BIE Planar Manual © 2012

Kendall Atkinson
University of Iowa

April 19, 2013

Abstract

This manual explains the mathematical theory, construction, and use of a general purpose MATLAB package for solving Laplace's equation for planar problems.

1 Introduction

This manual describes the programs in a MATLAB package for solving various boundary integral equation reformulations for solving

$$\Delta u(x,y) = 0, \qquad (x,y) \in D$$

with a variety of boundary condition. We consider both interior and exterior problems, Dirichlet and Neumann boundary conditions, and boundaries $S \equiv \partial D$ that can be either smooth or polygonal. We consider Nyström methods, collocation methods, and a discrete Galerkin method, thus giving the user the chance to experiment with different kinds of numerical methods.

The following notation is used in this manual. The region D can be either an interior (bounded) or an exterior (unbounded) open region. When referring to an interior problem, we use D_i, and for an exterior region, D_e. In all cases we assume the region D_i is simply-connected; solving Laplace's equation on multiply-connected regions is not included in this package. For convenience we often, but not always, assume the origin $\mathbf{0} = (0,0)$ is in the bounded portion into which D divides the plane: $\mathbf{0} \in D_i$ or $\mathbf{0} \in \mathbb{R}^2 \backslash \overline{D}_e$. The boundary S is assumed to have a parameterization $\mathbf{r}(t) = (\xi(t), \eta(t))$, $0 \le t \le L$, traversed in a counter-clockwise direction. For smooth boundaries, this parameterization is to be given explicitly by the user; and in this case, some of the programs allow the point $\mathbf{r}(0) = \mathbf{r}(L)$ to be a corner point with the interior angle in $(0, 2\pi)$. Also for smooth boundaries, the derivatives $\mathbf{r}^{(j)}(t)$ should be continuous on $[0, 2\pi]$ for $j \ge 2$, at a minimum. If there is a corner point at $\mathbf{r}(0) = \mathbf{r}(L)$, then $\mathbf{r}'(0) \ne \mathbf{r}'(L)$. With polygonal boundaries, the user need give only the coordinates of the corner points, and the programs will produce the needed parameterization.

The unit normal into D_i at $\mathbf{r}(t)$ is denoted by $\mathbf{n}(t)$,

$$\mathbf{n}(t) = \frac{1}{\sqrt{\xi'(t)^2 + \eta'(t)^2}} \left(-\eta'(t), \xi'(t)\right).$$

When the normal is at a point $(x, y) \in S$, we also may write $\mathbf{n}(x, y)$.

For the theoretical background to the boundary integral equation formulations and the numerical methods being used, see [2, Chaps 7, 8]. In our package of programs, there are four subdirectories of routines to address the four boundary value problems listed below; the subdirectory *IntDirProb* contains additional lower level subdirectories. Although there is some overlapping of these subdirectories, it seemed best to keep them separate for a variety of reasons. In the lowest level subdirectories there is always a program whose name begins with *demo*. It contains some explanations and sample commands. Read the explanations and try those commands as an introduction to the programs in the subdirectory. As is common in using MATLAB, the demonstration programs pause after plotting each figure, and the user should push any key to continue.

The MATLAB package for the programs discussed here can be downloaded from

http://www.math.uiowa.edu/~atkinson/ftp/PlanarBiePak/planar_bie.tar.zip

The author would appreciate any comments and suggestions for improving this package and this manual. Send an email to

Kendall-Atkinson@uiowa.edu

2 The Problems

The following boundary value problems are addressed by our programming package. The programs are divided first according to the type of problem being solved. A further division is used for the interior Dirichlet problem based on the nature of the boundary, and thence on the type of numerical method or the form of boundary integral equation being used. Most of the demonstration programs assume the interior region D_i is starlike with respect to a given point; but the main programs for solving the boundary integral equation do not make this assumption.

2.1 Interior Dirichlet Problem

$$\begin{aligned} \Delta u(x, y) &= 0, \qquad (x, y) \in D_i, \\ u(x, y) &= f(x, y), \qquad (x, y) \in S. \end{aligned} \tag{1}$$

This is addressed in the program subdirectory named *IntDirProb*. The boundary S is required to be smooth in some of the program packages (in *IntDirProb/Smooth*), although the package *IntDirProb/Smooth/Collocation* allows for a single corner. The boundary is assumed to be polygonal in *IntDirProb/Polygonal*.

2.2 Exterior Dirichlet Problem

$$\begin{aligned} &\Delta u\,(x,y) = 0, \qquad (x,y) \in D_e, \\ &u\,(x,y) = f\,(x,y)\,, \qquad (x,y) \in S, \\ &\sup_{(x,y)\in D_e} |u\,(x,y)| < \infty. \end{aligned} \tag{2}$$

See the package *ExtDirProb*.

2.3 Interior Neumann Problem

$$\begin{aligned} &\Delta u\,(x,y) = 0, \qquad (x,y) \in D_i, \\ &\frac{\partial u\,(x,y)}{\partial \mathbf{n}\,(x,y)} = f\,(x,y)\,, \qquad (x,y) \in S, \\ &u\,(\mathbf{0}) = 0. \end{aligned} \tag{3}$$

In this case we assume $\mathbf{0} \in D_i$. For a solution u to exist, unique up to the addition of an arbitrary constant, it is necessary and sufficient that

$$\int_S f\,ds = 0. \tag{4}$$

See the package *IntNmnProb*.

2.4 Exterior Neumann Problem

$$\begin{aligned} &\Delta u\,(x,y) = 0, \qquad (x,y) \in D_e, \\ &\frac{\partial u\,(x,y)}{\partial \mathbf{n}\,(x,y)} = f\,(x,y)\,, \qquad (x,y) \in S, \\ &\lim_{|(x,y)|\to\infty} u\,(x,y) = 0. \end{aligned} \tag{5}$$

In this case we assume $\mathbf{0} \in D_i$. Again, it is necessary that (4) hold, and then there is a unique solution u. See the package *ExtNmnProb*.

3 The Boundary Integral Equations

The various integral equations make use of the following special potentials, both of which satisfy $\Delta u\,(A) \equiv 0$, $A = (x,y) \in \mathbb{R}^2 \backslash S$. The *single layer potential* with density function ρ is given by

$$u(A) = \int_S \rho(Q)\,[\log|A-Q|]\;dS_Q, \qquad A \in \mathbb{R}^2. \tag{6}$$

The *double layer potential* with density function ρ is given by

$$u(A) = \int_S \rho(Q) \frac{\partial}{\partial \mathbf{n}_Q}\,[\log|A-Q|]\;dS_Q, \qquad A \in \mathbb{R}^2. \tag{7}$$

3.1 The interior Dirichlet problem

For interior problems, the following 'Green's identity' is used to obtain immediately *direct boundary integral equations*, and it is also used to create *indirect boundary integral equations*. For any sufficiently smooth function $u \in C^1(\overline{D}_i) \cap C^2(D_i)$ that satisfies Laplace's equation over D_i,

$$\int_S \left\{ \frac{\partial u(Q)}{\partial \mathbf{n}_Q} \log|P-Q| - u(Q) \frac{\partial}{\partial \mathbf{n}_Q} \left[\log|P-Q|\right] \right\} dS_Q = \begin{cases} 2\pi u(P), & P \in D_i \\ \Omega(P) u(P), & P \in S \\ 0, & P \in D_e \end{cases}, \tag{8}$$

with $\Omega(P)$ the interior angle at $P \in S$. For smooth curves S, of course $\Omega(P) \equiv \pi$. For corner points $P \in S$, we assume $\Omega(P) \in (0, 2\pi)$, avoiding cusps. The integrals on the left side of the equation are single and double layer potentials, respectively.

For the interior Dirichlet problem (1), we have the following boundary integral equation of the first kind for a smooth boundary S:

$$\int_S \rho(Q) \log|P-Q| \, dS_Q = \pi f(P) - \int_S f(Q) \frac{\partial}{\partial \mathbf{n}_Q} \left[\log|P-Q|\right] dS_Q, \qquad P \in S, \tag{9}$$

with $\rho(Q) \equiv \partial u(Q)/\partial \mathbf{n}_Q$ the unknown function. After solving for ρ, the identity (8) can be used to construct $u(P)$ for $P \in D_i$,

$$u(P) = \frac{1}{2\pi} \int_S \left\{ \rho(Q) \log|P-Q| - f(Q) \frac{\partial}{\partial \mathbf{n}_Q} \left[\log|P-Q|\right] \right\} dS_Q, \qquad P \in D_i. \tag{10}$$

This is the focus of *IntDirProb/Smooth/FirstKindDirect*. The only difficulty with this formulation is that (9) is not uniquely solvable for certain boundaries, those whose boundary is a Γ-contour. For a discussion of this, see [5]. The problem is avoided if $\operatorname{diam}(D_i) < 1$; and re-scaling of the Laplace equation will always make this possible.

As a side-note, the identity (8) can be used to create a boundary integral equation of the second kind to solve the Neumann problem (3):

$$\pi u(P) + \int_S u(Q) \frac{\partial}{\partial \mathbf{n}_Q} \left[\log|P-Q|\right] dS_Q = \int_S f(Q) \log|P-Q| \, dS_Q, \qquad P \in S. \tag{11}$$

Unfortunately this integral equation is not uniquely solvable, with $u(P) \equiv 1$ a solution of the homogeneous equation. This can be fixed as in [1]; but we apply an alternative approach, using the Kelvin transform in combination with solving the exterior Neumann problem (5). This is discussed below following (26).

The best known boundary integral equation for solving the interior Dirichlet problem (1) is obtained by assuming the solution u can be represented by the

double layer potential of (7). In general, the dipole density function ρ must satisfy

$$-\pi\rho\,(P) + \mathcal{K}\rho\,(P) = f, \tag{12}$$

$$\mathcal{K}\rho(P) = (-\pi + \Omega(P))\,\rho(P) + \int_S \rho(Q)\frac{\partial}{\partial \mathbf{n}_Q}\left[\log|P-Q|\right] dS_Q, \qquad P \in S. \tag{13}$$

For ρ a continuous function on S, the function $\mathcal{K}\rho$ is also continuous on S. The possible discontinuity in $(-\pi + \Omega(P))\,\rho(P)$ is matched by a corresponding discontinuity in the integral term.

For a smooth boundary S, this leads to solving the equation

$$-\pi\rho(P) + \int_S \rho(Q)\frac{\partial}{\partial \mathbf{n}_Q}\left[\log|P-Q|\right] dS_Q = f(P), \qquad P \in S. \tag{14}$$

This integral equation is uniquely solvable for all $f \in C\,(S)$ or $f \in L^2\,(S)$. The kernel may appear to be discontinuous, even singular, but in fact it is continuous. Using the representation $\mathbf{r}(t) = (\xi(t), \eta(t))$ for S, $0 \le t \le L$, we can rewrite (14) as

$$-\pi\rho(t) + \int_0^L K(t,s)\rho(s)\,ds = f(t), \qquad 0 \le t \le L, \tag{15}$$

$$\begin{aligned} K(t,s) &= \frac{\eta'(s)[\xi(t)-\xi(s)] - \xi'(s)[\eta(t)-\eta(s)]}{[\xi(t)-\xi(s)]^2 + [\eta(t)-\eta(s)]^2} \\ &= \frac{\eta'(s)\xi[s,s,t] - \xi'(s)\eta[s,s,t]}{|\mathbf{r}[s,t]|^2}, \qquad s \ne t. \end{aligned} \tag{16}$$

The quantity $\mathbf{r}[s,t]$ is a first order Newton divided difference, and $\xi[s,s,t]$ and $\eta[s,s,t]$ are second order divided differences. From this,

$$K(t,t) = \frac{\eta'(t)\xi''(t) - \xi'(t)\eta''(t)}{2\left\{\xi'(t)^2 + \eta'(t)^2\right\}}. \tag{17}$$

The kernel function $K\,(t,s)$ is continous over $[0,L]\times[0,L]$ provided the curve S is twice continously differentiable. The integral equation (15) is solved in the programs of *IntDirProb/Smooth/Collocation* and *IntDirProb/Smooth/Nystrom*. The potential (7) is then used to obtain further information about the solution u.

For a boundary S that is only piecewise smooth, the formula (12)-(13) is more difficult to solve numerically, and the associated theory is also more complicated; see [2, Chap. 8]. The formula (16) is still valid, but now $K\,(t,s)$ is increasingly singular as $\mathbf{r}\,(t)$ and $\mathbf{r}\,(s)$ approach a corner point from opposite sides.

Another popular alternative for solving the interior Dirichlet problem (1) is to assume the solution u can be represented as a single layer potential (6). The the unknown density function ρ must satisfy

$$\int_S \rho(Q)\left[\log|P-Q|\right]\,dS_Q = f\,(P)\,, \qquad P \in S, \tag{18}$$

an integral equation of the first kind. The programs for this approach to solving (1) are given in *IntDirProb/Smooth/FirstKindIndirect*. Begin with the program *demo_bie_FirstKind*. As discussed earlier following (10), this approach will not work if S is a Γ-contour; and this is avoided if $\text{diam}\,(D_i) < 1$.

3.2 The exterior Neumann problem

There is an analogue to (8) for potentials over the unbounded exterior region D_e. Assume $u \in C^1(\overline{D_e}) \cap C^2(D_e)$, $\Delta u \equiv 0$ on D_e, and

$$\sup_{P \in \overline{D_e}} |u(P)| < \infty.$$

Then

$$\begin{aligned} \int_S \left\{ \frac{\partial u(Q)}{\partial \mathbf{n}_Q} \log|P-Q| - u(Q) \frac{\partial}{\partial \mathbf{n}_Q} \left[\log|P-Q|\right] \right\} dS_Q \\ = \begin{cases} 2\pi \left[u(\infty) - u(P)\right], & P \in D_e \\ 2\pi u(\infty) - \left[2\pi - \Omega(P)\right] u(P), & P \in S \\ 2\pi u(\infty), & P \in D_i \end{cases} . \end{aligned} \tag{19}$$

Consider the exterior Neumann problem (5). To obtain a unique solution, we require $u(\infty) = 0$. For a smooth boundary S, we have

$$-\pi u(P) + \int_S u(Q) \frac{\partial}{\partial \mathbf{n}_Q} \left[\log|P-Q|\right] dS_Q = \int_S f(Q) \log|P-Q| \, dS_Q, \qquad P \in S. \tag{20}$$

The left side of the equation is exactly the same operator as in (14), and thus (20) is uniquely solvable for all possible right sides. This is implemented in *ExtNmnProb* as a solution to the exterior Neumann problem (5).

The right side of (20) must be evaluated numerically, and this must be done with some care as the integrand is singular. To simplify the discussion, assume the parameterization interval is $[0, 2\pi]$; then

$$\int_S f(Q) \log|P-Q| \, dS_Q = \int_0^{2\pi} \left[\log|\mathbf{r}(t) - \mathbf{r}(s)|\right] |\mathbf{r}'(s)| \, f(\mathbf{r}(s)) \, ds, \qquad 0 \le t \le 2\pi.$$

Introduce $\varphi(s) = |\mathbf{r}'(s)| \, f(\mathbf{r}(s))$,

$$\mathcal{A}\varphi(t) = \int_0^{2\pi} \varphi(s) \log \left| 2e^{-\frac{1}{2}} \sin\left(\frac{t-s}{2}\right) \right| ds,$$

$$\mathcal{B}\varphi(t) = \int_0^{2\pi} B(t,x)\varphi(s) \, ds,$$

$$B(t,s) = \begin{cases} \log \dfrac{|\sqrt{e}\,[\mathbf{r}(t) - \mathbf{r}(s)]|}{\left|2\sin\left(\dfrac{t-s}{2}\right)\right|}, & t - s \neq 2m\pi \\ \log|\sqrt{e}\mathbf{r}'(t)|, & t - s = 2m\pi \end{cases} .$$

Then

$$\int_S f(Q)\log|P-Q|\;dS_Q = -\pi\mathcal{A}\varphi(t) + \mathcal{B}\varphi(t), \qquad 0 \le t \le 2\pi. \tag{21}$$

The integral operator $\mathcal{B}$ is well-behaved and can be well-approximated using trapezoidal numerical integration. For the term $\mathcal{A}\varphi(t)$, let

$$\begin{aligned} \varphi(s) &= \frac{1}{\sqrt{2\pi}} \sum_{m=-\infty}^{\infty} \widehat{\varphi}(m) e^{ims}, \\ \widehat{\varphi}(m) &= \frac{1}{\sqrt{2\pi}} \int_0^{2\pi} \varphi(s) e^{-ims} ds \end{aligned}$$

be the Fourier expansion of φ. Then

$$\mathcal{A}\varphi(t) = \frac{1}{\sqrt{2\pi}} \left[\widehat{\varphi}(0) + \sum_{|m|>0} \frac{\widehat{\varphi}(m)}{|m|} e^{ims} \right]. \tag{22}$$

To appproximate $\mathcal{A}\varphi$, we truncate this expansion to some user-given degree and we approximate the Fourier coefficients using trapezoidal numerical integration, again for some user-given number of nodes. This is implemented in *ExtNmnProb*.

3.3 The Kelvin transform

There is a close connection between interior and exterior problems, based on using the inversion of the plane through the unit circle. For $\mathbf{0} \neq (x,y) \in \mathbb{R}^2$, the point

$$\mathcal{T}(x,y) \equiv (\xi,\eta) = \frac{1}{r^2}(x,y), \qquad r^2 = x^2 + y^2 \tag{23}$$

is the inverse of (x,y) with respect to the unit circle. Easily, $\mathcal{T}^2 = I$. For the Jacobian matrix

$$J(\mathcal{T}) \equiv \begin{bmatrix} \dfrac{\partial\xi}{\partial x} & \dfrac{\partial\xi}{\partial y} \\ \dfrac{\partial\eta}{\partial x} & \dfrac{\partial\eta}{\partial y} \end{bmatrix},$$

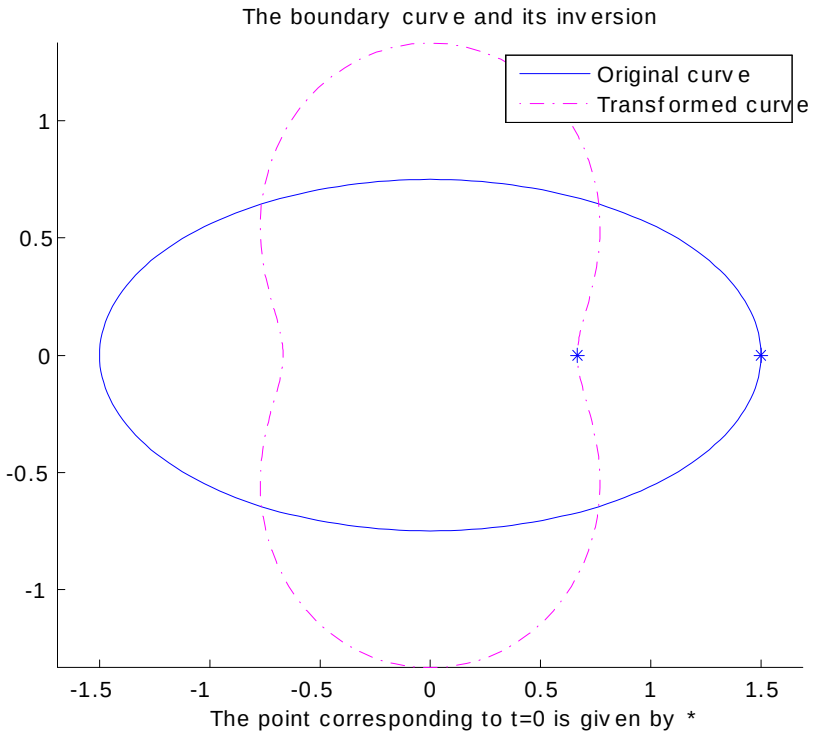
$$\det J(\mathcal{T}) = -\frac{1}{r^2}.$$

Introduce

$$\begin{aligned} \widehat{D}_i &= \mathcal{T}(D_e), \\ \widehat{D}_e &= \mathcal{T}(D_i), \\ \widehat{S} &= \mathcal{T}(S). \end{aligned}$$

Let S be the ellipse

$$(x,y) = (1.5\cos t,\, 0.75\sin t), \qquad 0 \le t \le 2\pi.$$



The image $\widehat{S}$ is shown in Figure 3.3.Given a function u defined over D_e, define

$$\widehat{u}(\xi,\eta) = u\left(\mathcal{T}^{-1}(\xi,\eta)\right), \qquad (\xi,\eta) \in \widehat{D}_i. \tag{24}$$

The function $\widehat{u}$ is defined over the interior region $\widehat{D}_i$ with boundary $\widehat{S}$. Also,

$$\Delta\widehat{u}(\xi,\eta) = r^4 \Delta u(x,y), \qquad (x,y) = \mathcal{T}(\xi,\eta) \in D_e. \tag{25}$$

If u is a potential over D_e with a limit as $(x,y) \to \infty$, then $\widehat{u}$ is a potential over $\widehat{D}_i$.

For normal derivative boundary conditions imposed on u over S, we have

$$\frac{\partial u(x,y)}{\partial \mathbf{n}(x,y)} = -\rho^2 \frac{\partial \widehat{u}(\xi,\eta)}{\partial \widehat{\mathbf{n}}(\xi,\eta)}, \qquad (x,y) = \mathcal{T}(\xi,\eta) \in S \tag{26}$$

with $\rho^2 = \xi^2 + \eta^2 = r^{-2}$, $\mathbf{n}(x,y)$ the inner normal to S at (x,y), and $\widehat{\mathbf{n}}(\xi,\eta)$ the inner normal to $\widehat{S}$ at (ξ,η). This leads to

$$\int_S \frac{\partial u}{\partial \mathbf{n}}\, dS = -\int_S \frac{\partial \widehat{u}}{\partial \widehat{\mathbf{n}}}\, dS.$$

Thus the first integral is zero if and only if the second integral is also zero, true if u is a potential with a limit as $(x, y) \to \infty$.

Using (24), we transform the exterior Dirichlet problem (2) over D_e to an interior Dirichlet problem over $\widehat{D}_i$, and we transform the interior Neumann problem (3) over D_i to an exterior Neumann problem over $\widehat{D}_e$. Boundary conditions over S convert to boundary conditions over $\widehat{S}$, using (24) and (26). The needed programs are given in *ExtDirProb* and *IntNmnProb*.

4 Evaluation of potential-related functions

In addition to evaluating the solution of a boundary integral equation, we often want to evaluate other functions associated with that solution. These vary with the particular boundary integral equation being solved. They include evaluation of the potential function over its domain of interest. Also, for Dirichlet problems, what is the normal derivative over the boundary S.

4.1 Evaluating the double layer potential

The double layer potential $u(A)$ of (7) must be evaluated numerically. We have chosen to use numerical integration over the boundary S. For smooth boundaries we have used primarily trazezoidal numerical integration.

Using the parameterization of the boundary $\mathbf{r}(t) = (\xi(t), \eta(t))$, $0 \le t \le L$, the double layer potential (7) over D_i can be written

$$u(x, y) = \int_0^L M(x, y, s)\rho(s)\, ds, \qquad (x, y) \in D_i, \tag{27}$$

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

This is increasingly singular as the point (x, y) approaches S. To see this more clearly, let S be the unit circle given by $\mathbf{r}(s) = (\cos s, \sin s)$, $0 \le s \le 2\pi$. Then

$$M(x, y, s) = \frac{-\cos s[\cos s - x] - \sin s[\sin s - y]}{[\cos s - x]^2 + [\sin s - y]^2}.$$

To see the near-singular behaviour more clearly, let (x, y) approach $(\cos s, \sin s)$ along the line

$$(x, y) = q(\cos s, \sin s), \qquad 0 \le q < 1.$$

Then after simplifying,

$$M(q\cos s, q\sin s, s) = \frac{1}{1 - q}, \tag{28}$$

showing the singular behaviour more explicitly.

Let $\widetilde{\rho}$ be an approximation of ρ, obtained, say, by solving the boundary integral equation (14), and let $\widetilde{u}(x, y)$ be the corresponding double layer potential,

as in (27). Let $\widetilde{u}_m(x,y)$ denote the result of approximating $\widetilde{u}(x,y)$ using the trapezoidal rule with m subdivisions of $[0,L]$:

$$\widetilde{u}_m(x,y) = h\sum_{i=1}^{m} M(x,y,t_i)\widetilde{\rho}(t_i), \qquad (x,y)\in D_i. \tag{29}$$

A discussion of the error in $\widetilde{u}_m$ is given in [2, §7.2.1] for S a smooth boundary. From the behaviour indicated in (28), the error $u(x,y)-\widetilde{u}_m(x,y)$ for fixed m becomes worse as (x,y) approaches the boundary S. This discussion also applies to the use of (8) and (19) to approximate a potential u when it and its normal derivative are known on the boundary S.

There are two variants to test the evaluation of the double layer potential. In *IntDirProb/Smooth/Collocation*, see the program *demo_ bie*. In both cases, it is assumed that the region D_i is starlike with respect to some given point; see *set_ bdy_ param* in *demo_ bie*. One test of approximating the double layer potential is to evaluate it along a line from the given interior point to something close to the boundary S, and then show a graph of the error along this line. A second test is to evaluate the potential over the entire region. To access these, use *demo_ bie*.

4.2 Approximating the normal derivative

For the Dirichlet problem, the normal derivative of a potential u can be found by solving the first kind boundary integral equation that results from using (8) or (19), as in (9). This is implemented in *IntDirProb/Smooth/FirstKindDirect*.

We now give an alternative for the case that the second kind equation (14) is used to solve the interior Dirichlet problem. Let ρ be the solution to (14) and let u be the associated double layer potential (7). Let $v(x,y)$ be the complex conjugate to u, unique up to the addition of a constant. This refers to creating the analytic function

$$f(z) = u(x,y) + iv(x,y), \qquad z = x+iy.$$

The normal derivative of $u(x,y)$ equals the tangential derivative of v with respect to arc-length on S. We calculate the function $v(\mathbf{r}(t))$, $0\le t\le L$; and then we use its derivative with respect to arc-length to obtain the normal derivative of u.

Calculating v on S uses the identity

$$v(\mathbf{r}(t)) = \frac{1}{\pi}\int_0^L [\rho(s)-\rho(t)]\,\frac{\xi'(s)[\xi(t)-\xi(s)]+\eta'(s)[\eta(t)-\eta(s)]}{|\mathbf{r}(t)-\mathbf{r}(s)|^2}\,ds. \tag{30}$$

For a discussion of this formula, see [4, p. 31]. We calculate this at a user-given number of parameterization points t, and this is then used to construct an interpolating cubic spline. Differentiating the spline is straightforward. We then correct this derivative to compensate for the lack of a arc-length parameterization.

4.3 Polygonal boundaries

The programs in *IntDirProb/Smooth/Collocation* allow the boundary to have a corner at $\mathbf{r}(0) = \mathbf{r}(L)$. Most of the remaining programs in the package assume that smooth boundaries are smooth at all points, including at $t = 0$, namely, $r^{(j)}(0) = r^{(j)}(L)$, $j = 0, 1, 2, \ldots$ To explore the case of piecewise smooth boundaries, we have a package of programs for polygonal boundaries, given in *IntDirProb/Polygonal/Collocation*. There are special routines to assist the user with the parameterization of the boundary. The user need give only the *xy*-coordinates of the corner points; see the demostration routine *create_polygon*. The parameterization function $\mathbf{r}(t)$ is then constructed by the routines *create_polygon_bdy* and *generate_curve_polygon*.

As can be seen from the Green's identities (8) and (19), and also from the formulas (12) and (13), the presence of a corner requires a change in the equation. More importantly, the density function ρ in any of the boundary integral equations being solved will have singularities in their first derivatives with respect to arc-length at the corner points of S. For an extensive discussion of solving boundary integral equations on planar regions with a boundary possessing corners, see [2, Chap. 8].

For a general piecewise smooth boundary S, we assume the interval $[0, L]$ can be subdivided as

$$0 = L_0 < L_1 < ... < L_J = L \tag{31}$$

with $\mathbf{r}(t)$ being several times continuously differentiable on each of the subintervals $[L_{i-1}, L_i]$, $i = 1, ..., J$. Assume the points $\mathbf{r}(L_i)$ are "corners" of the boundary. Let S_i denote the section of S obtained by traveling along it from $\mathbf{r}(L_{i-1})$ to $\mathbf{r}(L_i)$; and $S_J \equiv S_0$, $S_{J+1} \equiv S_1$. At each corner $\mathbf{r}(L_i)$, form tangents to the boundary at $\mathbf{r}(L_i)$ on both the section S_i and S_{i+1}; and let $(1 - \chi_i)\pi$ be the angle interior to D_i formed by these two tangent lines. Assume

$$-1 < \chi_i < 1, \qquad \chi_i \neq 0, \qquad i = 0, 1, ..., J, \tag{32}$$

and of course, $\chi_0 \equiv \chi_J$. The choice $\chi_i = 0$ would correspond to a smooth boundary; and the values $\chi_i = \pm 1$ would correspond to "cusps", yielding boundary value problems and BIE that are much more difficult to treat, both theoretically and numerically.

The solution ρ of (12) has the following behaviour in a neighborhood of the corner $\mathbf{r}(L_i)$:

$$\rho(t) - \rho(L_i) = \mathcal{O}\left(|\mathbf{r}(t) - \mathbf{r}(L_i)|^{\beta}\right), \tag{33}$$

$$\beta = \frac{1}{1 + |\chi_i|}. \tag{34}$$

With the restriction (32) on χ_i, we have $\frac{1}{2} < \beta < 1$, leading to a discontinuous first derivative for $\rho(t)$ for $t \approx L_i$. This affects the choice of a numerical method, as is discussed later.

5 Numerical methods

To solve boundary integral equations of the second kind, we use two main numerical methods: collocation with piecewise polynomial functions and Nyström's method with the trapezoidal rule. For an integral equation of the first kind, as in (9), we use a discrete Galerkin method, explained later. We begin with the collocation method for solving (12), and it will differ for a smooth boundary and for a polygonal boundary. We begin with the former. Afterwards, we discuss the Nyström method for solving (14) when S is smooth.

5.1 Collocation with a smooth boundary

The programs for solving (12) (or (14) when S is smooth at all points including $t = 0$) are given in *IntDirProb/Smooth/Collocation*; and the routine *demo_ bie* gives an overview and demonstration of the programs. For the following, look first at the routine *pw_ poly_ collocation*. The parameterization interval will be divided into N subintervals, and the approximation will be a polynomial on each subinterval. The boundary S is allowed to have a corner at $\mathbf{r}(0)$, and consequently, we also allow grading of the mesh in the neighborhood of that corner point.

If the input parameter `spacing` = *uniform*, then the interval $[0, L]$ is divided into N subintervals of equal length, say

$$0 = T_0 < T_1 < \cdots < T_N = L.$$

The parameter N is to be increased to obtain increased accuracy in the approximating solution $\widetilde{\rho}$. If the input parameter `spacing` = *graded*, then define a graded mesh as follows. For simplicity, assume N is even. For a curve S that allows grading, this will have been noted when defining the curve. For graded meshes, a grading parameter $q \geq 1$ will have been given when setting up the defining parameters for S; see the routine *set_ bdy_ param* inside the demonstration program *demo_ bie*. Then define

$$T_j = \left(\frac{2j}{N}\right)^q \frac{L}{2}, \tag{35}$$

$$T_{N-j} = L - T_j, \tag{36}$$

for $j = 0, 1, \ldots, \frac{1}{2}N$. The case $q = 1$ gives a uniform mesh; and $q > 1$ gives an 'algebraically graded mesh', the most widely studied form of grading.

The approximating solution will be a polynomial of degree $\leq d$, a user supplied parameter. If the input parameter `type` = *equal*, then the $d+1$ collocation points on $[T_{j-1}, T_j]$ will be equally spaced and will include the endpoints of the subinterval; it is assumed $d \geq 1$ in this case. If the input parameter `type` = *gauss*, then the $d+1$ collocation points on $[T_{j-1}, T_j]$ will be the Gauss-Legendre points of order $d + 1$ for the subinterval; it is assumed $d \geq 0$ in this case. If there is a corner point at $\mathbf{r}(0)$, then choose `type` = *gauss*; this avoids the corner point being chosen as a collocation node point. We denote the $d+1$ node points

on $[T_{j-1}, T_j]$ by $\{t_{0,j}, \ldots, t_{d,j}\}$. In the case `type` = *equal*, the endpoint nodes coincide: $t_{d,j} = t_{0,j+1}$, $j = 1, \ldots, N$.

For the collocation solution, we seek a piecewise polynomial function $\widetilde{\rho}$ of degree $\leq d$ on each of the subintervals $[T_{j-1}, T_j]$, and we require the integral equation, say to be satisfied by $\widetilde{\rho}$ at each of the collocation points on $[0, L]$. This leads to solving a linear system. For `type` = *gauss*, the order of the linear system is $(d+1)N$; and for `type` = *equal*, the order is $dN + 1$. The MATLAB program for solving dense linear systems is used. For a discussion of the error in $\rho - \widetilde{\rho}$, see [2, Chap. 3]. To say something more precise about the error in this collocation method, assume the true solution ρ is $(d+1)$-times continuously differentiable over $[0, L]$. The the error satisfies

$$\|\rho - \widetilde{\rho}\|_\infty \leq c\, N^{-(d+1)}, \qquad N \geq 1. \tag{37}$$

For $\mathbf{r}(0)$ a corner of S, see (40) below.

To better understand the linear system to be solved, let $\ell_{i,j}(t)$ denote the i^{th} Lagrange basis function with respect to the j^{th} subinterval $[T_{j-1}, T_j]$. The function $\widetilde{\rho}$ has the formula

$$\widetilde{\rho}(t) = \sum_{i=0}^{d} \widetilde{\rho}(t_{i,j})\, \ell_{i,j}(t), \qquad T_{j-1} < t < T_j, \quad j = 1, \ldots, N, \tag{38}$$

and the coefficients $\{\widetilde{\rho}(t_{i,j}) \mid 0 \leq i \leq d,\ 1 \leq j \leq N\}$ are to be determined. When `type` = *equal*, the function $\widetilde{\rho}$ is continuous over $[0, L]$; when `type` = *gauss*, the function $\widetilde{\rho}$ is generally discontinuous at the endpoints $\{T_j\}$. Recall (15), insert (38), and force equality at the node points $\{t_{i,j}\}$. This yields the linear system

$$-\pi\widetilde{\rho}(t_{i,j}) + \sum_{k=1}^{N} \sum_{m=0}^{d} \widetilde{\rho}(t_{m,k}) \int_{T_{k-1}}^{T_k} K(t_{i,j}, s)\, \ell_{m,k}(s)\, ds = f(t_{i,j})$$

for $0 \leq i \leq d$, $1 \leq j \leq N$. This system contains integrals which must be evaluated numerically. We use Gauss-Legendre quadrature over each subinterval $[T_{j-1}, T_j]$ with a user specified number of nodes, referred to as *quad_order* in the routine *pw_poly_collocation*. Generally only a small number such as 4 or 5 is needed; but the user of the software can experiment with varying choices.

The program *test_evaluate_potential* demonstrates the evaluation of the double layer potential (27) along a specified line in the region D_i, and the program *test_evaluate_potential2* demonstrates the evaluation of the double layer potential over the entire region D_i. The program *test_normal_deriv_method* demonstrates the approximation of the normal derivative over the boundary S. All of these routines are tested together in the program *demo_bie*; examples are given and the calling sequences are illustrated.

5.2 Collocation with a polygonal boundary

Much of the general structure of the preceding case, for S a smooth boundary, carries over to the solution of (12)-(13) for a polygonal boundary. The programs

for this case are contained in *IntDirProb/Polygonal*. Recall the division of $[0, L]$ given in (31). Introduce a further subdivision:

$$\begin{aligned} K_{2j} &= L_j, \qquad j = 0, 1, \ldots, J \\ K_{2j-1} &= \frac{1}{2}\left(L_{j-1} + L_j\right), \qquad j = 1, \ldots, J \end{aligned}$$

In analogy with (35)-(36), define a graded mesh on $[K_{2j-2}, K_{2j-1}]$ as follows. For a given $q \geq 1$ and an integer $M \geq 1$, define

$$T_{i,j} = K_{2j-2} + \left(\frac{i}{M}\right)^q \left(K_{2j-1} - K_{2j-2}\right), \qquad j = 0, 1, \ldots, M.$$

For the interval $[K_{2j-1}, K_{2j}]$, define

$$T_{i,j+M} = K_{2j} - \left(\frac{M-i}{M}\right)^q \left(K_{2j} - K_{2j-1}\right), \qquad j = 0, 1, \ldots, M.$$

Collectively we refer to these points $T_{i,j}$ as $\left\{T_0, T_1, \ldots, T_{N(M)}\right\}$. The parameter q can be varied so as to also vary the grading at the corners

$$\left\{\mathbf{r}\left(L_0\right), \mathbf{r}\left(L_1\right), \ldots, \mathbf{r}\left(L_J\right)\right\},$$

a reasonable strategy in light of the behaviour of the solution ρ indicated in (33). The parameter M will also need to increase to obtain increased accuracy in the collocation solution $\widetilde{\rho}$; and M can also be chosen to vary with the particular corner being considered in defining the mesh, although we do not do so in our programs. We discuss further the varying of q after giving an error result for the collocation solution $\widetilde{\rho}$. For the defining of these points $\{T_{i,j}\}$ in our programs for polygonal boundaries, see *IntDirProb/Polygonal/generate_ nodes_ polygon.*

With this definition of the mesh points $\left\{T_0, T_1, \ldots, T_{N(M)}\right\}$, we then proceed as was done in the smooth boundary case of the preceding subsection. The major difference is that we impose `type` = *gauss*, thus avoiding the use of corner points as interpolation points for the construction of $\widetilde{\rho}$; this ensures that $\Omega\left(P\right) = \pi$ at all collocation points $\mathbf{r}\left(P\right)$ on S.

For the error $\|\rho - \widetilde{\rho}\|_\infty$, we must look at the interpolation error in using our graded mesh for functions with the behaviour of (33). This is a complicated subject and is not easily summarized; see [2, §8.3] for a summary of the results. Assume the grading parameter about $\mathbf{r}\left(L_i\right)$, call it q_i, satisfies

$$q_i \geq \frac{d+1}{\beta_i} \tag{39}$$

with β_i given in (34), for $i = 1, \ldots, J$. Then, in essence,

$$\|\rho - \widetilde{\rho}\|_\infty \leq c\, N\left(M\right)^{-(d+1)}, \qquad M \geq 1, \tag{40}$$

as $M \to \infty$. For a more complete statement, see [2, pp. 410-414]. The assumption (39) is implemented in the definition of the mesh $\{T_{i,j}\}$ in *IntDirProb/Polygonal/generate_ nodes_ polygon.* There is an option to over-ride this choice of $\{q_i\}$ with a fixed value of q that is smaller than given by (39); for example, one could force q to simply equal 1, in which case the mesh is uniform.

5.3 The Nyström method for a smooth boundary

Consider the boundary integral equation (15) for the interior Dirichlet problem with a smooth boundary. Approximate the integral operator using trapezoidal numerical integration,

$$\mathcal{K}\rho\,(t) \equiv \int_0^L K(t,s)\rho(s)\,ds \approx h\sum_{j=1}^{N} K\,(t,t_j)\,\rho\,(t_j) \equiv \mathcal{K}_N\rho\,(t)$$

with N subdivisions of $[0,L]$, $t_j = jh$, and $h = L/N$. Note that we have combined the terms for the endpoints of $[0,L]$ due to the periodicity of the integrand. over the integration interval.

The Nyström method consists of solving

$$-\pi\rho_N(t) + h\sum_{j=1}^{N} K\,(t,t_j)\,\rho_N\,(t_j) = f(t), \qquad 0 \le t \le L, \tag{41}$$

Begin by collocating at the node points, thus solving for $\{\rho_N\,(t_i)\,, i = 1,\ldots,N\}$,

$$-\pi\rho_N(t_i) + h\sum_{j=1}^{N} K\,(ti,t_j)\,\rho_N\,(t_j) = f(t_i), \qquad i = 1,\ldots,N.$$

For the remaining points of $[0,L]$, it can be shown that

$$\rho_N\,(t) = -\frac{1}{\pi}\left\{ f(t) - h\sum_{j=1}^{N} K\,(t,t_j)\,\rho_N\,(t_j)\right\}, \qquad 0 \le t \le L.$$

An extensive discussion of the Nyström method is given in [2, Chap. 4]. For the error,

$$\|\rho - \rho_N\|_\infty \le c\,\|\mathcal{K}\rho - \mathcal{K}_N\rho\|_\infty \tag{42}$$

for some $c > 0$ and for all sufficiently large values of N. Thus the error in the solution ϱ_N is bounded by the numerical integration error in using the trapezoidal rule to approximate $\mathcal{K}\rho$. The programs that use the Nyström method to solve (15) are given in *IntDirProb/Smooth/Nystrom*. Begin with the routine *demo_bie_Nystrom*. Fortran programs using this method were also given in [3] to solve (14) and other boundary integral equations.

5.4 A discrete Galerkin method

Consider the boundary integral equation (9) for the interior Dirichlet problem, writing it as

$$\int_S \rho\,(Q)\log|P-Q|\;dS_Q = g\,(P)\,, \qquad P \in S$$

with g representing the right side of (9). The solution ρ is the normal derivative of u over S. Recall (6) and rewrite the above equation as

$$-\pi \mathcal{A}\rho(t) + \mathcal{B}\rho(t) = g(\mathbf{r}(t)), \qquad 0 \le t \le 2\pi.$$

Look for a solution

$$\rho_N(t) = \frac{1}{\sqrt{2\pi}} \sum_{m=-N}^{N} c_m e^{ims},$$

and define the residual $r_N = -\pi \mathcal{A}\rho_N + \mathcal{B}\rho_N - g$. Determine the coefficients $\{c_{-N}, \ldots, c_0, \ldots, c_N\}$ by requiring

$$\int_0^{2\pi} r_N(t)\, e^{-ikt}\, dt = 0, \qquad k = -N, \ldots, N.$$

Using (22), this leads to the linear system

$$\frac{c_k}{\max\{1,k\}} + \frac{1}{2\pi} \sum_{m=-N}^{N} c_m \int_0^{2\pi} \int_0^{2\pi} B(t,s)\, e^{ims} e^{-ikt}\, ds\, dt$$
$$= \frac{1}{\sqrt{2\pi}} \int_0^{2\pi} g(\mathbf{r}(t))\, e^{-ikt}\, dt, \qquad k = -N, \ldots, N.$$

The integrals must be evaluated numerically. Both the single integrals and the double integrals are approximated using the trapezoidal rule. The user specifies the degree N for ϱ_N and the number of subdivisions to be used for the trapezoidal rule. See the programs in *IntDirProb/Smooth/FirstKindDirect*, and in particular, *Drchlt_Direct_FirstKind*. The program *demo_Drchlt_Direct_FirstKind* demonstrates the various programs in the package *IntDirProb/Smooth/FirstKindDirect*.

5.5 Numerical methods for other boundary value problems

With all of the remaining boundary value problems, we consider only smooth boundaries S. We solve the exterior Neumann problem (5) using the direct boundary integral equation (20), and the programs are in *ExtNmnProb*. Begin with *demo_Nystrom_ExtNmn*. The approximation procedure is the Nyström method. For the interior Neumann problem (3), we apply the Kelvin transform of §3.3 to transform the problem to an exterior Neumann problem; and then the Nyström method is applied. The programs are in *IntNmnProb*. Begin with *demo_Nystrom_IntNmn*. For uniqueness, the potential is set to zero at the origin. For the exterior Dirichlet problem (2), we again use the Kelvin transform, to convert the problem to an interior Dirichlet problem. Then the Nyström method is used. The programs are in *IntDirProb*. Begin with The programs are in *IntNmnProb*. Begin with *demo_Nystrom_IntNmn*. Whenever the Kelvin transform is being used, we assume $\mathbf{0} \in D_i$.

6 Modifying the test problems

As stated earlier, look for the programs whose name begins with *demo*. In each one, there are subprograms named *set_ bdy_ param* and *print_ curve_ data*. These are used in defining the boundary curve S, with the actual definition of S given in *generate_ curve*. These can be modified to define your favorite boundary curve. For problems in which inversion through the unit circle and the Kelvin transform are being used, the origin $\mathbf{0}$ should be inside the interior region D_i. More detail about defining the boundary curve is given below. Boundary data is given in the programs *bdyfcn*, *normal_ deriv_ bdyf*, and *true_ conjugate_ fcn*, and these are discussed in more detail below.

6.1 Defining the boundary curve

The boundary curve is defined in the routine *generate_ curve*. To allow for additional parameters in defining the boundary curve, these are set in *set_ bdy_ param* and are transferred to *generate_ curve* and *print_ curve_ data* via a MATLAB structure, one we ordinarily call *curve_ parms*. As an example, an ellipse has the parametric form

$$(x, y) = (a\cos t, b\sin t)\,, \qquad 0 \le t \le 2\pi,$$

with $a, b > 0$, and this definition is given in *generate_ curve.* The structure *curve_ parms* would contain the parameters a and b. In contrast, the cardioid

$$(x, y) = (c\,\cos(t) + k)\,(a\cos t, b\sin t)\,, \qquad 0 \le t \le 2\pi, \quad k > c > 0,$$

would have *curve_ parms* contain the parameters a, b, k, and c. The present program *set_ bdy_ param* inputs the parameters at runtime from the user. Some labelling of graphs with curve dependent information is done in *print_ curve_ data*.

In addition to containing the curve parameters, other information can be included. For example, for starlike domains, an interior point *int_ pt,* with respect to which the domain is starlike, is given in *curve_ parms*; and information on the subdivision scheme may be included. The latter is expecially important with collocation methods in which a graded mesh may be used. The generating of the subdivision of the curve and the collocation node points is especially important with collocation methods for piecewise polynomial approximations, and a variety of choices are made possible by our schema.

The program *draw_ bdy* is used to draw a boundary curve, and the program *draw_ bdy_ with_ line* is used when evaluating the double layer potential along a line inside the region D_i. These are illustrated with many of the *demo* programs.

6.2 The boundary data programs

In our demonstration programs, the routine *bdyfcn* gives the true solution at all points inside the region $D_i \cup S$ for interior problems and inside the region $D_e \cup S$ for exterior problems. For our demonstrations, we give several examples.

In the example routines *normal_ deriv_ bdy,* it is necessary to give the gradient ∇u for the solution u. We also require the user to give the conjugate function v (named *true_ conjugate_ fcn*) for the several of the program subdirectories associated with solving the interior Dirichlet problem with solution u. As noted earlier, this is unique only up to the addition of an arbitrary constant. The particular subdirectories for which *true_ conjugate_ fcn* is required are *IntDirProb/Polygonal*, *IntDirProb/Smooth/Nystrom*, and *IntDirProb/Smooth/Collocation*.

For exterior problems, we have created our examples by applying the Kelvin transformation to harmonic functions defined over an interior region containing the origin. This results in functions with the desired behaviour at ∞, but it also complicates the calculation of the gradient of the true solution for our demonstration examples. See *bdyfcn* and *normal_ deriv_ bdy* in the subdirectory *ExtNmnProb;* and see *bdyfcn* and *normal_ deriv_ bdy_ Kelvin* in the subdirectory *ExtDirProb*.

References

[1] K. Atkinson. The solution of non-unique linear integral equations, *Numer. Math.* **10** (1967), pp. 117-124.

[2] K. Atkinson. *The Numerical Solution of Integral Equations of the Second Kind*, Cambridge University Press, 1997.

[3] Y.-M. Jeon and K. Atkinson. Algorithm 788: Automatic boundary integral equation programs for the planar Laplace equation, *ACM Trans. on Math. Software* **24** (1998), pp. 395-417.

[4] C. Miller. *Numerical solution of two-dimensional potential theory problems using integral equation techniques*, PhD thesis, Univ. of Iowa, 1979.

[5] Y. Yan and I. Sloan. On integral equations of the first kind with logarithmic kernels, *J. Integral Eqns & Applics* **1** (1988), pp. 517-548.