Homework 6
due Mon, May 8

Let $\Omega = \{\mathbf{x} \in \mathbb{R}^2 \ : \ \frac{x_1^2}{4} + x_2^2 \le 1\}$. Write a computer program to approximate the solution of

$$\begin{aligned} \Delta u &= 0 \quad \text{in } \Omega \\ u &= g \quad \text{on } \partial\Omega \end{aligned} \tag{1}$$

where $g(\mathbf{x}) = \log[(x_1+2)^2 + (x_2+1)^2]$. (Pretend we only know the values on the boundary. Later we can compare the result to the exact solution $u(\mathbf{x}) = \log[(x_1 + 2)^2 + (x_2+1)^2]$.)

We will use a vanilla version of the boundary element method to solve this problem. The idea is to represent the solution using a double layer potential

$$u(\mathbf{x}) = \int_\gamma \partial_{n_y} N(\mathbf{x}, \mathbf{y}) \mu(\mathbf{y}) \, ds(\mathbf{y}), \tag{2}$$

where the Newtonian potential in two dimensions is given by

$$N(\mathbf{x}, \mathbf{y}) = \frac{1}{2\pi} \log|\mathbf{x} - \mathbf{y}| \tag{3}$$

and $\partial_{n_y} N(\mathbf{x}, \mathbf{y}) = \mathbf{n}(\mathbf{y}) \cdot \nabla_y N(\mathbf{x}, \mathbf{y})$, where $\mathbf{n}(\mathbf{y})$ is the outward normal to the boundary at $\mathbf{y}$. We will discretize the problem by breaking the curve γ into n segments

$$\Delta_j = \{(2\cos t, \sin t) \ : \ t_{2j-1} \le t \le t_{2j+1}\}, \qquad j = 1, \dots, n \tag{4}$$

with midpoints (in parameter space)

$$\mathbf{x}_j = (2\cos t_{2j}, \sin t_{2j}), \qquad j = 1, \dots, n. \tag{5}$$

The points t_k are uniformly distributed from 0 to 2π with the last point equal to the first point (modulo 2π):

$$t_k = 2\pi k/(2n), \qquad k = 1, \dots, 2n+1. \tag{6}$$

We will approximate $\mu(\mathbf{y})$ to be constant on each segment, and enforce (2) only at the collocation points $\mathbf{x}_i$:

$$u(\mathbf{x}_i) = \sum_j \left(\int_{\Delta_j} \partial_{n_y} N(\mathbf{x}_i, \mathbf{y}) \, ds(\mathbf{y}) \right) \mu_j = \sum_j A_{ij} \mu_j. \tag{7}$$

This allows us to solve for μ in terms of g. (For the self term (when $j = i$), treat $\mathbf{x}_i$ to be slightly on the interior side of the curve segment Δ_j).

What to turn in:

(1) Find a simple formula for $\int_{\Delta_j} \partial_{n_y} N(\mathbf{x}, \mathbf{y})\, ds(\mathbf{y})$ in terms of $\mathbf{x}$ and the endpoints of the segment Δ_j.

Set $n = 96$ and write a matlab program to:

(2) Make a plot of μ_j vs. j

(3) Make a contour plot of your computed solution $u_{\text{approx}}(\mathbf{x})$.

(4) Make a contour plot of $u_{\text{approx}} - u_{\text{exact}}$.

The code snippet in the file hw6.m might help with the visualization. In this code, I construct a mesh on which to evaluate the solution, triangulate it, and make a contour plot. Before this code executes, I have already computed the vectors $\mathrm{xx}(k) = 2\cos(t_k)$ and $\mathrm{yy}(k) = \sin(t_k)$ for $k = 1..2n+1$, as well as the vector $\mathrm{mu}(j)$, $j = 1..n$. The routine "eval1(x,y,xx,yy,mu)" computes the approximation of the integral (2) at (x, y) using the segments stored in xx, yy and the moments mu. (Note that in this code I am using y to represent the second component of (x, y) rather than as the integration variable (which you can write as (ξ, η) if you find you need to – but you probably won't need to)).