Exercises on boundary integral equation methods for planar problems

February 13, 2012

These laboratory exercises will be decidedly practical in nature, but there will be occasional theoretical exercises. Some suggestions for convenient function to use in Matlab will be given, but other languages could well be used. We will mainly be looking at Laplace's equation on various domains with different boundary conditions, starting with the most basic setup.

1 Laplace's equation - Dirichlet boundary conditions

The problem we wish to solve is

$$\begin{aligned} \Delta U(P) &= 0, \qquad P \in \Omega, && (1) \\ U(P) &= f(P), \qquad P \in \Gamma, && (2) \end{aligned}$$

where Ω is a smooth interior domain in the plane with boundary Γ.

We begin by expressing the solution $U(P)$ as a double layer potential,

$$U(P) = \int_\Gamma \frac{\partial G(P,Q)}{\partial n_Q} \mu(Q) \mathrm{d}S_Q, \qquad P \in \Omega \tag{3}$$

with n_Q being the outward unit normal at Q and $G(P,Q) = \frac{1}{2\pi}\log(|P-Q|)$ is the Green's function for the Laplace operator in the plane. To compute the solution at some point P we need to first compute the unknown double layer density μ. As P approaches the boundary in (3), the jump relations together with the boundary conditions give the integral equation

$$\frac{1}{2}\mu(P) + \frac{1}{2\pi}\int_\Gamma \mu(Q) \frac{\partial}{\partial n_Q} \log(|P-Q|)\, \mathrm{d}S_Q = f(P), \qquad P \in \Gamma \tag{4}$$

for μ. It is common to absorb the factor 1/2 into the density μ, in which case (3) must be adjusted accordingly.

In complex notation the same equation is

$$\frac{1}{2}\mu(z) + \frac{1}{2\pi}\int_\Gamma \mu(\tau) \Im\left\{\frac{\mathrm{d}\tau}{\tau - z}\right\} = f(z), \qquad z \in \Gamma, \tag{5}$$

where $\Im$ denotes the imaginary part. Note that the density is still real. When (5) has been solved for μ the solution in Ω can be computed via

$$U(z) = \frac{1}{2\pi}\int_\Gamma \mu(\tau) \Im\left\{\frac{\mathrm{d}\tau}{\tau - z}\right\}, \qquad z \in \Omega. \tag{6}$$

Exercise 1 : Confirm that the real and complex equations indeed are equivalent, for example by parameterizing both equations and comparing the results.

We will use complex representations here. The equations generally turn out more compact that way, and also much of the literature on boundary integral equations in the plane use complex notation so it could perhaps be useful to have seen and worked with such notation a bit.

To solve (5) we need to evaluate integrals numerically, and to begin with we will use the trapezoidal rule. For periodic and smooth integrands the trapezoidal rule has very nice convergence properties(see Chapter 3.1.1 in the thesis). Using Nyström discretization/method (see Chapter 4 in Atkinson), we are now in a position to solve Laplace's equation. As for what domain to use, any smooth, reasonably well behaved one can be tried. Circles and such may be too boring, though, and a parametrization for a somewhat more interesting five-armed starfish domain is

$$\tau(t) = (1 + 0.3\cos(5t))\, e^{it}, \qquad t \in [0, 2\pi] \tag{7}$$

Furthermore, we would like to choose the boundary conditions in such a way that we know the correct solution for comparisons. For this we may use

$$f(z) = \Im\left\{\frac{z^2}{z - z_p}\right\}, \qquad z \in \Gamma, \tag{8}$$

where $z_p \notin \Omega$.

Exercise 2 : If (8) is used as right hand side what will $U(z), z \in \Omega$ be and why?

Exercise 3 : Parametrize the boundary of the five-armed starfish domain, and use Nyström discretization to set up a linear equation system for solving (5) with the right hand side as in 8, not forgetting the diagonal elements. Solve it using Gaussian elimination and use the discrete density to compute the solution at points along a line from the center of the starfish to the tip of one of its arms. Compare with the known solution. What happens and why? Try different numbers of discretization points.

Exercise 4 : Compute the condition number of the system matrix for increasing numbers of discretization points and plot them. What is the difference in behaviour compared to finite difference schemes? Also, plot the eigenvalues of the matrix for increasing numbers of discretization points. What does this suggest regarding the nature of the integral operator?

Exercise 5 : Solve the linear system to full accuracy using Matlab's in-built iterative solver `gmres` for increasing numbers of discretization points. What happens in terms of iterations required and what does this mean for the asymptotic time complexity of the solver? What would happen in this regard if the fast multipole method was employed?

2 Laplace's equation - Neumann boundary conditions

We will now look at exterior domains with Neumann boundary conditions. The problem is

$$\begin{aligned} \Delta U(P) &= 0 \qquad P \in \Omega_e, & (9) \\ \frac{\partial U(P)}{\partial n_P} &= f(P) \qquad P \in \Gamma, & (10) \end{aligned}$$

where Ω_e is the infinite domain in the plane outside the boundary Γ. We also need some control over the behaviour of $U(P)$ far away, and we prescribe that $|U(P)| \to 0$ as $|P| \to \infty$. This condition is actually equivalent to the condition that the net flux over the boundary should be zero, that is

$$\int_\Gamma f(Q) \mathrm{d}S_Q = 0, \tag{11}$$

which certainly is a reasonable restriction here.

The standard approach is for the solution to be represented as the single layer potential

$$U(P) = \int_\Gamma G(P,Q)\rho(Q)\mathrm{d}S_Q, \qquad P \in \Omega_e \tag{12}$$

where, again, $G(P,Q) = \frac{1}{2\pi}\log(|P-Q|)$ is the Green's function for the Laplace operator in the plane. In complex notation (12) becomes

$$U(z) = \frac{1}{2\pi}\int_\Gamma \rho(\tau)\log(|z-\tau|)\,|\mathrm{d}\tau|, \qquad z \in \Omega_e. \tag{13}$$

Exercise 6 : Derive the integral equation for the exterior Neumann-problem in complex notation. The result will be similar to the equation for the interior Dirichlet problem. Why is the double layer potential not used to represent $U(z)$?

3 Laplace's equation - Corners

Within the above framework it is also possible to treat domains having corners. There are a few problems that need addressing, though. The most pressing one is that the trapezoidal rule is no longer a good choice for the numerical quadrature. It was very accurate for smooth integrands but in this case the order drops to two. Worse, with the trapezoidal rule there will be quadrature nodes exactly on corners where the kernel is unbounded. Instead of the trapezoidal rule, one can use composite N point Gauss-Legendre quadrature, by dividing the boundary into segments, each containing a set of N quadrature nodes. If one makes sure that the corners are situated at the intersection between segments, no nodes will be exactly on the corners, and furthermore the order of the quadrature rule is $2N$. See 3.1.1 in the thesis.

Exercise 7 : Generate 16-point Gauss-Legendre quadrature nodes and weights for integration on the interval $[-1, 1]$. Look up the algorithm if you don't know how.

Exercise 8 : Set up and solve the integral equation for the interior Dirichlet problem on the unit square using composite 16-point Gauss-Legendre quadrature with four segments on each side of the square. Use the same right hand side as in the smooth case above, and generate a square grid of interior points to compute the solution at. The 10-logarithm of the error at these points can then be displayed using for example `imagesc` and `colorbar` in Matlab to produce a picture of the error over the square.

The result should not be encouraging, compared to how things looked with the smooth domain. A classical "quick fix" for improving matters is to refine the set of quadrature nodes by repeatedly dividing the segments closest to the corners in two. If the closest segments have been divided n times we call this an n-ply refined mesh. The idea is that most of the problems occur in the vicinity of the corners so we allocate the bulk of our resources there.

Exercise 9 : Compute the solution with various levels of refinement and check the error. Similarly as before, look at the spectrum of the system matrix when using an n-ply refined mesh for increasing n. What can be said about the integral operator on a non-smooth domain?

Exercise 10 : Try, by using a well refined mesh, to estimate how the density μ behaves in the vicinity of a corner. That is, if the parameter t goes to zero as we approach the corner from a certain direction, what is α in the leading term t^α of the density(disregarding constant terms)? When estimating the constant term, it is worth remembering that the error in the density is at its largest on the segment closest to the corner, so a better estimation might be found a bit further away.

Exercise 11 : Summarize, by appealing to your findings above, what theoretical and practical problems arise when boundary integral equation methods are applied to non-smooth domains.

By the way, the problems encountered here can be dealt with without excessive refinement(which of course leads to very large linear system for large-scale problems), for example by in a sense making the refinement process implicit, faster and more stable, or by constructing tailor made quadrature rules.

4 Stokes flow

As an example of another problem treatable by integral equation methods we will finally look briefly at Stokes flow. Omitting details(see for example Greengard and Kropinski, "Integral Equation Methods for Stokes flow and Isotropic Elasticity in the plane" for such), one possibility for solving the Stokes flow problem in the interior of a finite domain Ω, with prescribed fluid velocities on the boundary Γ, is to seek a complex density $\omega(z)$ such that

$$\omega(z)+\frac{1}{\pi}\int_{\Gamma}\omega(\tau)\Im\left\{\frac{\mathrm{d}\tau}{\tau-z}\right\}-\frac{1}{\pi}\int_{\Gamma}\overline{\omega(\tau)}\frac{\Im\left\{\mathrm{d}\tau\left(\bar{\tau}-\bar{z}\right)\right\}}{\left(\bar{\tau}-\bar{z}\right)^{2}}=-\mathrm{i}h(z),\qquad z\in\Gamma, \tag{14}$$

where bar denotes complex conjugation and $h(z)$ is the complex fluid velocity on the boundary. When setting up boundary conditions, we need to ensure zero net flux across Γ, which in complex notation is expressed as

$$\Im\left\{\int_{\Gamma}\overline{h(\tau)}\mathrm{d}\tau\right\}=0. \tag{15}$$

Once (14) has been solved, several quantities of interest can be computed using the density $\omega(z)$. The velocity in Ω can be found via

$$u(z)+\mathrm{i}v(z)=\frac{1}{\pi\mathrm{i}}\int_{\Gamma}\omega(\tau)\Im\left\{\frac{\mathrm{d}\tau}{\tau-z}\right\}-\frac{1}{\pi\mathrm{i}}\int_{\Gamma}\overline{\omega(\tau)}\frac{\Im\left\{\mathrm{d}\tau\left(\bar{\tau}-\bar{z}\right)\right\}}{\left(\bar{\tau}-\bar{z}\right)^{2}},\qquad z\in\Omega \tag{16}$$

where $u(z)$ and $v(z)$ are the horizontal and vertical components, respectively. Furthermore, the pressure p and the vorticity ζ are given by

$$p(z)+\frac{i}{\nu}\zeta(z)=\frac{2}{\mathrm{i}\pi}\int_{\Gamma}\frac{\omega(\tau)\mathrm{d}\tau}{\left(\tau-z\right)^{2}},\qquad z\in\Omega \tag{17}$$

where ν is the viscosity.

Solving (14) is a bit cumbersome. The reason is that both $\omega(\tau)$ and $\overline{\omega(\tau)}$ appears in the equation, so discretizing the system straight up will not work. One way to get around this is to expand the linear system to twice the size, treating the real and imaginary parts separately. The complex 1×1 equation system

$$ax=b \tag{18}$$

with $a=a_r+\mathrm{i}a_i$, $x=x_r+\mathrm{i}x_i$ and $b=b_r+\mathrm{i}b_i$, would then be expanded into

$$\begin{bmatrix}a_r & -a_i\\ a_i & a_r\end{bmatrix}\begin{bmatrix}x_r\\ x_i\end{bmatrix}=\begin{bmatrix}b_r\\ b_i\end{bmatrix},$$

following the rules of complex multiplication, and the conjugated variant $a\bar{x}=b$ would be expanded in a similar way.

It should be noted that there are well-defined limits for the integral kernels of (14) when $\tau\to z$. Both involve τ'' though, so on straight line segments as for the box they are both zero(except exactly on the corners).

Exercise 12 : Discretize and solve equation (14) on the unit square using Gauss-Legendre quadrature with refinements as above. As for boundary conditions and for checking the correctness of your code, you could try to simulate pipe flow. That is, setting the inflow and outflow on opposing sides both having quadratic profile, and setting no-slip conditions on the remaining sides. The flow in the domain should then be uniform. After you have verified the code, you could try tangential flow over one of the sides, i.e. lid-driven cavity flow.

NOTE! It should probably be admitted that the equation (14) is actually singular. If you inspect the spectrum of the resulting matrix you will find one eigenvalue ≈ 0. Due to the errors introduced by the corners, however, the matrix is not singular in a numerical sense, so we get sensible results anyway, the null-space does not affect the flow. The null-space can be ”patched up”, so that the resulting system is non-singular, but we skip that here.