1 Problem Statement

In this assignment we solve a *forward problem* from electrophysiology: given an electric potential on the surface of the heart, compute the electric potential on the surface of the torso. For simplicity we model the heart and torso as an annulus (Figure 1), and assume the conductivity of the body, $\sigma$, to be constant.

From the quasi-static Maxwell’s equations we have:

$$\nabla^2 \phi(\vec{x}) = -\frac{\rho_f}{\epsilon} = f(\vec{x}), \quad \vec{x} \in \Omega$$

(1)

where $\rho_f$ = free charge density and $\epsilon$ = permittivity.

This is a second order PDE, so to solve for $\phi(\vec{x})$ requires two boundary conditions. The first condition is the known distribution on the heart. For the second we require the flux of $\phi$ across the torso to be zero:

$$\phi(\vec{x}) = g_I(\vec{x}), \quad \vec{x} \in \Gamma_I$$

(2)

$$\frac{\partial \phi(\vec{x})}{\partial \vec{n}} = 0, \quad \vec{x} \in \Gamma_O$$

(3)

Physically, the second condition comes from assuming that the medium outside the torso is non-conducting, and therefore no currents cross the surface.

(assuming $J \cdot \vec{n} = 0$ implies $\frac{1}{\epsilon} \vec{E} \cdot \vec{n} = 0$, and so $\frac{\partial \phi}{\partial \vec{n}} = 0$).

As an aside, the *backward problem* is of practical interest to physicians: given the electric potential on the surface of the torso, determine the potential on the heart (or brain). The backward problem is more challenging, and is the subject of ongoing research.

![Figure 1: Computational Domain](image-url)
2 Description of the Mathematics

Equation (1), together with the boundary conditions (2) and (3) form a Poisson problem. Due to symmetry, this can actually be solved analytically on the annulus using the method of eigenfunction expansions.

Our focus here, however, will be the derivation of a second-order finite-difference (FD) approximation of Equation (1). Switching to polar coordinates, we write the Laplacian as:

$$
\nabla^2 \phi = \frac{\partial^2 \phi}{\partial r^2} + \frac{1}{r} \frac{\partial \phi}{\partial r} + \frac{1}{r^2} \frac{\partial^2 \phi}{\partial \theta^2}
$$

and applying second-order expansions term-by-term (as in class and in the text), yields the approximation:

$$
\nabla^2 \phi_{ij} = \frac{\phi_{i+1,j} - 2\phi_{i,j} + \phi_{i-1,j}}{\Delta r^2} + \frac{\phi_{i,j+1} - \phi_{i,j-1}}{2r_i \Delta r} + \frac{\phi_{i+1,j} - 2\phi_{i,j} + \phi_{i-1,j}}{r_i^2 \Delta \theta^2} + O(\Delta r^2 + \Delta \theta^2) \tag{4}
$$

which gives $\nabla^2 \phi$ as a linear combination of its neighbors, forming a five-point stencil. The coefficients of the stencil are $r$-dependent because of the cartesian-to-polar mapping.

There are many ways to form a $\phi_{ij}$ grid in polar coordinates, on the annular domain. We chose the layout below because then the solution on $\Gamma_O$ is easy to parse later – it’s just $\phi_{N1}, ..., \phi_{NN}$. Applying Equation (4) at each interior point yields an $N^2 \times N^2$ sparse linear system, $M \phi = F$, shown schematically in Figure (2).

The Dirichlet condition is needed for the first $N$ rows of $M$: the $L$ coefficient is subtracted from $F$. For the Neumann condition, we use a one-sided approximation of $\partial \phi / \partial n$:

$$
\frac{\partial \phi_{ij}}{\partial n} = 0 = \frac{3\phi_{ij} - 4\phi_{i-1,j} + \phi_{i+1,j}}{2\Delta r} + O(\Delta r^2)
$$

which appears in the last $N$ rows of $M$. Finally, the periodicity in $\theta$ causes a permutation of the $T$ and $B$ coefficients.

![Figure 2: Grid Layout and Linear System for N=3, where (L, T, R, B, C) denotes left, top, right bottom and center FD coefficients](image)

3 Description of the Algorithms

I implemented routines in Matlab to build the linear system (Appendix B), storing all matrices in Matlab’s sparse format (the full matrix for $N = 160$, it would be 5.24 GB in size!). Modular routines are used to build each boundary condition. Matlab contains highly optimized linear solvers. The direct solver (sparse LU decomposition and gaussian elimination) proved faster than `gmres` for this system.
4 Demonstration of Correctness of Implementation

To demonstrate that our implementation is correct, we choose a test potential, $\phi_0$ and compute $f_0$ and $g_{10}$:

\[
\begin{align*}
\phi_0 &= \cos(\pi r) \sin(\theta) \\
f_0 &= -\pi^2 \cos(\pi r) \sin(\theta) - \frac{1}{r} (\pi \sin(\pi r) \sin(\theta)) - \frac{1}{r^2} (\cos(\pi r) \sin(\theta)) \\
g_{10} &= -\sin(\theta)
\end{align*}
\]

We feed $f_0$ and $g_{10}$ into our solver and compare the result, $\tilde{\phi}$ with the known solution $\phi$ at each grid point, using the $\infty$-norm:

\[
e = \sum_{i,j} (\tilde{\phi}_{ij} - \phi(r_i, \theta_j))
\]

The results for the direct solver are plotted in Figure 3, for $N=10, 20, 40, ..., 200$. The line has a slope of approximately $m = -2$ for $N > 60$, proving $O(h^2)$ convergence. It is interesting that the slope for $N < 60$ shows some variation.

5 Results/Analysis of Results

Now for the meat of the assignment. We are given three right-hand-side functions:

\[
\begin{align*}
f_1(r, \theta) &= 2.5 - r \\
f_2(r, \theta) &= (10(r - 1)(2 - r) + 1) \sin(\theta) \\
f_3(r, \theta) &= 1 \text{ on } [1, 1.3], 0.5 \text{ on } [1.3, 1.7] \text{ and } 0.25 \text{ on } [1.7, 2.0]
\end{align*}
\]

and five input functions:
\[ g_1(\theta) = \sin(\theta) \]
\[ g_{2,3,4}(\theta) = e^{-\alpha(\theta-\pi)^2} \text{ with } \alpha = 1, 10, 100 \]
\[ g_5(\theta) = 1.0 \quad \theta \in [0, 0.2] \cup [1.0, 1.2] \cup [2.0, 2.2] \text{ and zero elsewhere} \]

to be solved for \( N = 10, 20, 40, 80, 160 \).

We choose to analyze these by putting \( f \) in the “outer loop”, i.e. considering all \( g \) for a given \( f \). Our main goal going into this is to determine if it is possible to reverse-engineer \( g_O \) by examining \( g_I \). For example, if \( f = 0 \) uniformly, then we would have \( g_I = g_O \), which can be verified by plugging \( \phi(\theta) = g_I(\theta) \) into Equation (1).

5.1 \[ f(r, \theta) = f_1 = 2.5 - r \]

Figure 4: Inputs \( f_1, g_I \) and solutions \( g_O \) and \( \tilde{\phi} \) for \( g_I = (a)g_1 \) (b)\( g_2 \) (c)\( g_3 \) (d)\( g_4 \) (e)\( g_5 \)
Clearly, the output is greatly influenced by the input. This $f$ is radially symmetric and positive, so the amplitude of $\phi$ must decrease with $r$. It appears there might be a simple linear relationship between $g_O$ and $g_I$, especially for the first input function. Sure enough, regression analysis yields:

$$g_O = 0.8g_I + 0.5$$

But this does not hold for localized input, especially the pulsed function, $g = g_5$. Figure 5 plots $g_O$ vs. $g_I$ for each input function.

The pulse function shows relatively poor convergence. That’s because in deriving the finite-differences via Taylor series, it was assumed that $\phi$ was differentiable in $\Omega$. Forcing $\phi$ to meet discontinuous Dirichlet B.C. breaks that assumption.

I thought that linearity might hold for $g_I =$ (any linear combination of sines and cosines). It turns out that this is (experimentally) true only for the case $g_I = a\cos(k\theta) + b\sin(k\theta)$. The attenuation factor seems to be frequency-dependent: higher frequencies are smoothed more, as in a lowpass filter. Figure (6) shows the output for an input containing two different frequencies (Note the change in scales on the y-axes).
5.2 \[ f(r, \theta) = f_2 = (10(r - 1)(2 - r) + 1) \sin(\theta) \]

The situation for \( f = f_2 \) is quite different. \( f_2 \) is \( \theta \)-dependent, and takes on both positive (\( \phi \) “concave down”) and negative (\( \phi \) “concave up”) values. The \( r \)-component of \( f \) is a parabola that has max value 2.5 at \( r = 1.5 \), which is large compared to the inputs. As a result, the effects of \( f \) overpower the inputs and the outputs all looks very similar. The results converge very, very quickly compared to the first \( f \) considered above.
5.3 \( f(r, \theta) = f_3 = 1 \) on \([1, 1.3]\), 0.5 on \([1.3, 1.7]\) and 0.25 on \([1.7, 2.0]\)

Finally, \( f = f_3 \) is sort of a discrete version of the first rhs, \( f_1 \), and the results are very similar qualitatively. Again there is linearity for the sinusoidal input function (Figure 9). One thing to note here is that discontinuities in \( f \) are handled ok by the integrator, unlike discontinuities in \( g_I \).
Figure 9: $g_O$ vs. $g_I = g_1$ for $f = f_3$

6 Summary and Conclusions

We have derived and implemented a second-order FD scheme for the Poisson problem on an annulus, and proved convergence properties. In analyzing the behavior of the Poisson solution for the inputs and rhs functions considered, a few key ideas emerge:

- Continuity in the boundary conditions is required for good convergence.
- Large values (and perhaps $\theta$-dependence?) in the right-hand-side function overwhelm the input, and also speed up convergence.
- For the radially symmetric right-hand-sides considered, the effects of the input were clearly visible in the output.
- For $g_I = a \sin(k\theta) + b \cos(k\theta)$ and radially symmetric $f$, there is a linear relationship between $g_I$ and $g_O$.