uniform two dimensional grid. The interpolated function is $f(x,y) = \sin y \cos x$. As mentioned before, Nearest Neighbour Interpolation is a fast interpolation method that can be used in any dimension. Shown in Figure 2, the interpolation function is also an piecewise constant function. Bilinear Interpolation is the extension of Linear Interpolation in two dimensions. The main process of Bilinear Interpolation is to do the interpolation first in the x -direction(or y -direction) and then find the estimation by doing interpolating in the y -direction(or x -direction). Bicubic Interpolation can construct a smooth interpolation curve, see also in figure 2.

Previous interpolation methods are commonly used in different areas, however, these methods ignored the physical meaning of the given data sets. Also, in real life problems, like rainfall or temperature estimation, the data sets are usually not distributed well. In this paper, we want to use harmonic function to interpolate the given data points. We discussed two situations: first, the data points are given on the boundary of the domain; second, we only know data at several points inside the domain.

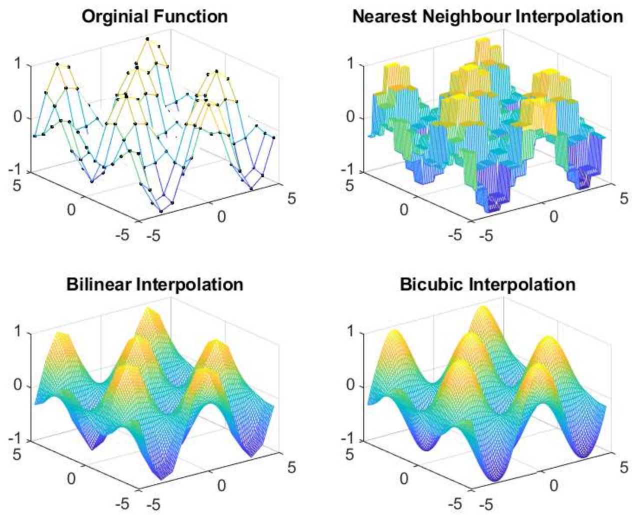


Figure 2: Three Two Dimension Interpolation Methods

3 Properties of Harmonic Function

In the previous section, we introduced several common interpolation methods. As we mentioned before, some interpolation methods have not considered the physical meaning of given data or its distribution. Here, we want to use harmonic functions, one of the most beautiful type of functions to interpolate the given data points. First, let's define what harmonic function is.

Definition 3.1. *A harmonic function is a twice continuously differentiable function $u : \Omega \to \mathbb{R}$ that satisfies Laplace's equation where Ω is the domain in $\mathbb{R}^n$. That is*

$$\Delta u = 0 \tag{3}$$

In other words, a harmonic function is a solution of Laplace equation. Some properties of harmonic function will be introduced in the following. It is these properties which make it possible to be considered as one kind of interpolation function. In this paper, we fix $\Omega \subset \mathbb{R}^2$ be a bounded smooth domain (open, connected set). Recall

Theorem 3.1. (Divergence Theorem) *Let $\Omega \subset \mathbb{R}^2$, and $F \in C^1(\overline{\Omega})$, then we have:*

$$\int_\Omega \nabla \cdot F\, dx = \int_{\partial\Omega} F \cdot n\, dS$$

where n is the unit outward normal.

From the above theorem, one can easily derive the following

Theorem 3.2. (Green's First Identity) *Let $\Omega \subset \mathbb{R}^2$ and $u \in C^2(\overline{\Omega})$, and let $v \in C^1(\overline{\Omega})$. Then we have*

$$\int_\Omega \nabla v \cdot \nabla u\, dx + \int_\Omega v\, \Delta u\, dx = \int_{\partial\Omega} v \frac{\partial u}{\partial n}\, dS. \tag{4}$$

Proof. From the equality,

$$\nabla v \cdot \nabla u + v\, \Delta u = \nabla \cdot (v \nabla u),$$

and Theorem 3.1,we have

$$\int_\Omega \nabla v \cdot \nabla u\, dx + \int_\Omega v \Delta u\, dx = \int_\Omega \nabla \cdot (v\nabla u) dx = \int_{\partial\Omega} (v\nabla u) \cdot n\, dS = \int_{\partial\Omega} v\, \frac{\partial u}{\partial n}\, dS.$$

By symmetry, (4) can be rewritten as the following:

$$\int_\Omega \nabla v \cdot \nabla u\, dx + \int_\Omega u\, \Delta v\, dx = \int_{\partial\Omega} u \frac{\partial v}{\partial n}\, dS. \tag{5}$$

□

Let g be a function defined on $\partial\Omega$. Define

$$\mathcal{A} = \{w \in C^2(\Omega) \,|\, w \equiv g \text{ on } \partial\Omega\}$$

Definition 3.2. *For $w \in \mathcal{A}$, define the energy functional of w to be*

$$I[w] = \frac{1}{2} \int_\Omega |\nabla w|^2. \tag{6}$$

Theorem 3.3. (Dirichlet's Principle)
For $u \in \mathcal{A}$, we have

$$u \text{ is a harmonic function if and only if } I[u] = \min_{w \in \mathcal{A}} I[w].$$

Proof. We first prove necessity. For fixed $v \in C_c^\infty(\Omega)$, we can define $\phi(t) = I[u+tv]$, note that $u + tv \in A$ because v has compact support. Note that

$$\begin{aligned} f(t) &= \frac{1}{2}\int_\Omega |\nabla(u+tv)|^2\,dx \\ &= \frac{1}{2}\int_\Omega |\nabla u + t\nabla v|^2\,dx \\ &= \frac{1}{2}\int_\Omega [\,|\nabla u|^2 + 2t\,\nabla u \nabla v + t^2\,|\nabla v|^2]\,dx \\ &= I[u] + t\int_\Omega \nabla u \nabla v + t^2\, I[v]. \end{aligned}$$

Based on our assumption, $f(u)$ is minimal at $t = 0$. Hence, consider its derivative with respect to t

$$\begin{aligned} 0 = \frac{d}{dt} f(t)|_{t=0} &= \int_\Omega \nabla u \nabla v\,dx \\ &= \int_{\partial\Omega} v\,\frac{\partial u}{\partial n}\,dS - \int_\Omega v\,\Delta u\,dx. \end{aligned}$$

Since v has compact support, therefore $v = 0$ on $\partial\Omega$, and it follows that

$$\int_\Omega v\,\Delta u\,dx = 0 \quad \forall v \in C_c^2(\Omega).$$

Here we claim that $\Delta u = 0$. Suppose $\Delta u(x) \neq 0$ for some $x \in \Omega$, we may assume $\Delta u(x) > 0$, also notice that Δu is continuous at x, therefore, there exists $B(x,r) \subset \Omega$ such that $u(y) > 0$, for all $y \in B(x,r)$. We can also pick $v \in C_c^2(B(x,r))$ such that $v(y) > 0 \ \ for\ y \in B(x,r).$ Then, we have:

$$\int_\Omega v\Delta u\,dx = \int_{B(x,r)} v\Delta u\,dx \; > \; 0.$$

Contradiction.

To prove the other direction, let $w \in A$, and we need to show that $I[u] \le I[w]$.
Let $v = w - u$, then we have:

$$
\begin{aligned}
I[w] = I[v+u] &= \int_\Omega |\nabla(v+u)|^2 \\
&= \int_\Omega (\nabla v + \nabla u)\cdot(\nabla v + \nabla u) \\
&= \int_\Omega |\nabla v|^2 + \int_\Omega |\nabla u|^2 + 2\int_\Omega (\nabla v \, \cdot \nabla u) \\
&= I[v] + I[u] + \int_\Omega \nabla(w-u)\cdot\nabla u \\
&= I[v] \ + I[u] + \int_\Omega (w-u)\cdot\Delta u - \int_{\partial\Omega}(w-u)\frac{\partial u}{\partial n} dS \\
&= I[v] + I[u] \geq I[u].
\end{aligned}
$$

□

The Dirichlet Principle states that if u is a harmonic function that satisfies (3), then it minimizes the energy. To find the approximation function for the given data or image, we usually want to find the minimizer or at least the local minimizer of some functional. By the Dirichlet Principle, the harmonic function is the minimizer of the gradient u in L^2 norm. Before using harmonic functions to interpolate the given Dirichlet boundary data, one question is why only one solution exists. The uniqueness can be explained by the Maximum Principle. I will prove it by using the following lemma called the Mean Value Property.

Lemma 1. (Mean Value Property) *Suppose u is a harmonic function, and $u \in C^2(\Omega)$. For $x \in \Omega$ and any open ball $B(x,r) \subset \Omega$, we have*

$$
\begin{aligned}
u(x) &= ⨍_{B(x,r)} u\,dy = \frac{1}{|B(x,r)|}\int_{B(x,r)} u\,dy \\
&= ⨍_{\partial B(x,r)} u\,dy = \frac{1}{|\partial B(x,r)|}\int_{\partial B(x,r)} u\,dy.
\end{aligned}
$$

Theorem 3.4. (Maximum Principle)
If u is a harmonic function in Ω, and $u \in C^2(\Omega) \cup C(\partial\Omega)$, then

1. *we have*

$$\max_{\overline{\Omega}} u = \max_{\partial\Omega} u.$$

2. *If there exist $x_0 \in \Omega$ such that $u(x_0) = \max_{\partial\Omega} u$, and Ω is connected, then $u \equiv u(x_0)$ in Ω.*

Suppose we have two harmonic functions u_1, u_2, and $u_1, u_2 \in C^2(\Omega) \cup C(\partial\Omega)$. Suppose both of them satisfy the given boundary data. Consider the function $u_1 - u_2$, it has to be zero on the boundary Ω. By the first part of the Maximal Principal, we know that $u_1 - u_2 \leq 0$ on Ω $\partial\Omega$. Now we switch order, and consider function $u_2 - u_1$. Apply Maximum Principal again we get $u_2 - u_1 \leq 0$. Hence, we know that $u_1 = u_2$ on Ω.

Theorem 3.5. (Green's Second Identity) *Suppose $u, v \in C^2(\Omega) \cup C^1(\overline{\Omega})$. Then,*

$$\int_\Omega u\Delta v - v\Delta u \, dx = \int_{\partial\Omega} u\frac{\partial v}{\partial n} - v\frac{\partial u}{\partial n} \, dS.$$

Here, $\partial/\partial n$ means the directional derivative with respect to the unit outer normal.

Proof. In view of Green's First Identity, we have

$$\int_\Omega \nabla u \nabla v \, dx + \int_\Omega u\Delta v \, dx = \int_{\partial\Omega} u\frac{\partial v}{\partial n} \, dS$$

and

$$\int_\Omega \nabla v \nabla u \, dx + \int_\Omega v\Delta u \, dx = \int_{\partial\Omega} v\frac{\partial u}{\partial n} \, dS.$$

The result follows from subtracting these two equations. □

In the reminder of this thesis, we focus our attention on $n = 2$.

Definition 3.3. *Define*

$$\Phi(x) = -\frac{1}{2\pi} \log |x|$$

for $x \in \mathbb{R}^2 \backslash \{0\}$. The function Φ is called ***the fundamental solution of Laplace equation*** *(in dimension 2). We also define $\Phi_x(y) = \Phi(x - y)$.*

It is easy to see that $\Delta_y \Phi_x(y) = 0$ for $y \neq x$. Now we can state our main theorem.

Theorem 3.6. *Let $u \in C^2(\Omega) \cup C^1(\overline{\Omega})$ be harmonic.*

1. *For $x \in \Omega$, we have*

$$u(x) = \int_{\partial\Omega} \Phi(y - x)\frac{\partial u}{\partial n}(y) - u(y)\frac{\partial \Phi}{\partial n}(y - x) \, dS(y)$$

2. Suppose $\partial\Omega$ is smooth. For $x \in \partial\Omega$, we have

$$\frac{1}{2}u(x) = \int_{\partial\Omega} \Phi(y-x)\frac{\partial u}{\partial n}(y) - u(y)\frac{\partial \Phi}{\partial n}(y-x)\, dS(y);$$

Proof. Suppose $x \in \Omega$. Since Ω is open, we can find $\epsilon > 0$ such that $\overline{B_{2\epsilon}(x)} \subset \Omega$. Let $\Omega_\epsilon = \Omega \backslash B_\epsilon(x)$ Now we apply Green's Second Identity to obtain

$$0 = \int_{\partial\Omega_\epsilon} u(y)\frac{\partial \Phi}{\partial n}(y-x) - \Phi(y-x)\frac{\partial u}{\partial n}(y)\, dS(y).$$

Because $\partial\Omega_\epsilon = \partial\Omega \cup \partial B_\epsilon(x)$, we have

$$\int_{\partial B_\epsilon(x)} u(y)\frac{\partial \Phi}{\partial n}(y-x) - \Phi(y-x)\frac{\partial u}{\partial n}(y)\, dS(y) = \int_{\partial\Omega} \Phi(y-x)\frac{\partial u}{\partial n}(y) - u(y)\frac{\partial \Phi}{\partial n}(y-x)\, dS(y).$$

Our aim is to show that the left-hand-side will converge to $u(x)$ as $\epsilon \to 0$. From the Mean Value Property, we obtain

$$\int_{\partial B_\epsilon(x)} u(y)\frac{\partial \Phi}{\partial n}(y-x)\, dS(y) = \int_{\partial B_\epsilon(x)} u(y)\frac{1}{2\pi\epsilon}\, dS(y) = u(x).$$

Remember that the harmonic function is smooth and that $\overline{B_{2\epsilon}(x)}$ is compact. For each $y \in \overline{B_\epsilon(x)}$ we have

$$\left|\frac{\partial u}{\partial n}(y)\right| \le |\nabla u(y)| \le M,$$

where M is the maximum value of $|\nabla u|$ on $\overline{B_{2\epsilon}(x)}$. Hence, we get

$$\begin{aligned}
\left|\int_{\partial B_\epsilon(x)} \Phi(y-x)\frac{\partial u}{\partial n}(y)\, dS(y)\right| &\le \int_{\partial B_\epsilon(x)} \left|\Phi(y-x)\frac{\partial u}{\partial n}(y)\right|\, dS(y) \\
&\le M \int_{\partial B_\epsilon(x)} |\Phi(y-x)| dS(y) \\
&= M \cdot \frac{\log \epsilon}{2\pi} \cdot 2\pi\epsilon = M(\epsilon \log \epsilon).
\end{aligned}$$

Because M is a uniform bound for all $\epsilon' \le \epsilon$ and because $\epsilon \log \epsilon \to 0$ as $\epsilon \to 0$, this integral approaches 0 as $\epsilon \to 0$. Hence, the first part of the theorem follows.

Now suppose $\partial\Omega$ is smooth and let $x \in \partial\Omega$. Pick $\epsilon > 0$ such that $B_{2\epsilon}(x) \cap \Omega \subset \Omega$. By similar argument, we derive

$$\int_{\partial B_\epsilon(x)} u(y)\frac{\partial \Phi}{\partial n}(y-x) - \Phi(y-x)\frac{\partial u}{\partial n}(y)\, dS(y) = \int_{\Gamma_\epsilon} \Phi(y-x)\frac{\partial u}{\partial n}(y) - u(y)\frac{\partial \Phi}{\partial n}(y-x)\, dS(y),$$

where $\Gamma_\epsilon = \partial\Omega_\epsilon \backslash B_\epsilon(x)$. Since $\partial\Omega$ is smooth, the limit of left-hand-side as $\epsilon \to 0$ will be

$$\lim_{\epsilon \to 0} \int_0^\pi \left[u(\epsilon, \theta)\frac{\partial \Phi}{\partial n}(\epsilon) - \Phi(\epsilon)\frac{\partial u}{\partial n}(\epsilon, \theta) \right] \epsilon\, d\theta.$$

Let's estimate this integral. The first part is

$$\int_0^\pi u(\epsilon, \theta)\frac{\partial \Phi}{\partial n}(\epsilon)\epsilon\, d\theta = \int_0^\pi u(\epsilon, \theta)\frac{1}{2\pi\epsilon}\epsilon\, d\theta = \frac{1}{2\pi}\int_0^\pi u(\epsilon, \theta) d\theta \to \frac{1}{2\pi}\pi u(x)$$

as $\epsilon \to 0$. Using a similar trick to find a uniform bound M, the second part is

$$\left| \int_0^\pi \Phi(\epsilon)\frac{\partial u}{\partial n}(\epsilon, \theta)\epsilon\, d\theta \right| \le M \cdot \frac{\log \epsilon}{2\pi} \cdot \pi\epsilon = \frac{M}{2}\epsilon \log \epsilon \to 0$$

as $\epsilon \to 0$. Because $\Gamma_\epsilon \to \partial\Omega$ as $\epsilon \to 0$, we conclude that

$$\frac{1}{2}u(x) = \int_{\partial\Omega} \Phi(y-x)\frac{\partial u}{\partial n}(y) - u(y)\frac{\partial \Phi}{\partial n}(y-x)\, dS(y);$$

the second part of the theorem follows. □

4 Method and Numerical Experiment

Our main goal is to find the solution of the following problem:

$$u(x_i) = v_i,\ \Delta u = 0, \tag{7}$$

where $v_i \in \mathbb{R}$ are given function values of an unknown smooth harmonic function which has fixed boundary or interior values at given data points $x_i \in \mathbb{R}^2$. Two different situations are considered here: given interpolating nodes are on the boundary, and when the given interpolating nodes are distributed on the whole domain.

4.1 Data are given on Boundary

4.1.1 Method

Now from both the Dirichlet Principle and Maximum Principle, we consider finding the unique interpolating harmonic function on the given boundary. However, it is not easy to find the harmonic function. Let's consider the following equation system:

$$\begin{cases} \frac{\partial u}{\partial t} = \Delta u \quad \text{for } (x, y, t) \in \Omega \times (0, \infty), \\ u(x, y, 0) = F(x, y) \quad \text{for } (x, y) \in \Omega \\ u(x, y, t) = f(x, y) \quad \text{for } (x, y, t) \in \partial\Omega \times [0, \infty) \end{cases} \tag{8}$$

Hence, instead of finding the harmonic function directly, we use the heat equation to find it. Why this is correct? For a function of two variable and the time variable t, like the heat equation in (8), we have the following:

$$\lim_{t \to \infty} u(x, y, t) = v(x, y).$$

Hence $v(x, y)$ satisfies:

$$\begin{cases} \Delta v(x, y) = 0 \quad \text{for } (x, y) \in \Omega \\ v(x, y) = f(x, y) \quad \text{for } (x, y) \in \partial\Omega. \end{cases}$$

To find the approximation by heat equation, a known method called finite differences, which replaces each derivative by a difference quotient, will be used in the experiment.To find the function satisfying (8) on Ω, we do the following steps:

- Draw a grid on Ω and choose mesh size $\Delta x, \Delta y, \Delta t$ for x, y, t.
- Approximate the value of $u(i\Delta x, j\Delta y, n\Delta t)$ for $x = i\Delta x, y = j\Delta y, t = n\Delta t$ by a number $u_{i,j}^n$ indexed by integers i, j, n:

$$u_{i,j}^n = u(i\Delta x, j\Delta y, n\Delta t)$$

Therefore we can immediately have the following approximations:

$$\frac{\partial u}{\partial t} \sim \frac{u_{i,j}^{n+1} - u_{i,j}^n}{\Delta t}$$

$$\frac{\partial^2 u}{\partial x^2} \sim \frac{u_{i+1,j}^n - 2u_{i,j}^n + u_{i-1,j}^n}{(\Delta x)^2}$$

$$\frac{\partial^2 u}{\partial y^2} \sim \frac{u_{i,j+1}^n - 2u_{i,j}^n + u_{i,j-1}^n}{(\Delta y)^2}$$

- One can quickly rewrite $\frac{\partial u}{\partial t} = \Delta u$, by using finite differences:

$$\frac{u_{i,j}^{n+1} - u_{i,j}^{n}}{\Delta t} = \frac{u_{i+1,j}^{n} - 2u_{i,j}^{n} + u_{i-1,j}^{n}}{(\Delta x)^2} + \frac{u_{i,j+1}^{n} - 2u_{i,j}^{n} + u_{i,j-1}^{n}}{(\Delta y)^2} \tag{9}$$

  Drawing a $(N+1) \times (N+1)$ grid on Ω, and choose the same mesh size for both x and y, that is $\Delta x = \Delta y$, (9) can be simplified to

$$u_{i,j}^{n+1} = \frac{\Delta t}{(\Delta x)^2}(u_{i+1,j}^{n} + u_{i-1,j}^{n} - 4u_{i,j}^{n} + u_{i,j+1}^{n} + u_{i,j-1}^{n}) + u_{i,j}^{n}, \tag{10}$$

- $u_{i,j}^{0}$ is given by the second equation in (8). Using (10) gives us $u_{i,j}^{1}$, then (10) gives $u_{i,j}^{2}$ and so on.
  Finally, (8) can be rewritten numerically as the following:

$$\begin{cases} u_{i,j}^{n+1} = \frac{\Delta t}{(\Delta x)^2}(u_{i+1,j}^{n} + u_{i-1,j}^{n} - 4u_{i,j}^{n} + u_{i,j+1}^{n} + u_{i,j-1}^{n}) + u_{i,j}^{n} \\ u_{i,j}^{0} = F_{i,j} \quad \text{for } i, j = 0, 1, 2, ...N \\ u_{i,j}^{n} = f_{i,j} \quad \text{for } i, j = 0, N, \end{cases} \tag{11}$$

4.1.2 Numerical Experiment

In all numerical simulations, we take as the domain Ω is the square with boundary values given. That is, Ω is set as $[-0.5, 0.5] \times [-0.5, 0.5]$ and mesh points are placed on $(N+1) \times (N+1)$ gridding points on it. As described in equation (11), $u_{i,j}^{0}$ is given on the whole domain, and the boundary value is fixed. In the following tables, variable k is the number of iterations, Δx is the mesh size of chosen domain, $\frac{\Delta t}{(\Delta x)^2}$ is the step size of each iteration. In all numerical simulation, step size is 0.25. The errors shown are the investigation of accuracy compared to the original harmonic function on Ω.

(1) $v(x, y) = x^2 - y^2 + xy$

Table 1: Experiment Report for Function (1)

k	deltx	Max	Mean	Var
2500	0.025	3.594×10^{-15}	1.383×10^{-15}	1.162×10^{-30}
5000	0.025	4.718×10^{-16}	1.629×10^{-16}	1.718×10^{-32}
2500	0.0125	4.583×10^{-5}	1.812×10^{-5}	1.846×10^{-10}
5000	0.0125	2.000×10^{-8}	7.906×10^{-9}	3.510×10^{-17}

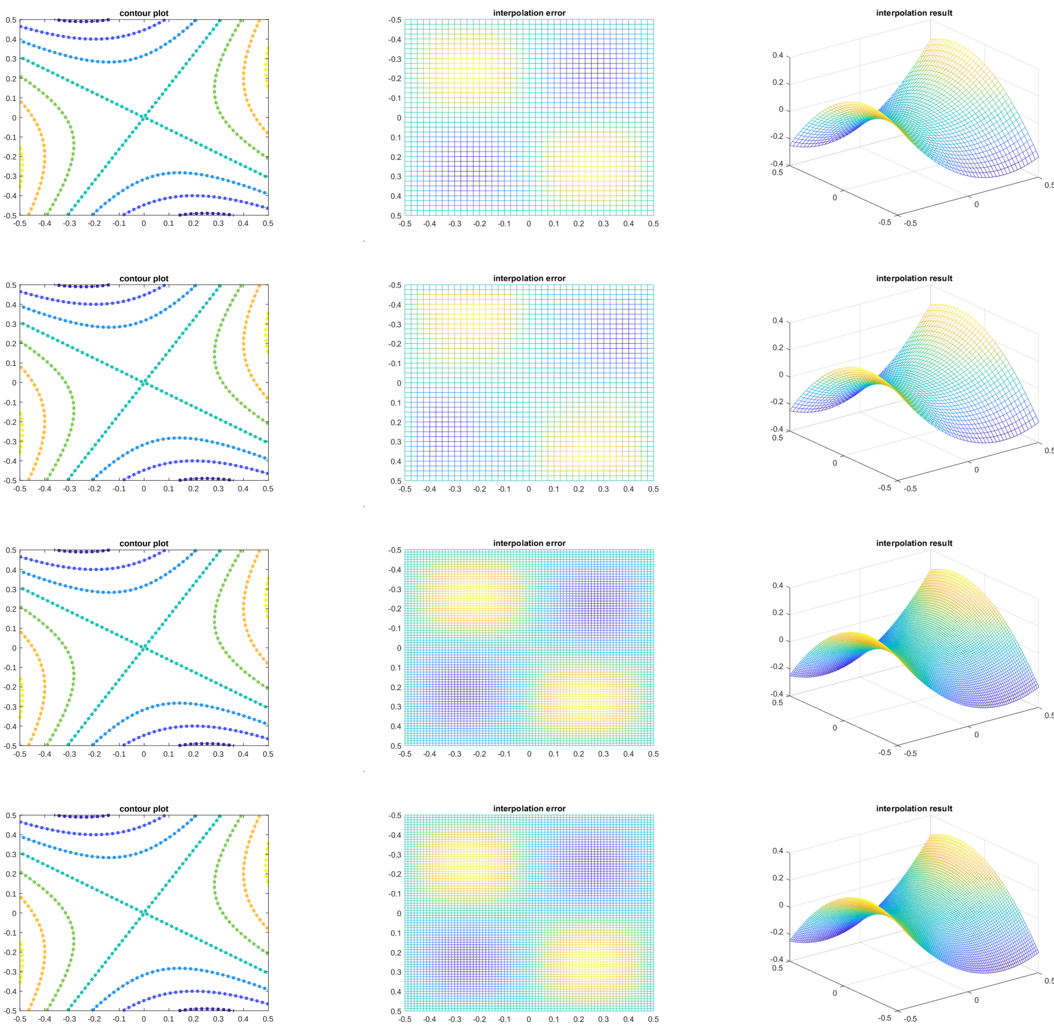
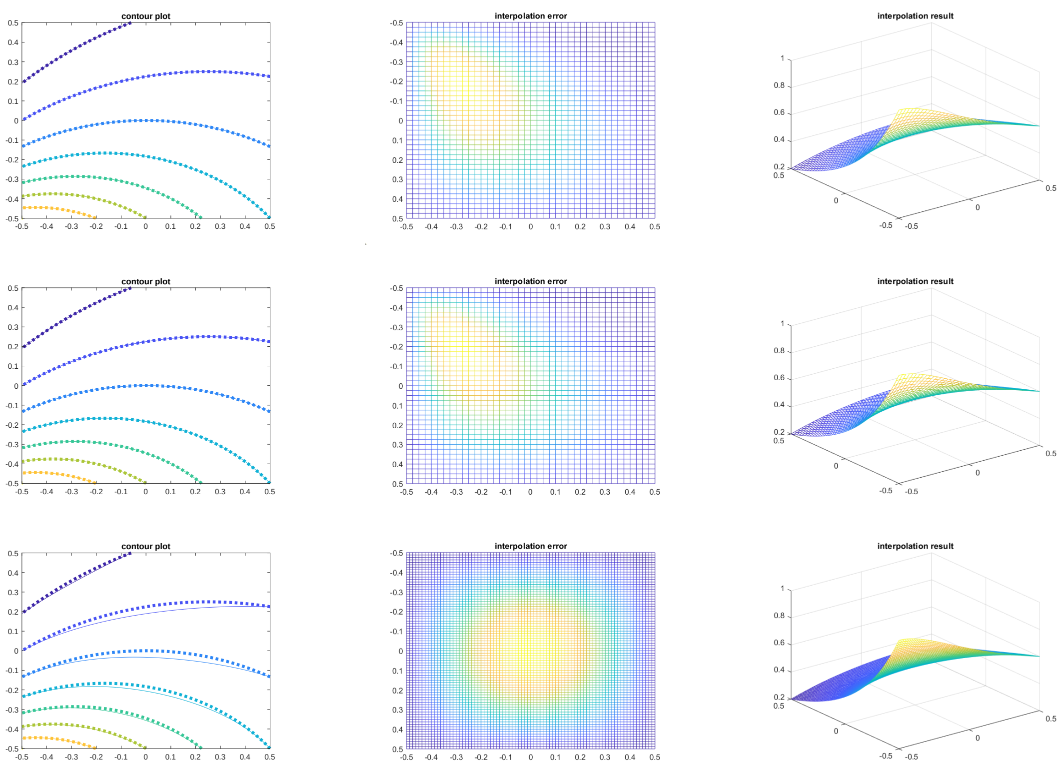
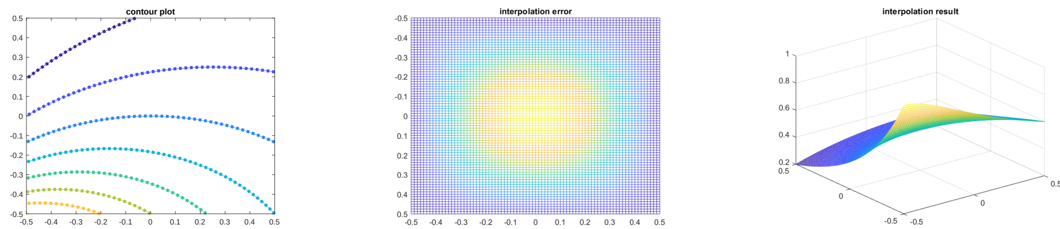

contour plot
interpolation error
interpolation result
contour plot
interpolation error
interpolation result
contour plot
interpolation error
interpolation result
contour plot
interpolation error
interpolation result

Table 2: Experiment Report for Function (2)

k	deltx	Max	Mean	Var
5000	0.025	3.594×10^{-5}	1.160×10^{-5}	1.170×10^{-10}
10000	0.025	2.785×10^{-5}	1.153×10^{-5}	1.163×10^{-10}
5000	0.0125	0.072	0.0068	2.580×10^{-5}
10000	0.0125	3.587×10^{-4}	1.463×10^{-4}	1.191×10^{-8}

(2) $v(x,y) = \frac{x+1}{(x+1)^2+(y+1)^2}$

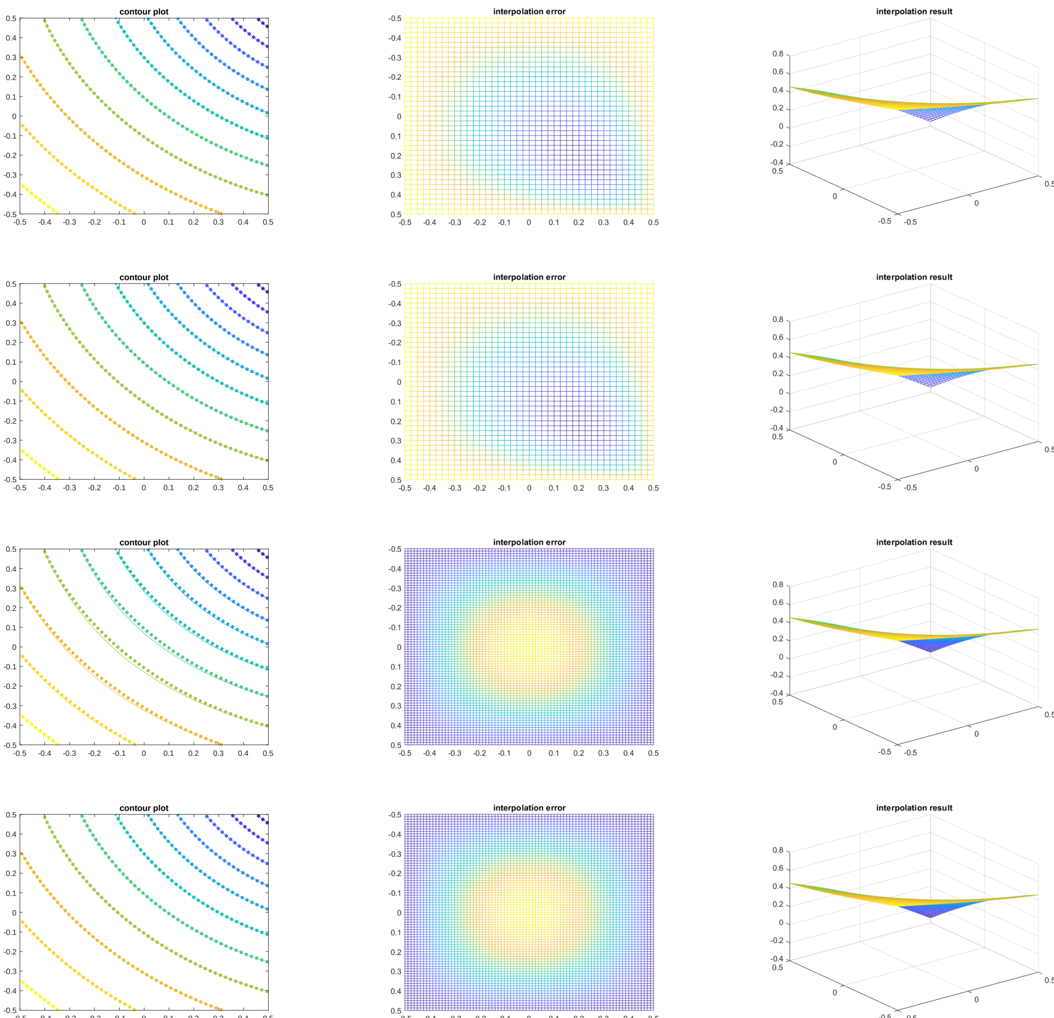




(3) $v(x,y) = \log(\sqrt{(x+1)^2 + (y-1)^2})$

Table 3: Experiment Report for Function (3)

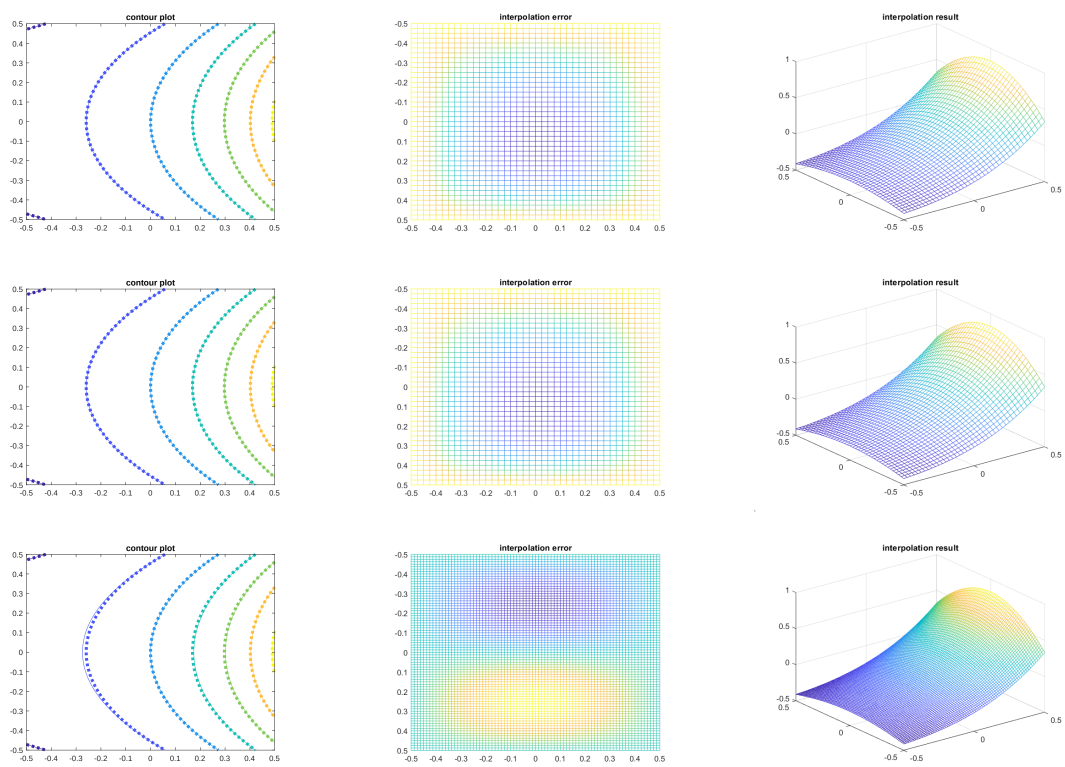
k	deltx	Max	Mean	Var
5000	0.025	1.454×10^{-5}	5.425×10^{-6}	1.813×10^{-11}
10000	0.025	1.461×10^{-5}	5.467×10^{-6}	1.836×10^{-11}
5000	0.0125	0.0119	0.0047	1.234×10^{-5}
10000	0.0125	2.481×10^{-4}	9.780×10^{-5}	5.396×10^{-9}

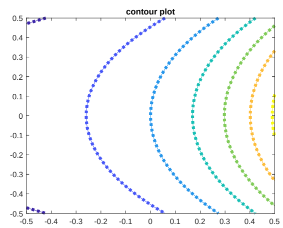

contour plot
interpolation error
interpolation result
contour plot
interpolation error
interpolation result
contour plot
interpolation error
interpolation result
contour plot
interpolation error
interpolation result

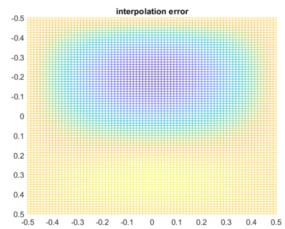
(4) $v(x, y) = e^x(x \cos y - y \sin y)$

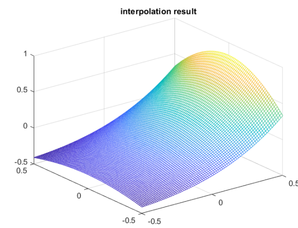
Table 4: Experiment Report for Function (4)

k	deltx	Max	Mean	Var
2000	0.025	3.342×10^{-5}	1.466×10^{-5}	1.031×10^{-10}
5000	0.025	3.156×10^{-5}	1.390×10^{-5}	9.203×10^{-11}
2000	0.0125	0.0085	0.0033	6.134×10^{-6}
5000	0.0125	4.682×10^{-5}	1.303×10^{-5}	1.837×10^{-10}









4.2 Scattered Data are given on whole domain

4.2.1 Method

Sometimes, data are not only given on boundaries, but can be given on the whole domain, and not distributed well. Here we use the Boundary Element Method to find the Harmonic function. The theorem (3) proved in the previous section tells us a harmonic function $u \in C^2(\Omega) \cup C^1(\overline{\Omega})$ can be described by the boundary integral equation:

$$\alpha(x)u(x) = \int \Phi_x \frac{\partial u}{\partial n} - u\frac{\partial \Phi_x}{\partial n} dS, \tag{12}$$

where $\alpha(x) = 1$ for $x \in \Omega$ and $\alpha(x) = \frac{1}{2}$ for $x \in \partial\Omega$.

Now let's replace $\partial\Omega$ by some straight line segments $\Gamma_1, \Gamma_2, ...\Gamma_n$. Also, let $y_1, y_2, ...y_n \in \mathbb{R}^2$ be the middle points of $\Gamma_1, \Gamma_2, ...\Gamma_n$. Therefore the approximation of (12) is:

$$\begin{aligned} \alpha(x)u(x) &= \int \Phi_x \frac{\partial u}{\partial n} - u\frac{\partial \Phi_x}{\partial n} dS \\ &\approx \int_{\Gamma_1 \cup \Gamma_2, ... \cup \Gamma_n} \Phi_x \frac{\partial u}{\partial n} - u\frac{\partial \Phi_x}{\partial n} dS \\ &= \sum_{j=1}^{n} \int_{\Gamma_j} \Phi_x \frac{\partial u}{\partial n} - \sum_{j=1}^{n} \int_{\Gamma_j} u \frac{\partial \Phi_x}{\partial n} \end{aligned} \tag{13}$$

If we use the value of $u(y_i)$ to represent the value of u on each Γ_i, then (13) can be approximated as:

$$\sum_{j=1}^{n} \frac{\partial u(y_j)}{\partial n} \int_{\Gamma_j} \Phi_{x_i} dS - \sum_{j=1}^{n} u(y_j) \int_{\Gamma_j} \frac{\partial \Phi_x}{\partial n}$$

Hence if we know the value of $u(y_j)$ and value of $\frac{\partial u(y_j)}{\partial n}$ for $i = 1, 2, ...n$ then for each $x \in \overline{\Omega}$ the value of $u(x)$ can be approximated.

Now for given data $x_1, x_2, ...x_n \in \mathbb{R}^2$, and $v_1, v_2, ...v_n \in \mathbb{R}$, to find $u \in C^2(\mathbb{R}^2)$ such that $\Delta u = 0, u(x_i) = v_i$ for all i, we do the following:

- Enclose $x_1, x_2, ...x_n$ by a rectangle Ω.
- Discretize $\partial\Omega$ into straight line segments $\Gamma_1, \Gamma_2, ...\Gamma_n$ and define y_j be the midpoints of Γ_j.
- For each $x_i \in R$ we consider:

$$v_i = \sum_{j=1}^{n} \frac{\partial u}{\partial n}(y_j) \int_{\Gamma_j} \Phi_{x_i} dS - \sum_{j=1}^{n} u(y_j) \int_{\Gamma_j} \frac{\partial \Phi_{x_i}}{\partial n} dS,$$

and

$$\frac{1}{2} u(y_i) = \sum_{j=1}^{n} \frac{\partial u}{\partial n}(y_j) \int_{\Gamma_j} \Phi_{x_i} dS - \sum_{j=1}^{n} u(y_j) \int_{\Gamma_j} \frac{\partial \Phi_{x_i}}{\partial n} dS,$$

In terms of matrix multiplication we have:

$$\left(\int_{\Gamma_j} \Phi_{x_i} dS\right) \begin{pmatrix} \frac{\partial u}{\partial n}(y_1) \\ \vdots \\ \frac{\partial u}{\partial n}(y_n) \end{pmatrix} - \left(\int_{\Gamma_j} \frac{\partial \Phi_{x_i}}{\partial n} dS\right) \begin{pmatrix} u(y_1) \\ \vdots \\ u(y_n) \end{pmatrix} = \begin{pmatrix} v(1) \\ \vdots \\ v(n) \end{pmatrix} \tag{14}$$

and,

$$\left(\int_{\Gamma_j} \Phi_{y_i} dS\right) \begin{pmatrix} \frac{\partial u}{\partial n}(y_1) \\ \vdots \\ \frac{\partial u}{\partial n}(y_n) \end{pmatrix} - \left(\int_{\Gamma_j} \frac{\partial \Phi_{y_i}}{\partial n} dS + \frac{1}{2}\delta_{i,j}\right) \begin{pmatrix} u(y_1) \\ \vdots \\ u(y_n) \end{pmatrix} = 0, \tag{15}$$

where

$$\delta_{i,j} = \begin{cases} 1 & if \quad i = j \\ 0 & if \quad i \neq j \end{cases}.$$

Let's combine (14) and (15), that is

$$\left(\begin{array}{c|c} \int_{\Gamma_j} \Phi_{x_i} dS & -\int_{\Gamma_j} \frac{\partial \Phi_{x_i}}{\partial n} dS \\ \hline \int_{\Gamma_j} \Phi_{y_i} dS & -\int_{\Gamma_j} \frac{\partial \Phi_{y_i}}{\partial n} dS - \frac{1}{2}\delta_{i,j} \end{array} \right) \begin{pmatrix} \frac{\partial u}{\partial n}(y_1) \\ \vdots \\ \frac{\partial u}{\partial n}(y_n) \\ \hline u(y_1) \\ \vdots \\ u(y_n) \end{pmatrix} = \begin{pmatrix} v_1 \\ \vdots \\ v_n \\ \hline 0 \\ \vdots \\ 0 \end{pmatrix} \tag{16}$$

As long as $u(y_1), ...u(y_n)$ and $\frac{\partial u}{\partial n}(y_1), \frac{\partial u}{\partial n}(y_2), ...\frac{\partial u}{\partial n}(y_n)$ are known, for each $x_i \in R$, $u(x_i)$ can be computed by

$$\alpha_{x_i} u(x_i) = \left(\begin{array}{c|c} \int_{\Gamma_j} \Phi_{x_i} dS & -\int_{\Gamma_j} \frac{\partial \Phi_{x_i}}{\partial n} dS \end{array} \right) \begin{pmatrix} \frac{\partial u}{\partial n}(y_1) \\ \hline u(y_n) \end{pmatrix} \tag{17}$$

- If the solution of (17) is not unique, more constraints are needed here. Remember we proved that lowest energy is attained if it is a harmonic function. Let's say $I(u)$ is energy of the steady field. Hence consider

$$I(u) = \int_R |\nabla u|^2 = \int_{\partial R} u \frac{\partial u}{\partial n} dS,$$

also,

$$\int_{\partial R} \frac{\partial u}{\partial n} = 0.$$

Discretizing these two equations, we get

$$I(u) = \sum_{j=1}^{n} u(y_j) \frac{\partial u}{\partial n}(y_j) l(\Gamma_j)$$

and,

$$\sum_{j=1}^{n} \frac{\partial u}{\partial n}(y_j) l(\Gamma_j) = 0$$

where $l(\Gamma_i)$ is the length of the i-th segment. Hence, we try to solve

$$\min\{\sum_{j=1}^{n} u(y_j)\frac{\partial u}{\partial n}(y_j)l(\Gamma_j)\}$$

with,

$$\begin{cases} \left(\begin{array}{c|c} \int_{\Gamma_j} \Phi_{x_i} dS & -\int_{\Gamma_j} \frac{\partial \Phi_{x_i}}{\partial n} dS \\ \hline \int_{\Gamma_j} \Phi_{y_i} dS & -\int_{\Gamma_j} \frac{\partial \Phi_{y_i}}{\partial n} dS \end{array}\right) \begin{pmatrix} \frac{\partial u}{\partial n}(y_1) \\ \vdots \\ \frac{\partial u}{\partial n}(y_1) \\ \hline u(y_1) \\ \vdots \\ u(y_n) \end{pmatrix} = \begin{pmatrix} v_1 \\ \vdots \\ v_n \\ \hline 0 \\ \vdots \\ 0 \end{pmatrix} \\ \sum_{j=1}^{n} \frac{\partial u}{\partial n}(y_j) l(\Gamma_j) = 0. \end{cases} \tag{18}$$

Notice that for (18), we do not solve for the exact solution, but a solution such that

$$\left| \left(\int_{\Gamma_j} \Phi_x dS \;\middle|\; -\int_{\Gamma_j} \frac{\partial \Phi_x}{\partial n} dS \right) \begin{pmatrix} \frac{\partial u}{\partial n}(y_1) \\ \hline u(y_1) \end{pmatrix} - \begin{pmatrix} v_1 \\ \vdots \\ v_n \end{pmatrix} \right| < \epsilon \tag{19}$$

for chosen ϵ.

4.2.2 Numerical Experiment

In this part we give the data sets from different harmonic functions and use our method to find the accuracy of proposed scheme. For each simulation, data set $(x_1, v_1), ...(x_n, v_n)$ is given on the $N \times N$ grid on $[-0.5, 0.5] \times [-0.5, 0.5]$. We enclose all the given data points in a bigger rectangular Ω. Then the interpolating result will be compared with the original values of the harmonic function on a 100×100 grid on Ω. The following are the result of the investigation of accuracy compared to the original harmonic function v on Ω.

(1) $v(x, y) = x^2 - y^2 + xy$

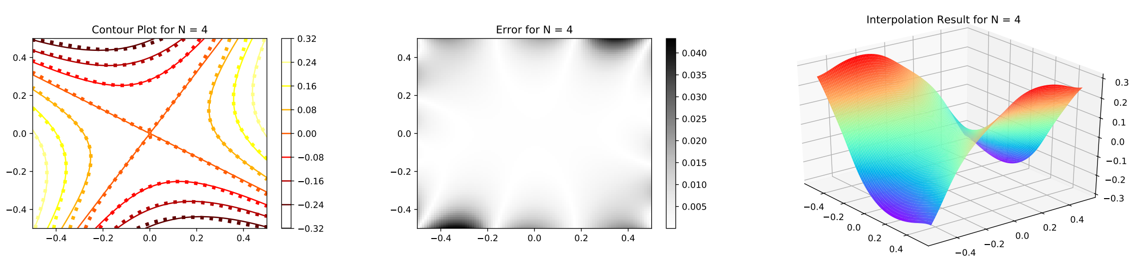
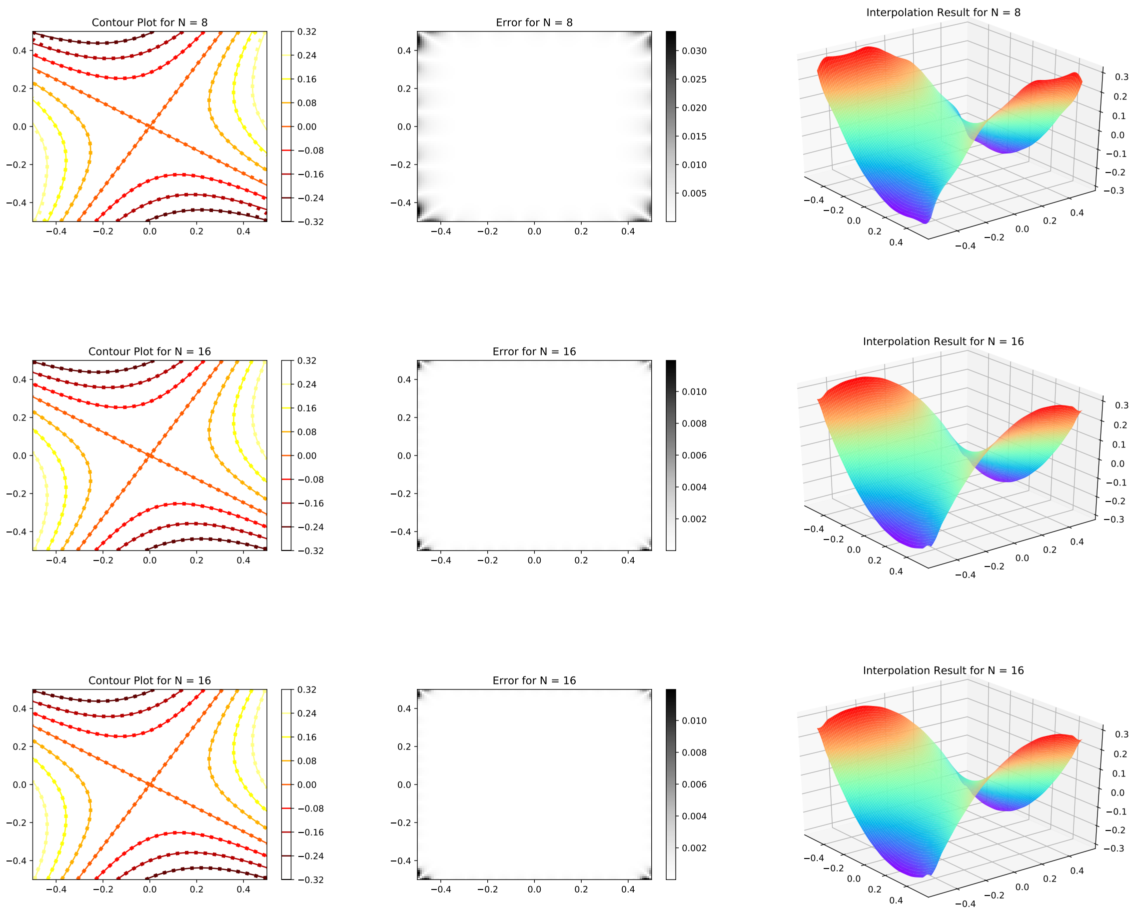
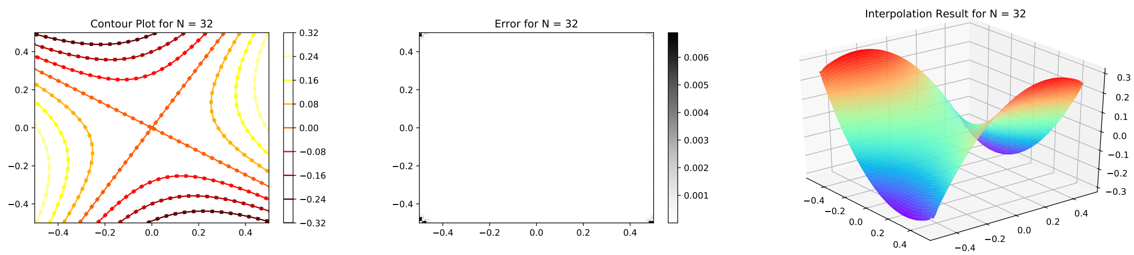


Table 5: Experiment Report for Function (1)

N	4	8	16	32
Max of Error	4.328×10^{-2}	3.337×10^{-2}	1.195×10^{-2}	6.891×10^{-3}
Min of Error	5.126×10^{-7}	6.251×10^{-10}	1.055×10^{-12}	5.557×10^{-14}
Mean of Error	4.514×10^{-3}	1.062×10^{-3}	1.110×10^{-4}	2.568×10^{-5}
Variance of Error	3.803×10^{-5}	8.667×10^{-6}	4.561×10^{-7}	7.194×10^{-8}

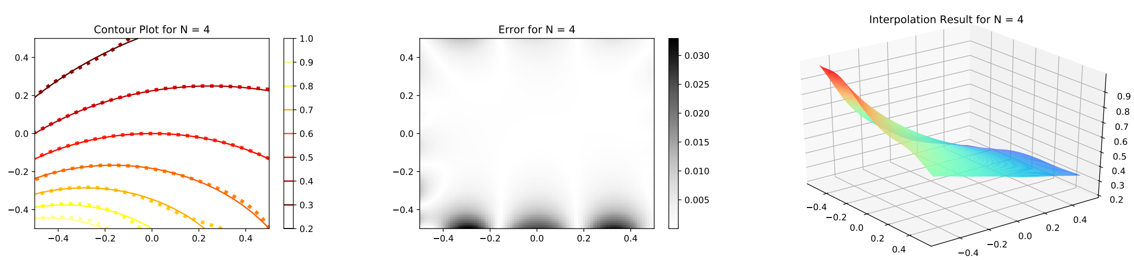


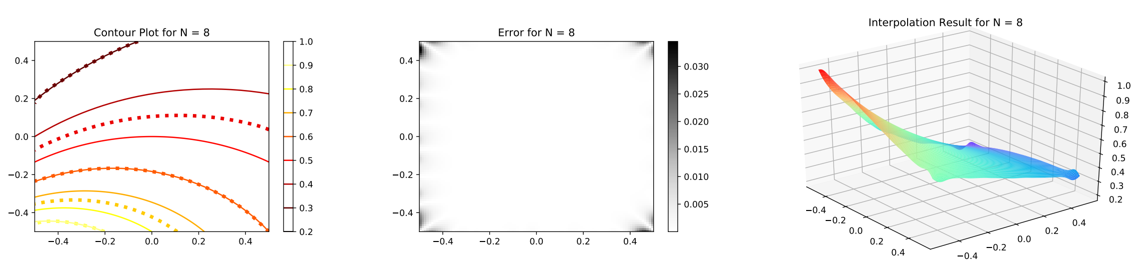


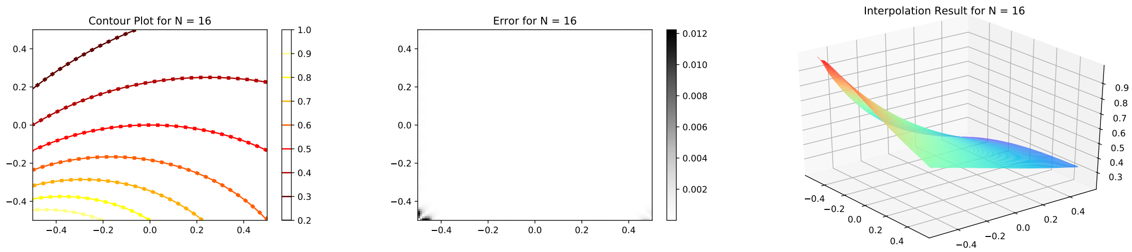
(2) $v(x, y) = \frac{x+1}{(x+1)^2+(y+1)^2}$

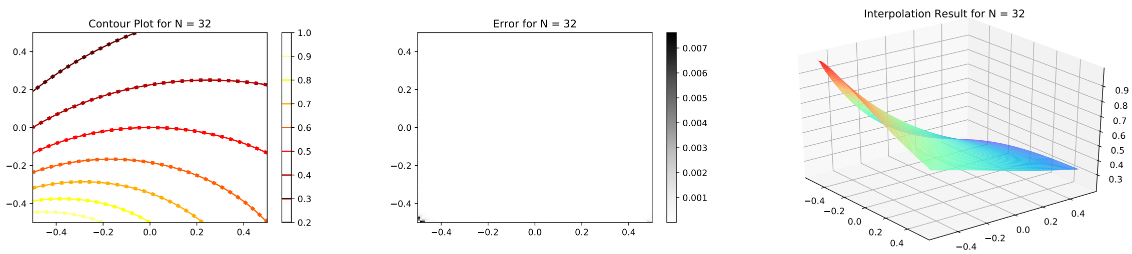
Table 6: Experiment Report for Function (2)

N	4	8	16	32
Max of Error	3.294×10^{-2}	3.453×10^{-2}	1.224×10^{-2}	7.619×10^{-3}
Min of Error	2.491×10^{-7}	1.852×10^{-10}	6.239×10^{-16}	8.882×10^{-16}
Mean of Error	2.574×10^{-3}	6.453×10^{-4}	3.918×10^{-5}	9.288×10^{-6}
Variance of Error	1.851×10^{-5}	5.781×10^{-6}	1.280×10^{-8}	2.010×10^{-8}





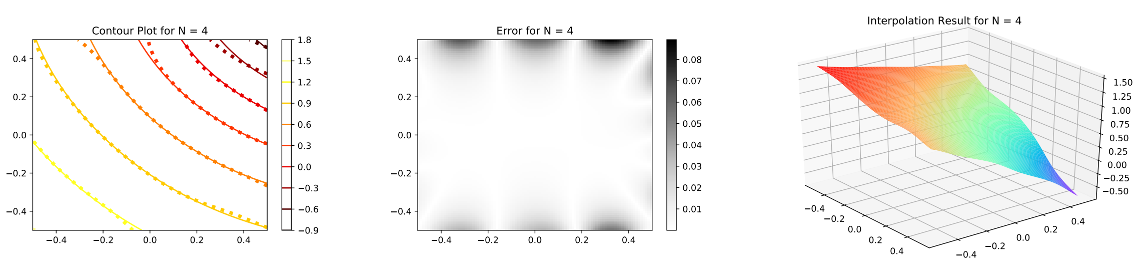


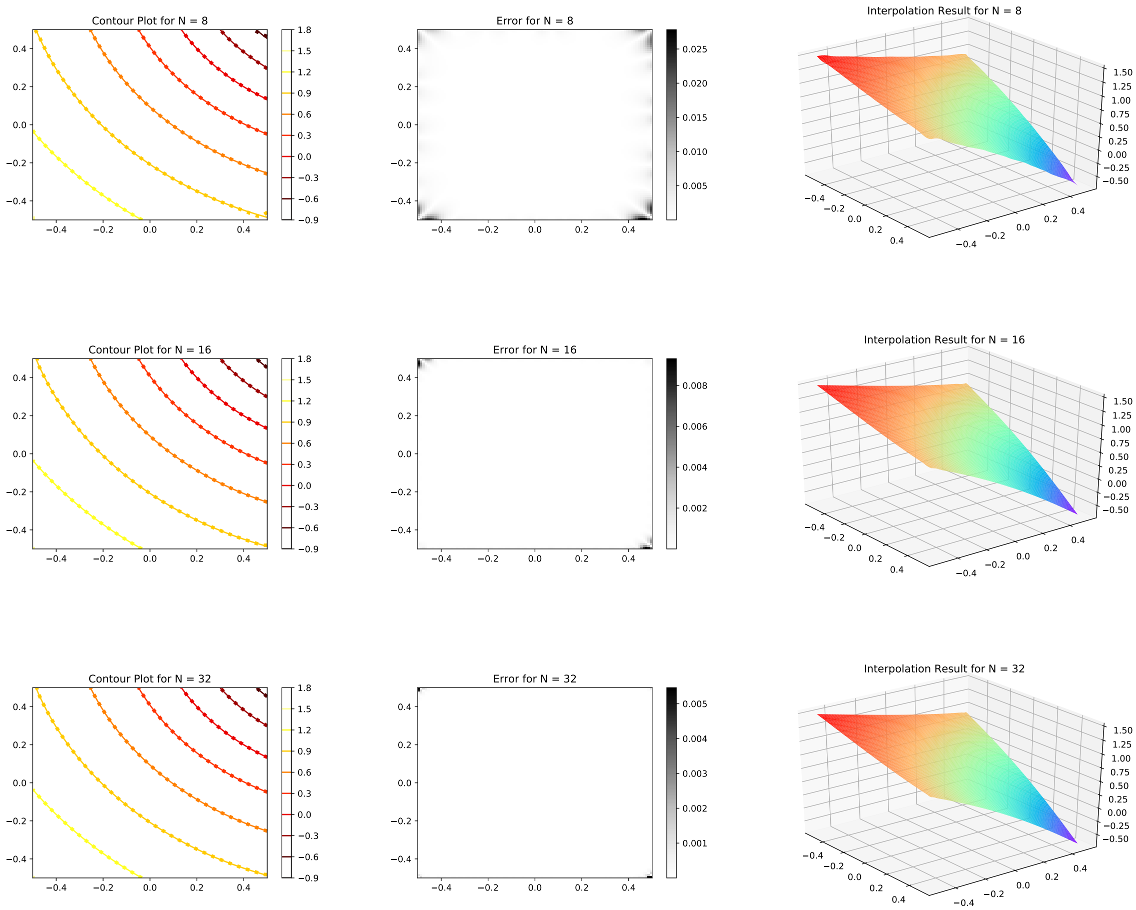


(3) $v(x,y) = ln[(x-1)^2 + (y-1)^2]$

Table 7: Experiment Report for Function (3)

N	4	8	16	32
Max of Error	9.915×10^{-2}	3.406×10^{-2}	9.309×10^{-3}	1.1219×10^{-2}
Min of Error	8.743×10^{-7}	1.004×10^{-11}	2.487×10^{-14}	5.863×10^{-13}
Mean of Error	7.969×10^{-3}	5.699×10^{-4}	4.437×10^{-5}	2.553×10^{-5}
Variance of Error	1.568×10^{-4}	4.134×10^{-6}	1.423×10^{-7}	1.1216×10^{-7}

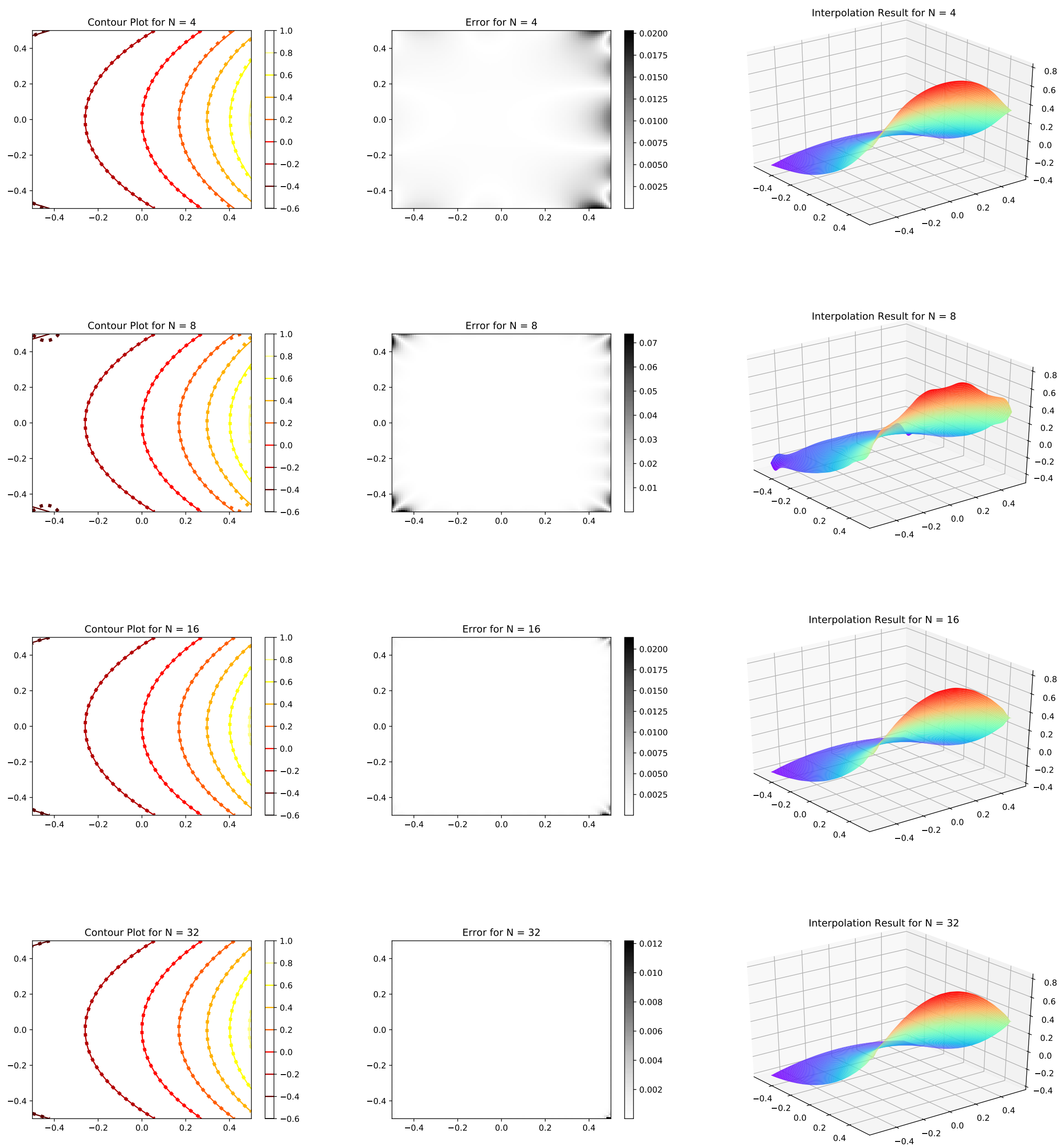




(4) $v(x, y) = e^x(x \cos y - y \sin y)$

Table 8: Experiment Report for Function (4)

N	4	8	16	32
Max of Error	2.031×10^{-2}	7.366×10^{-2}	2.140×10^{-2}	1.1219×10^{-2}
Min of Error	6.135×10^{-8}	3.293×10^{-9}	1.727×10^{-12}	5.863×10^{-13}
Mean of Error	1.577×10^{-3}	1.849×10^{-3}	1.111×10^{-4}	2.442×10^{-5}
Variance of Error	5.436×10^{-6}	4.014×10^{-5}	7.792×10^{-7}	1.216×10^{-7}



5 Conclusion

This paper focuses on the interpolation method for given data points on $\mathbb{R}^2$. We state that traditional interpolation methods usually do not consider the physical meaning of the interpolating function. Also consider real life problems like weather forecasting, in which data may not be distributed well, we found that using harmonic function as an interpolating function is a reasonable strategy. Two different interpolation methods are discussed here. The first method deals with the situation where data are given on the boundary. It uses the heat equation to approach the steady state on the domain. The experimental result indicated that after enough iterations, the error between the approximated result and the original function could be relatively small. In this method, data are given on the boundary. The second method is a convolution based method, called the boundary element method. In this situation, data are give on the whole domain but not distributed well. The second experiment shows that the approximated data points fit the original harmonic function well even when fewer data points are given. Some traditional interpolation methods in one and two dimensions are summarized in the previous section. There are two main reasons why we consider harmonic function as interpolation functions:

1. Some interpolation methods do not consider the physical meaning of data points.

2. Some interpolation methods cannot be used for scattered data points or data points given only on the boundary.

Hence, a smooth function, harmonic function defined on a open connected domain Ω are considered here. Some properties of the harmonic function are introduced in the third section. Theorem 3.2 can be showed easily based on Divergence Theorem. Then, using Theorem 3.2, we proved an important property of harmonic functions. Theorem 3.3 states that the harmonic function on a bounded open connected domain is the minimizer of the Dirichlet functional. The uniqueness theorem can be proved directly by using Theorem 3.4. These properties are the motivations and reasons for using harmonic function to interpolate the given data. To use the boundary element method, we introduced the fundamental solution of Laplace equation and Theorem 3.6. Theorem 3.6 tells us as that long as $u \in C^2(\Omega) \cup C^1(\overline{\Omega})$, then the values of u inside the domain on the boundary of domain can be given by a line integral on the boundary related to its fundamental solution. Two numerical methods are introduced step by step in the fourth section. Experiments are done by Matlab and Python. Both numerical experiment try to investigate the accuracy between the approximated values and the values generated by original harmonic functions. The results demonstrated in the fourth section reveal that both methods works if only boundary data or scatted data are given.

In the future, more work will be done on the application of harmonic function interpolation. There are some different types of interpolation used in different subjects. For example, the method could be used in weather forecasting. In recent years, more interdisciplinary methods have been applied to the study of interpolation methods. In [15], a neural network methods such as ANN(artificial neural network) was used in interpolation models to predict rainfall. In our future work, one of the topics should be how to create the interpolation model combining our method with different statistical techniques in different areas.

References

[1] Garrett Birkhoff. Interpolation to boundary data in triangles. *Journal of Mathematical Analysis and Applications*, 42(2):474–484, May 1973. ISSN 0022-247X. doi: 10.1016/0022-247X(73)90154-6. URL `http://www.sciencedirect.com/science/article/pii/0022247X73901546`.

[2] Gungahlin Weather Centre. Weather station to be built on Mt. Everest! URL `http://www.gungahlinweather.com/articles/everest.htm`.

[3] Lawrence C. Evans. *Partial Differential Equations*. American Mathematical Soc., 2010. ISBN 978-0-8218-4974-3. Google-Books-ID: Xnu0o_EJrCQC.

[4] Richard Franke. Scattered data interpolation: tests of some methods. *Mathematics of Computation*, 38(157):181–181, January 1982. ISSN 0025-5718. doi: 10.1090/S0025-5718-1982-0637296-4. URL `http://www.ams.org/jourcgi/jour-getitem?pii=S0025-5718-1982-0637296-4`.

[5] Yukio Ôhashi. ASTRONOMY AND MATHEMATICS OF YIXING. page 8.

[6] XueQing Jiang. Rouble's woes reap mixed harvest[1]- Chinadaily.com.cn. URL `http://www.chinadaily.com.cn/business/2015-04/13/content_20416109.htm`.

[7] A. D. Maude. Interpolation—mainly for graph plotters. *The Computer Journal*, 16(1):64–65, January 1973. ISSN 0010-4620. doi: 10.1093/comjnl/16.1.64. URL `https://academic.oup.com/comjnl/article/16/1/64/435132`.

[8] D. H. McLain. Two Dimensional Interpolation from Random Data. *The Computer Journal*, 19(2):178–181, May 1976. ISSN 0010-4620, 1460-2067. doi: 10.1093/comjnl/19.2.178. URL `https://academic.oup.com/comjnl/article-lookup/doi/10.1093/comjnl/19.2.178`.

[9] E. Meijering. A chronology of interpolation: from ancient astronomy to modern signal and image processing. *Proceedings of the IEEE*, 90(3):319–342, March 2002. ISSN 00189219. doi: 10.1109/5.993400. URL `http://ieeexplore.ieee.org/document/993400/`.

[10] Walter Rudin. *Principles of Mathematical Analysis*. McGraw-Hill, 1976. ISBN 978-0-07-085613-4. Google-Books-ID: kwqzPAAACAAJ.

[11] Timothy Sauer. *Numerical Analysis*. Addison-Wesley Publishing Company, USA, 2nd edition, 2011. ISBN 978-0-321-78367-7.

[12] Stefan Sauter and Christoph Schwab. *Boundary Element Methods*. Springer Series in Computational Mathematics. Springer-Verlag, Berlin Heidelberg, 2011. ISBN 978-3-540-68092-5. URL `https://www.springer.com/us/book/9783540680925`.

[13] Donald Shepard. A two-dimensional interpolation function for irregularly-spaced data. In *Proceedings of the 1968 23rd ACM national conference on -*, pages 517–524, Not Known, 1968. ACM Press. doi: 10.1145/800186.810616. URL `http://portal.acm.org/citation.cfm?doid=800186.810616`.

[14] Walter A. Strauss. *Partial Differential Equations: An Introduction*. John Wiley & Sons, December 2007. ISBN 978-0-470-05456-7. Google-Books-ID: m2hvDwAAQBAJ.

[15] K. W. Wong, P. M. Wong, T. D. Gedeon, and C. C. Fung. Rainfall prediction model using soft computing technique. *Soft Computing - A Fusion of Foundations, Methodologies and Applications*, 7(6):434–438, May 2003. ISSN 1432-7643, 1433-7479. doi: 10.1007/s00500-002-0232-4. URL `http://link.springer.com/10.1007/s00500-002-0232-4`.