Chapter 1
TWO–DIMENSIONAL LAPLACE'S EQUATION

1.1 Introduction

Perhaps a good starting point for introducing boundary element methods is through solving boundary value problems governed by the two-dimensional Laplace's equation

$$\frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} = 0. \tag{1.1}$$

The Laplace's equation occurs in the formulation of problems in many diverse fields of studies in engineering and physical sciences, such as thermostatics, elastostatics, electrostatics, magnetostatics, ideal fluid flow and flow in porous media.

An interior boundary value problem which is of practical interest requires solving Eq. (1.1) in the two-dimensional region R (on the Oxy plane) bounded by a simple closed curve C subject to the boundary conditions

$$\begin{aligned} \phi &= f_1(x,y) \text{ for } (x,y) \in C_1, \\ \frac{\partial \phi}{\partial n} &= f_2(x,y) \text{ for } (x,y) \in C_2, \end{aligned} \tag{1.2}$$

where f_1 and f_2 are suitably prescribed functions and C_1 and C_2 are non-intersecting curves such that $C_1 \cup C_2 = C$. Refer to Figure 1.1 for a geometrical sketch of the problem.

The normal derivative $\partial\phi/\partial n$ in Eq. (1.2) is defined by

$$\frac{\partial \phi}{\partial n} = n_x \frac{\partial \phi}{\partial x} + n_y \frac{\partial \phi}{\partial y}, \tag{1.3}$$

where n_x and n_y are respectively the x and y components of a unit normal vector to the curve C. Here the unit normal vector $[n_x, n_y]$ on C is taken to be pointing away from the region R. Note that the normal vector may vary from point to point on C. Thus, $[n_x, n_y]$ is a function of x and y.

The boundary conditions given in Eq. (1.2) are assumed to be properly posed so that the boundary value problem has a unique solution, that is, it is assumed that one can always find a function $\phi(x,y)$ satisfying Eqs. (1.1)-(1.2) and that there is only one such function.

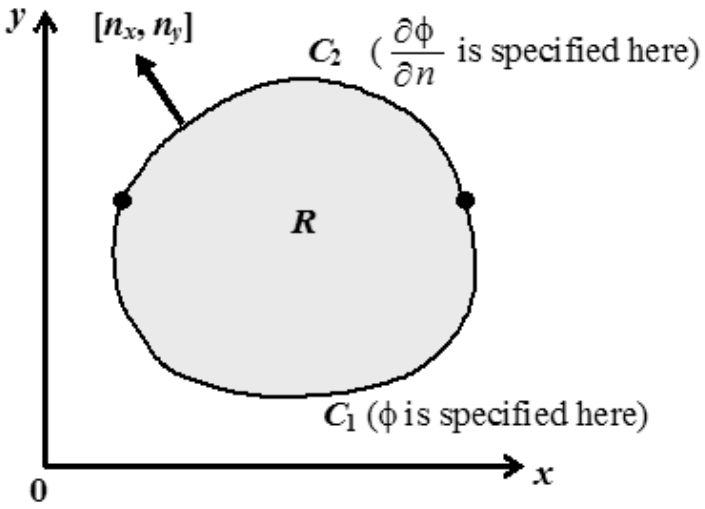

Figure 1.1

For a particular example of practical situations involving the boundary value problem above, one may mention the classical heat conduction problem where ϕ denotes the steady-state temperature in an isotropic solid. Eq. (1.1) is then the temperature governing equation derived, under certain assumptions, from the law of conservation of heat energy together with the Fourier's heat flux model. The heat flux out of the region R across the boundary C is given by $-\kappa\partial\phi/\partial n$, where κ is the thermal heat conductivity of the solid. Thus, the boundary conditions in Eq. (1.2) imply that at each and every given point on C either the temperature or the heat flux (but not both) is known. To determine the temperature field in the solid, one has to solve Eq. (1.1) in R to find the solution that satisfies the prescribed boundary conditions on C.

In general, it is difficult (if not impossible) to solve exactly the boundary value problem defined by Eqs. (1.1)-(1.2). The mathematical complexity involved depends on the geometrical shape of the region R and the boundary conditions given in Eq. (1.2). Exact solutions can only be found for relatively simple geometries of R (such as a square region) together with particular boundary conditions. For more complicated geometries or general boundary conditions, one may have to resort to numerical (approximate) techniques for solving Eqs. (1.1)-(1.2).

This chapter introduces a boundary element method for the numerical solution of the interior boundary value problem defined by Eqs. (1.1)-(1.2). We show how a boundary integral solution can be derived for Eq. (1.1) and applied to obtain a simple boundary element procedure for approximately solving the boundary value problem under consideration. The implementation of the numerical procedure on the computer, achieved through coding in FORTRAN 77, is discussed in detail.

1.2 Fundamental Solution

If we use polar coordinates r and θ centered about $(0,0)$, as defined by $x = r\cos\theta$ and $y = r\sin\theta$, and introduce $\psi(r,\theta) = \phi(r\cos\theta, r\sin\theta)$, we can rewrite Eq. (1.1) as

$$\frac{1}{r}\frac{\partial}{\partial r}(r\frac{\partial \psi}{\partial r}) + \frac{1}{r^2}\frac{\partial^2 \psi}{\partial \theta^2} = 0. \tag{1.4}$$

For the case in which ψ is independent of θ, that is, if ψ is a function of r alone, Eq. (1.4) reduces to the ordinary differential equation

$$\frac{d}{dr}(r\frac{d}{dr}[\psi(r)]) = 0 \text{ for } r \neq 0. \tag{1.5}$$

The ordinary differential equation in Eq. (1.5) can be easily integrated twice to yield the general solution

$$\psi(r) = A\ln(r) + B, \tag{1.6}$$

where A and B are arbitrary constants.

From (1.6), it is obvious that the two-dimensional Laplace's equation in Eq. (1.1) admits a class of particular solutions given by

$$\phi(x,y) = A\ln\sqrt{x^2+y^2} + B \text{ for } (x,y) \neq (0,0). \tag{1.7}$$

If we choose the constants A and B in (1.7) to be $1/(2\pi)$ and 0 respectively and shift the center of the polar coordinates from $(0,0)$ to the general point (ξ,η), a particular solution of Eq. (1.1) is

$$\phi(x,y) = \frac{1}{2\pi}\ln\sqrt{(x-\xi)^2+(y-\eta)^2} \text{ for } (x,y) \neq (\xi,\eta). \tag{1.8}$$

As we shall see, the particular solution in Eq. (1.8) plays an important role in the development of boundary element methods for the numerical solution of the interior boundary value problem defined by Eqs. (1.1)-(1.2). We specially denote this particular solution using the symbol $\Phi(x,y;\xi,\eta)$, that is, we write

$$\Phi(x,y;\xi,\eta) = \frac{1}{4\pi}\ln[(x-\xi)^2+(y-\eta)^2]. \tag{1.9}$$

We refer to $\Phi(x,y;\xi,\eta)$ in Eq. (1.9) as the fundamental solution of the two-dimensional Laplace's equation. Note that $\Phi(x,y;\xi,\eta)$ satisfies Eq. (1.1) everywhere except at (ξ,η) where it is not well defined.

1.3 Reciprocal Relation

If ϕ_1 and ϕ_2 are any two solutions of Eq. (1.1) in the region R bounded by the simple closed curve C then it can be shown that

$$\int_C (\phi_2 \frac{\partial \phi_1}{\partial n} - \phi_1 \frac{\partial \phi_2}{\partial n}) ds(x,y) = 0. \tag{1.10}$$

Eq. (1.10) provides a reciprocal relation between any two solutions of the Laplace's equation in the region R bounded by the curve C. It may be derived from the two-dimensional version of the Gauss-Ostrogradskii (divergence) theorem as explained below.

According to the divergence theorem, if $\underline{\mathbf{F}} = u(x,y)\underline{\mathbf{i}} + v(x,y)\underline{\mathbf{j}}$ is a well defined vector function such that $\underline{\nabla} \cdot \underline{\mathbf{F}} = \partial u/\partial x + \partial v/\partial y$ exists in the region R bounded by the simple closed curve C then

$$\int_C \underline{\mathbf{F}} \cdot \underline{\mathbf{n}} \; ds(x,y) = \iint_R \underline{\nabla} \cdot \underline{\mathbf{F}} \; dxdy,$$

that is,

$$\int_C [u n_x + v n_y] ds(x,y) = \iint_R [\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}] dxdy,$$

where $\underline{\mathbf{n}} = [n_x, n_y]$ is the unit normal vector to the curve C, pointing away from the region R.

Since ϕ_1 and ϕ_2 are solutions of Eq. (1.1), we may write

$$\begin{aligned} \frac{\partial^2 \phi_1}{\partial x^2} + \frac{\partial^2 \phi_1}{\partial y^2} &= 0, \\ \frac{\partial^2 \phi_2}{\partial x^2} + \frac{\partial^2 \phi_2}{\partial y^2} &= 0. \end{aligned}$$

If we multiply the first equation by ϕ_2 and the second one by ϕ_1 and take the difference of the resulting equations, we obtain

$$\frac{\partial}{\partial x}(\phi_2 \frac{\partial \phi_1}{\partial x} - \phi_1 \frac{\partial \phi_2}{\partial x}) + \frac{\partial}{\partial y}(\phi_2 \frac{\partial \phi_1}{\partial y} - \phi_1 \frac{\partial \phi_2}{\partial y}) = 0,$$

which can be integrated over R to give

$$\iint_R [\frac{\partial}{\partial x}(\phi_2 \frac{\partial \phi_1}{\partial x} - \phi_1 \frac{\partial \phi_2}{\partial x}) + \frac{\partial}{\partial y}(\phi_2 \frac{\partial \phi_1}{\partial y} - \phi_1 \frac{\partial \phi_2}{\partial y})] dxdy = 0.$$

Application of the divergence theorem to convert the double integral over R into a line integral over C yields

$$\int_C [(\phi_2 \frac{\partial \phi_1}{\partial x} - \phi_1 \frac{\partial \phi_2}{\partial x}) n_x + (\phi_2 \frac{\partial \phi_1}{\partial y} - \phi_1 \frac{\partial \phi_2}{\partial y}) n_y] ds(x,y) = 0$$

which is essentially Eq. (1.10).

Together with the fundamental solution given by Eq. (1.9), the reciprocal relation in Eq. (1.10) can be used to derive a useful boundary integral solution for the two-dimensional Laplace's equation.

1.4 Boundary Integral Solution

Let us take $\phi_1 = \Phi(x,y;\xi,\eta)$ (the fundamental solution as defined in Eq. (1.9)) and $\phi_2 = \phi$, where ϕ is the required solution of the interior boundary value problem defined by Eqs. (1.1)-(1.2).

Since $\Phi(x,y;\xi,\eta)$ is not well defined at the point (ξ,η), the reciprocal relation in Eq. (1.10) is valid for $\phi_1 = \Phi(x,y;\xi,\eta)$ and $\phi_2 = \phi$ only if (ξ,η) does not lie in the region $R \cup C$. Thus,

$$\int_C [\phi(x,y) \frac{\partial}{\partial n}(\Phi(x,y;\xi,\eta)) - \Phi(x,y;\xi,\eta) \frac{\partial}{\partial n}(\phi(x,y))] ds(x,y) = 0$$

$$\text{for } (\xi,\eta) \notin R \cup C. \qquad (1.11)$$

A more interesting and useful integral equation than Eq. (1.11) can be derived from Eq. (1.10) if we take the point (ξ,η) to lie in the region $R \cup C$.

For the case in which (ξ,η) lies in the interior of R, Eq. (1.10) is valid if we replace C by $C \cup C_\varepsilon$, where C_ε is a circle of center (ξ,η) and radius ε as shown in Figure 1.2*. This is because $\Phi(x,y;\xi,\eta)$ and its first order partial derivatives (with respect to x or y) are well defined in the region between C and C_ε. Thus, for C and C_ε in Figure 1.2, we can write

$$\int_{C \cup C_\varepsilon} [\phi(x,y) \frac{\partial}{\partial n}(\Phi(x,y;\xi,\eta)) - \Phi(x,y;\xi,\eta) \frac{\partial}{\partial n}(\phi(x,y))] ds(x,y) = 0,$$

that is,

$$\int_C [\phi(x,y) \frac{\partial}{\partial n}(\Phi(x,y;\xi,\eta)) - \Phi(x,y;\xi,\eta) \frac{\partial}{\partial n}(\phi(x,y))] ds(x,y)$$

$$= -\int_{C_\varepsilon} [\phi(x,y) \frac{\partial}{\partial n}(\Phi(x,y;\xi,\eta)) - \Phi(x,y;\xi,\eta) \frac{\partial}{\partial n}(\phi(x,y))] ds(x,y). \qquad (1.12)$$

*The divergence theorem is not only applicable for simply connected regions but also for multiply connected ones such as the one shown in Figure 1.2. For the region in Figure 1.2, the unit normal vector to C_ε (the inner boundary) points towards the center of the circle.

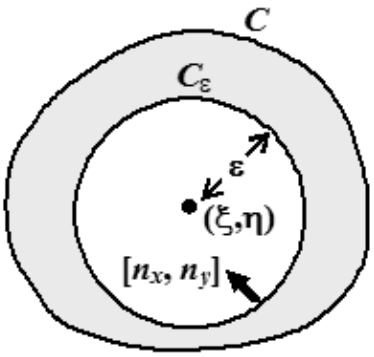

Figure 1.2

Eq. (1.12) holds for any radius $\varepsilon > 0$, so long as the circle C_ε (in Figure 1.2) lies completely inside the region bounded by C. Thus, we may let $\varepsilon \to 0^+$ in Eq. (1.12). This gives

$$\begin{aligned} & \int_C [\phi(x,y)\frac{\partial}{\partial n}(\Phi(x,y;\xi,\eta)) - \Phi(x,y;\xi,\eta)\frac{\partial}{\partial n}(\phi(x,y))]ds(x,y) \\ = & -\lim_{\varepsilon\to 0^+}\int_{C_\varepsilon} [\phi(x,y)\frac{\partial}{\partial n}(\Phi(x,y;\xi,\eta)) - \Phi(x,y;\xi,\eta)\frac{\partial}{\partial n}(\phi(x,y))]ds(x,y). \end{aligned} \tag{1.13}$$

Using polar coordinates r and θ centered about (ξ,η) as defined by $x-\xi = r\cos\theta$ and $y-\eta = r\sin\theta$, we may write

$$\begin{aligned} \Phi(x,y;\xi,\eta) &= \frac{1}{2\pi}\ln(r), \\ \frac{\partial}{\partial n}[\Phi(x,y;\xi,\eta)] &= n_x\frac{\partial}{\partial x}[\Phi(x,y;\xi,\eta)] + n_y\frac{\partial}{\partial y}[\Phi(x,y;\xi,\eta)] \\ &= \frac{n_x\cos\theta + n_y\sin\theta}{2\pi r}. \end{aligned} \tag{1.14}$$

The Taylor's series of $\phi(x,y)$ about the point (ξ,η) is given by

$$\phi(x,y) = \sum_{m=0}^{\infty}\sum_{k=0}^{m}\left(\frac{\partial^m\phi}{\partial x^k\partial y^{m-k}}\right)\Big|_{(x,y)=(\xi,\eta)}\frac{(x-\xi)^k(y-\eta)^{m-k}}{k!(m-k)!}.$$

On the circle C_ε, $r=\varepsilon$. Thus,

$$\phi(x,y) = \sum_{m=0}^{\infty}\sum_{k=0}^{m}\left(\frac{\partial^m}{\partial x^k\partial y^{m-k}}[\phi(x,y)]\right)\Big|_{(x,y)=(\xi,\eta)}\frac{\varepsilon^m\cos^k\theta\sin^{m-k}\theta}{k!(m-k)!} \quad \text{for } (x,y)\in C_\varepsilon. \tag{1.15}$$

Similarly, we may write

$$\frac{\partial}{\partial n}[\phi(x,y)] = \sum_{m=0}^{\infty}\sum_{k=0}^{m}\left(\frac{\partial^m}{\partial x^k \partial y^{m-k}}\{\frac{\partial}{\partial n}[\phi(x,y)]\}\right)\Bigg|_{(x,y)=(\xi,\eta)} \times\frac{\varepsilon^m \cos^k\theta \sin^{m-k}\theta}{k!(m-k)!} \quad \text{for } (x,y)\in C_\varepsilon. \tag{1.16}$$

Using Eqs. (1.14), (1.15) and (1.16) and writing $ds(x,y) = \varepsilon d\theta$ with θ ranging from 0 to 2π, we may now attempt to evaluate the limit on the right hand side of Eq. (1.13). On C_ε, the normal vector $[n_x, n_y]$ is given by $[-\cos\theta, -\sin\theta]$. Thus,

$$\begin{aligned}
&\int_{C_\varepsilon} \phi(x,y)\frac{\partial}{\partial n}[\Phi(x,y;\xi,\eta)]ds(x,y)\\
&= -\frac{1}{2\pi}\phi(\xi,\eta)\int_0^{2\pi} d\theta\\
&\quad -\frac{1}{2\pi}\sum_{m=1}^{\infty}\sum_{k=0}^{m}\frac{\varepsilon^m}{k!(m-k)!}\left(\frac{\partial^m\phi}{\partial x^k\partial y^{m-k}}\right)\Bigg|_{(x,y)=(\xi,\eta)}\int_0^{2\pi}\cos^k\theta\sin^{m-k}\theta d\theta\\
&\to -\phi(\xi,\eta) \quad \text{as } \varepsilon\to 0^+,
\end{aligned} \tag{1.17}$$

and

$$\begin{aligned}
&\int_{C_\varepsilon}\Phi(x,y;\xi,\eta)\frac{\partial}{\partial n}[\phi(x,y)]ds(x,y)\\
&= \frac{1}{2\pi}\sum_{m=0}^{\infty}\sum_{k=0}^{m}\left(\frac{\partial^m}{\partial x^k\partial y^{m-k}}\left(\frac{\partial}{\partial n}[\phi(x,y)]\right)\right)\Bigg|_{(x,y)=(\xi,\eta)}\\
&\quad \times\frac{\varepsilon^{m+1}\ln(\varepsilon)}{k!(m-k)!}\int_0^{2\pi}\cos^k\theta\sin^{m-k}\theta d\theta\\
&\to 0 \quad \text{as } \varepsilon\to 0^+,
\end{aligned} \tag{1.18}$$

since $\varepsilon^{m+1}\ln(\varepsilon)\to 0$ as $\varepsilon\to 0^+$ for $m = 0, 1, 2, \cdots$.

Consequently, as $\varepsilon\to 0^+$, Eq. (1.13) yields

$$\phi(\xi,\eta) = \int_C [\phi(x,y)\frac{\partial}{\partial n}(\Phi(x,y;\xi,\eta)) - \Phi(x,y;\xi,\eta)\frac{\partial}{\partial n}(\phi(x,y))]ds(x,y) \quad \text{for } (\xi,\eta)\in R. \tag{1.19}$$

Together with Eq. (1.9), Eq. (1.19) provides us with a boundary integral solution for the two-dimensional Laplace's equation. If both ϕ and $\partial\phi/\partial n$ are known

at all points on C, the line integral in Eq. (1.19) can be evaluated (at least in theory) to calculate ϕ at any point (ξ, η) in the interior of R. From the boundary conditions (1.2), at any given point on C, either ϕ or $\partial\phi/\partial n$, not both, is known, however.

To solve the interior boundary value problem, we must find the unknown ϕ and $\partial\phi/\partial n$ on C_2 and C_1 respectively. As we shall see later on, this may be done through manipulation of data on the boundary C only, if we can derive a boundary integral formula for $\phi(\xi, \eta)$, similar to the one in Eq. (1.19), for a general point (ξ, η) that lies on C.

For the case in which the point (ξ, η) lies on C, Eq. (1.10) holds if we replace the curve C by $D \cup D_\varepsilon$, where the curves D and D_ε are as shown in Figure 1.3. (If C_ε is the circle of center (ξ, η) and radius ε, then D is the part of C that lies outside C_ε and D_ε is the part of C_ε that is inside R.) Thus,

$$\begin{aligned} &\int_D [\phi(x,y)\frac{\partial}{\partial n}(\Phi(x,y;\xi,\eta)) - \Phi(x,y;\xi,\eta)\frac{\partial}{\partial n}(\phi(x,y))]ds(x,y) \\ = \; &-\int_{D_\varepsilon} [\phi(x,y)\frac{\partial}{\partial n}(\Phi(x,y;\xi,\eta)) - \Phi(x,y;\xi,\eta)\frac{\partial}{\partial n}(\phi(x,y))]ds(x,y). \end{aligned} \quad (1.20)$$

Let us examine what happens to Eq. (1.20) when we let $\varepsilon \to 0^+$.
As $\varepsilon \to 0^+$, the curve D tends to C. Thus, we may write

$$\begin{aligned} &\int_C [\phi(x,y)\frac{\partial}{\partial n}(\Phi(x,y;\xi,\eta)) - \Phi(x,y;\xi,\eta)\frac{\partial}{\partial n}(\phi(x,y))]ds(x,y) \\ = \; &-\lim_{\varepsilon\to 0^+}\int_{D_\varepsilon} [\phi(x,y)\frac{\partial}{\partial n}(\Phi(x,y;\xi,\eta)) - \Phi(x,y;\xi,\eta)\frac{\partial}{\partial n}(\phi(x,y))]ds(x,y). \end{aligned} \quad (1.21)$$

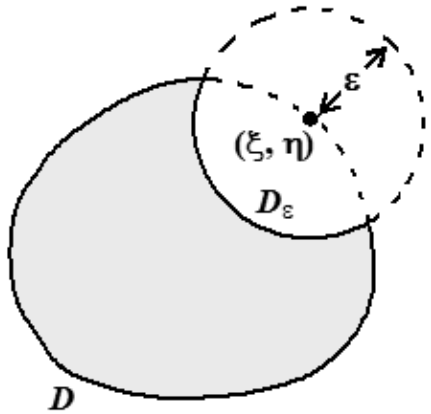

Figure 1.3

Note that, unlike in Eq. (1.13), the line integral over C in Eq. (1.21) is improper as its integrand is not well defined at (ξ,η) which lies on C. Strictly speaking, the line integration should be over the curve C without an infinitesimal segment that contains the point (ξ,η), that is, the line integral over C in Eq. (1.21) has to be interpreted in the Cauchy principal sense if (ξ,η) lies on C.

To evaluate the limit on the right hand side of Eq. (1.21), we need to know what happens to D_ε when we let $\varepsilon \to 0^+$. Now if (ξ,η) lies on a smooth part of C (not at where the gradient of the curve changes abruptly, that is, not at a corner point, if there is any), one can intuitively see that the part of C inside C_ε approaches an infinitesimal straight line as $\varepsilon \to 0^+$. Thus, we expect D_ε to tend to a semi-circle as $\varepsilon \to 0^+$, if (ξ,η) lies on a smooth part of C. It follows that in attempting to evaluate the limit on the right hand side of Eq. (1.21) we have to integrate over only half a circle (instead of a full circle as in the case of Eq. (1.13)).

Modifying Eqs. (1.17) and (1.18), we obtain

$$\begin{aligned}
\lim_{\varepsilon\to 0^+} \int_{D_\varepsilon} \phi(x,y)\frac{\partial}{\partial n}[\Phi(x,y;\xi,\eta)]ds(x,y) &= -\frac{1}{2}\phi(\xi,\eta), \\
\lim_{\varepsilon\to 0^+} \int_{D_\varepsilon} \Phi(x,y;\xi,\eta)\frac{\partial}{\partial n}[\phi(x,y)]ds(x,y) &= 0.
\end{aligned}$$

Hence Eq. (1.21) gives

$$\frac{1}{2}\phi(\xi,\eta) = \int_C [\phi(x,y)\frac{\partial}{\partial n}(\Phi(x,y;\xi,\eta)) - \Phi(x,y;\xi,\eta)\frac{\partial}{\partial n}(\phi(x,y))]ds(x,y)$$
$$\text{for } (\xi,\eta) \text{ lying on a smooth part of } C. \quad (1.22)$$

Together with the boundary conditions in Eq. (1.2), Eq. (1.22) may be applied to obtain a numerical procedure for determining the unknown ϕ and/or $\partial\phi/\partial n$ on the boundary C. Once ϕ and $\partial\phi/\partial n$ are known at all points on C, the solution of the interior boundary value problem defined by Eqs. (1.1)-(1.2) is given by Eq. (1.19) at any point (ξ,η) inside R. More details are given in Section 1.5 below.

For convenience, we may write Eqs. (1.11), (1.19) and (1.22) as a single equation given by

$$\lambda(\xi,\eta)\phi(\xi,\eta) = \int_C [\phi(x,y)\frac{\partial}{\partial n}(\Phi(x,y;\xi,\eta)) - \Phi(x,y;\xi,\eta)\frac{\partial}{\partial n}(\phi(x,y))]ds(x,y), \quad (1.23)$$

if we define

$$\lambda(\xi,\eta) = \begin{cases} 0 & \text{if } (\xi,\eta) \notin R\cup C, \\ 1/2 & \text{if } (\xi,\eta) \text{ lies on a smooth part of } C, \\ 1 & \text{if } (\xi,\eta) \in R. \end{cases} \quad (1.24)$$

1.5 Boundary Element Solution with Constant Elements

We now show how Eq. (1.23) may be applied to obtain a simple boundary element procedure for solving numerically the interior boundary value problem defined by Eqs. (1.1)-(1.2).

The boundary C is approximated as an N-sided polygon with sides $C^{(1)}$, $C^{(2)}$, $\cdots$, $C^{(N-1)}$ and $C^{(N)}$, that is,

$$C \simeq C^{(1)} \cup C^{(2)} \cup \cdots \cup C^{(N-1)} \cup C^{(N)}. \tag{1.25}$$

The sides or the boundary elements $C^{(1)}$, $C^{(2)}$, $\cdots$, $C^{(N-1)}$ and $C^{(N)}$ are constructed as follows. We put N well spaced out points $(x^{(1)}, y^{(1)})$, $(x^{(2)}, y^{(2)})$, $\cdots$, $(x^{(N-1)}, y^{(N-1)})$ and $(x^{(N)}, y^{(N)})$ on C, in the order given, following the counter clockwise direction. Defining $(x^{(N+1)}, y^{(N+1)}) = (x^{(1)}, y^{(1)})$, we take $C^{(k)}$ to be the boundary element from $(x^{(k)}, y^{(k)})$ to $(x^{(k+1)}, y^{(k+1)})$ for $k = 1, 2, \cdots, N$.

As an example, in Figure 1.4, the boundary $C = C_1 \cup C_2$ in Figure 1.1 is approximated using 5 boundary elements denoted by $C^{(1)}$, $C^{(2)}$, $C^{(3)}$, $C^{(4)}$ and $C^{(5)}$.

For a simple approximation of ϕ and $\partial\phi/\partial n$ on the boundary C, we assume that these functions are constants over each of the boundary elements. Specifically, we make the approximation:

$$\phi \simeq \overline{\phi}^{(k)} \text{ and } \frac{\partial \phi}{\partial n} = \overline{p}^{(k)} \text{ for } (x, y) \in C^{(k)} \ (k = 1, 2, \cdots, N), \tag{1.26}$$

where $\overline{\phi}^{(k)}$ and $\overline{p}^{(k)}$ are respectively the values of ϕ and $\partial\phi/\partial n$ at the midpoint of $C^{(k)}$.

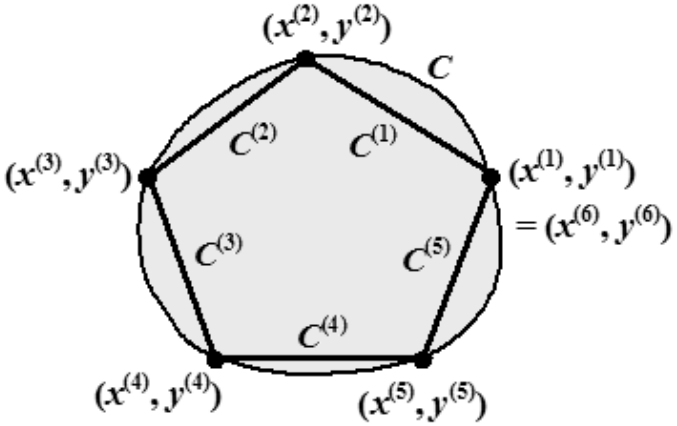

Figure 1.4

With Eqs. (1.25) and (1.26), we find that Eq. (1.23) can be approximately written as

$$\lambda(\xi,\eta)\phi(\xi,\eta) = \sum_{k=1}^{N}\{\overline{\phi}^{(k)}\mathcal{F}_2^{(k)}(\xi,\eta) - \overline{p}^{(k)}\mathcal{F}_1^{(k)}(\xi,\eta)\}, \tag{1.27}$$

where

$$\begin{aligned}\mathcal{F}_1^{(k)}(\xi,\eta) &= \int_{C^{(k)}} \Phi(x,y;\xi,\eta)ds(x,y),\\ \mathcal{F}_2^{(k)}(\xi,\eta) &= \int_{C^{(k)}} \frac{\partial}{\partial n}[\Phi(x,y;\xi,\eta)]ds(x,y).\end{aligned} \tag{1.28}$$

For a given k, either $\overline{\phi}^{(k)}$ or $\overline{p}^{(k)}$ (not both) is known from the boundary conditions in Eq. (1.2). Thus, there are N unknown constants on the right hand side of Eq. (1.27). To determine their values, we have to generate N equations containing the unknowns.

If we let (ξ,η) in Eq. (1.27) be given in turn by the midpoints of $C^{(1)}$, $C^{(2)}$, $\cdots$, $C^{(N-1)}$ and $C^{(N)}$, we obtain

$$\begin{aligned}\frac{1}{2}\overline{\phi}^{(m)} &= \sum_{k=1}^{N}\{\overline{\phi}^{(k)}\mathcal{F}_2^{(k)}(\overline{x}^{(m)},\overline{y}^{(m)}) - \overline{p}^{(k)}\mathcal{F}_1^{(k)}(\overline{x}^{(m)},\overline{y}^{(m)})\}\\ &\qquad\qquad \text{for } m = 1,2,\cdots,N,\end{aligned} \tag{1.29}$$

where $(\overline{x}^{(m)},\overline{y}^{(m)})$ is the midpoint of $C^{(m)}$.

In the derivation of Eq. (1.29), we take $\lambda(\overline{x}^{(m)},\overline{y}^{(m)}) = 1/2$, since $(\overline{x}^{(m)},\overline{y}^{(m)})$ being the midpoint of $C^{(m)}$ lies on a smooth part of the approximate boundary $C^{(1)} \cup C^{(2)} \cup \cdots \cup C^{(N-1)} \cup C^{(N)}$.

Eq. (1.29) constitutes a system of N linear algebraic equations containing the N unknowns on the right hand side of Eq. (1.27). We may rewrite it as

$$\sum_{k=1}^{N} a^{(mk)}z^{(k)} = \sum_{k=1}^{N} b^{(mk)} \text{ for } m = 1,2,\cdots,N, \tag{1.30}$$

where $a^{(mk)}$, $b^{(mk)}$ and $z^{(k)}$ are defined by

$$
\begin{aligned}
a^{(mk)} &= \begin{cases} -\mathcal{F}_1^{(k)}(\overline{x}^{(m)}, \overline{y}^{(m)}) & \text{if } \phi \text{ is specified over } C^{(k)}, \\ \mathcal{F}_2^{(k)}(\overline{x}^{(m)}, \overline{y}^{(m)}) - \frac{1}{2}\delta^{(mk)} & \text{if } \partial\phi/\partial n \text{ is specified over } C^{(k)}, \end{cases} \\
b^{(mk)} &= \begin{cases} \overline{\phi}^{(k)}(-\mathcal{F}_2^{(k)}(\overline{x}^{(m)}, \overline{y}^{(m)}) + \frac{1}{2}\delta^{(mk)}) & \text{if } \phi \text{ is specified over } C^{(k)}, \\ \overline{p}^{(k)}\mathcal{F}_1^{(k)}(\overline{x}^{(m)}, \overline{y}^{(m)}) & \text{if } \partial\phi/\partial n \text{ is specified over } C^{(k)}, \end{cases} \\
\delta^{(mk)} &= \begin{cases} 0 & \text{if } m \neq k, \\ 1 & \text{if } m = k, \end{cases} \\
z^{(k)} &= \begin{cases} \overline{p}^{(k)} & \text{if } \phi \text{ is specified over } C^{(k)}, \\ \overline{\phi}^{(k)} & \text{if } \partial\phi/\partial n \text{ is specified over } C^{(k)}. \end{cases}
\end{aligned} \tag{1.31}
$$

Note that $z^{(1)}$, $z^{(2)}$, $\cdots$, $z^{(N-1)}$ and $z^{(N)}$ are the N unknown constants on the right hand side of Eq. (1.27), while $a^{(mk)}$ and $b^{(mk)}$ are known coefficients.

Once Eq. (1.30) is solved for the unknowns $z^{(1)}$, $z^{(2)}$, $\cdots$, $z^{(N-1)}$ and $z^{(N)}$, the values of ϕ and $\partial\phi/\partial n$ over the element $C^{(k)}$, as given by $\overline{\phi}^{(k)}$ and $\overline{p}^{(k)}$ respectively, are known for $k = 1, 2, \cdots, N$. Eq. (1.27) with $\lambda(\xi, \eta) = 1$ then provides us with an explicit formula for computing ϕ in the interior of R, that is,

$$
\phi(\xi, \eta) \simeq \sum_{k=1}^{N} \{\overline{\phi}^{(k)}\mathcal{F}_2^{(k)}(\xi, \eta) - \overline{p}^{(k)}\mathcal{F}_1^{(k)}(\xi, \eta)\} \quad \text{for } (\xi, \eta) \in R. \tag{1.32}
$$

To summarize, a boundary element solution of the interior boundary value problem defined by Eqs. (1.1)-(1.2) is given by Eq. (1.32) together with Eqs. (1.28), (1.30) and (1.31). Because of the approximation in Eqs. (1.25) and (1.26), the solution is said to be obtained using constant elements. Analytical formulae for calculating $\mathcal{F}_1^{(k)}(\xi, \eta)$ and $\mathcal{F}_2^{(k)}(\xi, \eta)$ in Eq. (1.28) are given in Eqs. (1.37), (1.38), (1.40) and (1.41) (together with Eq. (1.35)) in the section below.

1.6 Formulae for Integrals of Constant Elements

The boundary element solution above requires the evaluation of $\mathcal{F}_1^{(k)}(\xi, \eta)$ and $\mathcal{F}_2^{(k)}(\xi, \eta)$. These functions are defined in terms of line integrals over $C^{(k)}$ as given in Eq. (1.28). The line integrals can be worked out analytically as follows.

Points on the element $C^{(k)}$ may be described using the parametric equations

$$
\left.\begin{aligned} x &= x^{(k)} - t\ell^{(k)}n_y^{(k)} \\ y &= y^{(k)} + t\ell^{(k)}n_x^{(k)} \end{aligned}\right\} \text{ from } t = 0 \text{ to } t = 1, \tag{1.33}
$$

where $\ell^{(k)}$ is the length of $C^{(k)}$ and $[n_x^{(k)}, n_y^{(k)}] = [y^{(k+1)} - y^{(k)}, x^{(k)} - x^{(k+1)}]/\ell^{(k)}$ is the unit normal vector to $C^{(k)}$ pointing away from R.

For $(x,y) \in C^{(k)}$, we find that $ds(x,y) = \sqrt{(dx)^2 + (dy)^2} = \ell^{(k)} dt$ and

$$(x-\xi)^2 + (y-\eta)^2 = A^{(k)} t^2 + B^{(k)}(\xi,\eta) t + E^{(k)}(\xi,\eta), \tag{1.34}$$

where

$$\begin{aligned} A^{(k)} &= [\ell^{(k)}]^2, \\ B^{(k)}(\xi,\eta) &= [-n_y^{(k)}(x^{(k)} - \xi) + (y^{(k)} - \eta) n_x^{(k)}](2\ell^{(k)}), \\ E^{(k)}(\xi,\eta) &= (x^{(k)} - \xi)^2 + (y^{(k)} - \eta)^2. \end{aligned} \tag{1.35}$$

The parameters in Eq. (1.35) satisfy $4A^{(k)}E^{(k)}(\xi,\eta) - [B^{(k)}(\xi,\eta)]^2 \geq 0$ for any point (ξ,η). To see why this is true, consider the straight line defined by the parametric equations $x = x^{(k)} - t\ell^{(k)} n_y^{(k)}$ and $y = y^{(k)} + t\ell^{(k)} n_x^{(k)}$ for $-\infty < t < \infty$. Note that $C^{(k)}$ is a subset of this straight line (given by the parametric equations from $t = 0$ to $t = 1$). Eq. (1.34) also holds for any point (x,y) lying on the extended line. If (ξ,η) does not lie on the line then $A^{(k)}t^2 + B^{(k)}(\xi,\eta)t + E^{(k)}(\xi,\eta) > 0$ for all real values of t (that is, for all points (x,y) on the line) and hence $4A^{(k)}E^{(k)}(\xi,\eta) - [B^{(k)}(\xi,\eta)]^2 > 0$. On the other hand, if (ξ,η) is on the line, we can find exactly one point (x,y) such that $A^{(k)}t^2 + B^{(k)}(\xi,\eta)t + E^{(k)}(\xi,\eta) = 0$. As each point (x,y) on the line is given by a unique value of t, we conclude that $4A^{(k)}E^{(k)}(\xi,\eta) - [B^{(k)}(\xi,\eta)]^2 = 0$ for (ξ,η) lying on the line.

From Eqs. (1.28), (1.33) and (1.34), $\mathcal{F}_1^{(k)}(\xi,\eta)$ and $\mathcal{F}_2^{(k)}(\xi,\eta)$ may be written as

$$\begin{aligned} \mathcal{F}_1^{(k)}(\xi,\eta) &= \frac{\ell^{(k)}}{4\pi} \int_0^1 \ln[A^{(k)}t^2 + B^{(k)}(\xi,\eta)t + E^{(k)}(\xi,\eta)]dt, \\ \mathcal{F}_2^{(k)}(\xi,\eta) &= \frac{\ell^{(k)}}{2\pi} \int_0^1 \frac{n_x^{(k)}(x^{(k)} - \xi) + n_y^{(k)}(y^{(k)} - \eta)}{A^{(k)}t^2 + B^{(k)}(\xi,\eta)t + E^{(k)}(\xi,\eta)} dt. \end{aligned} \tag{1.36}$$

The second integral in Eq. (1.36) is the easiest one to work out for the case in which $4A^{(k)}E^{(k)}(\xi,\eta) - [B^{(k)}(\xi,\eta)]^2 = 0$. For this case, the point (ξ,η) lies on the straight line of which the element $C^{(k)}$ is a subset. Thus, the vector $[x^{(k)} - \xi, y^{(k)} - \eta]$ is perpendicular to $[n_x^{(k)}, n_y^{(k)}]$, that is, $n_x^{(k)}(x^{(k)} - \xi) + n_y^{(k)}(y^{(k)} - \eta) = 0$, and we obtain

$$\mathcal{F}_2^{(k)}(\xi,\eta) = 0 \quad \text{for } 4A^{(k)}E^{(k)}(\xi,\eta) - [B^{(k)}(\xi,\eta)]^2 = 0. \tag{1.37}$$

From the integration formula

$$\begin{aligned} \int \frac{dt}{at^2 + bt + c} &= \frac{2}{\sqrt{4ac - b^2}} \arctan\left(\frac{2at + b}{\sqrt{4ac - b^2}}\right) + \text{ constant} \\ &\quad \text{for real constants } a,\ b \text{ and } c \text{ such that } 4ac - b^2 > 0, \end{aligned}$$

we find that

$$\begin{aligned}\mathcal{F}_2^{(k)}(\xi,\eta) &= \frac{\ell^{(k)}[n_x^{(k)}(x^{(k)}-\xi)+n_y^{(k)}(y^{(k)}-\eta)]}{\pi\sqrt{4A^{(k)}E^{(k)}(\xi,\eta)-[B^{(k)}(\xi,\eta)]^2}} \\ &\quad\times[\arctan(\frac{2A^{(k)}+B^{(k)}(\xi,\eta)}{\sqrt{4A^{(k)}E^{(k)}(\xi,\eta)-[B^{(k)}(\xi,\eta)]^2}}) \\ &\quad-\arctan(\frac{B^{(k)}(\xi,\eta)}{\sqrt{4A^{(k)}E^{(k)}(\xi,\eta)-[B^{(k)}(\xi,\eta)]^2}})] \\ &\quad\text{for } 4A^{(k)}E^{(k)}(\xi,\eta)-[B^{(k)}(\xi,\eta)]^2>0. \end{aligned} \tag{1.38}$$

If $4A^{(k)}E^{(k)}(\xi,\eta)-[B^{(k)}(\xi,\eta)]^2=0$, we may write

$$A^{(k)}t^2+B^{(k)}(\xi,\eta)t+E^{(k)}(\xi,\eta)=A^{(k)}(t+\frac{B^{(k)}(\xi,\eta)}{2A^{(k)}})^2.$$

Thus,

$$\begin{aligned}\mathcal{F}_1^{(k)}(\xi,\eta) &= \frac{\ell^{(k)}}{4\pi}\int_0^1 \ln[A^{(k)}(t+\frac{B^{(k)}(\xi,\eta)}{2A^{(k)}})^2]dt \\ &\quad\text{for } 4A^{(k)}E^{(k)}(\xi,\eta)-[B^{(k)}(\xi,\eta)]^2=0. \end{aligned} \tag{1.39}$$

Now if (ξ,η) lies on a smooth part of $C^{(k)}$, the integral in Eq. (1.39) is improper, as its integrand is not well defined at the point $t=t_0\equiv -B^{(k)}(\xi,\eta)/(2A^{(k)})\in(0,1)$. Strictly speaking, the integral should then be interpreted in the Cauchy principal sense, that is, to evaluate it, we have to integrate over $[0,t_0-\varepsilon]\cup[t_0+\varepsilon,1]$ instead of $[0,1]$ and then let $\varepsilon\to 0$ to obtain its value. However, in this case, it turns out that the limits of integration $t=t_0-\varepsilon$ and $t=t_0+\varepsilon$ eventually do not contribute anything to the integral. Thus, for $4A^{(k)}E^{(k)}(\xi,\eta)-[B^{(k)}(\xi,\eta)]^2=0$, the final analytical formula for $\mathcal{F}_1^{(k)}(\xi,\eta)$ is the same irrespective of whether (ξ,η) lies on $C^{(k)}$ or not. If (ξ,η) lies on $C^{(k)}$, we may ignore the singular behaviour of the integrand and apply the fundamental theorem of integral calculus as usual to evaluate the definite integral in Eq. (1.39) directly over $[0,1]$.

The integration required in Eq. (1.39) can be easily done to give

$$\begin{aligned}\mathcal{F}_1^{(k)}(\xi,\eta) &= \frac{\ell^{(k)}}{2\pi}\{\ln(\ell^{(k)})+(1+\frac{B^{(k)}(\xi,\eta)}{2A^{(k)}})\ln|1+\frac{B^{(k)}(\xi,\eta)}{2A^{(k)}}| \\ &\quad-\frac{B^{(k)}(\xi,\eta)}{2A^{(k)}}\ln|\frac{B^{(k)}(\xi,\eta)}{2A^{(k)}}|-1\} \\ &\quad\text{for } 4A^{(k)}E^{(k)}(\xi,\eta)-[B^{(k)}(\xi,\eta)]^2=0. \end{aligned} \tag{1.40}$$

Using

$$\int \ln(at^2+bt+c)dt = t[\ln(a)-2]+(t+\frac{b}{2a})\ln[t^2+\frac{b}{a}t+\frac{c}{a}] +\frac{1}{a}\sqrt{4ac-b^2}\arctan(\frac{2at+b}{\sqrt{4ac-b^2}})+ \text{ constant}$$
$$\text{for real constants } a,\ b \text{ and } c \text{ such that } 4ac-b^2>0,$$

we obtain

$$\begin{aligned}\mathcal{F}_1^{(k)}(\xi,\eta) = \frac{\ell^{(k)}}{4\pi}\{&2[\ln(\ell^{(k)})-1]-\frac{B^{(k)}(\xi,\eta)}{2A^{(k)}}\ln|\frac{E^{(k)}(\xi,\eta)}{A^{(k)}}|\\ &+(1+\frac{B^{(k)}(\xi,\eta)}{2A^{(k)}})\ln|1+\frac{B^{(k)}(\xi,\eta)}{A^{(k)}}+\frac{E^{(k)}(\xi,\eta)}{A^{(k)}}|\\ &+\frac{\sqrt{4A^{(k)}E^{(k)}(\xi,\eta)-[B^{(k)}(\xi,\eta)]^2}}{A^{(k)}}\\ &\times[\arctan(\frac{2A^{(k)}+B^{(k)}(\xi,\eta)}{\sqrt{4A^{(k)}E^{(k)}(\xi,\eta)-[B^{(k)}(\xi,\eta)]^2}})\\ &-\arctan(\frac{B^{(k)}(\xi,\eta)}{\sqrt{4A^{(k)}E^{(k)}(\xi,\eta)-[B^{(k)}(\xi,\eta)]^2}})]\}\\ &\text{for } 4A^{(k)}E^{(k)}(\xi,\eta)-[B^{(k)}(\xi,\eta)]^2>0. \end{aligned} \tag{1.41}$$

1.7 Implementation on Computer

We attempt now to develop double precision FORTRAN 77 codes which can be used to implement the boundary element procedure described in Section 1.5 on the computer. In our discussion here, syntaxes, variables and statements in FORTRAN 77 are written in typewriter fonts, for example, `xi`, `eta` and `A=L**2d0`.

One of the tasks involved is the setting up of the system of linear algebraic equations given in Eqs. (1.30) and (1.31). To do this, the functions $\mathcal{F}_1^{(k)}(\xi,\eta)$ and $\mathcal{F}_2^{(k)}(\xi,\eta)$ have to be computed using the formulae in Section 1.6. We create a subroutine called `CPF` which accepts the values of ξ, η, $x^{(k)}$, $y^{(k)}$, $n_x^{(k)}$, $n_y^{(k)}$ and $\ell^{(k)}$(stored in the real variables `xi`, `eta`, `xk`, `yk`, `nkx`, `nky` and `L`) in order to calculate and return the values of $\pi\mathcal{F}_1^{(k)}(\xi,\eta)$ and $\pi\mathcal{F}_2^{(k)}(\xi,\eta)$ (in the real variables `PF1` and `PF2`).

The subroutine `CPF` is listed below.

```
      subroutine CPF(xi,eta,xk,yk,nkx,nky,L,PF1,PF2)

      double precision xi,eta,xk,yk,nkx,nky,L,PF1,PF2,
     & A,B,E,D,BA,EA
```

```
      A=L**2d0
      B=2d0*L*(-nky*(xk-xi)+nkx*(yk-eta))
      E=(xk-xi)**2d0+(yk-eta)**2d0
      D=dsqrt(dabs(4d0*A*E-B**2d0))
      BA=B/A
      EA=E/A

      if (D.lt.0.0000000001d0) then
      PF1=0.5d0*L*(dlog(L)
     & +(1d0+0.5d0*BA)*dlog(dabs(1d0+0.5d0*BA))
     & -0.5d0*BA*dlog(dabs(0.5d0*BA))-1d0)
      PF2=0d0
      else
      PF1=0.25d0*L*(2d0*(dlog(L)-1d0)-0.5d0*BA*dlog(dabs(EA))
     & +(1d0+0.5d0*BA)*dlog(dabs(1d0+BA+EA))
     & +(D/A)*(datan((2d0*A+B)/D)-datan(B/D)))
      PF2=L*(nkx*(xk-xi)+nky*(yk-eta))/D
     & *(datan((2d0*A+B)/D)-datan(B/D))
      endif

      return
      end
```

CPF is repeatedly called in the subroutine CELAP1. CELAP1 reads in the number of boundary elements (N) in the real variable N, the midpoints $(\overline{x}^{(k)}, \overline{y}^{(k)})$ in the real arrays xm(1:N) and ym(1:N), the boundary points $(x^{(k)}, y^{(k)})$ in the real arrays xb(1:N+1) and yb(1:N+1), the normal vectors $(n_x^{(k)}, n_y^{(k)})$ in the real arrays nx(1:N) and ny(1:N), the lengths of the boundary elements in the real array lg(1:N) and the types of boundary conditions (on the boundary elements) in the integer array BCT(1:N) together with the corresponding boundary values in the real array BCV(1:N), set up and solve Eq. (1.30), and return all the values of $\overline{\phi}^{(k)}$ and $\overline{p}^{(k)}$ in the arrays phi(1:N) and dphi(1:N) respectively. (More details on the arrays BCT(1:N) and BCV(1:N) will be given later on in Section 1.8.) Thus, a large part of the boundary element procedure (with constant elements) for the numerical solution of the boundary value problem is executed in CELAP1.

The subroutine CELAP1 is listed as follows.

```
      subroutine CELAP1(N,xm,ym,xb,yb,nx,ny,lg,BCT,BCV,phi,dphi)

      integer m,k,N,BCT(1000)

      double precision xm(1000),ym(1000),xb(1000),yb(1000),
```

```
     & nx(1000),ny(1000),lg(1000),BCV(1000),A(1000,1000),
     & B(1000),pi,PF1,PF2,del,phi(1000),dphi(1000),F1,F2,
     & Z(1000)

       pi=4d0*datan(1d0)

       do 10 m=1,N
       B(m)=0d0
       do 5 k=1,N
       call CPF(xm(m),ym(m),xb(k),yb(k),nx(k),ny(k),lg(k),PF1,PF2)
       F1=PF1/pi
       F2=PF2/pi
       if (k.eq.m) then
       del=1d0
       else
       del=0d0
       endif
       if (BCT(k).eq.0) then
       A(m,k)=-F1
       B(m)=B(m)+BCV(k)*(-F2+0.5d0*del)
       else
       A(m,k)=F2-0.5d0*del
       B(m)=B(m)+BCV(k)*F1
       endif
     5 continue
    10 continue

       call solver(A,B,N,1,Z)

       do 15 m=1,N
       if (BCT(m).eq.0) then
       phi(m)=BCV(m)
       dphi(m)=Z(m)
       else
       phi(m)=Z(m)
       dphi(m)=BCV(m)
       endif
    15 continue

     return
     end
```

The values of $a^{(mk)}$ in Eq. (1.30) are kept in the real array A(1:N,1:N), the sum $b^{(m1)} + b^{(m2)} + \cdots + b^{(mN)}$ on the right hand side of the equation in the real array B(1:N) and the solution $z^{(k)}$ in the real array Z(1:N). To solve for $z^{(k)}$, an LU decomposition is performed on the matrix containing the coefficients $a^{(mk)}$ to obtain a simpler system that may be easily solved by backward substitutions. This is done in the subroutine SOLVER (listed below together with supporting subprograms DAXPY, DSCAL and IDAMAX†) which accepts the integer N (giving the number of unknowns), the real arrays A(1:N,1:N) and B(1:N) and the integer lud to return Z(1:N). In general, the integer lud may be given any value except 0. However, if we are solving two different systems of linear algebraic equations with the same square matrix $[a^{(mk)}]$, one after the other, lud may be given the value 0 the second time SOLVER is called. This is because it is not necessary to perform the LU decomposition on the same square matrix again to solve the second system after solving the first. If lud is given the value 0, SOLVER assumes that the square matrix has already been properly decomposed before and avoids the time consuming decomposition process. In CELAP1, since the square matrix has not been decomposed yet, the value of 1 is passed into lud when we call SOLVER.

The subroutine SOLVER and its supporting programs are listed as follows.

```
      subroutine SOLVER(A,B,N,lud,Z)

      integer lda,N,ipvt(1000),info,lud,IDAMAX,
     & j,k,kp1,l,nm1,kb

      double precision A(1000,1000),B(1000),Z(1000),t,AMD(1000,1000)

      common /ludcmp/ipvt,AMD

      nm1=N-1

      do 5 i=1,N
      Z(i)=B(i)
    5 continue

      if (lud.eq.0) goto 99
```

†The main part of SOLVER for decomposing the square matrix A and solving AX = B is respectively taken from the codes in the LINPACK subroutines DGEFA and DGESL written by Cleve Moler. The supporting subprograms DAXPY, DSCAL and IDAMAX written by Jack Dongarra are also from LINPACK. DGEFA, DGESL, DAXPY, DSCAL and IDAMAX are all in the public domain and may be downloaded from Netlib website at http://www.netlib.org. Permission for reproducing the codes here was granted by Netlib's editor-in-chief Jack Dongarra.

```
      do 6 i=1,N
      do 6 j=1,N
      AMD(i,j)=A(i,j)
    6 continue

      info=0

      if (nm1.lt.1) go to 70

      do 60 k=1,nm1
      kp1=k+1
      l=IDAMAX(N-k+1,AMD(k,k),1)+k-1
      ipvt(k)=l
      if (AMD(l,k).eq.0.0d0) goto 40
      if (l.eq.k) goto 10
      t=AMD(l,k)
      AMD(l,k)=AMD(k,k)
      AMD(k,k)=t
   10 continue
      t=-1.0d0/AMD(k,k)
      call DSCAL(N-k,t,AMD(k+1,k),1)
      do 30 j=kp1,N
      t=AMD(l,j)
      if (l.eq.k) go to 20
      AMD(l,j)=AMD(k,j)
      AMD(k,j)=t
   20 continue
      call DAXPY(N-k,t,AMD(k+1,k),1,AMD(k+1,j),1)
   30 continue
      goto 50
   40 continue
      info=k
   50 continue
   60 continue

   70 continue

      ipvt(N)=N

      if (AMD(N,N).eq.0.0d0) info=N
      if (info.ne.0) pause 'Division by zero in SOLVER!'
```

```
99 continue

   if (nm1.lt.1) goto 130

   do 120 k=1,nm1
   l=ipvt(k)
   t=Z(l)
   if (l.eq.k) goto 110
   Z(l)=Z(k)
   Z(k)=t
110 continue
   call DAXPY(N-k,t,AMD(k+1,k),1,Z(k+1),1)
120 continue

130 continue

   do 140 kb=1,N
   k=N+1-kb
   Z(k) = Z(k)/AMD(k,k)
   t=-Z(k)
   call DAXPY(k-1,t,AMD(1,k),1,Z(1),1)
140 continue

   return
   end

   subroutine DAXPY(N,da,dx,incx,dy,incy)

   double precision dx(1000),dy(1000),da

   integer i,incx,incy,ix,iy,m,mp1,N

   if(N.le.0) return
   if (da .eq.  0.0d0) return
   if(incx.eq.1.and.incy.eq.1) goto 20

   ix=1
   iy=1

   if(incx.lt.0) ix=(-N+1)*incx+1
   if(incy.lt.0) iy=(-N+1)*incy+1
```

```
      do 10 i=1,N
      dy(iy)=dy(iy)+da*dx(ix)
      ix=ix+incx
      iy=iy+incy
   10 continue

      return

   20 m=mod(N,4)

      if( m.eq.  0 ) go to 40

      do 30 i=1,m
      dy(i)=dy(i)+da*dx(i)
   30 continue

      if(N.lt.4) return

   40 mp1=m+1

      do 50 i=mp1,N,4
      dy(i)=dy(i)+da*dx(i)
      dy(i+1)=dy(i+1)+da*dx(i+1)
      dy(i+2)=dy(i+2)+da*dx(i+2)
      dy(i+3)=dy(i+3)+da*dx(i+3)
   50 continue

      return
      end

      subroutine DSCAL(N,da,dx,incx)

      double precision da,dx(1000)

      integer i,incx,m,mp1,N,nincx

      if(N.le.0.or.incx.le.0) return
      if(incx.eq.1) goto 20
      nincx = N*incx

      do 10 i=1,nincx,incx
      dx(i)=da*dx(i)
```

```
      10 continue

         return

      20 m=mod(N,5)

         if(m.eq.0) goto 40

         do 30 i=1,m
         dx(i) = da*dx(i)
      30 continue

         if(N.lt.5) return

      40 mp1=m+1

         do 50 i=mp1,N,5
         dx(i)=da*dx(i)
         dx(i+1)=da*dx(i+1)
         dx(i+2)=da*dx(i+2)
         dx(i+3)=da*dx(i+3)
         dx(i+4)=da*dx(i+4)
       50 continue

         return
         end

         function IDAMAX(N,dx,incx)

         double precision dx(1000),dmax

         integer i,incx,ix,N,IDAMAX

         IDAMAX = 0
         if(N.lt.1.or.incx.le.0 ) return
         IDAMAX = 1

         if(N.eq.1)return
         if(incx.eq.1) goto 20
         ix = 1
         dmax = dabs(dx(1))
         ix = ix + incx
```

```
      do 10 i=2,N
      if(dabs(dx(ix)).le.dmax) goto 5
      IDAMAX=i
      dmax=dabs(dx(ix))
    5 ix=ix+incx
   10 continue

      return

   20 dmax=dabs(dx(1))

      do 30 i=2,N
      if(dabs(dx(i)).le.dmax) goto 30
      IDAMAX=i
      dmax=dabs(dx(i))
   30 continue

      return
      end
```

Once the values of $\overline{\phi}^{(k)}$ and $\overline{p}^{(k)}$ are returned in the arrays `phi(1:N)` and `dphi(1:N)` by `CELAP1`, they can be used by the subroutine `CELAP2` to compute the value of ϕ at any chosen point (ξ, η) in the interior of the solution domain. In the listing of `CELAP2` below, `xi` and `eta` are the real variables which carry the values of ξ and η respectively. The computed value of $\phi(\xi, \eta)$ is returned in the real variable `pint`. Note that the subroutine `CPF` is called in `CELAP2` to compute $\pi\mathcal{F}_1^{(k)}(\xi, \eta)$ and $\pi\mathcal{F}_2^{(k)}(\xi, \eta)$.

```
      subroutine CELAP2(N,xi,eta,xb,yb,nx,ny,lg,phi,dphi,pint)

      integer N,i

      double precision xi,eta,xb(1000),yb(1000),nx(1000),ny(1000),
     & lg(1000),phi(1000),dphi(1000),pint,sum,pi,PF1,PF2

      pi=4d0*datan(1d0)
      sum=0d0

      do 10 i=1,N
      call CPF(xi,eta,xb(i),yb(i),nx(i),ny(i),lg(i),PF1,PF2)
      sum=sum+phi(i)*PF2-dphi(i)*PF1
```

```
10 continue

  pint=sum/pi

  return
  end
```

1.8 Numerical Examples

We now show how the subroutines CELAP1 and CELAP2 may be used to solve two specific examples of the interior boundary value problem described in Section 1.1.

Example 1.1

The solution domain is the square region $0 < x < 1$, $0 < y < 1$. The boundary conditions are

$$\left.\begin{array}{ll} \phi = 0 & \text{on } x = 0 \\ \phi = \cos(\pi y) & \text{on } x = 1 \end{array}\right\} \text{ for } 0 < y < 1$$

$$\frac{\partial \phi}{\partial n} = 0 \text{ on } y = 0 \text{ and } y = 1 \text{ for } 0 < x < 1.$$

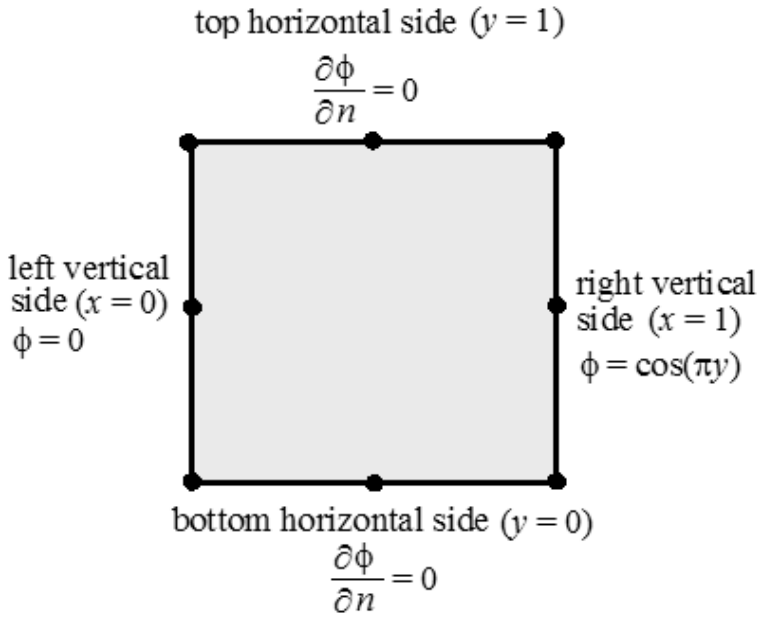

Figure 1.5

The sides of the square are discretized into boundary elements of equal length. To do this, we choose N evenly spaced out points on the sides as follows. The boundary points on the sides $y = 0$ (bottom horizontal), $x = 1$ (right vertical), $y = 1$

(top horizontal) and $x = 0$ (left vertical) are respectively given by $(x^{(m)}, y^{(m)}) = ([m-1]\ell, 0)$, $(x^{(m+N_0)}, y^{(m+N_0)}) = (1, [m-1]\ell)$, $(x^{(m+2N_0)}, y^{(m+2N_0)}) = (1-[m-1]\ell, 1)$ and $(x^{(m+3N_0)}, y^{(m+3N_0)}) = (0, 1-[m-1]\ell)$ for $m = 1, 2, \cdots, N_0$, where N_0 is the number of boundary elements per side (so that $N = 4N_0$) and $\ell = 1/N_0$ is the length of each element. For example, the boundary points for $N_0 = 2$ (that is, 8 boundary elements) are shown in Figure 1.5.

The input points $(x^{(1)}, y^{(1)})$, $(x^{(2)}, y^{(2)})$, $\cdots$, $(x^{(N-1)}, y^{(N-1)})$, $(x^{(N)}, y^{(N)})$ and $(x^{(N+1)}, y^{(N+1)})$, arranged in counter clockwise order on the boundary of the solution domain, are stored in the real arrays `xb(1:N+1)` and `yb(1:N+1)`. (Recall that $(x^{(N+1)}, y^{(N+1)}) = (x^{(1)}, y^{(1)})$.) The values in these arrays are input data defining the geometry of the solution domain, to be generated by the user of the subroutines `CELAP1` and `CELAP2`. As the geometry in this example is a simple one, the input data for the boundary points may be generated by writing a simple code as follows.

```
      N=4*N0
      dl=1d0/dfloat(N0)
      do 10 i=1,N0
      xb(i)=dfloat(i-1)*dl
      yb(i)=0d0
      xb(N0+i)=1d0
      yb(N0+i)=xb(i)
      xb(2*N0+i)=1d0-xb(i)
      yb(2*N0+i)=1d0
      xb(3*N0+i)=0d0
      yb(3*N0+i)=1d0-xb(i)
   10 continue
      xb(N+1)=xb(1)
      yb(N+1)=yb(1)
```

Note that `N0` is an integer variable which gives the number of boundary elements per side and `dl` is a real variable giving the length of an element. The value of `N0` is a given input. The boundary points in Figure 1.5 may be generated by the code above if we give `N0` the value of 2.

In order to call `CELAP1` and `CELAP2`, the midpoints of the elements (in the real arrays `xm(1:N)` and `ym(1:N)`), the lengths of the elements (in the real array `lg(1:N)`) and the unit normal vectors to the elements (in the real arrays `nx(1:N)` and `ny(1:N)`) are required. These can be calculated from the input data stored in the arrays `xb(1:N+1)` and `yb(1:N+1)`. The general code for the calculation (which is valid for any geometry of the solution domain) is as follows.

```
      do 20 i=1,N
      xm(i)=0.5d0*(xb(i)+xb(i+1))
```

```
      ym(i)=0.5d0*(yb(i)+yb(i+1))
      lg(i)=dsqrt((xb(i+1)-xb(i))**2d0+(yb(i+1)-yb(i))**2d0)
      nx(i)=(yb(i+1)-yb(i))/lg(i)
      ny(i)=(xb(i)-xb(i+1))/lg(i)
   20 continue
```

The type of boundary conditions on an element (that is, whether ϕ or $\partial\phi/\partial n$ is specified) and the corresponding specified value of either ϕ or $\partial\phi/\partial n$ are input data. The integer array `BCT(1:N)` is used to keep track of the types of boundary conditions on the elements. If ϕ is specified on the 5-th boundary element $C^{(5)}$ then `BCT(5)` is given the value 0. If `BCT(5)` is not 0, then we know that $\partial\phi/\partial n$ is specified on $C^{(5)}$. The values of either ϕ or $\partial\phi/\partial n$ prescribed on the boundary elements are stored in the real array `BCV(1:N)`. For the boundary points in Figure 1.5, the input boundary values of ϕ on the two elements on the right vertical sides are given by $\cos(\pi\eta)$ with η being the y coordinates of the midpoints of the elements. For the boundary value problem here, the code for generating the input data for `BCT` and `BCV` are as follows.

```
      do 30 i=1,N
      if (i.le.N0) then
      BCT(i)=1
      BCV(i)=0d0
      else if ((i.gt.N0).and.(i.le.(2*N0))) then
      BCT(i)=0
      BCV(i)=dcos(pi*ym(i))
      else if ((i.gt.(2*N0)).and.(i.le.(3*N0))) then
      BCT(i)=1
      BCV(i)=0d0
      else
      BCT(i)=0
      BCV(i)=0d0
      endif
   30 continue
```

We may now invoke `CELAP1` using the statement

```
call CELAP1(N,xm,ym,xb,yb,nx,ny,lg,BCT,BCV,phi,dphi)
```

to give us the (approximate) values of ϕ and $\partial\phi/\partial n$ on the boundary elements. The boundary values of ϕ and $\partial\phi/\partial n$ (that is, $\overline{\phi}^{(k)}$ and $\overline{p}^{(k)}$) are respectively stored in the real arrays `phi(1:N)` and `dphi(1:N)`. For example, if the variable `BCT(5)` has the value 0, we know that ϕ is specified on the 5-th boundary element and hence the variable `dphi(5)` gives us the approximate value of $\partial\phi/\partial n$ on $C^{(5)}$.

Once CELAP1 is called, we may use CELAP2 to calculate the value of ϕ at any interior point inside the square. For example, if we wish to calculate ϕ at $(0.50, 0.70)$, we may use the call statement

```
call CELAP2(N,0.50,0.70,xb,yb,nx,ny,lg,phi,dphi,pint)
```

to return us the approximate value of $\phi(0.50, 0.70)$ in the real variable pint.

An example of a complete program for the boundary value problem presently under consideration is given below.

```
      program EX1PT1

      integer N0,BCT(1000),N,i,ians

      double precision xb(1000),yb(1000),xm(1000),ym(1000),
     & nx(1000),ny(1000),lg(1000),BCV(1000),
     & phi(1000),dphi(1000),pint,dl,xi,eta,pi

      print*,'Enter number of elements per side (<250):'
      read*,N0
      N=4*N0

      pi=4d0*datan(1d0)
      dl=1d0/dfloat(N0)

      do 10 i=1,N0
      xb(i)=dfloat(i-1)*dl
      yb(i)=0d0
      xb(N0+i)=1d0
      yb(N0+i)=xb(i)
      xb(2*N0+i)=1d0-xb(i)
      yb(2*N0+i)=1d0
      xb(3*N0+i)=0d0
      yb(3*N0+i)=1d0-xb(i)
   10 continue
      xb(N+1)=xb(1)
      yb(N+1)=yb(1)

      do 20 i=1,N
      xm(i)=0.5d0*(xb(i)+xb(i+1))
      ym(i)=0.5d0*(yb(i)+yb(i+1))
      lg(i)=dsqrt((xb(i+1)-xb(i))**2d0+(yb(i+1)-yb(i))**2d0)
      nx(i)=(yb(i+1)-yb(i))/lg(i)
```

```
      ny(i)=(xb(i)-xb(i+1))/lg(i)
   20 continue

      do 30 i=1,N
      if (i.le.N0) then
      BCT(i)=1
      BCV(i)=0d0
      else if ((i.gt.N0).and.(i.le.(2*N0))) then
      BCT(i)=0
      BCV(i)=dcos(pi*ym(i))
      else if ((i.gt.(2*N0)).and.(i.le.(3*N0))) then
      BCT(i)=1
      BCV(i)=0d0
      else
      BCT(i)=0
      BCV(i)=0d0
      endif
   30 continue

      call CELAP1(N,xm,ym,xb,yb,nx,ny,lg,BCT,BCV,phi,dphi)

   50 print*,'Enter coordinates xi and eta of an interior point:'

      read*,xi,eta

      call CELAP2(N,xi,eta,xb,yb,nx,ny,lg,phi,dphi,pint)

      write(*,60)pint,(dexp(pi*xi)-dexp(-pi*xi))*dcos(pi*eta)
     & /(dexp(pi)-dexp(-pi))
   60 format('Numerical and exact values are:',
     & F14.6,' and',F14.6,' respectively')

      print*,'To continue with another point enter 1:'
      read*,ians

      if (ians.eq.1) goto 50

      end
```

All the subprograms needed for compiling `EX1PT1` into an executable program are the subroutines `CELAP1`, `CELAP2`, `CPF` and `SOLVER` (together with its supporting subprograms `DAXPY`, `DSCAL` and `IDAMAX`).

It is easy to check that boundary value problem here has the exact solution

$$\phi = \frac{\sinh(\pi x)\cos(\pi y)}{\sinh(\pi)}.$$

In the program `EX1PT1` above, the numerical value of ϕ (as calculated by the boundary element procedure with constant elements) at an input interior point (ξ, η) is compared with the exact solution.

Table 1.1

(ξ, η)	20 elements	80 elements	Exact
$(0.10, 0.20)$	0.022605	0.022397	0.022371
$(0.10, 0.30)$	0.016454	0.016279	0.016254
$(0.10, 0.40)$	0.008681	0.008560	0.008545
$(0.50, 0.20)$	0.163153	0.161521	0.161212
$(0.50, 0.30)$	0.118290	0.117325	0.117127
$(0.50, 0.40)$	0.062107	0.061673	0.061577
$(0.90, 0.20)$	0.586250	0.590103	0.589941
$(0.90, 0.30)$	0.427451	0.428609	0.428618
$(0.90, 0.40)$	0.223159	0.225308	0.225338

The numerical values of ϕ at various interior points obtained by `EX1PT1` using 20 and 80 boundary elements are compared with the exact solution in Table 1.1. There is a significant improvement in the accuracy of the numerical results when the number of boundary elements used is increased from 20 to 80.

Table 1.2

a	0.900	0.950	0.990	0.995	0.999
20 elements	0.136%	2.830%	8.504%	9.563%	10.601%
80 elements	0.111%	0.144%	0.716%	1.403%	2.213%

We also examine the accuracy of the numerical value of ϕ at the interior point (a, a) as a approaches 1 from below, that is, as the point (a, a) gets closer and closer to the point $(1, 1)$ on the boundary of the square domain. The percentage errors in the numerical values of ϕ from calculations using 20 and 80 boundary elements are shown in Table 1.2 for various values of a. In each of the two sets of results, it is interesting to note that the percentage error grows as a approaches 1. For a fixed value of a near 1, the percentage error of the numerical value of ϕ calculated with 80 elements are lower than that obtained using 20 elements. It is a 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.

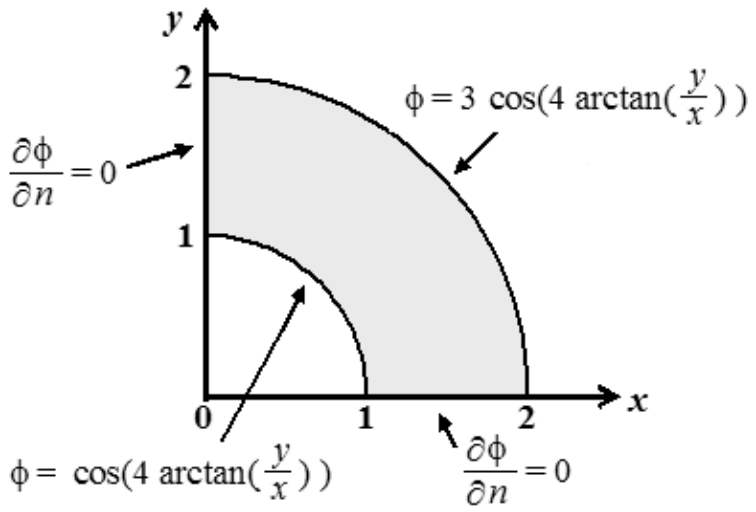

Figure 1.6

Example 1.2

Take the solution domain to be the region bounded between the circles $x^2 + y^2 = 1$ and $x^2 + y^2 = 4$ in the first quadrant of the Oxy plane as shown in Figure 1.6. The boundary conditions are given by

$$\begin{aligned}
\frac{\partial \phi}{\partial n} &= 0 \text{ on the straight side } x = 0,\ 1 < y < 2, \\
\frac{\partial \phi}{\partial n} &= 0 \text{ on the straight side } y = 0,\ 1 < x < 2, \\
\phi &= \cos(4 \arctan(\frac{y}{x})) \text{ on the arc } x^2 + y^2 = 1,\ x > 0,\ y > 0, \\
\phi &= 3\cos(4 \arctan(\frac{y}{x})) \text{ on the arc } x^2 + y^2 = 4,\ x > 0,\ y > 0.
\end{aligned}$$

This boundary value problem may be solved numerically using the boundary element procedure with constant elements as in Example 1.1. To do this, we only have to modify the parts in the program `EX1PT1` that generate input data for the arrays `xb(1:N+1)`, `yb(1:N+1)`, `BCT(1:N)` and `BCV(1:N)`. Before we modify the program, we have to work out formulae for the boundary points $(x^{(1)}, y^{(1)})$, $(x^{(2)}, y^{(2)})$, $\cdots$, $(x^{(N-1)}, y^{(N-1)})$ and $(x^{(N)}, y^{(N)})$.

Let us discretize each of the straight sides of the boundary into N_0 elements and the arcs on $x^2 + y^2 = 1$ and $x^2 + y^2 = 4$ into $2N_0$ and $8N_0$ elements respectively, so that $N = 12N_0$. Specifically, the boundary points are given by

$$\begin{aligned}
(x^{(m)}, y^{(m)}) &= (1 + \frac{[m-1]}{N_0}, 0) \text{ for } m = 1, 2, \cdots, N_0, \\
(x^{(m+N_0)}, y^{(m+N_0)}) &= (2\cos(\frac{[m-1]\pi}{16N_0}), 2\sin(\frac{[m-1]\pi}{16N_0})) \text{ for } m = 1, 2, \cdots, 8N_0,
\end{aligned}$$

$$(x^{(m+9N_0)}, y^{(m+9N_0)}) = (0, 2 - \frac{[m-1]}{N_0}) \text{ for } m = 1, 2, \cdots, N_0,$$
$$(x^{(m+10N_0)}, y^{(m+10N_0)}) = (\sin(\frac{[m-1]\pi}{4N_0}), \cos(\frac{[m-1]\pi}{4N_0})) \text{ for } m = 1, 2, \cdots, 2N_0.$$

Thus, for the boundary value problem presently under consideration, the code for generating the input data for the boundary points in the real arrays `xb(1:N+1)` and `yb(1:N+1)` is as given below. Note that we are required to supply an input value for the integer `N0`.

```
      N=12*N0
      pi=4d0*datan(1d0)

      do 10 i=1,8*N0
      dl=pi/dfloat(16*N0)
      xb(i+N0)=2d0*dcos(dfloat(i-1)*dl)
      yb(i+N0)=2d0*dsin(dfloat(i-1)*dl)
      if (i.le.N0) then
      dl=1d0/dfloat(N0)
      xb(i)=1d0+dfloat(i-1)*dl
      yb(i)=0d0
      xb(i+9*N0)=0d0
      yb(i+9*N0)=2d0-dfloat(i-1)*dl
      endif
      if (i.le.(2*N0)) then
      dl=pi/dfloat(4*N0)
      xb(i+10*N0)=dsin(dfloat(i-1)*dl)
      yb(i+10*N0)=dcos(dfloat(i-1)*dl)
      endif
   10 continue
      xb(N+1)=xb(1)
      yb(N+1)=yb(1)
```

The code for generating the input data for the integer array `BCT(1:N)` and the real array `BCV(1:N)` is as given below.

```
      do 30 i=1,N
      if ((i.le.N0).or.((i.gt.(9*N0)).and.(i.le.(10*N0)))) then
      BCT(i)=1
      BCV(i)=0d0
      else if ((i.gt.N0).and.(i.le.(9*N0))) then
      BCT(i)=0
```

```
      BCV(i)=3d0*dcos(4d0*datan(ym(i)/xm(i)))
      else
      BCT(i)=0
      BCV(i)=dcos(4d0*datan(ym(i)/xm(i)))
      endif
   30 continue
```

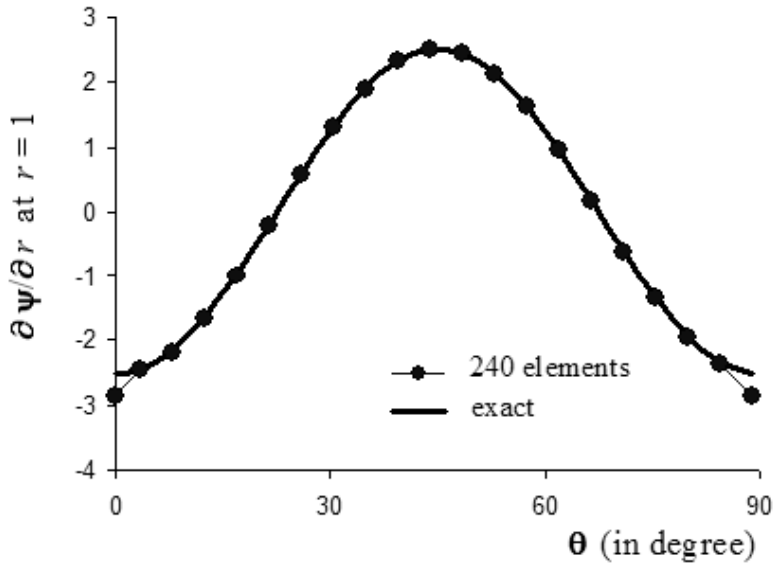

Figure 1.7

As ϕ is specified on the arc $x^2+y^2=1$, $x>0$, $y>0$, the last $2N_0$ variables in the array `dphi(1:N)` returned by `CELAP1` give us the numerical values of $\partial\phi/\partial n$ at the midpoints of the last $2N_0$ boundary elements, that is, $-\partial\psi/\partial r$ at those midpoints if we define $\psi(r,\theta)=\phi(r\cos\theta, r\sin\theta)$, where the polar coordinates r and θ are given by by $x=r\cos\theta$ and $y=r\sin\theta$. We may print out these variables to obtain the approximate values of $\partial\psi/\partial r$ at the midpoints of the last $2N_0$ boundary elements. In Figure 1.7, the numerical $\partial\psi/\partial r$ at $r=1$, $0<\theta<\pi/2$, obtained using 240 elements (that is, using $N_0=20$) are compared graphically against the values obtained from the exact solution[‡] given by

$$\phi=[\frac{16}{85}([x^2+y^2]^2-\frac{1}{[x^2+y^2]^2})-\frac{16}{255}(\frac{[x^2+y^2]^2}{16}-\frac{16}{[x^2+y^2]^2})]\cos(4\arctan(\frac{y}{x})).$$

The numerical values show a good agreement with the exact ones except at points that are extremely close to the corner points $(0,1)$ and $(1,0)$, that is, except at near $\theta=0$ and $\theta=\pi/2$.

[‡]Refer to page 202 of the book *Partial Differential Equations in Mechanics 1* by APS Selvadurai (Springer-Verlag, 2000).

The numerical values of ϕ at selected points in the interior of the solution domain, obtained using 240 elements, are compared with the exact solution in Table 1.3. There is a good agreement between the two sets of results. The interior points in the last two rows of Table 1.3 are close to the corner point $(1, 0)$. Note that the errors of the numerical values at these two points are higher compared with those at the other points. When we repeat the same calculation using 480 elements ($N_0 = 40$), the numerical values of ϕ are 0.826108 and 0.974111 at $(1.099998, 0.001920)$ and $(1.010000, 0.000176)$ respectively, that is, we observe a significant improvement in the accuracy of the numerical values at the two points.

Table 1.3

(ξ, η)	240 elements	Exact
$(1.082532, 0.625000)$	-0.392546	-0.392045
$(0.875000, 1.515544)$	-0.908254	-0.907816
$(1.060660, 1.060660)$	-1.094489	-1.094211
$(1.099998, 0.001920)$	0.824548	0.826958
$(1.010000, 0.000176)$	0.960174	0.975656

1.9 Summary and Discussion

A boundary element solution for the interior boundary value problem defined by Eqs. (1.1)-(1.2) is given by Eq. (1.32) together with Eqs. (1.28), (1.30) and (1.31). The solution is constructed from the boundary integral solution in Eq. (1.23). Constant elements are used, that is, the boundary (of the solution domain) is discretized into straight line elements and the solution ϕ and its normal derivative $\partial\phi/\partial n$ on the boundary are approximated as constants over a boundary element.

As no discretization of the entire solution domain is required, the boundary element solution may be easily implemented on the computer for problems involving complicated geometries and general boundary conditions. The boundary may be easily discretized into line elements by merely placing on it well spaced out points. We have discussed in detail how the numerical procedure can be coded in FORTRAN 77. In spite of the specific programming language used, our discussion may still be useful to readers who are interested in developing the method using other software tools (such as C++ and MATLAB), as FORTRAN 77 codes are relatively easy to decipher.

The term "direct boundary element method" is often used to describe the boundary element procedure given in this chapter. This is because the unknowns in the formulation given by Eq. (1.30) can be directly interpreted as values of ϕ or $\partial\phi/\partial n$ on the boundary. An alternative boundary element method may be obtained

from the simpler boundary integral solution

$$\phi(x,y) = \int_C A(\xi,\eta)\ln([x-\xi]^2+[y-\eta]^2)ds(\xi,\eta),$$

where $A(\xi,\eta)$ is a (boundary) weight function yet to be determined. To determine $A(\xi,\eta)$ approximately, we discretize C into boundary elements $C^{(1)}$, $C^{(2)}$, $\cdots$, $C^{(N-1)}$ and $C^{(N)}$as before, and approximate $A(\xi,\eta)$ as a constant $A^{(m)}$ over $C^{(m)}$, in order to obtain the approximation

$$\phi(x,y) \simeq \sum_{m=1}^{N} A^{(m)} \int_{C^{(m)}} \ln([x-\xi]^2+[y-\eta]^2)ds(\xi,\eta).$$

The constants $A^{(m)}$ are to be determined by using the given boundary conditions. We shall not go into further details here other than pointing out that such as an approach gives rise to a so called indirect boundary element method as the unknowns $A^{(m)}$ are not related to ϕ or $\partial\phi/\partial n$ on the boundary in a simple and direct manner.

1.10 Exercises

1. If ϕ satisfies the two-dimensional Laplace's equation in the region R bounded by a simple closed curve C, use the divergence theorem to show that

$$\int_C \frac{\partial}{\partial n}[\phi(x,y)]ds(x,y) = 0.$$

(Note. This implies that if we prescribe $\partial\phi/\partial n$ at all points on C in our boundary value problem we have to be careful to ensure the above equation is satisfied. Otherwise, the boundary value problem does not have a solution.)

2. If ϕ satisfies the two-dimensional Laplace's equation in the region R bounded by the curve C, use the divergence theorem to derive the relation

$$\iint_R |\underline{\nabla}\phi(x,y)|^2 dxdy = \int_C \phi(x,y)\frac{\partial}{\partial n}[\phi(x,y)]ds(x,y).$$

Hence show that: (a) if $\phi = 0$ at all points on C then $\phi = 0$ at all points in R, that is, show that if the boundary conditions are given by $\phi = 0$ on C then the solution of our boundary value problem is uniquely given by $\phi = 0$ for $(x,y) \in R$, and (b) if $\partial\phi/\partial n = 0$ at all points on C then ϕ can be any arbitrary constant function in R, that is, if the boundary conditions are given by $\partial\phi/\partial n = 0$ on C, then our boundary value problem has infinitely many solutions given by $\phi = c$ for $(x,y) \in R$, where c is an arbitrary constant.

3. Use the result in Exercise 2(a) above to show that if the boundary conditions are given by $\phi = f(x, y)$ at all points on the simple closed curve C then the boundary value problem governed by the two-dimensional Laplace's equation in the region R has a unique solution. [Hint. Show that if ϕ_1 and ϕ_2 are any two solutions satisfying the Laplace's equation and the boundary conditions under consideration then $\phi_1 = \phi_2$ at all points in R.] (Notes. (1) In general, for the interior boundary value problem defined by Eqs. (1.1)-(1.2) to have a unique solution, ϕ must be specified at **at least one** point on C. (2) For the case in which $\partial\phi/\partial n$ is specified at all points on C, ϕ is only determined to within an arbitrary constant. In such a case, the boundary element procedure in this chapter may still work to give us one of the infinitely many solutions.)

4. Eq. (1.8) is not the only solution of the two-dimensional Laplace's equation that is not well defined at the single point (ξ, η). By differentiating Eq. (1.8) partially with respect to x and/or y as many times as we like, we may generate other solutions that are not well defined at (ξ, η). An example of these other solutions is

$$\phi(x, y) = \frac{(x - \xi)}{2\pi[(x - \xi)^2 + (y - \eta)^2]}.$$

   If we denote this solution by $\Phi(x, y; \xi, \eta)$ (like what we had done before for the solution in Eq. (1.8)), investigate whether we can still derive the boundary integral solution as given by Eq. (1.19) from the reciprocal relation in Eq. (1.10) or not.

5. Explain why the parameter $\lambda(\xi, \eta)$ in Eq. (1.23) can be calculated using

$$\lambda(\xi, \eta) = \int_C \frac{\partial}{\partial n}(\Phi(x, y; \xi, \eta))ds(x, y).$$

   Taking C to be the boundary of the triangular region $y < -x + 1$, $x > 0$, $y > 0$, evaluate the line integral above to check that: (a) $\lambda(2, 1) = 0$, (b) $\lambda(1, 0) = 1/8$, (c) $\lambda(0, 0) = 1/4$, (d) $\lambda(1/2, 1/2) = 1/2$, and (e) $\lambda(1/2, 1/4) = 1$.

6. The boundary element solution given in this chapter provides us with an approximate but explicit formula for calculating ϕ at any interior point (ξ, η) in the solution domain. We may also be interested in computing the vector quantity $\underline{\nabla}\phi$. Can an approximate explicit formula be obtained for $\underline{\nabla}\phi$ at (ξ, η)? How can we obtain one?

7. Modify the program `EX1PT1` in Section 1.7 to solve numerically the Laplace's equation given by Eq. (1.1) in the region $x^2 + y^2 < 1$, $x > 0$, $y > 0$, subject to