Chapter 3
The dual reciprocity boundary element method

This chapter presents the dual reciprocity boundary element method for solving non-homogenous problems. The first section introduces the difficulty associated with domain integrals which may occur in the solution of Poisson's equation. The second section surveys some features of Poisson's equation. The third section gives the foundation of the dual reciprocity method to handle the domain integral problem. The fourth section discusses the modification of the programs described in previous chapters in order to deal with such problems by transforming to Laplace problem. The fifth section details the formulation of the dual reciprocity boundary element method with the two frequently-used radial basis functions. The sixth section describes a program applying both radial basis functions in the method. The seventh section reports on a program to solve variation on Poisson's equation and Poisson-type equations. The last section gives concluding remarks.

3.1 Introduction

There are three classical numerical methods for solving problems in engineering and applied science. The first approach is the Finite Difference Method (FDM). This technique approximates the derivatives in the governing differential equation, using some type of truncated Taylor expansion. The second is the Finite Element Method (FEM). This method involves the approximation of variables over small parts of the domain, called elements, in terms of polynomial interpolation functions. The disadvantages of FEM are that large quantities of data are required to discretise the full domain. The third one is the Boundary Element Method (BEM). This approach is developed as a response to the FEM discretisation problem. The method requires discretisation of the boundary only, thus reducing the quantity of data necessary to run a program.

However, there are some difficulties in extending the technique to applications such as non-homogeneous, non-linear and time-dependent problems. The main drawback in these cases is the need to discretise the domain into a series of internal cells to deal
with the terms not taken to the boundary by application of the fundamental solution. This additional discretisation destroys some of the attraction of the method in terms of the data required to run the program and the complexity of the extra operations involved.

A new approach is needed to deal with domain integrals in boundary elements. Several methods have been proposed by different authors. The Dual Reciprocity Method (DRM) is essentially a generalised way of constructing particular solutions that can be used to solve non-linear and time-dependent problems as well as to represent any internal source distribution.

### 3.2 Poisson's equation

Consider the Poisson's equation

\[
\nabla^2 u = b
\]

(3.1)

in a domain \( \Omega \) (as shown in Figure 3.2.1) where \( b \) is a known function of position.

![Figure 3.2.1 Geometric definition of the problem](image-url)
As, for the Laplace equation, we have

$$\int_\Omega (\nabla^2 u - b)u^* d\Omega = \int_{\Gamma_2} (\bar{q} - \bar{q}'\bar{u}) u' d\Gamma - \int_{\Gamma_1} (\bar{u} - \bar{u}') q' d\Gamma$$  \hspace{1cm} (3.2)$$

which can be integrated by parts twice to produce (Partridge and Brebbia, 1992)

$$\int_\Omega (\nabla^2 u^*)u d\Omega - \int_\Omega bu^* d\Omega = -\int_{\Gamma_2} \bar{q}u^* d\Gamma - \int_{\Gamma_1} qu^* d\Gamma + \int_{\Gamma_2} \bar{u}q^* d\Gamma + \int_{\Gamma_1} \bar{b}q^* d\Gamma$$  \hspace{1cm} (3.3)$$

After substituting the fundamental solution $u^*$ of the Laplace operator into (3.3) and grouping all boundary terms together we obtain

$$c_i u_i + \int_{\Gamma} uq^* d\Gamma + \int_{\Omega} bu^* d\Omega = \int_{\Gamma} qu^* d\Gamma$$  \hspace{1cm} (3.4)$$

Notice that, although the function $b$ is known and consequently the integral in $\Omega$ does not introduce any new unknowns, the problem has changed in character as we need now to carry out a domain integral as well as the boundary integral. The constant $c_i$ depends only on the boundary geometry at the point $i$ under consideration.

The simplest way of computing the domain integral term in equation (3.4) is by subdividing the region into a series of internal cells, over each of which a numerical integration scheme such as Gauss quadrature can be applied. However, this technique loses the attraction of its boundary only character. Several methods have been proposed by different authors (Partridge et al., 1992). The Dual Reciprocity Method (DRM) is essentially a generalised way of constructing particular solutions that can be used to solve non-linear and time-dependent problems as well as to represent any internal source distribution. The method can be applied to define sources over the whole domain or only on part of it. Before discussing the method, we investigate how to approximate the function $b$ in case of a position function.
3.3 Foundation of the dual reciprocity method

Consider the Poisson equation

$$\nabla^2 u = b$$  \hspace{1cm} (3.5)

where $b = b(x, y)$. The solution to equation (3.5) can be expressed as the sum of the solution of a homogeneous and a particular solution as

$$u = u_h + \hat{u}$$  \hspace{1cm} (3.6)

where $u_h$ is the solution of the homogeneous equation $\nabla^2 u = 0$ and $\hat{u}$ is a particular solution of (3.5) such that

$$\nabla^2 \hat{u} = b.$$  \hspace{1cm} (3.7)

We approximate $b$ as a linear combination of interpolation functions for each of which we can find a particular solution, $\hat{u}$. The interpolation functions are defined over a set of interpolation points in the domain and on its boundary.

If there are $N$ boundary nodes and $L$ internal nodes, as in Figure 3.3.1, there will be $N + L$ interpolation functions, $f_j$, and $N + L$ particular solutions, $\hat{u}_j$.

![Figure 3.3.1 Interpolation points from boundary and internal nodes](image-url)
The approximation of $b$ over $\Omega$ is written in the form

$$b(x, y) \approx \sum_{j=1}^{N+L} \alpha_j f_j(x, y). \quad (3.8)$$

If $b$ and $f_j$ are the values of $b$ and $f_j$ at node $i$ respectively, then we have a matrix equation for the unknown coefficients $\{\alpha_j : j = 1, 2, \ldots, N + L\}$, i.e.

$$\mathbf{b} = \mathbf{F}\alpha. \quad (3.9)$$

The particular solutions, $\hat{u}_j$, and the interpolation functions, $f_j$, are linked through the relation

$$\nabla^2 \hat{u}_j = f_j. \quad (3.10)$$

Substituting equation (3.10) into (3.8) gives

$$b \approx \sum_{j=1}^{N+L} \alpha_j (\nabla^2 \hat{u}_j). \quad (3.11)$$

Equation (3.11) can be substituted into the original equation (3.5) to give the following expression

$$\nabla^2 u = \sum_{j=1}^{N+L} \alpha_j (\nabla^2 \hat{u}_j). \quad (3.12)$$

Multiplying by the fundamental solution and integrating by parts over the domain, we obtain

$$\int_{\Omega} (\nabla^2 u) u^* d\Omega = \sum_{j=1}^{N+L} \alpha_j \int_{\Omega} \left(\nabla^2 \hat{u}_j\right) u^* d\Omega. \quad (3.13)$$
Note that the same result may be obtained from equation
\[
\int_{\Omega} (\nabla^2 u)u^r d\Omega = \int_{\Omega} bu^r d\Omega. \tag{3.14}
\]

Integrating by parts the Laplacian terms in (3.13) produces the following integral equation for each source node \(i\) (Partridge et al., 1992):
\[
c_i u_i + \int_{\Gamma_i} q^* u d\Gamma - \int_{\Gamma_i} u^* q d\Gamma = \sum_{j=1}^{N+L} \alpha_j \left( c_i \hat{u}_{ij} + \int_{\Gamma_i} q^* \hat{u}_j d\Gamma - \int_{\Gamma_i} u^* \hat{q}_j d\Gamma \right) \tag{3.15}
\]
The term \(\hat{q}_j\) in equation (3.15) is defined as \(\hat{q}_j = \frac{\partial \hat{u}_j}{\partial n}\), where \(n\) is the unit outward normal to \(\Gamma\), and can be written as
\[
\hat{q}_j = \frac{\partial \hat{u}_j}{\partial x} \frac{\partial}{\partial n} + \frac{\partial \hat{u}_j}{\partial y} \frac{\partial}{\partial n} \tag{3.16}
\]
Note that equation (3.15) involves no domain integrals. The next step is to write equation (3.15) in discretised form, with summations over the boundary elements replacing the integrals. This gives, for a source node \(i\), the expression
\[
c_i u_i + \sum_{k=1}^{N} \int_{\Gamma_k} q^* u d\Gamma - \sum_{k=1}^{N} \int_{\Gamma_k} u^* q d\Gamma = \sum_{j=1}^{N+L} \alpha_j \left( c_i \hat{u}_{ij} + \sum_{k=1}^{N} \int_{\Gamma_k} q^* \hat{u}_j d\Gamma - \sum_{k=1}^{N} \int_{\Gamma_k} u^* \hat{q}_j d\Gamma \right) \tag{3.17}
\]
After introducing the interpolation function and integrating over each boundary element, the above equation can be written in terms of nodal values as
\[
c_i u_i + \sum_{k=1}^{N} H_{ik} u_k - \sum_{k=1}^{N} G_{ik} q_k = \sum_{j=1}^{N+L} \alpha_j \left( c_i \hat{u}_{ij} + \sum_{k=1}^{N} H_{ik} \hat{u}_j - \sum_{k=1}^{N} G_{ik} \hat{q}_j \right) \tag{3.18}
\]
where the terms \(H_{ik}\) and \(G_{ik}\) are defined as in Chapter 1. The index \(k\) is used for the boundary nodes which are the field points. After application to all boundary nodes, using a collocation technique, equation (3.18) can be expressed in matrix form as
\[
\mathbf{H} \mathbf{u} - \mathbf{G} \mathbf{q} = \sum_{j=1}^{N+L} \alpha_j (\mathbf{H} \mathbf{u}_j - \mathbf{G} \mathbf{q}_j)
\]  

(3.19)

If each of the vectors \( \mathbf{u}_j \) and \( \mathbf{q}_j \) is considered to be one column of the matrices \( \mathbf{U} \) and \( \mathbf{Q} \) respectively, then equation (3.19) may be written without the summation in the form

\[
\mathbf{H} \mathbf{u} - \mathbf{G} \mathbf{q} = (\mathbf{H} \mathbf{U} - \mathbf{G} \mathbf{Q}) \alpha.
\]  

(3.20)

Note that equation (3.18) contains no domain integrals. The source term \( b \) in (3.5) has been replaced by equivalent boundary integrals. This was done by first approximating \( b \) using equation (3.11), and then expressing both the right and left hand sides of the resulting expression as boundary integrals using the second form of Green's theorem or a reciprocity principle. It is this operation which gives the name to the method: reciprocity has been applied to both sides of (3.13) to take all the terms to the boundary, hence Dual Reciprocity Boundary Element Method.

3.4 Transformation to Laplace's equation and the MULDRM program

In this section we develop the method in a manner, such that we can easily adapt the existing MULBEM code to solve the Poisson equation. We name the modified program MULDRM. This program firstly uses the radial basis function \( f = 1 + r \).

3.4.1 Mathematical concepts

Consider the Poisson equation (3.5) with the discretisation as shown in Figure 3.3.1. From equation (3.8) we have, for each node

\[
b \approx \sum_{j=1}^{N+L} \alpha_j f_j
\]  

(3.21)
with $\nabla^2 \hat{u}_j = f_j$ where $f_j$ and $\hat{u}_j$ are known, as in equation (3.10).

Following the work of Partridge and Brebbia (1989) we shall expand $b$ in a series of radial basis functions. If $P$ is any point in the domain then a radial basis function may be written in the form $\phi(r)$ where $r$ is the distance of the point $(x, y)$ from $P$ i.e. $r$ is given by $r^2 = (x - x_p)^2 + (y - y_p)^2$.

In the first instance we use the radial basis functions

$$f_j = 1 + r_j$$

Using the relation (3.10) we obtain

$$\hat{u}_j = \frac{r_j^2}{4} + \frac{r_j^3}{9}. \quad (3.23)$$

Set

$$U = u - \sum_{j=1}^{N+L} \alpha_j \hat{u}_j$$

and so

$$\nabla^2 U = \nabla^2 u - \sum_{j=1}^{N+L} \alpha_j \nabla^2 (\hat{u}_j) \quad (3.25)$$

Substituting (3.5) and (3.10) into equation (3.25) we have

$$\nabla^2 U = b - \sum_{j=1}^{N+L} \alpha_j f_j. \quad (3.26)$$

Finally, by substituting (3.8) into equation (3.24), we obtain a new Laplace equation as
\[ \nabla^2 U = 0 \quad (3.27) \]

with boundary conditions

\[ U = \overline{U} = \overline{u} - \sum_{j=1}^{N+L} \alpha_j \hat{u}_j \quad \text{on} \quad \Gamma_1 \quad (3.28) \]

and

\[ Q = \overline{Q} = \overline{q} - \sum_{j=1}^{N+L} \alpha_j \hat{q}_j \quad \text{on} \quad \Gamma_2 \quad (3.29) \]

where \( Q \) is the normal derivative of \( U \).

After the Laplace equation (3.27) has been solved, the values of \( U \) and \( Q \) are known and hence we also obtain the solution of the Poisson equation (3.5) via

\[ u = U + \sum_{j=1}^{N+L} \alpha_j \hat{u}_j \quad (3.30) \]

on the boundary and inside the region

and

\[ q = Q + \sum_{j=1}^{N+L} \alpha_j \hat{q}_j \quad (3.31) \]

on the boundary of the region.

### 3.4.2 The MULDRM program

The MULBEM code is developed using the mathematical idea as described above. We name the modified program MULDRM. This program firstly uses the radial basis function \( f = 1 + r \). Other radial basis functions will be discussed in section 3.5. We now test MULDRM in the following example.
Example 3.4.1 The Poisson problem on an elliptical domain

We refer to the problem given by Partridge and Brebbia (1989). Consider the Poisson equation

\[ \nabla^2 u = b(x, y) \]  \hspace{1cm} (3.32)

on an elliptical domain. The discretisation is shown in Figure 3.4.1. The equation of the ellipse is

\[ \frac{x^2}{4} + y^2 = 1 \]  \hspace{1cm} (3.33)

In this example we take the position function as a constant i.e. \( b(x, y) = -2 \) (torsion problem). The governing equation is

\[ \nabla^2 u = -2 \]  \hspace{1cm} (3.34)

---

Figure 3.4.1 Discretisation of the boundary into 16 elements, 17 internal points, and 4 multiple nodes in elliptical domain
The Dirichlet boundary condition for this example sets \( u = 0 \) on the boundary.

The exact solution is

\[
u = -0.8 \left( \frac{x^2}{4} + y^2 - 1 \right)
\]

(3.35)

The normal derivative is

\[
q = -0.2(x^2 + 8y^2).
\]

(3.36)

We use the number of multiple nodes as 4, 8 and 16. The four multiple nodes are shown in Figure 3.4.1. The internal solutions are compared with the cell integration method (Partridge and Brebbia, 1990) and the exact solution as shown in Table 3.4.1.

Table 3.4.1 Internal solutions for the case \( b(x, y) = -2 \)

<table>
<thead>
<tr>
<th>Internal point</th>
<th>Cell integration</th>
<th>MULDRM (No. of multiple nodes)</th>
<th>Exact solution</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>(1.5,0.0)</td>
<td>0.331</td>
<td>0.344</td>
<td>0.347</td>
</tr>
<tr>
<td>(1.2,-0.35)</td>
<td>0.401</td>
<td>0.420</td>
<td>0.419</td>
</tr>
<tr>
<td>(0.6,-0.45)</td>
<td>0.557</td>
<td>0.576</td>
<td>0.574</td>
</tr>
<tr>
<td>(0.0,-0.45)</td>
<td>0.629</td>
<td>0.648</td>
<td>0.646</td>
</tr>
<tr>
<td>(0.9,0.0)</td>
<td>0.626</td>
<td>0.646</td>
<td>0.644</td>
</tr>
<tr>
<td>(0.3,0.0)</td>
<td>0.772</td>
<td>0.793</td>
<td>0.790</td>
</tr>
<tr>
<td>(0.0,0.0)</td>
<td>0.791</td>
<td>0.810</td>
<td>0.808</td>
</tr>
</tbody>
</table>

We observe from Table 3.4.1 that the results using MULDRM are better than those using cell integration. The points referred to as "shaded" correspond to number of multiple nodes. Percentage errors in potential at the internal points are shown in Figure 3.4.2 for the multiple node in the figure we use 4 multiple nodes.
We see that from Figure 3.4.2 that the MULDRM result is excellent since errors are less than 2\% and the solution using the method is better than that using the cell integration method. However, the multiple node approach does not seem to help very much. In this problem the boundary is smooth and does not contain "real" corners. The normal derivative on the boundary from MULDRM is shown in Figure 3.4.3.

Although MULDRM gives excellent results for the internal solution, we observe from Figure 3.4.3 that it gives poor normal derivatives. We shall see in section 3.5 that the direct use of the dual reciprocity method (not transformed to Laplace's equation) does not suffer in the same way.
We have shown that the Poisson equation can be solved using the transformation to Laplace's equation. However, the normal derivative from MULDRM using this approach is quite poor. Furthermore, the transformation to Laplace's equation cannot apply to the problem whose source terms contain the problem variable or its derivative. The dual reciprocity method together with radial basis functions will be extended to handle such problems in section 3.5 and those which follow.

3.5 Formulation of the dual reciprocity boundary element method

In the previous section we developed the DRM from Poisson's equation in the case $b = b(x, y)$ for which we can transform the partial differential equation to Laplace's equation. In this section we discuss the formulation of the dual reciprocity boundary element method for cases which cannot be transformed to Laplace's equation. We will implement a program using this formulation in section 3.6. Furthermore, it is suitable to be modified for solving non-linear problems later on.

We would expect the approximation of $b$ to be unsatisfactory if we use only boundary nodes. Consequently, we also use internal nodes for the approximation of $b$.

The $\alpha$ vector in equation (3.20) will now be considered. Equation (3.8) gives

$$ b \approx \sum_{j=1}^{N+L} \alpha_j f_j \quad (3.37) $$

and the equation for the unknown, $\alpha_j$, may be written as before

$$ \mathbf{b} = \mathbf{F} \alpha \quad (3.38) $$

where each column of $\mathbf{F}$ consists of a vector $\mathbf{f}_j$ containing the values of the function $f_j$ at the $(N + L)$ DRM collocation points. In the case of the problems considered in this section, the function $b$ in (3.5) and (3.8) is a known function of position. Thus equation (3.38) may be inverted to obtain $\alpha$, i.e.
\[ \mathbf{\alpha} = \mathbf{F}^{-1} \mathbf{b} \]  
(3.39)

The right-hand side of equation (3.20) is thus a known vector. It can be written as

\[ \mathbf{H} \mathbf{u} - \mathbf{G} \mathbf{q} = \mathbf{d} \]  
(3.40)

where the vector \( \mathbf{d} \) is defined as

\[ \mathbf{d} = \left( \mathbf{H} \mathbf{U} - \mathbf{G} \mathbf{Q} \right) \mathbf{\alpha} \]  
(3.41)

Applying boundary conditions to (3.40), this equation reduces to the form

\[ \mathbf{A} \mathbf{x} = \mathbf{y} \]  
(3.42)

where \( \mathbf{x} \) contains \( N \) unknown boundary values of \( u \) and \( q \).

After equation (3.42) is solved using standard techniques such as Gaussian elimination, the values at any internal node can be calculated from equation (3.18), each one involving a separate multiplication of known vectors and matrices. In the case of internal nodes, as explained in section 3.3, \( c_i = 1 \) and (3.18) becomes

\[ u_i = -\sum_{k=1}^{N} H_{ik} u_k + \sum_{k=1}^{N} G_{ik} q_k + \sum_{j=1}^{N} \mathbf{\alpha}_j \left( \hat{u}_{ij} + \sum_{k=1}^{N} H_{ik} \hat{u}_{kj} - \sum_{k=1}^{N} G_{ik} \hat{q}_{kj} \right) \]  
(3.43)

The particular solution, \( \hat{u}_j \), its normal derivative, \( \hat{q}_j \), and the corresponding interpolation functions \( f_j \) used in DRM analysis are not limited by formulation except that the resulting \( \mathbf{F} \) matrix, equation (3.38), should be non-singular.

In order to define these functions it is customary to propose an expansion for \( f \) and then compute \( \hat{u} \) and \( \hat{q} \) using equations (3.10) and (3.16), respectively. The originators of the method have proposed the following types of functions for \( f \)
1. Elements of the Pascal triangle
2. Trigonometric series
3. The distance function $r$ used in the definition of the fundamental solution

The $r$ function was adopted first by Nardini and Brebbia (1983) and then by most researchers as the simplest and most accurate alternative. In this case the function $f$ of $r$ is the radial basis function as used in section 3.4.

Let $r$ be given by $r = r_x i + r_y j$

so that

$$r^2 = r_x^2 + r_y^2$$  \hspace{1cm} (3.44)

For the radial basis function $f = r$, it can easily be shown that the corresponding $\hat{u}$ function is $\frac{r^2}{g}$, in the two-dimensional case.

The function $\hat{q}$ is

$$\hat{q} = \frac{r}{3} \left[ r_x \cos(n, x) + r_y \cos(n, y) \right]$$  \hspace{1cm} (3.45)

In the above, the direction cosine refers to the outward normal at the boundary with respect to the $x$ and $y$ axes. Formula (3.45) may be easily obtained using (3.16) and remembering that $\frac{\partial r}{\partial x} = \frac{r_x}{r}$ and $\frac{\partial r}{\partial y} = \frac{r_y}{r}$.

Furthermore, Partridge and Brebbia (1992) suggest that $f = r$ can in fact be regarded as just one component of the series

$$f = 1 + r + r^2 + ... + r^m$$  \hspace{1cm} (3.46)

The $\hat{u}$ and $\hat{q}$ functions corresponding to (3.46) are:
\[
\hat{u} = \frac{r^2}{4} + \frac{r^3}{9} + \ldots + \frac{r^{m+2}}{(m+2)^2} \quad (3.47)
\]

\[
\hat{q} = \left( r, \frac{\partial x}{\partial n} + r, \frac{\partial y}{\partial n} \right) \left( \frac{1}{2} + \frac{r}{3} + \ldots + \frac{r^{m}}{(m+2)} \right) \quad (3.48)
\]

However, in this work we concentrate on the two frequently-used radial basis functions: the linear function \( f = 1 + r \) and the augmented thin plate spline \( f = r^2 \log r + ax + by + c \).

For the linear function \( f = 1 + r \), we obtain directly from equation (3.47)

\[
\hat{u} = \frac{r^2}{4} + \frac{r^3}{9} ;
\quad (3.49)
\]

and from equation (3.48) we set

\[
\hat{q} = \left( r, \frac{\partial x}{\partial n} + r, \frac{\partial y}{\partial n} \right) \left( \frac{1}{2} + \frac{r}{3} \right)
\quad (3.50)
\]

The augmented thin plate spline, \( f = r^2 \log r + ax + by + c \), is used by Agnantiaris, et al (1996). Powell (1994) proves the uniform convergence of the thin plate spline when augmented with a linear term for function approximation. Golberg et al. (1998) suggest that the poor results in Agnantiaris et al. (1994) are due to the fact that the authors do not augment the thin plate spline with a linear term. Toutip (2000) also investigates both radial basis functions and finds that, in general, the approximation using the augmented thin plate spline is better than that using \( f = 1 + r \).

The augmented thin plate spline is employed to approximate the function \( b(x, y) \) as

\[
b(x, y) \approx \sum_{j=1}^{N+L} \alpha_j r_j^2 \log r_j + a + bx + cy
\quad (3.51)
\]

61
where \( \sum_{j=1}^{N+L} \alpha_j = \sum_{j=1}^{N+L} \alpha_j x_j = \sum_{j=1}^{N+L} \alpha_j y_j = 0 \)

To compute \( \alpha_j \) we use the \( N + L + 3 \) radial basis function

\[
f_k = r_k^2 \log r_k \quad \text{for} \quad k = 1, 2, 3, \ldots, N + L \quad (3.52)
\]

augmented with the linear functions

\[
f_{N+L+1} = 1 \quad (3.53)
\]

\[
f_{N+L+2} = x \quad (3.54)
\]

\[
f_{N+L+3} = y \quad (3.55)
\]

Using relation (3.10) we obtain

\[
\hat{u}_k = \frac{r_k^4 \log r_k - r_k^4}{16} \quad \text{for} \quad k = 1, 2, 3, \ldots, N + L \quad (3.56)
\]

\[
\hat{u}_{N+L+1} = \frac{x^2 + y^2}{4} \quad (3.57)
\]

\[
\hat{u}_{N+L+2} = \frac{x^3}{6} \quad (3.58)
\]

\[
\hat{u}_{N+L+3} = \frac{y^3}{6} \quad (3.59)
\]

and applying equation (3.16) we obtain

\[
\hat{q}_k = \left( \frac{r_k^2 \log r_k - r_k^2}{4} \right) \left( r_x \frac{\partial \hat{u}}{\partial n} + r_y \frac{\partial \hat{u}}{\partial n} \right) \quad \text{for} \quad k = 1, 2, 3, \ldots, N + L \quad (3.60)
\]

\[
\hat{q}_{N+L+1} = \frac{x \frac{\partial \hat{u}}{\partial n}}{2} \quad (3.61)
\]

\[
\hat{q}_{N+L+2} = \frac{x^2 \frac{\partial \hat{u}}{\partial n}}{2} \quad (3.62)
\]

\[
\hat{q}_{N+L+3} = \frac{y^2 \frac{\partial \hat{u}}{\partial n}}{2} \quad (3.63)
\]
There are various comments about these two radial basis functions. Golberg (1995) suggested that the augmented thin plate spline might be a better and more mathematically defensible choice because of its convergence to the exact solution and the accuracy of solutions. However, several authors indicated that in practice the thin plate spline showed little or no improvement over the traditional one (Agnantiaris et al., 1996). More comments can be found in Golberg et al. (1998). To make our comments about this subject we will use both radial basis functions in the program in section 3.6.

### 3.6 The DRBEM1 program

DRBEM1 program is based on the idea of the previous section. We note that in this section the implementation of the boundary and internal solutions are separate processes. The first part of this section details the scheme of matrix equations of the program, using both radial basis functions described in section 3.5. The second part discusses computational results from DRBEM1. They are compared with those obtained using MULDRM. The last sub-section investigates the convergence of solutions from both radial basis functions. In this section and subsequently, Linear stands for the linear radial basis function $f = 1 + r$ and ATPS for the augmented thin plate spline $f = r^2 \log r + ax + by + c$.

#### 3.6.1 Schematic form of the matrix equation

The basic system of equations for the DRM is derived from equation (3.20) as

$$Hu - Gq = (H\hat{U} - G\hat{Q})\alpha$$  \hspace{1cm} (3.64)

It can be schematized as shown in Figure 3.6.1.

![Figure 3.6.1 Scheme of system of equations for the boundary solution](image-url)
Notice that the matrices on the right hand side are known. \( \mathbf{H} \) and \( \mathbf{G} \) are from the main frame work of the standard collocation method. The matrix \( \hat{\mathbf{U}} \) is derived from equation (3.49) for Linear and from equations (3.56) to (3.59) for ATPS. The matrix \( \hat{\mathbf{Q}} \) is derived from equation (3.50) for Linear and from equations (3.60) to (3.63) for ATPS. The vector \( \mathbf{\alpha} \) is obtained by solving equation (3.38). The result on the right hand side of (3.64) will sometimes be stored by \( \mathbf{d} \) as in (3.40). After applying the boundary conditions, the system (3.64) will be solved. Having obtained the values of unknowns on the boundary, the next step is to calculate values at the internal points. For this, equation (3.63)

\[
\mathbf{u}_i = - \sum_{k=1}^{N} H_{ik} u_k + \sum_{k=1}^{N} G_{ik} q_k + \sum_{j=1}^{N+L} \alpha_j \left( \hat{u}_j + \sum_{k=1}^{N} H_{jk} \hat{u}_k - \sum_{k=1}^{N} G_{jk} \hat{q}_k \right)
\]  

(3.65)

is used as in the schematized form in Figure 3.6.2.

![Figure 3.6.2 Scheme of system of equations for the internal solution](image)

The scheme will be described term by term. On the left hand side the identity matrix \( \mathbf{I} \) of size \( L \times L \) multiplies the \( L \) values of \( \mathbf{u} \) at internal nodes. This is due to the \( c_i \) terms, which are all unity at internal nodes. Obviously, values of \( \mathbf{u} \) and \( \mathbf{q} \) are obtained by solving equation (3.64). The same matrix appears in the last term on the right hand side. The \( \mathbf{H} \) and \( \mathbf{G} \) matrices partitions of size \( L \times N \) which appear on both sides of equation (3.65) are produced by integrating over the boundary from each internal node. The matrices \( \hat{\mathbf{U}} \) and \( \hat{\mathbf{Q}} \) together with the vector \( \mathbf{\alpha} \) are the same as used in equation (3.64).

We emphasise that DRBEM1 uses both schemes for its implementation. However, the process to execute the internal solution is separated from the boundary solution. This

64
can be done only in the case that the function $b$ is known. For the case when the function $b$ contains the unknown, $u$, the two schemes will be integrated in one global scheme. We will discuss this case in section 3.7. Next we will investigate the computational results of the program.

3.6.2 Computational results

In this sub-section we test DRBEM1 on a problem already solved by MULDRM in section 3.4.

**Example 3.6.1** The Poisson problem on an elliptical domain

This is the same problem as in Example 3.4.1.

The internal solutions obtained using DRBEM1 are compared with those from MULDRM, and the exact solution as shown in Table 3.6.1.

Table 3.6.1 Internal solutions for the problem in Example 3.6.1

<table>
<thead>
<tr>
<th>Internal point</th>
<th>MULDRM</th>
<th>DRBEM1</th>
<th>Exact solution</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Standard</td>
<td>Multiple</td>
<td>Linear</td>
</tr>
<tr>
<td>(-1.5,0)</td>
<td>0.344</td>
<td>0.347</td>
<td>0.349</td>
</tr>
<tr>
<td>(-0.9,0)</td>
<td>0.647</td>
<td>0.644</td>
<td>0.643</td>
</tr>
<tr>
<td>(-0.3,0)</td>
<td>0.793</td>
<td>0.790</td>
<td>0.789</td>
</tr>
<tr>
<td>(0,0)</td>
<td>0.811</td>
<td>0.808</td>
<td>0.807</td>
</tr>
<tr>
<td>(0.3,0)</td>
<td>0.793</td>
<td>0.790</td>
<td>0.789</td>
</tr>
<tr>
<td>(0.9,0)</td>
<td>0.646</td>
<td>0.644</td>
<td>0.643</td>
</tr>
<tr>
<td>(1.5,0)</td>
<td>0.344</td>
<td>0.347</td>
<td>0.349</td>
</tr>
</tbody>
</table>

We can see from Table 3.6.1 that it is difficult to distinguish which is the better result. Percentage errors of internal solutions are calculated to clarify as shown in Figure 3.6.3.
We can see from Figure 3.6.3 that all solutions give relative errors less than 2% but the MULDRM result using the standard collocation is poorer among the other methods.

Normal derivatives are shown in Figure 3.6.4 and we see that the normal derivative on the boundary from DRBEM1 for both radial basis functions agrees well with the exact solution.
In the rest of this section we investigate the convergence of solutions obtained from *Linear* and *ATPS*.

### 3.6.3 Convergence of solution by radial basis functions

This subsection we examine the convergence of the solution obtained by both radial basis functions: *Linear* and *ATPS*. We investigate two problems. The first problem is from Example 3.6.1. We note that in such problems the boundary $\Gamma$ is different from $\Gamma_N$. The second problem is a mixed problem with both Dirichlet and Neumann boundary conditions. Furthermore, some parts of the boundary $\Gamma_N$ are the same as $\Gamma$.

#### Example 3.6.2 The Poisson problem on an elliptical domain

This is the same problem as in Example 3.6.1. To examine the convergence, we partition the boundary into 8, 16, 32, 64 and 128 elements with 17 internal points.

The internal solution for each partition is considered for both radial basis functions. Results are shown in Table 3.6.2 and Figure 3.6.5.

We can see that all solutions are covered to within 3% when the number of elements is greater than or equal to 16. The results from *ATPS* are better than those from *Linear*. The convergence of the internal solution at the point (1.5, 0) is shown in Figure 3.6.5.

<table>
<thead>
<tr>
<th>Point</th>
<th>Number of elements</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>8</td>
<td>16</td>
<td>32</td>
<td>64</td>
<td>128</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Linear</td>
<td>ATPS</td>
<td>Linear</td>
<td>ATPS</td>
<td>Linear</td>
<td>ATPS</td>
<td>Linear</td>
<td>ATPS</td>
</tr>
<tr>
<td>(1.5,0.0)</td>
<td></td>
<td>5.259</td>
<td>4.079</td>
<td>0.749</td>
<td>0.885</td>
<td>0.123</td>
<td>0.212</td>
<td>0.052</td>
<td>0.052</td>
</tr>
<tr>
<td>(1.2,-0.35)</td>
<td></td>
<td>7.039</td>
<td>6.282</td>
<td>1.809</td>
<td>1.700</td>
<td>0.193</td>
<td>0.407</td>
<td>0.100</td>
<td>0.100</td>
</tr>
<tr>
<td>(0.6,-0.45)</td>
<td></td>
<td>8.895</td>
<td>8.362</td>
<td>2.279</td>
<td>2.053</td>
<td>0.359</td>
<td>0.501</td>
<td>0.123</td>
<td>0.123</td>
</tr>
<tr>
<td>(0.0,-0.45)</td>
<td></td>
<td>9.048</td>
<td>7.615</td>
<td>2.265</td>
<td>2.034</td>
<td>0.372</td>
<td>0.500</td>
<td>0.123</td>
<td>0.123</td>
</tr>
<tr>
<td>(0.0,0.0)</td>
<td></td>
<td>7.495</td>
<td>6.207</td>
<td>1.68</td>
<td>1.534</td>
<td>0.25</td>
<td>0.374</td>
<td>0.092</td>
<td>0.092</td>
</tr>
<tr>
<td>(0.3,0.0)</td>
<td></td>
<td>7.597</td>
<td>6.300</td>
<td>1.663</td>
<td>1.525</td>
<td>0.242</td>
<td>0.371</td>
<td>0.091</td>
<td>0.091</td>
</tr>
<tr>
<td>(0.9,0.0)</td>
<td></td>
<td>7.575</td>
<td>6.245</td>
<td>1.501</td>
<td>1.415</td>
<td>0.175</td>
<td>0.343</td>
<td>0.084</td>
<td>0.084</td>
</tr>
</tbody>
</table>
We see from Figure 3.6.5 that the errors using ATPS are less than those using Linear at almost every internal point. However, the solution using Linear or ATPS converges to the exact solution as the number of elements increases.

Example 3.6.3 The mixed problem with exponential function right-hand side

Consider the Poisson equation

\[
\nabla^2 u = 5e^{x+y}
\]

(3.66)

on the positive quadrant of a unit-circle \( x^2 + y^2 = 1 \). Discretisation and boundary conditions are shown in Figure 3.6.6.

To examine the convergence, we partition the boundary into 12, 24, 48, 96 and 144 elements with 8 internal points. The partition into 24 elements is shown in Figure 3.6.6. The internal solution for each partition is obtained for both radial basis functions. Results are shown in Table 3.6.3.
We can see from Table 3.6.3 that when the number of elements is more than or equal to 24 errors are less than 3%. The percentage errors from ATPS are less than those from Linear except for some points near corners such as (0.25, 0.25). The convergence of the internal solution at the point (0.75, 0.25) is shown in Figure 3.6.7.

Table 3.6.3 Percentage errors in the internal solution for each partition

<table>
<thead>
<tr>
<th>Point</th>
<th>12</th>
<th>24</th>
<th>48</th>
<th>96</th>
<th>144</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear</td>
<td>ATPS</td>
<td>Linear</td>
<td>ATPS</td>
<td>Linear</td>
<td>ATPS</td>
</tr>
<tr>
<td>(0.75,0.25)</td>
<td>1.837</td>
<td>1.065</td>
<td>0.559</td>
<td>0.227</td>
<td>0.361</td>
</tr>
<tr>
<td>(0.50,0.25)</td>
<td>2.354</td>
<td>2.012</td>
<td>1.028</td>
<td>0.9</td>
<td>0.581</td>
</tr>
<tr>
<td>(0.25,0.25)</td>
<td>4.482</td>
<td>5.187</td>
<td>1.999</td>
<td>2.619</td>
<td>0.901</td>
</tr>
<tr>
<td>(0.25,0.50)</td>
<td>1.95</td>
<td>2.412</td>
<td>0.996</td>
<td>1.218</td>
<td>0.577</td>
</tr>
<tr>
<td>(0.50,0.50)</td>
<td>1.793</td>
<td>1.648</td>
<td>0.822</td>
<td>0.614</td>
<td>0.533</td>
</tr>
<tr>
<td>(0.75,0.50)</td>
<td>1.598</td>
<td>3.612</td>
<td>0.521</td>
<td>0.366</td>
<td>0.269</td>
</tr>
<tr>
<td>(0.50,0.75)</td>
<td>0.996</td>
<td>3.606</td>
<td>0.297</td>
<td>0.316</td>
<td>0.195</td>
</tr>
</tbody>
</table>
We can see from Figure 3.6.7 that the error using ATPS is less than that of using Linear. The solution converges to the exact solution as the number of elements increases.

We have shown that the solution from DRBEM1 using ATPS is better than that using Linear provided we have enough elements. However, DRBEM1 needs to improve in two ways. The first is the treatment of corners. This feature will be discussed in section 4.1. The second is to solve Poisson-type problems when the right-hand side function $b$ contains the problem variable. We discuss this feature in section 3.7.

### 3.7 The DRBEM2 Program

In this section we discuss how to modify DRBEM1 in order to solve various Poisson-type problems. We note that the process of finding the internal solution using DRBEM1 is separate from finding the boundary solution. For Poisson-type problems in which the right-hand side contains the problem variable or its derivative, we are not able to simply expand this term. We set up a solution process in which the boundary solution and the internal solution are obtained simultaneously. There is a cost,
however, in that this approach requires the inverse of an \((N+L) \times (N+L)\) matrix.

The program DRBEM2 is a modification of the DRBEM1 which finds all variables at the same time. Furthermore, it includes both radial basis functions. We explain how to implement it in the following subsection.

### 3.7.1 Schematic form of the matrix equation

Consider the Poisson equation

\[
\nabla^2 u = b
\]

(3.67)

As described in the formulation of the dual reciprocity method, the matrix form of the system of equations is obtained from equations (3.39) to (3.41) as

\[
Hu - Gq = (H\hat{U} - G\hat{Q})F^{-1}b
\]

(3.68)

For a discretisation with \(N\) boundary nodes and \(L\) internal points, this may be represent in schematic form as shown in Figure 3.7.1.

![Figure 3.7.1 Scheme of the whole system of equations, after Partridge et al. (1992)](image)

The definition of the symbols in the scheme is as follows:

- **0**: The zero submatrices
- **I**: The identity submatrices
- **BS**: The matrices associated exclusively with boundary nodes
- **IS**: The matrices associated exclusively with internal nodes
- **BS+IS**: The matrices associated with both boundary nodes and internal nodes
We note that the scheme in Figure 3.7.1 is derived from the two schemes in Figure 3.6.1 and Figure 3.6.2.

In general, the right hand side of equation (3.67) is an unknown function so that the right hand side of equation (3.68) is unknown.

Define

\[ S = (H \hat{U} - G \hat{Q})F^{-1} \]  

then equation (3.68) becomes

\[ Hu - Gq = Sb \]  

Equation (3.70) is the formulation for a program to solve Poisson-type problems. We discuss how to solve the problem for different forms of the function \( b \) as follows:

3.7.2 Poisson's equation \( \nabla^2 u = b(x, y) \)

In this case, the vector \( b \) in formulation (3.70), is known.

Setting

\[ R = Sb \]  

and substituting equation (3.71) in equation (3.70) we obtain

\[ Hu = Gq + R \]  

where the matrix \( R \) is known.

Applying the boundary condition as described in equation (1.30) we obtain the system of equations in matrix form as

\[ Ax = y \]
To test the program, we examine a few examples for this kind of Poisson equation as follows:

**Example 3.7.1** The case \( b(x, y) = 5e^{2x+y} \)

As in Example 3.6.3 we consider the Poisson equation

\[
\nabla^2 u = 5e^{2x+y}
\]

where the boundary conditions are as in the mixed problem of Example 3.6.3.

The potential solutions at the internal nodes are shown in Table 3.7.1. We observe from Table 3.7.1 that the solutions from both programs are almost identical. This fact is true for all problems in section 3.6. We note that the simultaneous solution for internal and boundary values yields the same solution as the separated approach. Hence it is sufficient to investigate only DRBEM2 for the rest of this section.

Table 3.7.1 The potential at the internal nodes of Example 3.7.1

<table>
<thead>
<tr>
<th>x</th>
<th>y</th>
<th>DRBEM 1</th>
<th>DRBEM 2</th>
<th>Exact solution</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>f = 1+r</td>
<td>ATPS</td>
<td>f = 1+r</td>
</tr>
<tr>
<td>0.75</td>
<td>0.25</td>
<td>5.7219</td>
<td>5.7410</td>
<td>5.7219</td>
</tr>
<tr>
<td>0.50</td>
<td>0.25</td>
<td>3.4545</td>
<td>3.4589</td>
<td>3.4544</td>
</tr>
<tr>
<td>0.25</td>
<td>0.25</td>
<td>2.0747</td>
<td>2.0616</td>
<td>2.0746</td>
</tr>
<tr>
<td>0.25</td>
<td>0.50</td>
<td>2.6912</td>
<td>2.6852</td>
<td>2.6912</td>
</tr>
<tr>
<td>0.50</td>
<td>0.50</td>
<td>4.4449</td>
<td>4.4542</td>
<td>4.4448</td>
</tr>
<tr>
<td>0.75</td>
<td>0.50</td>
<td>7.3506</td>
<td>7.3620</td>
<td>7.3506</td>
</tr>
<tr>
<td>0.50</td>
<td>0.75</td>
<td>5.7375</td>
<td>5.7364</td>
<td>5.7377</td>
</tr>
</tbody>
</table>

The normal derivative on the boundary as shown in Figure 3.7.2 agrees well with the exact value except at the corners. We shall discuss how to handle such problems in section 4.1.
3.7.3 The Poisson-type equation \( \nabla^2 u = b(x, y, u, \frac{\partial u}{\partial x}, \frac{\partial u}{\partial y}) \)

Consider the case

\[
b = c_1(x, y)u + c_2(x, y) \frac{\partial u}{\partial x} + c_3(x, y) \frac{\partial u}{\partial y}
\]

where \( c_k(x, y), \ k = 1, 2, 3 \) is a function of position. If we collocate at the nodes then we can rewrite (3.75) in the matrix form as

\[
b = C_ku + C_2 \frac{\partial u}{\partial x} + C_3 \frac{\partial u}{\partial y}
\]

where \( C_k, \ k = 1, 2, 3 \) is the diagonal matrix

\[
C_k = \begin{bmatrix}
c_k(x_1, y_1) & 0 & 0 & \ldots & 0 \\
0 & c_k(x_2, y_2) & 0 & \ldots & 0 \\
0 & 0 & c_k(x_3, y_3) & \ldots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & 0 & \ldots & c_k(x_{N+L}, y_{N+L})
\end{bmatrix}
\]
\[
\frac{\partial \mathbf{u}}{\partial x} = \left[ \frac{\partial u}{\partial x}(x_1, y_1), \frac{\partial u}{\partial x}(x_2, y_2), \ldots, \frac{\partial u}{\partial x}(x_{N+L}, y_{N+L}) \right]^T \tag{3.78}
\]

and
\[
\frac{\partial \mathbf{u}}{\partial y} = \left[ \frac{\partial u}{\partial y}(x_1, y_1), \frac{\partial u}{\partial y}(x_2, y_2), \ldots, \frac{\partial u}{\partial y}(x_{N+L}, y_{N+L}) \right]^T \tag{3.79}
\]

The basic approximation of the DRM technique, as described in section 3.5, is

\[
\mathbf{b} = \mathbf{F} \alpha \tag{3.80}
\]

A similar equation may be written for the nodal value of \( u \) in matrix form as

\[
\mathbf{u} = \mathbf{F} \beta \tag{3.81}
\]

Differentiating (3.81) produces

\[
\frac{\partial \mathbf{u}}{\partial x} = \frac{\partial \mathbf{F}}{\partial x} \beta \tag{3.82}
\]

where \( \frac{\partial \mathbf{F}}{\partial x} \) is a matrix where the \( j \)th column is

\[
\left[ \frac{\partial f_j}{\partial x}(x_1, y_1), \frac{\partial f_j}{\partial x}(x_2, y_2), \ldots, \frac{\partial f_j}{\partial x}(x_{N+L}, y_{N+L}) \right]^T \tag{3.83}
\]

N.B. With \( f = 1 + r \)

\[
\frac{\partial f_j}{\partial x}(x, y) = \frac{x - x_j}{r_j}
\]

Now at node \( j \), \( r_j = 0 \), but also \( x - x_j = 0 \) and we take

\[
\frac{\partial f_j}{\partial x}(x, y) = 0
\]
Rewriting equation (3.81) as $\beta = F^{-1}u$, then (3.82) becomes

$$\frac{\partial u}{\partial x} = \frac{\partial F}{\partial x} F^{-1}u$$  \hspace{1cm} (3.84)

In a similar manner for $\frac{\partial u}{\partial y}$, we obtain

$$\frac{\partial u}{\partial y} = \frac{\partial F}{\partial y} F^{-1}u$$  \hspace{1cm} (3.85)

Substituting equation (3.84) in (3.76) and then in (3.70) we obtain

$$Hu - Gq = S(C_1u + C_2 \frac{\partial F}{\partial x} F^{-1}u + C_3 \frac{\partial F}{\partial y} F^{-1}u)$$  \hspace{1cm} (3.86)

Setting

$$R = S(C_1 + C_2 \frac{\partial F}{\partial x} F^{-1} + C_3 \frac{\partial F}{\partial y} F^{-1})$$  \hspace{1cm} (3.87)

and substituting equation (3.87) in equation (3.86) and rearranging we obtain

$$(H - R)u = Gq$$  \hspace{1cm} (3.88)

Applying the boundary condition as described in equation (1.30) we obtain the system of equations in matrix form as

$$Ax = y$$  \hspace{1cm} (3.89)

**Example 3.7.2** The case $c_1 = -1, \; c_2 = c_3 = 0$

Consider the Poisson-type equation

$$\nabla^2 u = -u$$  \hspace{1cm} (3.90)
on the domain shown in Figure 3.4.4 of Example 3.4.1. We consider the Dirichlet problem for which the exact solution is

\[ u = \sin x \]  \hspace{1cm} (3.91)

The solution of the problem is also compared with the DRM solution given by Partridge and Brebbia (1990) (written \( P \) and \( B \) in the table that follow) and the exact solution in Table 3.7.2.

We see from Table 3.7.2 that DRBEM2 works slightly better than the reference program. In this example, it is clear that the solution using the thin plate spline is better than that using the other radial basis functions.

Table 3.7.2  The potential at the internal nodes of the problem

<table>
<thead>
<tr>
<th>Point</th>
<th>P and B</th>
<th>DRBEM 2</th>
<th>Exact</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>x</td>
<td>y</td>
<td>( f = 1 + r )</td>
<td>( f = r + 2 + 3 )</td>
</tr>
<tr>
<td>1.50</td>
<td>0.00</td>
<td>0.994</td>
<td>0.995</td>
</tr>
<tr>
<td>1.20</td>
<td>-0.35</td>
<td>0.928</td>
<td>0.932</td>
</tr>
<tr>
<td>0.60</td>
<td>-0.45</td>
<td>0.562</td>
<td>0.566</td>
</tr>
<tr>
<td>0.00</td>
<td>-0.45</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>0.90</td>
<td>0.00</td>
<td>0.780</td>
<td>0.784</td>
</tr>
<tr>
<td>0.30</td>
<td>0.00</td>
<td>0.294</td>
<td>0.296</td>
</tr>
<tr>
<td>0.00</td>
<td>0.00</td>
<td>0.000</td>
<td>0.000</td>
</tr>
</tbody>
</table>

We see from Figure 3.7.3 that the normal derivative on the boundary agrees with the exact value.
Example 3.7.3 The case \( c_1 = 0, \ c_2 = c_3 = -1 \)

Consider the Poisson-type equation

\[
\nabla^2 u = - \frac{\partial u}{\partial x} - \frac{\partial u}{\partial y}
\]

(3.92)

on the domain as in Figure 3.4.1 of Example 3.4.1. We consider the Dirichlet problem for which

\[
u = e^{-x} + e^{-y}
\]

(3.93)

The solution of the problem is also compared with the DRM solution given by Partridge and Brebbia (1990) and the exact solutions in Table 3.7.3.

As we expected, we can see from Table 3.7.3 that the solution using \textit{ATPS} is better at almost every internal point.
Table 3.7.3 The internal solution of the problem in Example 3.7.3

<table>
<thead>
<tr>
<th>Point</th>
<th>P and B</th>
<th>DRBEM 2</th>
<th>Exact solution</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>x</td>
<td>y</td>
<td>f = 1+r</td>
</tr>
<tr>
<td>1.50</td>
<td>0.00</td>
<td>1.231</td>
<td>1.214</td>
</tr>
<tr>
<td>1.20</td>
<td>-0.35</td>
<td>1.714</td>
<td>1.669</td>
</tr>
<tr>
<td>0.60</td>
<td>-0.45</td>
<td>2.107</td>
<td>2.057</td>
</tr>
<tr>
<td>0.00</td>
<td>-0.45</td>
<td>2.557</td>
<td>2.547</td>
</tr>
<tr>
<td>0.90</td>
<td>0.00</td>
<td>1.400</td>
<td>1.345</td>
</tr>
<tr>
<td>0.30</td>
<td>0.00</td>
<td>1.731</td>
<td>1.691</td>
</tr>
<tr>
<td>0.00</td>
<td>0.00</td>
<td>1.989</td>
<td>1.963</td>
</tr>
</tbody>
</table>

As in the previous example we also see from Figure 3.7.4 that the normal derivative agrees well with the exact solution.

![Figure 3.7.4 Normal derivative along the boundary from DRBEM2](image)

From Example 3.7.1 to 3.7.3, we have shown that, when the boundary is smooth, DRBEM2 using *ATPS* works slightly better than Partridge and Brebbia method (1990) using *Linear*. However, we need to resolve corner problems as we mentioned previously. We discuss a modification of DRBEM2 to deal with such problems in Chapter 4.
3.8 Concluding remarks

We have developed programs to solve a variety of linear problems. We begin with Laplace's equation. The existing program LINBEM using the standard collocation method works well with smooth boundary problems. However, for corner and discontinuous boundary condition problems, LINBEM gives poor solutions at the corners and the points with discontinuous boundary conditions. MULBEM, using the multiple node method with auxiliary boundary collocation approach, can resolve such problems. GRABEM using the gradient approach is an alternative to resolve such problems. Furthermore, it is suitable to apply in the dual reciprocity method.

For Poisson's equations, with the right-hand side function of position, there are two ways to handle this problem. The first one uses the idea of transformation from Poisson's equation to Laplace's equation. In this case, MULBEM is modified to MULDRM. It gives a good result for the potential function but quite poor in the normal derivative. The second one uses the direct dual reciprocity method. In this case the DRBEM1 is implemented and we found that the normal derivative from DRBEM1 is better than that from MULDRM.

For Poisson-type equations, the transformation to Laplace's equation is not available. DRBEM1 is modified to DRBEM2 using the formulation of the dual reciprocity method. DRBEM2 can solve various Poisson-type equations but needs to be improved to handle the corner problems and we shall discussed in Chapter 4.

For radial basis functions, so far we have shown that the augmented thin plate spline $f = r^2 \log r + a + bx + cy$ works, in general, better than the linear function $f = 1 + r$. We will investigate more in Chapter 4.