2.097J/6.339J/16.920J
Numerical Methods for Partial Differential Equations
Massachusetts Institute of Technology - Fall 2016

Project 6 - Boundary-Element (aka Integral Equation) Methods

Issued: 2 December 2016 *Due: 8 December 2016*

Project Instructions

We encourage you to work in pairs, it is more fun and you get more out of it, but working in pairs is NOT required for this problem set.

Note: Problem Set Sessions: Wed, Thurs, 12/7, 12/8, 7-9pm.

Given the modified schedule, we require that you only turn in brief answers and the plot of the solution for the first problem (indicated as "required") on Nystrom methods. Of course, you are more than welcome to submit solutions for the remaining problems if you like. Attempting more of the problems below is highly encouraged and will prove to be illuminating! The first two are short, the last one is MUCH longer, and you should select only if it matches your interests.

Required Problem: Nystrom methods

In this problem, we ask you to investigate using a Nystrom method to solve a 2-D exterior and an interior Laplace problem with Neumann boundary conditions.

For this problem, the boundary is the unit circle (all points x, y such that $\sqrt{x^2+y^2} = 1$). The boundary condition on the circle is of Neumann type, specifically, the normal derivative of the potential (with the normal pointing out of the circle) is

$$\frac{\partial u}{\partial n}\Big|_\Gamma = \frac{1}{3 + 2\cos\theta + \cos 2\theta}$$

where $\theta \in [-\pi, \pi]$, $x = \cos(\theta)$, and $y = \sin(\theta)$ (Note, the right-hand side for this problem is particularly good for demonstrating the properties of the Nystrom method, and is due to Fan Wang, a former 16.920 student).

For the 2-D Laplace's equation, the monopole integral formulation of the interior Neumann problem is given by

$$\frac{\partial u_\Gamma(\vec{x})}{\partial n_{\vec{x}}} = +\pi\sigma(\vec{x}) - \int_\Gamma^{PV} \frac{(\vec{x}-\vec{x}')^T n_{\vec{x}}}{\|\vec{x}-\vec{x}'\|^2}\sigma(\vec{x}')d\Gamma' \qquad \vec{x} \in \Gamma$$

and for the exterior Neumann problem, the monopole integral formulation is given by

$$\frac{\partial u_\Gamma(\vec{x})}{\partial n_{\vec{x}}} = -\pi\sigma(\vec{x}) - \int_\Gamma^{PV} \frac{(\vec{x}-\vec{x}')^T n_{\vec{x}}}{\|\vec{x}-\vec{x}'\|^2}\sigma(\vec{x}')d\Gamma' \qquad \vec{x}\in\Gamma.$$

a) Use a Nystrom method to solve the exterior Neumann problem, using equally spaced quadrature points on the circle and equal quadrature weights. What do you notice about the matrix elements?

b) When using the Nystrom method with equally spaced points to solve the exterior Neumann problem, how quickly does the method converge. That is, how do the solution errors decrease with increasing numbers of points.

c) Now try solving the *interior* Neumann problem with the equally spaced point Nystrom method. Is the interior problem easy to solve? What goes wrong? Does a solution exist?

d) Instead of using the backslash operator, write instead an iterative solver for your linear system. Attempt to accelerate the solution of your linear system by using the Fast Fourier Transform (FFT). The Discrete Fourier Transform (DFT) computes the following sum:

$$X_k = \sum_{n=0}^{N-1} x_n \exp\left(-2\pi i k n/N\right) \tag{1}$$

The FFT provides an $\mathcal{O}(N\log N)$ method for evaluating N such sums ($0 \le k \le N-1$).

Hint: Relate the matrix you obtain to the FFT matrix and perform the $\mathcal{O}(N^2)$ matrix-vector product in $\mathcal{O}(N\log N)$ time, using the FFT, at every iteration.

Option 1: Extending the Nystrom method to Ellipses

Try solving the *exterior* Neumann problem with the Nystrom method from the first problem, but change the boundary from a circle to an ellipse. Demonstrate that your program achieves exponential convergence, and show how the error behaves with increasing ratio of long to short axis. How did you select where to put the points?

Option 2: Panel Integral Short Problem

In this project, we ask you to investigate using a Gauss-Chebyshev quadrature product rule to evaluate the potential due to a unit monopole density on a flat panel, and to modify the

scheme to be more efficient in the limit as the evaluation point approaches the center of the panel. Such evaluations arise in boundary-element methods for solving three-dimensional Laplace problems, as noted in the first lecture.

More precisely, you will be computing the potential at a distance z above the center of a 2×2 square panel in the $x - y$ plane, $\Phi(z)$, given by

$$\Phi(z) = \int_{-1}^{1} \int_{-1}^{1} \frac{1}{\sqrt{x^2 + y^2 + z^2}} dxdy. \tag{2}$$

To get started, please:

a) Download and install the chebfun toolbox from http://www.chebfun.org/. Make sure that the directory in which you install chebfun is in your MATLAB path.

b) Please download our matlab function chebrect.m from stellar.

c) Try running chebrect with various values for number of quadrature points (try values for n between 5 and 50), and for various distances from the panel (try values for z between zero and one).

Consider generating plots or examining the code to answer the following questions, as they will help you decide how to modify the code.

a) What is the function *kron* doing in the chebrect function?

b) What does the plot of $\Phi(z)$ versus z look like?

c) Where are the quadrature points in relation to the evaluation point?

This problem is quite open-ended, there are lots of approaches to pursue. Please talk with us about it.

Option 3: Long Problem Potential Flow 3D exterior problem

In this problem we will be using the potential flow model of the air velocity $\mathbf{v}(\mathbf{x})$ generated by a wing moving at velocity $\mathbf{v}_w(\mathbf{x})$. In this model we consider the air velocity to be irrotational i.e. $\nabla \times \mathbf{v}(\mathbf{x}) = \mathbf{0}$. The irrotational velocity field can be expressed as a scalar velocity potential $u(\mathbf{x})$, as in

$$\mathbf{v}(\mathbf{x}) = \nabla u(\mathbf{x}). \tag{3}$$

If the air is assumed to be incompressible, the air velocity must have zero divergence as in

$$\nabla \cdot \mathbf{v}(\mathbf{x}) = 0. \tag{4}$$

Combining the above two equations yields Laplace's equation for the velocity potential u,

$$\nabla^2 u(\mathbf{x}) = 0. \tag{5}$$

Since the wing is moving with a velocity $\mathbf{v}_w(\mathbf{x})$, and air can not penetrate the wing, the air velocity in the direction normal to the wing must match the wing velocity. Nonpenetration therefore implies that

$$\frac{\partial u(\mathbf{x})}{\partial \mathbf{n}(\mathbf{x})} = \nabla u(\mathbf{x}) \cdot \mathbf{n}(\mathbf{x}) = \mathbf{v}_w(\mathbf{x}) \cdot \mathbf{n}(\mathbf{x}) \tag{6}$$

where $\mathbf{n}(\mathbf{x})$ is the normal to the wing at point $\mathbf{x}$. Note that the potential flow problem is an *exterior* Neumann problem, and the boundary condition enforces that the normal component of the potential flow velocity at the wing surface matches the normal component of the wing velocity.

The potential flow model is an approximate model, and one surprising aspect of the approximation becomes evident when considering the tangential velocity predicted by the potential flow model. The tangential component of the potential flow velocity *on* the wing surface can be computed using nearby values of the potential on the wing, and that velocity typically does not match the tangential component of the wing velocity. This seems to violate the commonly assumed *no slip* boundary conditions, which require matching the tangential components of the air and wing velocity on the wing surface. The explanation is that the potential flow model is predicting the velocity a small distance away from the wing. The physics of what is happening to the fluid velocity between the wing surface and the distance where potential flow model is valid is beyond the scope of this course. In addition, we should note that this simple potential flow model can not predict lift, as lift can not be predicted by a model with purely irrotational velocity. However, we can calculate the pressure distribution near the wing by using the tangential velocity calculated with the potential flow model and Bernoulli's equation.

Your job in this problem will be to develop a method to solve the exterior Neumann potential flow problem for the wing surface values of the velocity potential u.

a) What is the appropriate radiation condition for the potential flow problem. That is, how rapidly do you expect the velocity to decay with distance from the wing.

b) In this part of the problem we will formulate a boundary integral equation for the potential flow problem by using the Ansatz formulation method. Assume that the velocity potential is given in the form

$$u(\mathbf{x}) = \int_{\Gamma_w} \frac{\sigma(\mathbf{x}')}{\|\mathbf{x} - \mathbf{x}'\|} dS'. \tag{7}$$

If we were given $u(\mathbf{x})$ we would have an exterior Dirichlet problem analogous to the problem solved when calculating capacitance in electrostatic problems. Given $u(\mathbf{x})$ we

would have a first kind integral equation that we could solve for $\sigma(\mathbf{x})$ on the boundary. However, the potential flow problem generates Neumann boundary conditions,

$$\frac{\partial u(\mathbf{x})}{\partial \mathbf{n}_w(\mathbf{x})} = \nabla u(\mathbf{x}) \cdot \mathbf{n}_w(\mathbf{x}) = \mathbf{v}_w(\mathbf{x}) \cdot \mathbf{n}_w(\mathbf{x}). \tag{8}$$

Please formulate a second kind integral equation for the exterior Neumann potential flow problem by differentiating the the Ansatz formulation. Be careful about the "jump" conditions discussed in class.

c) We have developed a set of matlab routines that use piecewise-constant centroid collocation to solve the exterior Dirichlet problem, and would like you to modify the matlab routines to generate a piecewise-constant centroid collocation to solve the exterior Neumann problem you derived in part (b). Our code solves for the charge density on a conductor surface by solving the integral equation

$$u(\mathbf{x}) = \int_\Gamma \frac{\sigma(\mathbf{x}')}{\|\mathbf{x} - \mathbf{x}'\|} d\Gamma'$$

where $\mathbf{x}$ is a point in on the conductor surface, $u(\mathbf{x}) = 1$ for all points $\mathbf{x}$ on the conductor surface, and σ is the unknown surface charge density. Since the potential is fixed at 1 over the conductor surface, our code can be used to compute the capacitance of the conductor.

The matlab code includes several files listed below.

calccap.m:	This is the main routine which calculates capacitance.
calcp.m:	This file contains the routine which analytically computes $\int_{\text{panel}} \frac{1}{\|x - x'\|} dS'$ and can be used to compute panel integrals of other functions.
readpanels.m:	This routines reads a set of panels describing the discretized geometry.
readpanel.m:	This routine is called by `r`eadpanels and does the actual reading.
gencolloc.m:	This routine computes panel centroids.
collocation.m:	This routine sets up the collocation matrix.

We have also provided two examples: a sphere mesh for calibrating your results, and six wing meshes corresponding to four successively refined surface discretizations of the NACA0012 airfoil with and without endcaps. The airfoil mesh files are named `naca0012_*.qif`, the sphere mesh file is named `sphere.qif`. Once you have downloaded the files from the web, you can run an example by typing

```
[C, matrix] = calccap('naca0012_very_coarse_grid_with_endings.qif')
```

and then you can also examine the matrix. We have also included several utility scripts to plot panels, data distribution on panels and panels normals.

c.i) The routine in `calcp.m` can be used to compute the directional derivative of a panel potential at any point $\mathbf{x}$. What result does `calcp.m` produce when the point $\mathbf{x}$ is actually on the panel and the derivative is in the direction of the panel normal. What is the result when the point $\mathbf{x}$ is a little below the panel or a little above the panel. Is `calcp.m` performing correctly?

c.ii) Implement the piecewise-constant collocation scheme for computing σ in the integral equation you derived in part (b), but keep in mind the result in part (c.i) and please pay attention to the direction of the normal used in the derivatives of the potential. You can use `calcp.m` to calculate the panel integrals that you need.

c.iii) To validate your code, set $\mathbf{v}_w$ to be the unit vector along the x direction and calculate the charge distribution on the `sphere` meshes available from the class website. From each charge distribution calculate the potential $u(\mathbf{x})$ on the panel centroids and compare that to the value of the potential at the surface of a sphere with the same area. Theoretically, the value of the velocity potential on a sphere of radius a translating at velocity $\mathbf{V}$ in a uniform medium is given by

$$u(\mathbf{r}) = \mathbf{V} \cdot \frac{\mathbf{r}}{\|r\|_2} \left(-\frac{a^3}{2r^2} \right). \tag{9}$$

Again, please be sure to check on the normal directions used in `calcp.m` are the same directions that you use in your formulation. You can plot the normals using the script `plot_panel_normals.m`.

d) Once you have developed a matlab program for computing σ and the surface potentials, it is still necessary to compute the tangential velocities. Determine a scheme for evaluating the needed velocities, and implement it in Matlab. What happens when you take tangential derivatives of the Ansatz formulation potential, do you observe the same "jump" behavior? Why?

e) Use the script `plot_velocity_field_and_panels.m` to plot the velocity profile at a cross-section on the center of the wing for both the fine and coarse discretization of the wing, and then plot the difference between the potential flow velocity and the wing velocity. Do the results make sense to you?

f) Would switching to a formulation based on using Green's theorem have any advantages?