



JOINT INSTITUTE
交大密西根学院

VV557 Methods of Applied Mathematics II

Term Project:

A Short Introduction to Boundary Element Method

Written by

Zhou Zhanpeng 518021910594

Instructed by

Dr. Horst Hohberger

May 3, 2020

University of Michigan - Shanghai Jiao Tong University Joint Institute
(UM-SJTU JI)
800 Dongchuan Road, Shanghai, China 200240

Abstract

After nearly 40 years of research and development, the boundary element method has become an accurate and efficient engineering numerical analysis method to solve boundary value problem in such fields as mathematics and physics. Aiming to give a short introduction to BEM, in this paper, we will explore the internal principle and performance of boundary element method by solving a simple Dirichlet problem, discuss boundary element method using Green function, and then investigate further on BEM applied mixed boundary value problem.

Contents

1	Introduction	2
2	Dirichlet Problem for the Laplace Equation On The Half-disk	2
3	The Analytic Solution to the Dirichlet Problem	2
4	Boundary Element Methods	4
4.1	Boundary Integral solution	4
4.2	Boundary Element Method	6
4.3	Boundary Element Solution of Pre-defined Problem and Comparison with Analytic Solution	8
5	Boundary Element Method Using Green Function	11
5.1	Boundary Element Solution Using Green Function	11
5.2	Comparison with Analytic Solution	13
5.3	Comparison with Original Boundary Element Solution	15
6	Boundary Element Solution With Mixed Boundary Conditions	15
6.1	Mixed Boundary Value Problem for the Laplace Equation On The Half-disk .	15
6.2	Boundary Element Solution to Mixed Boundary Value Problem	16
7	Conclusion and Discussion	17
7.1	Conclusion	17
7.2	Further Discussion	18
8	Works Cited	18
9	Appendix	18

1 Introduction

The boundary element method is a relatively new numerical method developed after the finite element method. With nearly 40 years of research and development, the boundary element method has become an accurate and efficient engineering numerical analysis method in such fields as mathematics and physics. Different from the basic idea of finite element method to divide elements in the continuum domain, boundary element method only divides elements on the boundary of the domain and approximates the boundary condition with the function satisfying the governing equation. Therefore, compared with the finite element method, BEM has the advantages of fewer elements and simpler calculations.

In this paper, we are aimed to give a short introduction to BEM. Therefore, in the following sections, we will first explore the internal principle and performance of BEM by solving a simple Dirichlet problem, then discuss the performance of BEM using Green function, and finally investigate further on BEM applied to mixed boundary value problem.

2 Dirichlet Problem for the Laplace Equation On The Half-disk

To explain the principle of BEM more clearly, we will focus on a simple Dirichlet problem governed by the two-dimensional Laplace's equation on the half-disk in the following several sections. Therefore, it is convenient for us to define the problem in advance.

Thus, we set

$$\Omega = \{(x_1, x_2) \in R^2 : x_1^2 + x_2^2 < 1, x_2 > 0\} \quad (1)$$

Also,

$$\begin{aligned} -\Delta u &= 0 \text{ on } \Omega & u(x_1, x_2)|_{x_1^2+x_2^2=1} &= 1 \\ u(x_1, 0) &= 0 \text{ for } -1 < x_1 < 1 & u(x_1, 0) &= 0 \end{aligned} \quad (2)$$

3 The Analytic Solution to the Dirichlet Problem

Before we really get started with BEM, we need to have a basic idea of the solution to the Dirichlet problem. Therefore, we will first use method of images[1] to find the green function and then find the analytic solution to the Dirichlet problem.

First, according to method of images[1], we could find our images as shown in figure 1.

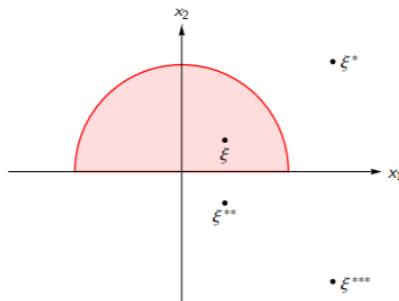


Figure 1: Images of the half-disk problem[1]

Supposing $\xi = (\xi_1, \xi_2) \in \Omega$, then

$$\begin{aligned}\xi^* &= \frac{\xi}{|\xi|^2} = \frac{1}{|\xi|^2}(\xi_1, \xi_2) \\ \xi^{**} &= (\xi_1, -\xi_2) \\ \xi^{***} &= \frac{1}{|\xi|^2}(\xi_1, -\xi_2)\end{aligned}$$

Also, it is known that the fundamental solution to $-\Delta E = \delta(x - \xi)$ is

$$E(x; \xi) = -\frac{1}{2\pi} \ln |x - \xi| \quad [1]$$

Thus, it is not hard for us to construct the green function of this problem

$$\begin{aligned}g(x; \xi) &= E(x; \xi) - E(x; \xi^*) - E(x; \xi^{**}) + E(x; \xi^{***}) \\ &= -\frac{1}{2\pi} \ln |x - \xi| + \frac{1}{2\pi} \ln |x - \xi^*| + \frac{1}{2\pi} \ln |x - \xi^{**}| - \frac{1}{2\pi} \ln |x - \xi^{***}|\end{aligned}$$

which obviously satisfy the boundary conditions that

$$g(x, y)|_{x_1^2 + x_2^2} = 0 \quad g(x_1, 0) = 0 \text{ for } -1 < x_1 < 1$$

Then, based on the solution formula for Elliptic problem[1], the solution formula for our specific problem can be found as

$$u(\varrho, \vartheta) = - \int_0^\pi \frac{\partial}{\partial r} g(r, \theta; \varrho, \vartheta)|_{r=1} d\theta \quad [1]$$

However, due to our solution formula is defined in polar coordinates, it is necessary to transform the green function into polar coordinates

$$\begin{aligned}\xi &= (\xi_1, \xi_2) = (\varrho \cos \vartheta, \varrho \sin \vartheta) \\ x &= (x_1, x_2) = (r \cos \theta, r \sin \theta)\end{aligned}$$

Now, we can calculate $\frac{\partial g}{\partial r}$. For example,

$$\frac{1}{2\pi} \ln |x - \xi| = \frac{1}{4\pi} \ln((r \cos \theta - \varrho \cos \vartheta)^2 + (r \sin \theta - \varrho \sin \vartheta)^2)$$

Then

$$\begin{aligned}\frac{\partial \frac{1}{2\pi} \ln |x - \xi|}{\partial r} &= \frac{\partial \frac{1}{4\pi} \ln((r \cos \theta - \varrho \cos \vartheta)^2 + (r \sin \theta - \varrho \sin \vartheta)^2)}{\partial r} \\ &= \frac{1}{2\pi} \frac{r - \varrho(\cos(\theta - \vartheta))}{r^2 - 2\varrho r(\cos(\theta - \vartheta)) + \varrho^2}\end{aligned}$$

Similarly, it is easy for us to find

$$\frac{\partial \frac{1}{2\pi} \ln |x - \xi^*|}{\partial r} = \frac{1}{2\pi} \frac{\varrho^2 r - \varrho(\cos(\theta - \vartheta))}{\varrho^2 r^2 - 2\varrho r(\cos(\theta - \vartheta)) + 1}$$

And

$$\frac{\partial \frac{1}{2\pi} \ln |x - \xi^{**}|}{\partial r} = \frac{1}{2\pi} \frac{r - \varrho(\cos(\theta + \vartheta))}{r^2 - 2\varrho r(\cos(\theta + \vartheta)) + \varrho^2}$$

Also,

$$\frac{\partial \frac{1}{2\pi} \ln |x - \xi^{***}|}{\partial r} = \frac{1}{2\pi} \frac{\varrho^2 r - \varrho(\cos(\theta + \vartheta))}{\varrho^2 r^2 - 2\varrho r(\cos(\theta + \vartheta)) + 1}$$

Finally, the analytic solution to this problem could be found:

$$\begin{aligned} u(\varrho, \vartheta) &= - \int_0^\pi \frac{\partial}{\partial r} g(r, \theta; \varrho, \vartheta) \Big|_{r=1} d\theta \\ &= \frac{\varrho^2 - 1}{2\pi} \int_0^\pi \left(\frac{1}{\varrho^2 + 1 - 2\varrho \cos(\theta + \vartheta)} - \frac{1}{\varrho^2 + 1 - 2\varrho \cos(\theta - \vartheta)} \right) d\theta \\ &= \left(\frac{1}{\pi} \arctan\left(\frac{1+\varrho}{1-\varrho} \tan \frac{\theta - \vartheta}{2}\right) - \frac{1}{\pi} \arctan\left(\frac{1+\varrho}{1-\varrho} \tan \frac{\theta + \vartheta}{2}\right) \right) \Big|_0^\pi - 1 \\ &= \frac{2}{\pi} \arctan\left(\frac{1+\varrho}{1-\varrho} \tan \frac{\vartheta}{2}\right) + \frac{2}{\pi} \arctan\left(\frac{1+\varrho}{1-\varrho} \cot \frac{\vartheta}{2}\right) - 1 \end{aligned}$$

Just to make the solution to this problem more intuitive, using Mathematica, a scientific computational tool, we could plot the solution on a three-dimensional plane as shown in figure 2.

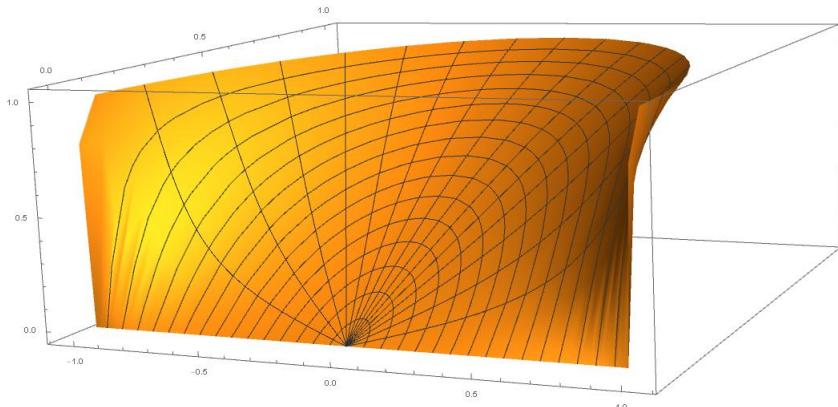


Figure 2: The analytic solution to the Dirichlet problem

4 Boundary Element Methods

However, how does BEM actually work? In this section, we will first start from the derivation of BEM, then apply BEM to the Dirichlet problem which we defined earlier and compare the obtained numerical solution with analytic solution to see how BEM performs.

4.1 Boundary Integral solution

To explain the internal principle of BEM, we should start from the basis of BEM, the boundary integral solution[2].

Supposing ϕ_1 and ϕ_2 are two solution of Eq. (2) in the region Ω , then, it is known that

$$\int_{\partial\Omega} (\phi_2 \frac{\partial\phi_1}{\partial n} - \phi_1 \frac{\partial\phi_2}{\partial n}) ds = 0 \quad (3)$$

which is called reciprocal relation[2] that can be easily derived from divergence theorem.

If we take $\phi_1 = E(x; \xi)$, the fundamental solution to the Eq. (2) and $\phi_2 = u$, where u is the required solution[2] of the boundary value problem. Then, the reciprocal relation[2] can be written as

$$\int_{\partial\Omega} [u(x) \frac{\partial E(x; \xi)}{\partial n} - E(x; \xi) \frac{\partial u(x)}{\partial n}] ds = 0 \quad [2] \quad (4)$$

where $\xi \notin \Omega$, due to $E(x; \xi)$ is not well-defined on the point ξ [2].

However, we can still make ξ in the interior of Ω as long as we modify the boundary as $\partial\Omega \cup \partial\Omega_\varepsilon$ where Ω_ε is a circle of center ξ and radius ε [2]. Thus, we can rewrite Eq. (4) as

$$\int_{\partial\Omega \cup \partial\Omega_\varepsilon} [u(x) \frac{\partial E(x; \xi)}{\partial n} - E(x; \xi) \frac{\partial u(x)}{\partial n}] ds = 0 \quad [2] \quad (5)$$

which can be also written as

$$\int_{\partial\Omega} [u(x) \frac{\partial E(x; \xi)}{\partial n} - E(x; \xi) \frac{\partial u(x)}{\partial n}] ds = - \int_{\partial\Omega_\varepsilon} [u(x) \frac{\partial E(x; \xi)}{\partial n} - E(x; \xi) \frac{\partial u(x)}{\partial n}] ds \quad [2] \quad (6)$$

if we use polar coordinates instead, the coordinates should be defined as

$$\begin{aligned} x_1 - \xi_1 &= r \cos \theta \\ x_2 - \xi_2 &= r \sin \theta \end{aligned} \quad (7)$$

Then, it is easy for us to obtain the fundamental solution and its normal derivative in polar coordinates

$$E(x; \xi) = \frac{1}{2\pi} \ln(r) \quad \frac{\partial E}{\partial n} = \frac{n_{x_1} \cos \theta + n_{x_2} \sin \theta}{2\pi r} \quad [2] \quad (8)$$

Also, we know the Taylor's series of u about the point ξ is

$$u = \sum_{m=0}^{\infty} \sum_{k=0}^m \left(\frac{\partial u}{\partial x_1^k \partial x_2^{m-k}} \right) |_{(x_1, x_2) = (\xi_1, \xi_2)} \frac{(x_1 - \xi_1)^k (x_2 - \xi_2)^{m-k}}{k! (m-k)!} \quad [2]$$

Then, for all the points on the circle $\partial\Omega_\varepsilon$, with polar coordinates, u can be also written as

$$u = \sum_{m=0}^{\infty} \sum_{k=0}^m \left(\frac{\partial u}{\partial x_1^k \partial x_2^{m-k}} \right) |_{(x_1, x_2) = (\xi_1, \xi_2)} \frac{\varepsilon^m \cos^k \theta \sin^{m-k} \theta}{k! (m-k)!} \quad [2] \quad (9)$$

Similarly, we can get

$$\frac{\partial u}{\partial n} = \sum_{m=0}^{\infty} \sum_{k=0}^m \left(\frac{\partial}{\partial x_1^k \partial x_2^{m-k}} \frac{\partial u}{\partial n} \right) |_{(x_1, x_2) = (\xi_1, \xi_2)} \frac{\varepsilon^m \cos^k \theta \sin^{m-k} \theta}{k! (m-k)!} \quad [2] \quad (10)$$

With Eq. (8), Eq. (9), Eq. (10) and knowing $ds = \varepsilon d\theta$ [2], we can then rewrite the right

hand side of Eq. (6) as

$$\begin{aligned} & \int_{\partial\Omega_\varepsilon} u(x) \frac{\partial E(x; \xi)}{\partial n} ds \\ &= -u(\xi) - \frac{1}{2\pi} \sum_{m=1}^{\infty} \sum_{k=0}^m \left(\frac{\partial u}{\partial x_1^k \partial x_2^{m-k}} \right) \Big|_{(x_1, x_2) = (\xi_1, \xi_2)} \frac{\varepsilon^m}{k!(m-k)!} \int_0^{2\pi} \cos^k \theta \sin^{m-k} \theta \theta d\theta \quad [2] \end{aligned} \quad (11)$$

which will be approximately equal to $-u(\xi)$ as $\varepsilon \rightarrow 0^+$ [2].

Also,

$$\begin{aligned} & \int_{\partial\Omega_\varepsilon} E(x) \frac{\partial u(x; \xi)}{\partial n} ds \\ &= \frac{1}{2\pi} \sum_{m=0}^{\infty} \sum_{k=0}^m \left(\frac{\partial}{\partial x_1^k \partial x_2^{m-k}} \left(\frac{\partial u}{\partial n} \right) \right) \Big|_{(x_1, x_2) = (\xi_1, \xi_2)} \frac{\varepsilon^{m+1} \ln(\varepsilon)}{k!(m-k)!} \int_0^{2\pi} \cos^k \theta \sin^{m-k} \theta \theta d\theta \quad [2] \end{aligned} \quad (12)$$

which will be approximately equal to 0 as $\varepsilon \rightarrow 0^+$ [2].

Therefore, as $\varepsilon \rightarrow 0^+$ Eq. (6) yields

$$u(\xi) = \int_{\partial\Omega} [u(x) \frac{\partial E(x; \xi)}{\partial n} - E(x; \xi) \frac{\partial u(x)}{\partial n}] ds \quad [2] \quad (13)$$

However, so far, ξ can be only be the points inside the Ω . If ξ are exactly at the boundary, $\partial\Omega_\varepsilon$ should be modified as a semicircle [2]. Correspondingly, for those ξ at the boundary, our Eq. (13) should be modified as

$$\frac{1}{2}u(\xi) = \int_{\partial\Omega} [u(x) \frac{\partial E(x; \xi)}{\partial n} - E(x; \xi) \frac{\partial u(x)}{\partial n}] ds \quad [2] \quad (14)$$

What's more, for those the points are even not contained in Ω , it is obvious that

$$\int_{\partial\Omega} [u(x) \frac{\partial E(x; \xi)}{\partial n} - E(x; \xi) \frac{\partial u(x)}{\partial n}] ds = 0 \quad [2] \quad (15)$$

Then, for convenience, we combine Eq. (13), Eq. (14), Eq. (15) together as a single equation which is

$$\lambda(\xi)u(\xi) = \int_{\partial\Omega} [u(x) \frac{\partial E(x; \xi)}{\partial n} - E(x; \xi) \frac{\partial u(x)}{\partial n}] ds \quad [2] \quad (16)$$

where $\lambda(\xi)$ is defined as

$$\lambda(\xi) = \begin{cases} 0 & \text{if } \xi \notin \Omega \\ \frac{1}{2} & \text{if } \xi \in \partial\Omega \\ 1 & \text{if } \xi \in \Omega \cap \xi \notin \partial\Omega \end{cases} \quad [2] \quad (17)$$

As shown above, the combined mathematical formula is just the boundary integral solution[2].

4.2 Boundary Element Method

We have shown the basis of boundary element method, the boundary integral solution, in the last section. Now, we will introduce a simple procedure of implementing BEM to show how BEM Works.

Step 1: Discretize Boundary C

First, we should divide the boundary into a few elements. For simplicity, we define $C = \partial\Omega$ [2], then, take arbitrary points on the boundary, as illustrated in the figure 3, $x^k \in C$ where $k = 1, 2, \dots, N$, to divides C into the N small line segments, C^1, C^2, \dots, C^N [2].

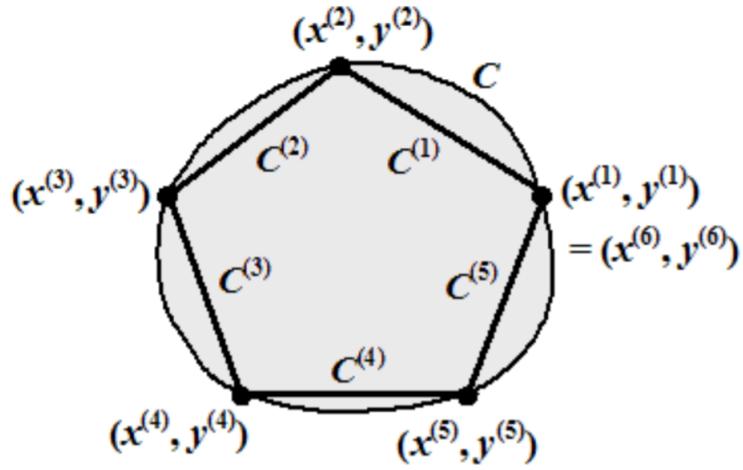


Figure 3: Discretize the boundary C [2]

Based on that, we can approximate the boundary C as the combination of all the line segments. That is

$$C \approx C^1 \cup C^2 \cup \dots \cup C^{N-1} \cup C^N \quad [2] \quad (18)$$

Step 2: Approximate Boundary Integral Solution

Then, we will make an approximation of u and $\frac{\partial u}{\partial n}$ on each line segment C^k . If the line segments are small enough, it is sufficient for us to take the u and $\frac{\partial u}{\partial n}$ at the mid point of the segment as a good approximation to the real value of u and $\frac{\partial u}{\partial n}$ at each point of the line segment[2]. Thus, we use following approximations,

$$u \approx \bar{u}^k \quad \frac{\partial u}{\partial n} \approx \bar{p}^k \text{ for } (x_1, x_2) \in C^k \quad (k = 1, 2, \dots, N) \quad [2] \quad (19)$$

where \bar{u}^k and \bar{p}^k are the values of u and $\frac{\partial u}{\partial n}$ at the midpoint of C^k .

Then based on our approximations, the boundary integral solution[2] can be written as

$$\lambda(\xi)u(\xi) \approx \sum_{k=1}^N (\bar{u}^k F_2^k(\xi) - \bar{p}^k F_1^k(\xi)) \quad [2] \quad (20)$$

where

$$\begin{aligned} F_1^k(\xi) &= \int_{C^k} E(x; \xi) ds \\ F_2^k(\xi) &= \int_{C^k} \frac{\partial}{\partial n} E(x; \xi) ds \end{aligned} \quad [2] \quad (21)$$

Now, we have derived the boundary element solution, which is just Eq. (20) and in most cases, F_1^k and F_2^k are easily calculated. However, we are not done, some unknown constants in the Eq. (20) still need to be found.

Step 3: Calculate Unknown Constants

Due to constrain of the boundary condition, either \bar{u}^k or \bar{p}^k is known, therefore, there are N unknown constants in our boundary element solution. Thus, we should construct a linear system with N independent equations to solve all the N unknown constants.

Here, by substituting the midpoints of C^1, C^2, \dots, C^N into the boundary element solution, we obtain the linear system

$$\frac{1}{2} \bar{u}^m = \sum_{k=1}^m (\bar{u}^k F_2^k(\bar{x}_1^m, \bar{x}_2^m) - \bar{p}^m F_1^k(\bar{x}_1^m, \bar{x}_2^m)) \text{ for } m = 1, 2, \dots, N \quad [2] \quad (22)$$

where $(\bar{x}_1^m, \bar{x}_2^m)$ is the midpoint of C^m .

However, to simplify our calculations, we can also rewrite Eq. (22) as

$$\sum_{k=1}^N a^{mk} z^k = \sum_{k=1}^N b^{mk} \text{ for } m = 1, 2, \dots, N \quad [2] \quad (23)$$

where a^{mk}, b^{mk}, z^k are

$$\begin{aligned} a^{mk} &= \begin{cases} -F_1^k(\bar{x}_1^m, \bar{x}_2^m) & \text{if } u \text{ is specified over } C^k \\ F_2^k(\bar{x}_1^m, \bar{x}_2^m) - \frac{1}{2} \delta^{mk} & \text{if } \frac{\partial u}{\partial n} \text{ is specified over } C^k \end{cases} \\ b^{mk} &= \begin{cases} \bar{u}^k (-F_2^k(\bar{x}_1^m, \bar{x}_2^m) + \frac{1}{2} \delta^{mk}) & \text{if } u \text{ is specified over } C^k \\ \bar{p}^k F_1^k(\bar{x}_1^m, \bar{x}_2^m) & \text{if } \frac{\partial u}{\partial n} \text{ is specified over } C^k \end{cases} \quad [2] \quad (24) \\ \delta^{mk} &= \begin{cases} 0 & \text{if } m \neq k \\ 1 & \text{if } m = k \end{cases} \\ z^k &= \begin{cases} \bar{p}^k & \text{if } u \text{ is specified over } C^k \\ \bar{u}^k & \text{if } \frac{\partial u}{\partial n} \text{ is specified over } C^k \end{cases} \end{aligned}$$

Summary

Applying BEM to solve a boundary value problem, we should first divide the boundary into many line segments and use Eq. (21) to solve both values of F_1 and F_2 on each line segment. Then, we should solve the linear system Eq. (23,24) to get all the unknown constants in the boundary element solution. Finally, we can plug all the calculation results into Eq. (20) to obtain the numerical solution to the BVP.

4.3 Boundary Element Solution of Pre-defined Problem and Comparison with Analytic Solution

In previous section, we have talked about the detailed principle and derivation of BEM. However, someone may ask, how does BEM perform in reality. In this section, we will apply BEM to the pre-defined problem, the Dirichlet problem governed by the two-dimensional Laplace's equation on the half-disk, and compare the numerical solution with analytic solution which have been obtained in previous section.

Following the steps in the last section, we firstly divide the semi-circle into 120 line segments as roughly shown in figure 4.

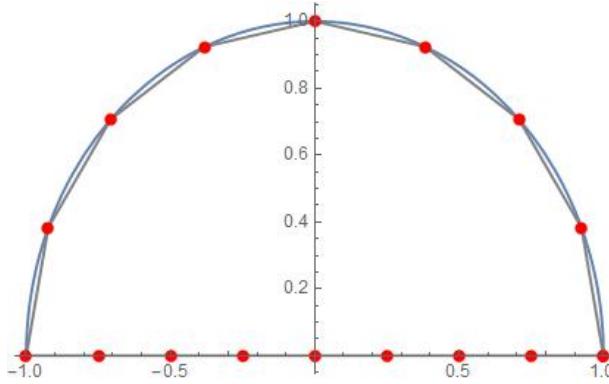


Figure 4: Rough idea to divide the boundary of half-disk

Then, we use Mathematica to calculate both values of F_1 and F_2 on each line segment and all the unknown constants (detailed code are attached in **Appendix**). Finally, we obtain the numerical solution (with 120 constant elements) and plot the solution in 3D-plane as shown in the figure 5 below.

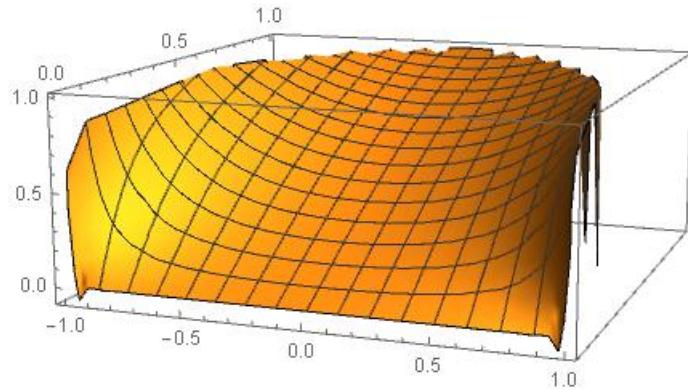


Figure 5: Boundary element solution to pre-defined problem

Comparing our plot of the numerical solution (figure 5) with previous plot of analytic solution (Figure 1), it is obvious that although both plots are almost same, the plot of numerical solution at the boundary is not as smooth as that of analytic solution. However, just drawing rough statements from visual observation is not sufficient, we will compare the two solution into details.

In the table 1 below, the numerical value of u (as calculated by the boundary element procedure with 60 and 120 constant elements) at an input interior point (ξ_1, ξ_2) is compared

with the exact solution.

(ξ_1, ξ_2)	60 Elements	120 Elements	Exact
(0.10,0.20)	0.253818	0.253738	0.253707
(0.10,0.30)	0.374508	0.374381	0.374334
(0.10,0.40)	0.488517	0.488345	0.488284
(0.50,0.20)	0.326754	0.326665	0.326623
(0.50,0.30)	0.469958	0.469776	0.469708
(0.50,0.40)	0.595803	0.595548	0.595458
(0.90,0.20)	0.773119	0.771894	0.771599
(0.90,0.30)	0.896111	0.895153	0.894863
(0.90,0.40)	0.977072	0.976404	0.976138

Table 1: Comparing numerical value of u with exact value at given interior points

From the table, it is clear that there is a significant improvement in the accuracy of the numerical results when the number of constant elements increases from 60 to 120. And when we use 120 constant elements, the numerical values has almost same number with the exact values.

We also examine the accuracy of the numerical value of u (as calculated by the boundary element procedure with 120 constant elements) at the interior point $(0.1, a)$ as a approaching 1 from 0.95. The results are listed in table 2 below.

Error	0.95	0.96	0.97	0.98	0.99
120 Elements	0.000102288	0.000102576	0.000103059	0.00010473	0.000112379

Table 2: Errors of numerical solution with 60 constant elements at some given interior points

From the table 2, it is clear that as interior points approaching to boundary, the errors get larger and larger. The results coincide with the well-known fact that the accuracy of a boundary element solution may deteriorate significantly at a point whose distance from the boundary is very small compared with the lengths of nearby boundary elements[2].

However, we only talked about interior points so far. What about the points which are exactly on the boundary? Will u at boundary points differ a lot with exact value? Here, we plot u (with 120 constant elements) at the boundary points where $\sqrt{x_1^2 + x_2^2} = 1$ and $x_2 > 0$ in the figure 6.

From the graph, it is clear that values of u at given boundary points vary around 0. However, the real values of u on the boundary should be equal to 1, which differ a lot with numerical results. One possible reason of the problem might be that when applying the BEM to solve the problem, the value on the boundary is undefined. Therefore, the difference between boundary element solution and analytic solution at the boundary points could be large. However, due to the continuity of the solution, we can still get a rough idea about the value u at boundary points by taking the value of u at interior point which is fairly close to the boundary point as a approximation to real value.

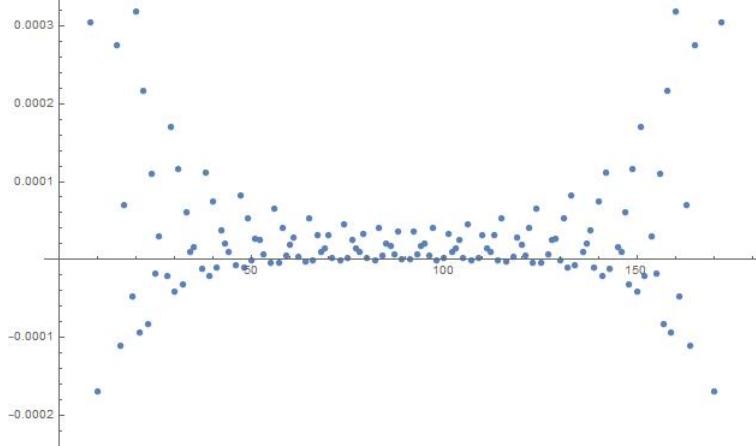


Figure 6: Numerical value of u at boundary points

To summarize the performance of BEM, we can conclude boundary element solution have a highly accuracy at interior points and more constant elements indicate a higher accuracy of the numerical solution. Although at boundary points, boundary element solution is not well-defined, we can still approximate the value at boundary points by taking the value at an interior point which is fairly close to the boundary point.

In real problems, the boundaries are often complex and irregular, where the analytical solution is often difficult to obtain. But the boundary element method are still valid for those problem and could hold a fairly high accuracy. Therefore, boundary element method is a quite general method to solve boundary value problems and is widely used in the engineering field.

5 Boundary Element Method Using Green Function

Now that we know the basic BEM principle and have successfully solved a simple boundary value problem with BEM, but can we explore more about BEM? As we all known, the Green function is a special kind of fundamental solution with homogeneous boundary conditions, so what would boundary element solution be if we substitute the Green function instead of ordinary fundamental solution? In this section, we will study with boundary element solution using Green function and based on that solve the pre-defined Dirichlet problem.

5.1 Boundary Element Solution Using Green Function

According to previous section, one of the fundamental solutions of the operator $-\Delta$ is

$$E(x; \xi) = -\frac{1}{2\pi} \ln |x - \xi| \quad [1]$$

Then, applying methods of images, it is easy to find the green function of Dirichlet problem governed by $-\Delta$ on the upper half-plane as

$$g(x; \xi) = E(x; \xi) - E(x; \xi^*) \quad [1] \quad (25)$$

where $\xi^* = (\xi_1, -\xi_2)$ supposing $\xi = (\xi_1, \xi_2)$ [1].

Thus, for the domain Ω defined in Eq. (1), we suppose $\partial\Omega = L \cup C$ where L denotes the bottom line of the semi circle, $x_2 = 0$, and C denotes the upper curve of the semi-circle, $\sqrt{x_1^2 + x_2^2} = 1$ [1]. Since $g(x; \xi)|_{x \in L} = 0$ and $\frac{\partial g(x; \xi)}{\partial n}|_{x \in L} = 0$, substituting the Green function into the boundary integral solution, we can get

$$\lambda(\xi)u(\xi) = \int_C [u(x)\frac{\partial g(x; \xi)}{\partial n} - g(x; \xi)\frac{\partial u(x)}{\partial n}]ds \quad [1] \quad (26)$$

where $\lambda(\xi)$ is defined as

$$\lambda(\xi) = \begin{cases} 0 & \text{if } \xi \in L \cup \bar{\Omega}^c \\ \frac{1}{2} & \text{if } \xi \in C \quad [1] \\ 1 & \text{if } \xi \in \Omega \cap \xi \notin C \end{cases}$$

As usual, to obtain the boundary element solution, we will divide C into N small line segments which can be roughly illustrated as the figure below.

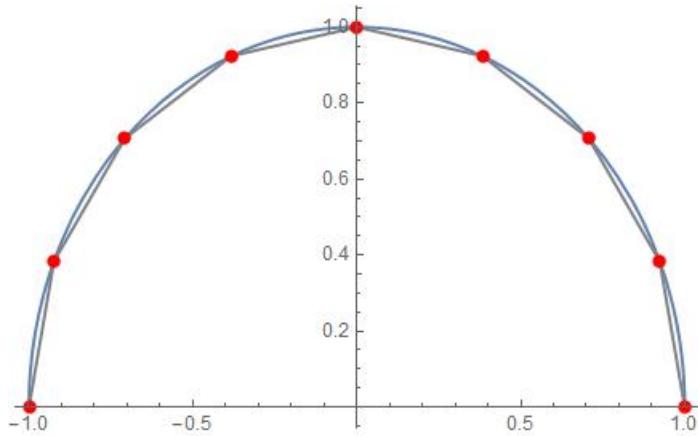


Figure 7: Rough Idea to Divide Boundary C

Thus, the boundary C can be approximately equal to the combination of all the line segments. That is

$$C \approx C^1 \cup C^2 \cup \dots \cup C^{N-1} \cup C^N \quad [2] \quad (28)$$

where C^k represent each line segment.

Also, as before, assuming that the value of u and $\frac{\partial u}{\partial n}$ on each line segment is constant, we use the both values at the midpoint of each line segments as approximations to the real values. That is

$$u \approx \bar{u}^k \quad \frac{\partial u}{\partial n} \approx \bar{p}^k \text{ for } (x_1, x_2) \in C^k \quad (k = 1, 2, \dots, N) \quad [2] \quad (29)$$

where \bar{u}^k and \bar{p}^k are the values of u and $\frac{\partial u}{\partial n}$ at the midpoint of C^k [2].

Then, it is easy for us to derive that

$$\lambda(\xi)u(\xi) \approx \sum_{k=1}^N (\bar{u}^k \int_{C^k} \frac{\partial g(\cdot; \xi)}{\partial n} ds - \bar{p}^k \int_{C^k} g(\cdot; \xi) ds) \quad [1] \quad (30)$$

where

$$\begin{aligned} \int_{C^k} \frac{\partial g(\cdot; \xi)}{\partial n} ds &= F_1^k(\xi) - F_1^k(\xi^*) & [1] \\ \int_{C^k} g(\cdot; \xi) ds &= F_2^k(\xi) - F_2^k(\xi^*) \end{aligned} \quad (31)$$

As before, we can also find N unknown parameters.

Thus, for the specific problem, Dirichlet Problem with Laplace operator on the half-disk, we rewrite our previous boundary element solution by changing the ordinary fundamental solution with green function of upper half plane. Then, to make the solution to this problem more intuitive, we plot the numerical solution with 120 constant elements in 3D-plane as shown in the figure 8.

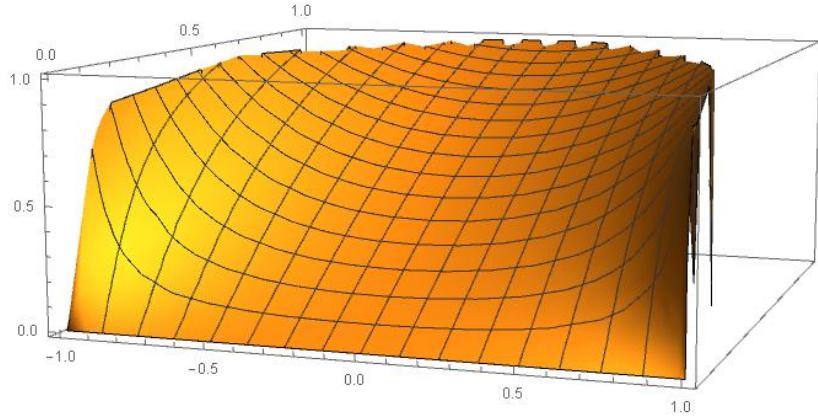


Figure 8: Boundary element Solution using Green function to pre-defined Dirichlet Problem

5.2 Comparison with Analytic Solution

Comparing the plot of modified boundary element solution (figure 8) with the plot of analytic solution (figure 2), still, both plots are quite similar with each other at interior points and the plot of boundary element solution using Green function at the boundary is not as smooth as the plot of analytic solution. Then, to provide more evidence to support our statement drawing from the visual observation, we will compare these two solutions into details.

Similarly as before, in the table 3, the value of u given by boundary element solution using Green function (with 60 and 120 constant elements) at input interior points are compared with exact value. From the table, we can see that both boundary element solutions give good approximations to the exact value u at given points. And we also notice that the one with a larger number of constant elements give a more accurate approximation. However, it should be mentioned that when applying boundary element solution using Green function, the differences of accuracies of two numerical solutions with different constant elements is not as large as those of previous numerical solutions without using Green function. Thus, we guess that with equal constant elements, BEM using Green function is more accurate than the previous one without using Green function. Without going into too much detail here, I will make a detailed comparison of the two methods in the next section.

(ξ_1, ξ_2)	60 Elements	120 Elements	Exact
(0.10,0.20)	0.253793	0.253729	0.253707
(0.10,0.30)	0.374455	0.374364	0.374334
(0.10,0.40)	0.488432	0.488321	0.488284
(0.50,0.20)	0.326824	0.326673	0.326623
(0.50,0.30)	0.469954	0.469769	0.469708
(0.50,0.40)	0.595719	0.595523	0.595458
(0.90,0.20)	0.772824	0.771899	0.771599
(0.90,0.30)	0.895567	0.895035	0.894863
(0.90,0.40)	0.976611	0.976256	0.976138

Table 3: Comparing numerical value with exact value of u at given points

Then, we also examine the accuracy of numerical value given by the boundary element solution using Green function with 80 constant elements at the interior point $(0.1, a)$ as a approaching 1. The results are show in the following table.

Error	0.95	0.96	0.97	0.98	0.99
80 Elements	0.000104368	0.000104222	0.000104274	0.000105507	0.000112669

Table 4: Errors of numerical solution with 80 constant elements

From the table 4, it is clear that as the interior point getting close to the boundary, error will also get large. Although the error at point $(0.1, 0.96)$ is a bit smaller than that at point $(0.1, 0.95)$, the overall trend is still on the rise.

We are also interested in how it performs at the boundary, therefore, we plot the value of u given by modified boundary element solution at some points on the boundary and the plot is shown in the figure below.

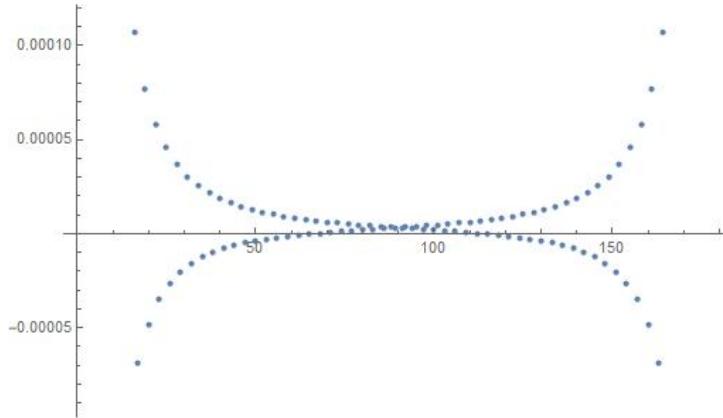


Figure 9: u Given by Modified Boundary Element Solution at Some Boundary Points

Still, we can see from the graph, the value u given by boundary element solution using Green function on the boundary points is undefined.

Now, we have compared both boundary element solutions with the analytic solution. Both of them have a good accuracy and almost behaves in a same way, which is under our expectation because Green function is just one special fundamental solution and all the features that original boundary element solution holds should also be hold for the boundary element solution using Green function.

5.3 Comparison with Original Boundary Element Solution

How does the boundary element solution using Green function perform compared with original boundary element solution. Based on the previous data, we guess that the one using Green function should have better accuracy than the one without using green function. Therefore, to prove our hypothesis, we compared the accuracy of both boundary element solutions with equal constant elements 120 at some given interior points. The results are given in the following table.

Error	(0.10,0.85)	(0.10,0.88)	(0.10,0.90)	(0.10,0.93)	(0.10,0.95)
With Green Fun.	4.68×10^{-5}	4.67×10^{-5}	4.66×10^{-5}	4.64×10^{-5}	4.63×10^{-5}
Original	9.90×10^{-5}	1.00×10^{-4}	1.01×10^{-4}	1.02×10^{-4}	1.02×10^{-4}

Table 5: Comparison of errors of BEM using Green function and original BEM

From the above table, it is clear that with the same number of constant elements, the boundary element solution using Green function gives lower errors. Thus, we can conclude that at interior points, the boundary element solution using Green function gives a better approximation compared with original boundary element solution.

6 Boundary Element Solution With Mixed Boundary Conditions

So far, we only focused on the Dirichlet problem which is simple and easy to calculate, but we know BEM can also be applied to complex BVP with even mixed boundary conditions like the mixture of Neumann condition and Dirichlet condition. Thus, in this section, we will investigate further on the performance of BEM when applied to mixed boundary value problem.

6.1 Mixed Boundary Value Problem for the Laplace Equation On The Half-disk

To study the BVP with mixed conditions, we will modified the original Dirichlet problem into mixed boundary value problem. That is

$$\Omega = \{(x_1, x_2) \in R^2 : x_1^2 + x_2^2 < 1, x_2 > 0\} \quad (32)$$

We set

$$\begin{aligned} -\Delta u &= 0 \text{ on } \Omega & u(x_1, x_2)|_{x_1^2 + x_2^2 = 1} &= 1 \\ & \frac{\partial u(x_1, 0)}{\partial n} &= -1 \text{ for } -1 < x_1 < 1 \end{aligned} \quad (33)$$

In the following sections, we will focus on the newly defined problem to study the performance of BEM when solving mixed boundary value problem.

6.2 Boundary Element Solution to Mixed Boundary Value Problem

Similarly as before, we use Mathematica to obtain the boundary element solution (with 120 constant elements) to the mixed boundary value problem (detailed code can be found in **Appendix**). And we plot our solution in the three-dimensional plane as shown in the figure 10 below.,

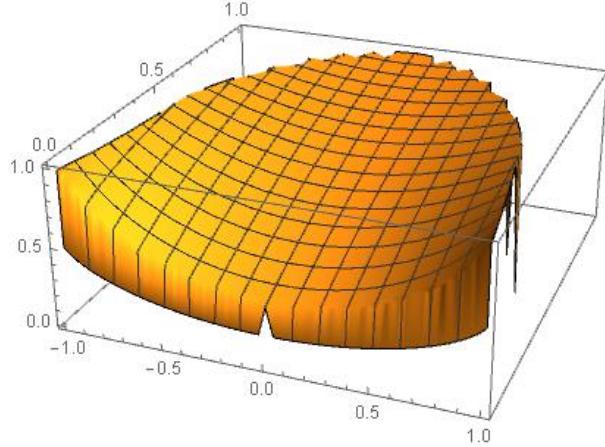


Figure 10: Boundary Element Solution to Mixed Boundary Value Problem

Also, to make a comparison, we use the finite element methods built inside Mathematica to obtain another numerical solution which holds sufficient high accuracy and can be approximately considered as the real solution to this problem. The plot of the numerical solution is given below.

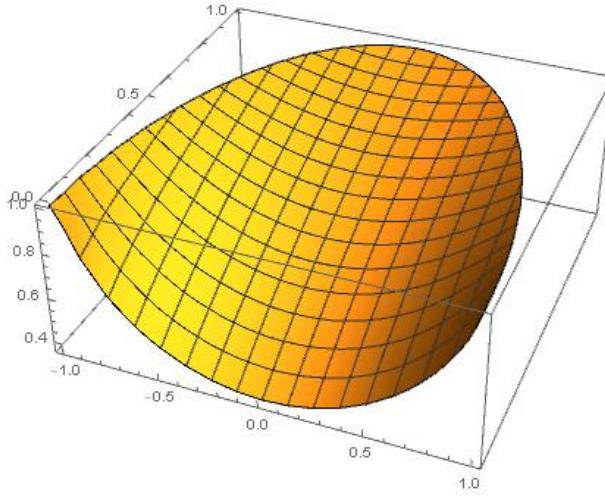


Figure 11: Approximate Real solution to Mixed Boundary Value Problem

Comparing the two plots, although their shapes look similar, it is hard for us to see if the boundary element method provides an accurate solution to the mixed boundary element problem. Therefore, we should look deeply into data and compare these two solutions into details.

In the table below, the numerical value of u of the boundary element solution with 60 and 120 constant elements at some given interior points (ξ_1, ξ_2) is compared with numerical

value of u of finite element solution which can be approximately considered as the real value of u at those points.

(ξ_1, ξ_2)	60 Element	120 Element	Approximate	Exact value
(0.10,0.20)	0.551203	0.550764	0.551841	
(0.10,0.30)	0.630278	0.629848	0.630815	
(0.10,0.40)	0.701632	0.701225	0.702062	
(0.50,0.20)	0.653107	0.652429	0.652624	
(0.50,0.30)	0.725203	0.724614	0.724802	
(0.50,0.40)	0.788329	0.787831	0.788019	
(0.90,0.20)	0.924841	0.924011	0.923646	
(0.90,0.30)	0.959895	0.959372	0.959226	
(0.90,0.40)	0.990241	0.989907	0.989775	

Table 6: Comparing numerical value with approximate exact value

From the table above, we can see that as the number of constant elements increases, the accuracy of our boundary element solution also increases. And both two boundary element solutions gives a good approximation to the real solution of this problem.

As before, we then examine the accuracy of value of u given by boundary element solution as the points getting closer to the boundary. Here, we use the boundary element solution with 120 constant elements and examine the accuracy at interior point $(0.1, a)$ where a will approach 1 gradually. The observed results are listed in the following table.

Error	0.95	0.96	0.97	0.98	0.99
120 Elements	3.74×10^{-6}	7.97×10^{-6}	2.07×10^{-5}	2.57×10^{-5}	3.87×10^{-5}

Table 7: Errors of numerical solution with 120 constant elements

From the table above, it is clear that as the points approaching to the boundary, the errors get larger and larger.

However, it should be still noticed that here we use the numerical solution obtained by finite element method as the real solution to the problem, although it holds a fairly high accuracy, it can not really replace the real solution to the problem. Therefore, we cannot conclude any reliable statement from the data, but roughly, we can still say that boundary element solution should be very close to the real solution to the problem.

In terms of above results, we can summarize that BEM is still valid for those complex mixed boundary value problems and the numerical solutions of mixed boundary value problems still behaves in the same way as those of simple Dirichlet problem.

7 Conclusion and Discussion

7.1 Conclusion

In this paper, we first analyze the internal principle and performance of BEM in detail by applying BEM to a simple Dirichlet problem. Then, we discuss the modified boundary element solution using Green function, which has a better performance than the previous one

without using Green function. Finally, we change our pre-defined Dirichlet problem to mixed boundary value problem and try to use BEM to solve it which turns out to be valid and could give a fairly accurate numerical solution to the problem. Although during the analysis we found the numerical solution given by BEM is not well-defined at the boundary points, we can still take the value of nearby points as a proper approximation to the real value at the boundary.

Thus, it could be concluded that BEM is a general method that can solve most boundary problems.

7.2 Further Discussion

The part of the article that I wanted to explore in detail is over, however, there are a few things more worth noting about BEM.

As a numerical method, compared with the finite element method, the boundary element method has many excellent characteristics. Instead of divide the continuum domain into many small elements, the boundary element method only needs to divide the boundary into a finite number of line segments which could turn a high-dimensional problem into a lower-dimensional problem. Therefore, BEM has greatly reduced the computation of solve a boundary value problem and highly improved the accuracy of the numerical solution.

However, BEM also has a big disadvantage, which is that BEM requires the corresponding green's function with an equation[3]. Because the boundary element method relates boundary conditions and internal unknowns through green's function, it is only applicable equations where green's function is easy to know[3]. Although there have been various ways to loosen the restrictions in recent times, this has fundamentally restricted the application of boundary element method to other problems such as hyperbolic equations[3].

Nevertheless, because of its simplicity and accuracy, BEM is still widely used in many fields.

8 Works Cited

- [1] H. Hohberger. “vv557 video slides.pdf”. UMJI-SJTU, Shanghai. [Online; accessed 2-May-2020].
- [2] ANG, W. T. *A Beginner’s Course in Boundary Element Methods*. Universal Publishers, Boca Raton, USA, 2007.
- [3] “Boundary element method,” Wikipedia, 12-Apr-2020. [Online]. Available: https://en.wikipedia.org/wiki/Boundary_element_method. [Accessed: 02-May-2020].

9 Appendix

The detailed code are attached here.

Code Part

Analytic Solution to the Dirichlet Problem

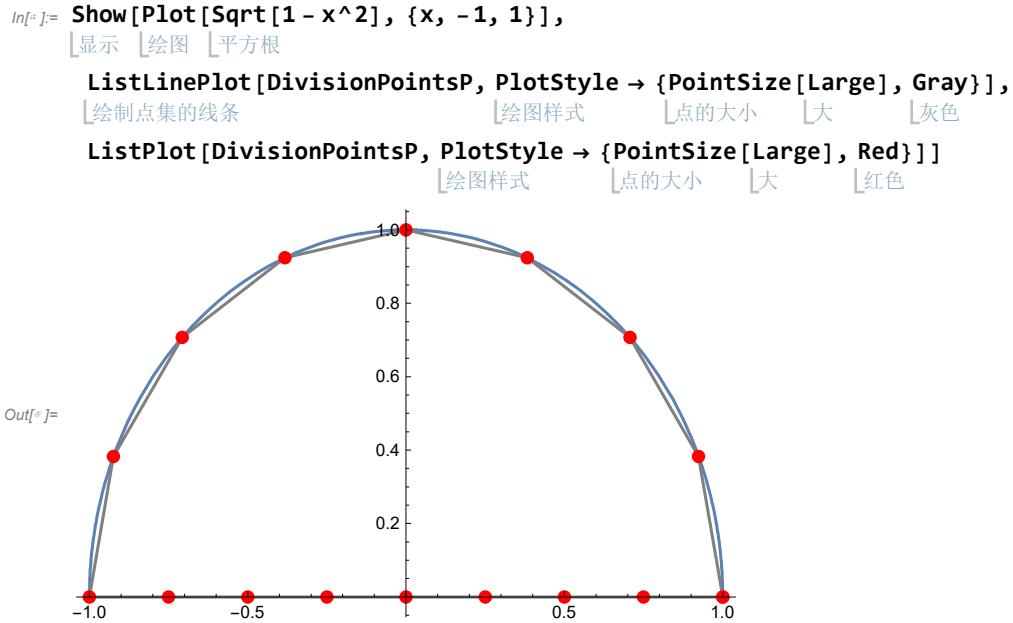
```
In[9]:= AnalyticSolution =
Function[{r, θ},  $\frac{2}{\pi} \operatorname{ArcTan}\left[\frac{1+r}{1-r} * \operatorname{Tan}\left[\frac{\theta}{2}\right]\right] + \frac{2}{\pi} \operatorname{ArcTan}\left[\frac{1+r}{1-r} * \operatorname{Cot}\left[\frac{\theta}{2}\right]\right] - 1$ ];
纯函数
```

```
ParametricPlot3D[{r * Cos[θ], r * Sin[θ], AnalyticSolution[r, θ]}, {r, 0, 1}, {θ, 0, π}, BoxRatios -> {2.5, 2.5, 1}]
绘制三维参数图 余弦 正弦
BoxRatios -> {2.5, 2.5, 1}
边界框比例
```

Out[9]=

Rough Idea of Dividing the Boundary

```
In[10]:= DivisionPointsP = {};
For[i = 0, i ≤ 8, i++,
For循环
DivisionPointsP = Insert[DivisionPointsP,
插入
N[CoordinateTransform["Polar" → "Cartesian", {1,  $\frac{i}{8}\pi$ }], 10], -1]
坐标变换
]
DivisionPointsP = Join[DivisionPointsP,
连接
{{{-0.75, 0}, {-0.5, 0}, {-0.25, 0}, {0, 0}, {0.25, 0}, {0.5, 0}, {0.75, 0}, {1, 0}}};
```



Implementation of Boundary Element Methods

```

In[1]:= NumberDivideP = 40;
TotalNumberPointsP = 3 * NumberDivideP;

In[3]:= PointsOnLineP = {};
For[i = 0, i < NumberDivideP - 1, i++,
  PointsOnLineP = Insert[PointsOnLineP, -1 +  $\frac{2 * i}{NumberDivideP}$ , -1]
  ];
PointsOnCurveP = {};
For[i = 0, i < 2 * NumberDivideP, i++,
  PointsOnCurveP = Insert[PointsOnCurveP,  $\frac{i}{2 * NumberDivideP}$ , -1]
  ];
PointsXP = PointsOnLineP;
For[i = 1, i < Length[PointsOnCurveP], i++,
  PointsXP = Insert[PointsXP, N[Cos[PointsOnCurveP[[i]] * π]], -1]
  ];
PointsYP = ConstantArray[0, Length[PointsOnLineP]];
For[i = 1, i < Length[PointsOnCurveP], i++,
  PointsYP = Insert[PointsYP, N[Sin[PointsOnCurveP[[i]] * π]], -1]
  ];

```

```

];
SegmentsLenP = {};
For[i = 1, i ≤ TotalNumberPointsP, i++,
[For循环
  SegmentsLenP = Insert[SegmentsLenP,
    [插入
      Norm[N[{PointsXP[[i + 1]] - PointsXP[[i]], PointsYP[[i + 1]] - PointsYP[[i]]}]], -1]
    [数值运算
  ];
NormalPointsXP = {};
For[i = 1, i ≤ TotalNumberPointsP, i++,
[For循环
  NormalPointsXP = Insert[NormalPointsXP,  $\frac{\text{PointsYP}[[i + 1]] - \text{PointsYP}[[i]]}{\text{SegmentsLenP}[[i]]}$ , -1]
  [插入
];
NormalPointsYP = {};
For[i = 1, i ≤ TotalNumberPointsP, i++,
[For循环
  NormalPointsYP = Insert[NormalPointsYP, - $\frac{\text{PointsXP}[[i + 1]] - \text{PointsXP}[[i]]}{\text{SegmentsLenP}[[i]]}$ , -1]
  [插入
];

```

```

In[6]:= FP = Function[{i, k, x, y},
  纯函数

  FA = Function[kk, SegmentsLenP[[kk]]^2][k];
  纯函数

  FB = Function[{kk, xx, yy},
  纯函数

  (-NormalPointsYP[[kk]] * (PointsXP[[kk]] - xx) + NormalPointsXP[[kk]] *
    (PointsYP[[kk]] - yy)) * 2 * SegmentsLenP[[kk]]][k, x, y];

  FE = Function[{kk, xx, yy}, (PointsXP[[kk]] - xx)^2 + (PointsYP[[kk]] - yy)^2][k, x, y];
  纯函数

  Decision = 4 * FA * FE - FB^2;

  If[i == 1,
  如果

    Re[If[Decision == 0,  $\frac{\text{SegmentsLenP}[[k]]}{2 \pi} * \left( \log[\text{SegmentsLenP}[[k]]] + \right.$ 
    ... 如果
     $\left( 1 + \frac{FB}{2 * FA} \right) * \log[\text{Abs}\left[1 + \frac{FB}{2 * FA}\right]] - \frac{FB}{2 * FA} * \log[\text{Abs}\left[\frac{FB}{2 * FA}\right]] - 1\right),$ 

     $\frac{\text{SegmentsLenP}[[k]]}{4 \pi} * \left( 2 * (\log[\text{SegmentsLenP}[[k]]] - 1) - \right.$ 
    对数 绝对值
     $\left. \frac{FB}{2 * FA} * \log[\text{Abs}\left[\frac{FE}{FA}\right]] + \left( 1 + \frac{FB}{2 * FA} \right) * \log[\text{Abs}\left[1 + \frac{FB + FE}{FA}\right]] + \right.$ 
    对数 绝对值
     $\left. \frac{\text{Sqrt}[Decision]}{FA} * \left( \text{ArcTan}\left[\frac{2 * FA + FB}{\text{Sqrt}[Decision]}\right] - \text{ArcTan}\left[\frac{FB}{\text{Sqrt}[Decision]}\right] \right) \right)$ 
    反正切 反正切
    ]]
    , Re[If[Decision == 0, 0,
    ... 如果

      SegmentsLenP[[k]] * ((NormalPointsXP[[k]] * (PointsXP[[k]] - x) +
        NormalPointsYP[[k]] * (PointsYP[[k]] - y)) / ( $\pi * \text{Sqrt}[Decision]$ )) *
      平方根
       $\left( \text{ArcTan}\left[\frac{2 * FA + FB}{\text{Sqrt}[Decision]}\right] - \text{ArcTan}\left[\frac{FB}{\text{Sqrt}[Decision]}\right] \right)$ 
      反正切 反正切
    ]]
    ]
  ];
  MidPointsXP = PointsOnLineP +  $\frac{1}{\text{NumberDivideP} * 2}$ ;
  For[i = 1, i < Length[PointsOnCurveP], i++,
  For循环 长度

    MidPointsXP =
  
```

```

Insert[MidPointsXP, Cos[(PointsOnCurveP[[i]] +  $\frac{1}{\text{NumberDivideP} * 4}$ ) * π], -1]
插入 余弦

];
MidPointsYP = ConstantArray[0, Length[PointsOnLineP]];
常量数组 长度

For[i = 1, i ≤ Length[PointsOnCurveP], i++,
For循环 长度

  MidPointsYP =
    Insert[MidPointsYP, Sin[(PointsOnCurveP[[i]] +  $\frac{1}{\text{NumberDivideP} * 4}$ ) * π], -1]
插入 正弦

];
BoundsP = {};
For[i = 1, i ≤ TotalNumberPointsP, i++,
For循环

  BoundsP =
    Insert[BoundsP, N[If[N[MidPointsXP[[i]]² + MidPointsYP[[i]]²] == 1, 1, 0]], -1];
插入 … 数值运算

];
FactorXP = {};
For[i = 1, i ≤ TotalNumberPointsP, i++,
For循环

  FactorXP = Insert[FactorXP,  $\frac{\text{PointsXP}[[i+1]] - \text{PointsXP}[[i]]}{2}$  + PointsXP[[i]], -1];
插入

];
FactorYP = {};
For[i = 1, i ≤ TotalNumberPointsP, i++,
For循环

  FactorYP = Insert[FactorYP,  $\frac{\text{PointsYP}[[i+1]] - \text{PointsYP}[[i]]}{2}$  + PointsYP[[i]], -1];
插入

];
aP = {};
For[i = 1, i ≤ TotalNumberPointsP, i++,
For循环

  Temp = {};
  For[j = 1, j ≤ TotalNumberPointsP, j++,
For循环

    Temp = Insert[Temp, N[-FP[1, j, FactorXP[[i]], FactorYP[[i]]]], -1];
插入 数值运算

  ];
  aP = Insert[aP, Temp, -1];
插入

];
bP = {};

```

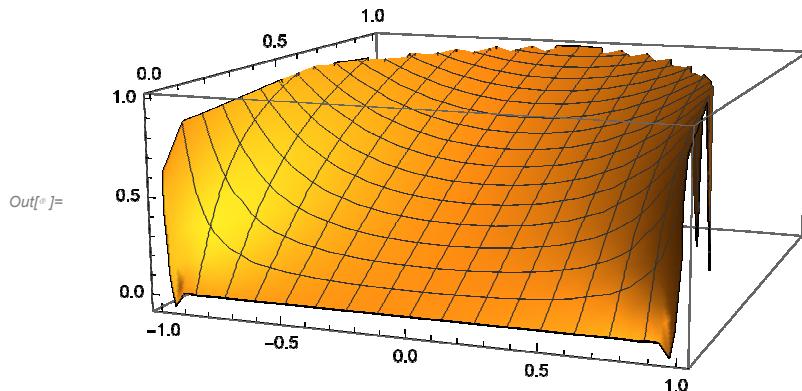
```

For[i = 1, i ≤ TotalNumberPointsP, i++,
  For循环
    Temp = {};
    For[j = 1, j ≤ TotalNumberPointsP, j++,
      For循环
        Temp = Insert[Temp, N[BoundsP[[j]] * (-FP[2, j, FactorXP[[i]], FactorYP[[i]]]) +
          插入 数值运算
          Function[{ii, jj}, If[ii == jj, 1, 0][i, j] / 2], -1];
          如果
        ];
        bP = Insert[bP, Sum[Temp[[k]], {k, 1, TotalNumberPointsP}], -1];
        插入 求和
      ];
      zP = LinearSolve[aP, bP];
      线性求解
    ];
    BoundsUP = {};
    For[i = 1, i ≤ TotalNumberPointsP, i++,
      For循环
        BoundsUP = Insert[BoundsUP, N[BoundsP[[i]]], -1];
        插入 数值运算
      ];
    ];
    BoundsNP = {};
    For[i = 1, i ≤ TotalNumberPointsP, i++,
      For循环
        BoundsNP = Insert[BoundsNP, zP[[i]], -1];
        插入
      ];
    ];
    In[6]:= BEMP = Function[{x, y}, Sum[BoundsUP[[i]] * FP[2, i, x, y] -
      纯函数 求和
      BoundsNP[[i]] * FP[1, i, x, y], {i, 1, TotalNumberPointsP}]];
  
```

```

In[7]:= Plot3D[BEMP[x, y], {x, -1, 1}, {y, 0, Sqrt[1 - x^2]}]
  绘制三维图形 平方根

```



Comparison Between Boundary Element Solution and Analytic Solution

```

In[8]:= PointsPA = {{0.10, 0.20}, {0.10, 0.30}, {0.1, 0.40}, {0.50, 0.20},
  {0.50, 0.30}, {0.50, 0.40}, {0.90, 0.20}, {0.90, 0.30}, {0.90, 0.40}};

```

```

In[5]:= ValueP120 = {};
For[i = 1, i ≤ Length[PointsPA], i++,
  |For循环 |长度
  ValueP120 = Insert[ValueP120, BEMP[PointsPA[[i, 1]], PointsPA[[i, 2]]], -1]
    |插入
]
ValueP120

Out[5]= {0.253738, 0.374381, 0.488345, 0.326665,
  0.469776, 0.595548, 0.771894, 0.895153, 0.976404}

In[6]:= ValueP60 = {};
For[i = 1, i ≤ Length[PointsPA], i++,
  |For循环 |长度
  ValueP60 = Insert[ValueP60, BEMP[PointsPA[[i, 1]], PointsPA[[i, 2]]], -1]
    |插入
]
ValueP60

Out[6]= {0.253818, 0.374508, 0.488517, 0.326754,
  0.469958, 0.595803, 0.773119, 0.896111, 0.977072}

In[7]:= PointsPAPolar = {};
For[i = 1, i ≤ Length[PointsPA], i++,
  |For循环 |长度
  PointsPAPolar = Insert[PointsPAPolar,
    |插入
    CoordinateTransform["Cartesian" → "Polar", PointsPA[[i]]], -1]
    |坐标变换
]
PointsPAPolar

Out[7]= {{0.223607, 1.10715}, {0.316228, 1.24905}, {0.412311, 1.32582},
  {0.538516, 0.380506}, {0.583095, 0.54042}, {0.640312, 0.674741},
  {0.921954, 0.218669}, {0.948683, 0.321751}, {0.984886, 0.418224} }

In[8]:= ValueExact = {};
For[i = 1, i ≤ Length[PointsPAPolar], i++,
  |For循环 |长度
  ValueExact = Insert[ValueExact,
    |插入
    AnalyticSolution[PointsPAPolar[[i, 1]], PointsPAPolar[[i, 2]]], -1]
]
ValueExact

Out[8]= {0.253707, 0.374334, 0.488284, 0.326623,
  0.469708, 0.595458, 0.771599, 0.894863, 0.976138}

In[9]:= PointsPA2 = {{0.1, 0.95}, {0.1, 0.96}, {0.1, 0.97}, {0.1, 0.98}, {0.1, 0.99}}
Out[9]= {{0.1, 0.95}, {0.1, 0.96}, {0.1, 0.97}, {0.1, 0.98}, {0.1, 0.99}}

```

```

In[°]:= PointsPAPolar2 = {};
For[i = 1, i ≤ Length[PointsPA2], i++,
  For循环   长度
  PointsPAPolar2 = Insert[PointsPAPolar2,
    插入
    CoordinateTransform["Cartesian" → "Polar", PointsPA2[[i]]], -1]
    坐标变换
  ]
PointsPAPolar2

Out[°]= {{0.955249, 1.46592}, {0.965194, 1.467},
{0.975141, 1.46807}, {0.985089, 1.46911}, {0.995038, 1.47013}}


In[°]:= ValueExact2 = {};
For[i = 1, i ≤ Length[PointsPAPolar2], i++,
  For循环   长度
  ValueExact2 = Insert[ValueExact2,
    插入
    AnalyticSolution[PointsPAPolar2[[i, 1]], PointsPAPolar2[[i, 2]]], -1]
  ]
ValueExact2

Out[°]= {0.970703, 0.97733, 0.983891, 0.990386, 0.996817}

In[°]:= ValueP1202 = {};
For[i = 1, i ≤ Length[PointsPA2], i++,
  For循环   长度
  ValueP1202 = Insert[ValueP1202, BEMP[PointsPA2[[i, 1]], PointsPA2[[i, 2]]], -1]
    插入
  ]
ValueP1202

Out[°]= {0.970805, 0.977433, 0.983994, 0.990491, 0.996929}

In[°]:= (ValueP1202 - ValueExact2)

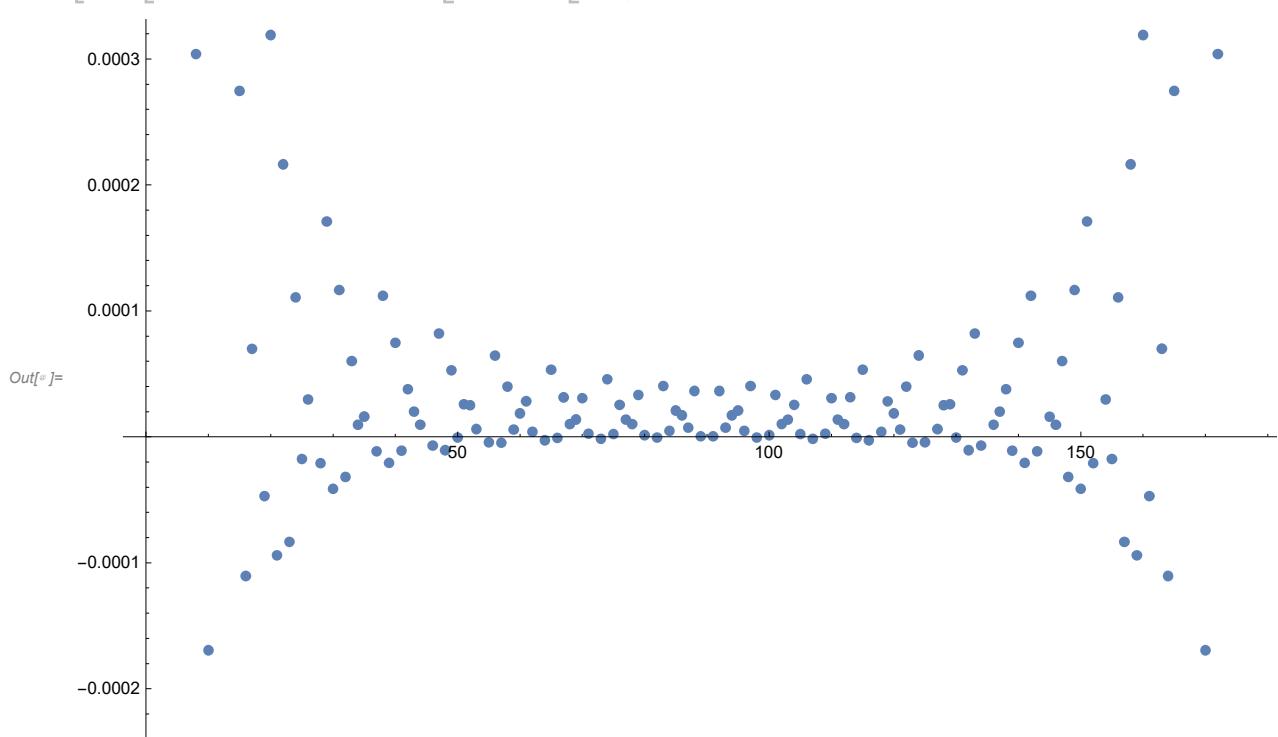
Out[°]= {0.000102288, 0.000102576, 0.000103059, 0.00010473, 0.000112379}

In[°]:= PointsPA3 = {};
For[i = 1, i ≤ 180, i++,
  For循环
  PointsPA3 = Insert[PointsPA3,
    插入
    N[CoordinateTransform["Polar" → "Cartesian", {1, (i/180)π}], 10], -1]
    坐标变换
  ]
PointsPA3

In[°]:= ValueP1203 = {};
For[i = 1, i ≤ Length[PointsPA3], i++,
  For循环   长度
  ValueP1203 = Insert[ValueP1203, BEMP[PointsPA3[[i, 1]], PointsPA3[[i, 2]]], -1]
    插入
  ]
ValueP1203

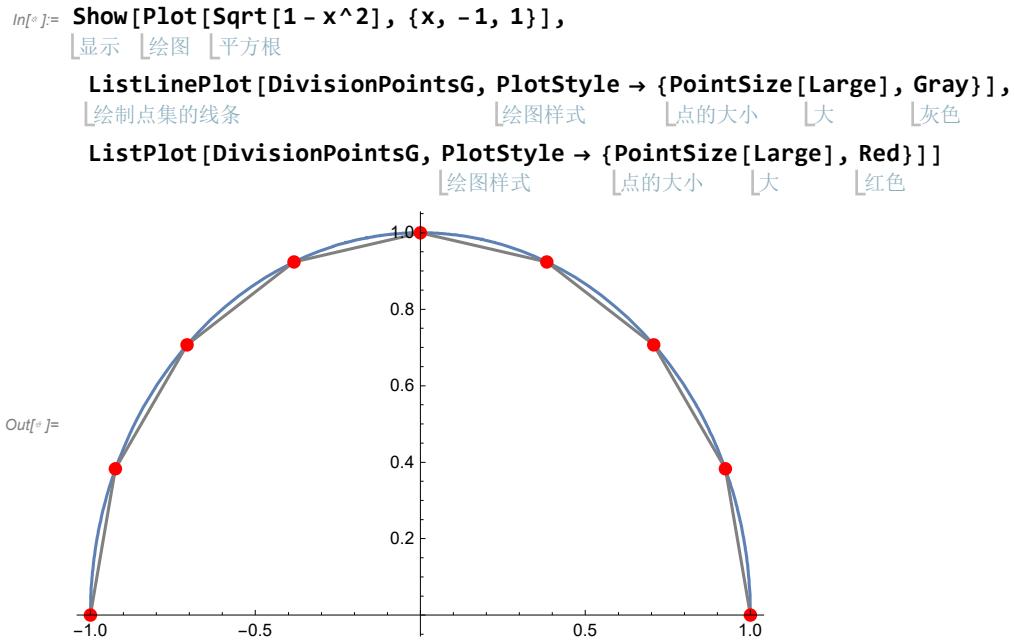
```

```
In[5]:= Show[ListPlot[ValueP1203], ListPlot[ConstantArray[1, 180]]]
```



Rough Idea of Dividing Boundary When Applying BEM Using Green Function

```
In[6]:= DivisionPointsG = {};
For[i = 0, i <= 8, i++,
 $\text{DivisionPointsG} = \text{Insert}[\text{DivisionPointsG},$ 
 $\text{N}[\text{CoordinateTransform}["\text{Polar"} \rightarrow "\text{Cartesian}", \{1, \frac{i}{8}\pi\}], 10], -1]$ 
]
```



Implementation of BEM Using Green Function

```
In[7]:= NumberDivideG = 60;
TotalNumberPointsG = 2 * NumberDivideG;
```

```

PointsOnCurveG = {};
For[i = 0, i ≤ 2 * NumberDivideG, i++,
[For循环
    PointsOnCurveG = Insert[PointsOnCurveG,  $\frac{i}{2 * NumberDivideG}$ , -1]
    [插入
];
PointsXG = {};
For[i = 1, i ≤ Length[PointsOnCurveG], i++,
[长度
    PointsXG = Insert[PointsXG, N[Cos[PointsOnCurveG[[i]] * π]], -1]
    [插入      [..]余弦
];
PointsYG = {};
For[i = 1, i ≤ Length[PointsOnCurveG], i++,
[长度
    PointsYG = Insert[PointsYG, N[Sin[PointsOnCurveG[[i]] * π]], -1]
    [插入      [..]正弦
];
SegmentsLenG = {};
For[i = 1, i ≤ TotalNumberPointsG, i++,
[For循环
    SegmentsLenG = Insert[SegmentsLenG,
    [插入
        Norm[N[{PointsXG[[i + 1]] - PointsXG[[i]], PointsYG[[i + 1]] - PointsYG[[i]]}]], -1]
        [数值运算
];
NormalPointsXG = {};
For[i = 1, i ≤ TotalNumberPointsG, i++,
[For循环
    NormalPointsXG = Insert[NormalPointsXG,  $\frac{PointsYG[[i + 1]] - PointsYG[[i]]}{SegmentsLenG[[i]]}$ , -1]
    [插入
];
NormalPointsYG = {};
For[i = 1, i ≤ TotalNumberPointsG, i++,
[For循环
    NormalPointsYG = Insert[NormalPointsYG, - $\frac{PointsXG[[i + 1]] - PointsXG[[i]]}{SegmentsLenG[[i]]}$ , -1]
    [插入
];
];

```

```

FG = Function[{i, k, x, y},
  纯函数

FA = Function[kk, SegmentsLenG[[kk]]^2][k];
  纯函数

FB = Function[{kk, xx, yy},
  纯函数
  (-NormalPointsYG[[kk]] * (PointsXG[[kk]] - xx) + NormalPointsXG[[kk]] *
   PointsYG[[kk]] - yy)) * 2 * SegmentsLenG[[kk]][k, x, y];
  纯函数

FE = Function[{kk, xx, yy}, (PointsXG[[kk]] - xx)^2 + (PointsYG[[kk]] - yy)^2][k, x, y];
  纯函数

Decision = 4 * FA * FE - FB^2;

If[i == 1,
  如果
  Re[If[Decision == 0,  $\frac{\text{SegmentsLenG}[k]}{2 \pi} * \left( \log[\text{SegmentsLenG}[k]] + \right.$ 
     $\left. \left( 1 + \frac{FB}{2 * FA} \right) * \log\left[\text{Abs}\left[1 + \frac{FB}{2 * FA}\right]\right] - \frac{FB}{2 * FA} * \log\left[\text{Abs}\left[\frac{FB}{2 * FA}\right]\right] - 1 \right),$ 
     $\frac{\text{SegmentsLenG}[k]}{4 \pi} * \left( 2 * (\log[\text{SegmentsLenG}[k]] - 1) - \right.$ 
     $\left. \frac{FB}{2 * FA} * \log\left[\text{Abs}\left[\frac{FE}{FA}\right]\right] + \left( 1 + \frac{FB}{2 * FA} \right) * \log\left[\text{Abs}\left[1 + \frac{FB + FE}{FA}\right]\right] + \right.$ 
     $\left. \frac{\text{Sqrt}[Decision]}{FA} * \left( \text{ArcTan}\left[\frac{2 * FA + FB}{\text{Sqrt}[Decision]}\right] - \text{ArcTan}\left[\frac{FB}{\text{Sqrt}[Decision]}\right] \right) \right)$ 
  ],
  如果
  , Re[If[Decision == 0, 0,
     $\text{SegmentsLenG}[k] * ((\text{NormalPointsXG}[k] * (\text{PointsXG}[k] - x) +$ 
     $\text{NormalPointsYG}[k] * (\text{PointsYG}[k] - y)) / (\pi * \text{Sqrt}[Decision])) * \sqrt{\left( \text{ArcTan}\left[\frac{2 * FA + FB}{\text{Sqrt}[Decision]}\right] - \text{ArcTan}\left[\frac{FB}{\text{Sqrt}[Decision]}\right] \right)}$ 
  ],
  如果
  ];
  如果
  MidPointsXG = {};
  For[i = 1, i <= Length[PointsOnCurveG], i++,
    For循环
    MidPointsXG =
  ];
  If
];

```

```

Insert[MidPointsXG, Cos[(PointsOnCurveG[[i]] +  $\frac{1}{\text{NumberDivideG} * 4}$ ) * π], -1]
插入 余弦

];
MidPointsYG = {};
For[i = 1, i ≤ Length[PointsOnCurveG], i++,
For循环 长度

MidPointsYG =
Insert[MidPointsYG, Sin[(PointsOnCurveG[[i]] +  $\frac{1}{\text{NumberDivideG} * 4}$ ) * π], -1]
插入 正弦

];
BoundsG = {};
For[i = 1, i ≤ TotalNumberPointsG, i++,
For循环

BoundsG = Insert[BoundsG, 1, -1];
插入

];
FactorXG = {};
For[i = 1, i ≤ TotalNumberPointsG, i++,
For循环

FactorXG = Insert[FactorXG,  $\frac{\text{PointsXG}[[i+1]] - \text{PointsXG}[[i]]}{2}$  + PointsXG[[i]], -1];
插入

];
FactorYG = {};
For[i = 1, i ≤ TotalNumberPointsG, i++,
For循环

FactorYG = Insert[FactorYG,  $\frac{\text{PointsYG}[[i+1]] - \text{PointsYG}[[i]]}{2}$  + PointsYG[[i]], -1];
插入

];
aG = {};
For[i = 1, i ≤ TotalNumberPointsG, i++,
For循环

Temp = {};
For[j = 1, j ≤ TotalNumberPointsG, j++,
For循环

Temp = Insert[Temp, N[-(FG[1, j, FactorXG[[i]], FactorYG[[i]]] -
插入 数值运算

FG[1, j, FactorXG[[i]], -FactorYG[[i]]]), -1];
插入

];
aG = Insert[aG, Temp, -1];
插入

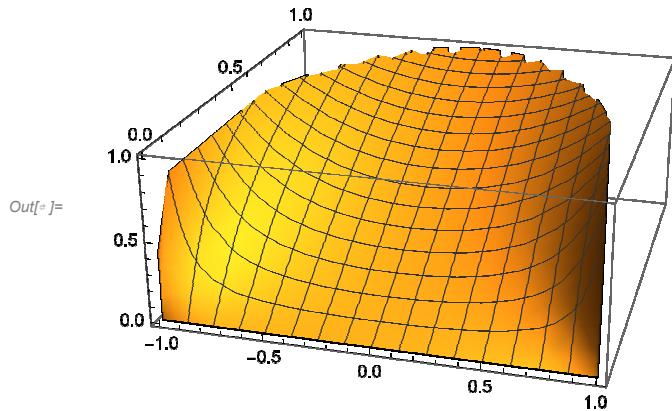
];
bG = {};
For[i = 1, i ≤ TotalNumberPointsG, i++,
For循环

```

```

Temp = {};
For[j = 1, j < TotalNumberPointsG, j++,
  For循环
    Temp = Insert[Temp, N[BoundsG[[j]] * 
      FG[2, j, FactorXG[[i]], FactorYG[[i]]] + FG[2, j, FactorXG[[i]],
      -FactorYG[[i]]] + Function[{ii, jj}, If[ii == jj, 1, 0]][i, j] / 2], -1];
      插入 数值运算
      纯函数 如果
    ];
  bG = Insert[bG, Sum[Temp[[k]], {k, 1, TotalNumberPointsG}], -1];
  插入 求和
];
zG = LinearSolve[aG, bG];
线性求解
BoundsUG = {};
For[i = 1, i < TotalNumberPointsG, i++,
  For循环
    BoundsUG = Insert[BoundsUG, N[BoundsG[[i]]], -1];
    插入 数值运算
  ];
];
BoundsNG = {};
For[i = 1, i < TotalNumberPointsG, i++,
  For循环
    BoundsNG = Insert[BoundsNG, zG[[i]], -1];
    插入
  ];
];
In[5]:= BEMG = Function[{x, y}, Sum[BoundsUG[[i]] * (FG[2, i, x, y] - FG[2, i, x, -y]) -
  纯函数 求和
  BoundsNG[[i]] * (FG[1, i, x, y] - FG[1, i, x, -y]), {i, 1, TotalNumberPointsG}]];
Out[5]= Plot3D[BEMG[x, y], {x, -1, 1}, {y, 0, Sqrt[1 - x^2]}]
绘制三维图形 平方根

```



Comparison Between BES Using Green Function and Analytic Solution

```

In[6]:= PointsGA = {{0.10, 0.20}, {0.10, 0.30}, {0.1, 0.40}, {0.50, 0.20},
{0.50, 0.30}, {0.50, 0.40}, {0.90, 0.20}, {0.90, 0.30}, {0.90, 0.40}};

```

```

In[°]:= ValueG60 = {};
For[i = 1, i ≤ Length[PointsGA], i++,
  |For循环 |长度
  ValueG60 = Insert[ValueG60, BEMG[PointsGA[[i, 1]], PointsGA[[i, 2]]], -1]
    |插入
]
ValueG60

Out[°]= {0.253793, 0.374455, 0.488432, 0.326824,
  0.469954, 0.595719, 0.772824, 0.895567, 0.976611}

In[°]:= ValueG120 = {};
For[i = 1, i ≤ Length[PointsGA], i++,
  |For循环 |长度
  ValueG120 = Insert[ValueG120, BEMG[PointsGA[[i, 1]], PointsGA[[i, 2]]], -1]
    |插入
]
ValueG120

Out[°]= {0.253729, 0.374364, 0.488321, 0.326673,
  0.469769, 0.595523, 0.771899, 0.895035, 0.976256}

In[°]:= PointsGA2 = {{0.10, 0.95}, {0.10, 0.96}, {0.10, 0.97}, {0.10, 0.98}, {0.10, 0.99}}
Out[°]= {{0.1, 0.95}, {0.1, 0.96}, {0.1, 0.97}, {0.1, 0.98}, {0.1, 0.99}}

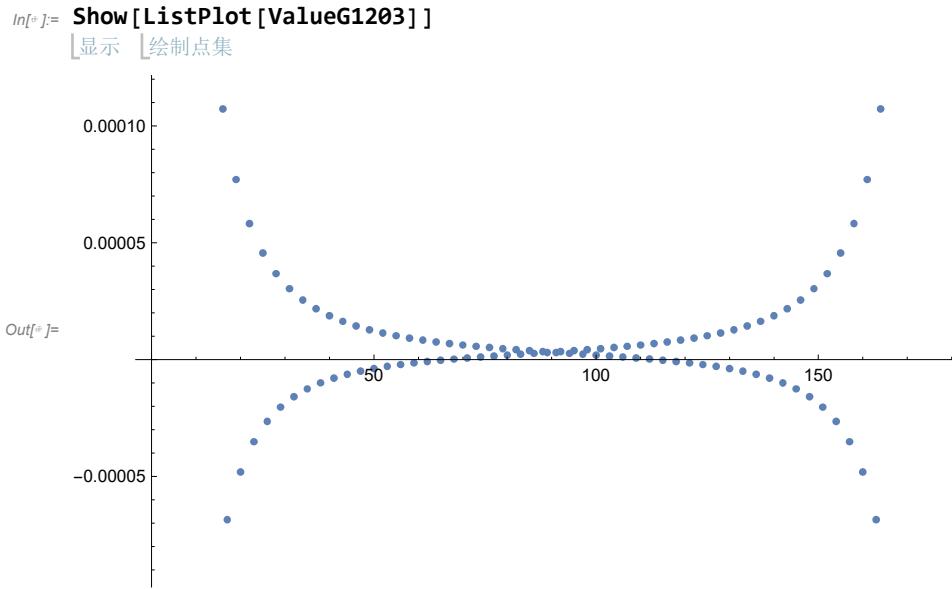
In[°]:= ValueG802 = {};
For[i = 1, i ≤ Length[PointsGA2], i++,
  |For循环 |长度
  ValueG802 = Insert[ValueG802, BEMG[PointsGA2[[i, 1]], PointsGA2[[i, 2]]], -1]
    |插入
]
ValueG802

Out[°]= {0.970807, 0.977434, 0.983995, 0.990492, 0.99693}

In[°]:= ValueG802 - ValueExact2
Out[°]= {0.000104368, 0.000104222, 0.000104274, 0.000105507, 0.000112669}

In[°]:= ValueG1203 = {};
For[i = 1, i ≤ Length[PointsPA3], i++,
  |For循环 |长度
  ValueG1203 = Insert[ValueG1203, BEMG[PointsPA3[[i, 1]], PointsPA3[[i, 2]]], -1]
    |插入
]

```



Comparison Between BES and BES Using Green Function

```

In[7]:= PointsGP = {{0.10, 0.85}, {0.10, 0.88}, {0.10, 0.90}, {0.10, 0.93}, {0.10, 0.95}};

In[7]:= ValueG1204 = {};
For[i = 1, i ≤ Length[PointsGP], i++,
  ValueG1204 = Insert[ValueG1204, BEMG[PointsGP[[i, 1]], PointsGP[[i, 2]]], -1];
]
ValueG1204

Out[7]= {0.900688, 0.922447, 0.936596, 0.957293, 0.970749}

In[8]:= ValueP1204 = {};
For[i = 1, i ≤ Length[PointsGP], i++,
  ValueP1204 = Insert[ValueP1204, BEMP[PointsGP[[i, 1]], PointsGP[[i, 2]]], -1];
]
ValueP1204

Out[8]= {0.90074, 0.922501, 0.93665, 0.957348, 0.970805}

In[9]:= PointsGPPolar = {};
For[i = 1, i ≤ Length[PointsGP], i++,
  PointsGPPolar = Insert[PointsGPPolar,
    CoordinateTransform["Cartesian" → "Polar", PointsGP[[i]]], -1];
]
PointsGPPolar

Out[9]= {{0.855862, 1.45369}, {0.885664, 1.45765},
{0.905539, 1.46014}, {0.935361, 1.46368}, {0.955249, 1.46592}}

```

```

In[6]:= ValueExactGP = {};
For[i = 1, i ≤ Length[PointsGPPolar], i++,
  | For循环 | 长度
  ValueExactGP = Insert[ValueExactGP,
    | 插入
    AnalyticSolution[PointsGPPolar[[i, 1]], PointsGPPolar[[i, 2]]], -1];
]
ValueExactGP

Out[6]= {0.900641, 0.922401, 0.936549, 0.957247, 0.970703}

In[7]:= ErrorG120 = ValueG1204 - ValueExactGP
Out[7]= {0.0000468309, 0.0000467238, 0.0000466232, 0.0000464328, 0.0000462819}

In[8]:= ErrorP120 = ValueP1204 - ValueExactGP
Out[8]= {0.0000989915, 0.00010014, 0.000100826, 0.000101743, 0.000102288}

```

Approximate Real Solution to Mixed Boundary Value Problem

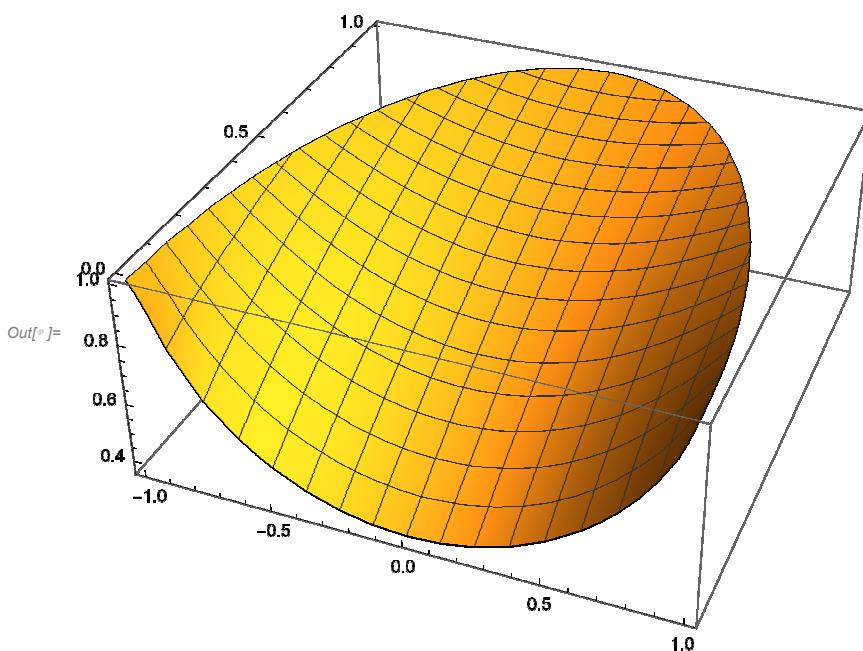
```

In[9]:= AnalyticSolution2 = NDSolveValue[{-\nabla_{x,y}^2 u[x, y] == NeumannValue[-1, y == 0],
  | 数值解的值 | 莫伊曼边值
  DirichletCondition[u[x, y] == 1, x^2 + y^2 == 1],
  | 狄里克雷条件
  u, {x, y} ∈ Disk[{0, 0}, 1, {0, π}]]
| 圆盘

Out[9]= InterpolatingFunction[+  Domain: {{-1., 1.}, {0., 1.}} ]
| 绘制三维图形 Output: scalar

```

In[¹⁰]:= Plot3D[AnalyticSolution2[x, y], {x, y} ∈ Disk[{0, 0}, 1, {0, π}]]



Implementation of BEM to Solve Mixed Boundary Value Problem

```

In[°]:= NumberDivideM = 40;
TotalNumberPointsM = 3 * NumberDivideM;

PointsOnLineM = {};
For[i = 0, i ≤ NumberDivideM - 1, i++,
|For循环
    PointsOnLineM = Insert[PointsOnLineM, -1 +  $\frac{2 * i}{NumberDivideM}$ , -1]
    |插入
];
PointsOnCurveM = {};
For[i = 0, i ≤ 2 * NumberDivideM, i++,
|For循环
    PointsOnCurveM = Insert[PointsOnCurveM,  $\frac{i}{2 * NumberDivideM}$ , -1]
    |插入
];
PointsXM = PointsOnLineM;
For[i = 1, i ≤ Length[PointsOnCurveM], i++,
|For循环 |长度
    PointsXM = Insert[PointsXM, N[Cos[PointsOnCurveM[[i]] * π]], -1]
    |插入 |余弦
];
PointsYM = ConstantArray[0, Length[PointsOnLineM]];
|常量数组 |长度
For[i = 1, i ≤ Length[PointsOnCurveM], i++,
|长度
    PointsYM = Insert[PointsYM, N[Sin[PointsOnCurveM[[i]] * π]], -1]
    |插入 |正弦
];
SegmentsLenM = {};
For[i = 1, i ≤ TotalNumberPointsM, i++,
|For循环
    SegmentsLenM = Insert[SegmentsLenM,
    |插入
        Norm[N[{PointsXM[[i + 1]] - PointsXM[[i]], PointsYM[[i + 1]] - PointsYM[[i]]}]], -1]
    |数值运算
];
NormalPointsXM = {};
For[i = 1, i ≤ TotalNumberPointsM, i++,
|For循环
    NormalPointsXM = Insert[NormalPointsXM,  $\frac{PointsYM[[i + 1]] - PointsYM[[i]]}{SegmentsLenM[[i]]}$ , -1]
    |插入
];
NormalPointsYM = {};

```

```
For[i = 1, i < TotalNumberPointsM, i++,  
  [For循环  
  NormalPointsYM = Insert[NormalPointsYM, - $\frac{\text{PointsXM}[[i + 1]] - \text{PointsXM}[[i]]}{\text{SegmentsLenM}[[i]]}$ , -1]  
  [插入  
];
```

```

In[1]:= FM = Function[{i, k, x, y},
  纯函数

  FA = Function[kk, SegmentsLenM[[kk]]^2][k];
  纯函数

  FB = Function[{kk, xx, yy},
  纯函数

  (-NormalPointsYM[[kk]] * (PointsXM[[kk]] - xx) + NormalPointsXM[[kk]] *
   (PointsYM[[kk]] - yy)) * 2 * SegmentsLenM[[kk]][k, x, y];

  FE = Function[{kk, xx, yy}, (PointsXM[[kk]] - xx)^2 + (PointsYM[[kk]] - yy)^2][k, x, y];
  纯函数

  Decision = 4 * FA * FE - FB^2;

  If[i == 1,
  如果

    Re[If[Decision == 0,  $\frac{\text{SegmentsLenM}[[k]]}{2 \pi} * \left( \log[\text{SegmentsLenM}[[k]]] + \right.$ 
    ... 如果
     $\left( 1 + \frac{FB}{2 * FA} \right) * \log\left[\text{Abs}\left[1 + \frac{FB}{2 * FA}\right]\right] - \frac{FB}{2 * FA} * \log\left[\text{Abs}\left[\frac{FB}{2 * FA}\right]\right] - 1\right),$ 

     $\frac{\text{SegmentsLenM}[[k]]}{4 * \pi} * \left( 2 * (\log[\text{SegmentsLenM}[[k]]] - 1) - \right.$ 
     $\frac{FB}{2 * FA} * \log\left[\text{Abs}\left[\frac{FE}{FA}\right]\right] + \left( 1 + \frac{FB}{2 * FA} \right) * \log\left[\text{Abs}\left[1 + \frac{FB + FE}{FA}\right]\right] +$ 

     $\frac{\text{Sqrt}[Decision]}{FA} * \left( \text{ArcTan}\left[\frac{2 * FA + FB}{\text{Sqrt}[Decision]}\right] - \text{ArcTan}\left[\frac{FB}{\text{Sqrt}[Decision]}\right] \right)\right)$ 
    ]]
  , Re[If[Decision == 0, 0,
  ... 如果

    SegmentsLenM[[k]] * ((NormalPointsXM[[k]] * (PointsXM[[k]] - x) +
    NormalPointsYM[[k]] * (PointsYM[[k]] - y)) / ( $\pi * \text{Sqrt}[Decision]$ )) *
    平方根
     $\left( \text{ArcTan}\left[\frac{2 * FA + FB}{\text{Sqrt}[Decision]}\right] - \text{ArcTan}\left[\frac{FB}{\text{Sqrt}[Decision]}\right] \right)$ 
    ]]
  ]
];

```

MidPointsXM = PointsOnLineM + $\frac{1}{\text{NumberDivideM} * 2}$;

For[i = 1, i ≤ Length[PointsOnCurveM], i++,
 For循环 长度

MidPointsXM =

```

Insert[MidPointsXM, Cos[(PointsOnCurveM[[i]] +  $\frac{1}{NumberDivideM * 4}$ ) * π], -1]
插入 余弦

];
MidPointsYM = ConstantArray[0, Length[PointsOnLineM]];
常量数组 长度

For[i = 1, i ≤ Length[PointsOnCurveM], i++,
For循环 长度

  MidPointsYM =
    Insert[MidPointsYM, Sin[(PointsOnCurveM[[i]] +  $\frac{1}{NumberDivideM * 4}$ ) * π], -1]
插入 正弦

];
BoundsM = {};
For[i = 1, i ≤ NumberDivideM, i++,
For循环

  BoundsM = Insert[BoundsM, -1, -1];
插入

];
For[i = 1, i ≤ 2 * NumberDivideM, i++,
For循环

  BoundsM = Insert[BoundsM, 1, -1];
插入

];
FactorXM = {};
For[i = 1, i ≤ TotalNumberPointsM, i++,
For循环

  FactorXM = Insert[FactorXM,  $\frac{PointsXM[[i + 1]] - PointsXM[[i]]}{2}$  + PointsXM[[i]], -1];
插入

];
FactorYM = {};
For[i = 1, i ≤ TotalNumberPointsM, i++,
For循环

  FactorYM = Insert[FactorYM,  $\frac{PointsYM[[i + 1]] - PointsYM[[i]]}{2}$  + PointsYM[[i]], -1];
插入

];
aM = {};
For[i = 1, i ≤ TotalNumberPointsM, i++,
For循环

  Temp = {};
  For[j = 1, j ≤ NumberDivideM, j++,
For循环

    Temp = Insert[Temp, N[FM[2, j, FactorXM[[i]], FactorYM[[i]]] -
插入 数值运算

      Function[{ii, jj}, If[ii == jj, 1, 0]][i, j] / 2], -1];
如果

  ];

```

```


$$\text{For}[j = \text{NumberDivideM} + 1, j \leq \text{NumberDivideM} * 3, j++,$$


$$\quad \text{Temp} = \text{Insert}[\text{Temp}, N[-\text{FM}[1, j, \text{FactorXM}[[i]], \text{FactorYM}[[i]]]], -1];$$


$$\quad \quad \quad \text{插入} \quad \quad \quad \text{数值运算}$$


$$\quad ];$$


$$\quad \text{aM} = \text{Insert}[\text{aM}, \text{Temp}, -1];$$


$$\quad \quad \quad \text{插入}$$


$$\quad ];$$


$$\quad \text{bM} = \{\};$$


$$\quad \text{For}[i = 1, i \leq \text{TotalNumberPointsM}, i++,$$


$$\quad \quad \quad \text{[For循环}$$


$$\quad \quad \quad \text{Temp} = \{\};$$


$$\quad \quad \quad \text{For}[j = 1, j \leq \text{NumberDivideM}, j++,$$


$$\quad \quad \quad \quad \quad \text{[For循环}$$


$$\quad \quad \quad \quad \quad \text{Temp} = \text{Insert}[\text{Temp}, N[\text{BoundsM}[[j]] * \text{FM}[1, j, \text{FactorXM}[[i]], \text{FactorYM}[[i]]]], -1];$$


$$\quad \quad \quad \quad \quad \text{插入} \quad \quad \quad \text{数值运算}$$


$$\quad \quad ];$$


$$\quad \quad \text{For}[j = \text{NumberDivideM} + 1, j \leq \text{NumberDivideM} * 3, j++,$$


$$\quad \quad \quad \text{[For循环}$$


$$\quad \quad \quad \text{Temp} = \text{Insert}[\text{Temp}, N[\text{BoundsM}[[j]] * (-\text{FM}[2, j, \text{FactorXM}[[i]], \text{FactorYM}[[i]]]) +$$


$$\quad \quad \quad \quad \quad \text{插入} \quad \quad \quad \text{数值运算}$$


$$\quad \quad \quad \quad \quad \text{Function}[\{ii, jj\}, \text{If}[ii == jj, 1, 0][i, j] / 2)], -1];$$


$$\quad \quad \quad \quad \quad \text{如果}$$


$$\quad \quad ];$$


$$\quad \quad \text{bM} = \text{Insert}[\text{bM}, \text{Sum}[\text{Temp}[[k]], \{k, 1, \text{TotalNumberPointsM}\}], -1];$$


$$\quad \quad \quad \text{插入} \quad \quad \quad \text{求和}$$


$$\quad \quad ];$$


$$\quad \text{zM} = \text{LinearSolve}[\text{aM}, \text{bM}];$$


$$\quad \quad \quad \text{线性求解}$$


$$\quad \text{BoundsUM} = \{\};$$


$$\quad \text{For}[i = 1, i \leq \text{NumberDivideM}, i++,$$


$$\quad \quad \quad \text{[For循环}$$


$$\quad \quad \quad \text{BoundsUM} = \text{Insert}[\text{BoundsUM}, \text{zM}[[i]], -1];$$


$$\quad \quad \quad \text{插入}$$


$$\quad \quad ];$$


$$\quad \quad \text{For}[i = \text{NumberDivideM} + 1, i \leq \text{NumberDivideM} * 3, i++,$$


$$\quad \quad \quad \text{[For循环}$$


$$\quad \quad \quad \text{BoundsUM} = \text{Insert}[\text{BoundsUM}, N[\text{BoundsM}[[i]]], -1];$$


$$\quad \quad \quad \text{插入} \quad \quad \quad \text{数值运算}$$


$$\quad \quad ];$$


$$\quad \text{BoundsNM} = \{\};$$


$$\quad \text{For}[i = 1, i \leq \text{NumberDivideM}, i++,$$


$$\quad \quad \quad \text{[For循环}$$


$$\quad \quad \quad \text{BoundsNM} = \text{Insert}[\text{BoundsNM}, N[\text{BoundsM}[[i]]], -1];$$


$$\quad \quad \quad \text{插入} \quad \quad \quad \text{数值运算}$$


$$\quad \quad ];$$


$$\quad \quad \text{For}[i = \text{NumberDivideM} + 1, i \leq \text{NumberDivideM} * 3, i++,$$


$$\quad \quad \quad \text{[For循环}$$

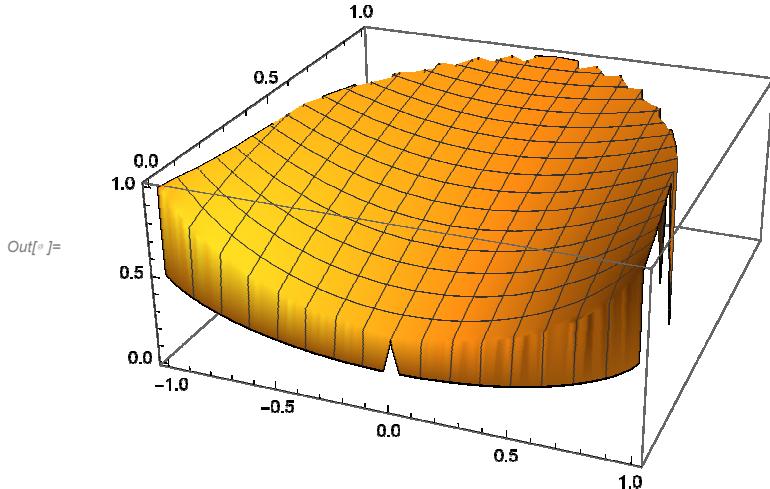

$$\quad \quad \quad \text{BoundsNM} = \text{Insert}[\text{BoundsNM}, \text{zM}[[i]], -1];$$


$$\quad \quad \quad \text{插入}$$


$$\quad \quad ];$$


```

```
In[1]:= BEMM = Function[{x, y}, Sum[BoundsUM[[i]] * FM[2, i, x, y] -  
    纯函数  
    求和  
    BoundsNM[[i]] * FM[1, i, x, y], {i, 1, TotalNumberPointsM}]];  
  
In[2]:= Plot3D[BEMM[x, y], {x, -1, 1}, {y, 0, Sqrt[1 - x^2]}]  
    绘制三维图形  
    平方根
```



Comparison Between Boundary Element Solution and Approximate Real Solution

```
In[3]:= PointsMA = {{0.10, 0.20}, {0.10, 0.30}, {0.1, 0.40}, {0.50, 0.20},  
    {0.50, 0.30}, {0.50, 0.40}, {0.90, 0.20}, {0.90, 0.30}, {0.90, 0.40}};  
  
In[4]:= ValueM120 = {};  
For[i = 1, i < Length[PointsMA], i++,  
    For循环  
    长度  
    ValueM120 = Insert[ValueM120, BEMM[PointsMA[[i, 1]], PointsMA[[i, 2]]], -1]  
        插入  
    ]  
ValueM120  
  
Out[4]= {0.550764, 0.629848, 0.701225, 0.652429,  
    0.724614, 0.787831, 0.924011, 0.959372, 0.989907}  
  
In[5]:= ValueM60 = {};  
For[i = 1, i < Length[PointsMA], i++,  
    For循环  
    长度  
    ValueM60 = Insert[ValueM60, BEMM[PointsMA[[i, 1]], PointsMA[[i, 2]]], -1]  
        插入  
    ]  
ValueM60  
  
Out[5]= {0.551203, 0.630278, 0.701632, 0.653107,  
    0.725203, 0.788329, 0.924841, 0.959895, 0.990241}
```

```

In[6]:= ValueExactMA = {};
For[i = 1, i ≤ Length[PointsMA], i++,
|For循环 |长度
ValueExactMA =
Insert[ValueExactMA, AnalyticSolution2[PointsMA[[i, 1]], PointsMA[[i, 2]]], -1]
|插入
]
ValueExactMA

Out[6]= {0.551841, 0.630815, 0.702062, 0.652624,
0.724802, 0.788019, 0.923646, 0.959226, 0.989775}

In[7]:= PointsMA2 = {{0.10, 0.95}, {0.10, 0.96}, {0.10, 0.97}, {0.0, 0.98}, {0, 0.99}};

In[8]:= ValueM1202 = {};
For[i = 1, i ≤ Length[PointsMA2], i++,
|For循环 |长度
ValueM1202 = Insert[ValueM1202, BEMM[PointsMA2[[i, 1]], PointsMA2[[i, 2]]], -1]
|插入
]
ValueM1202

Out[8]= {0.983365, 0.987142, 0.99088, 0.992721, 0.996405}

In[9]:= ValueExactMA2 = {};
For[i = 1, i ≤ Length[PointsMA2], i++,
|For循环 |长度
ValueExactMA2 =
Insert[ValueExactMA2, AnalyticSolution2[PointsMA2[[i, 1]], PointsMA2[[i, 2]]], -1]
|插入
]
ValueExactMA2

Out[9]= {0.983369, 0.987134, 0.99086, 0.992695, 0.996367}

In[10]:= ValueM1202 - ValueExactMA2
Out[10]= {-3.74495 × 10-6, 7.97227 × 10-6, 0.0000207005, 0.0000256709, 0.0000387372}

In[11]:= PointsMA3 = {};
For[i = 1, i ≤ 180, i++,
|For循环
PointsMA3 = Insert[PointsMA3,
|插入
N[CoordinateTransform["Polar" → "Cartesian", {1,  $\frac{i}{180}\pi$ }], 10], -1]
|坐标变换
]

```

```
In[5]:= ValueM1203 = {};
For[i = 1, i < Length[PointsMA3], i++,
  ValueM1203 = Insert[ValueM1203, BEMM[PointsMA3[[i, 1]], PointsMA3[[i, 2]]], -1]
]

```

```
In[6]:= Show[ListPlot[ValueM1203]]

```

