# AME 535a Introduction to Computational Fluid Dynamics
## University of Southern California – Fall 2018

---

**Project 3: Iterative Solution Methods**
Inverse Problem

*Handed Out: 25 Oct. 2018* *Due: 08 Nov. 2018*

---

## Problem Statement

It is of interest to researching neuroscientists to determine which regions of the brain are active during a given neurological response. In order to non-invasively determine active parts of the brain, an electroencephalogram (EEG) is used, in which scalp-mounted electrodes measure the electrical pulses created by active regions of the brain. In order to determine which regions of the brain are active (sources of electrical activity), an **inverse problem** must be solved.

In another example of an inverse problem, a researcher has an array of reaction chambers. Due to the experimental constraints, only the temperature and heat flux at the boundaries of the test chamber array can be measured. The boundary measurements of temperature and heat flux can be used, and an inverse problem may be solved, to determine the heat source locations which indicate the array positions in which reactions have occurred.

In this project, we use a rudimentary representation and approach to 'solving' the inverse problem. Rather than directly solving the inverse problem (which in itself is an interesting problem), we consider the forward problem: we guess some positions for the sources, calculate the corresponding boundary flux, and compare this computed boundary flux with the given boundary flux for which we are trying to determine source positions. By trial and error, or by some more advanced methods, we will eventually be able to determine the unknown source configurations by matching the given and computed boundary flux values.

The governing equation for the problem is Poisson's equation,

$$-\nabla^2\phi(x,y) = f \ , \tag{1}$$

with boundary condition $\phi(x,y) = 0$ on the entire boundary, and the square domain shown in Figure 1. Given that we will be solving the equation many times in the trial-and-error approach, we will exercise iterative solution techniques to speed up the process.

The domain is divided into 36 block regions, arranged in a $6 \times 6$ array of blocks. The 16 blocks not on the domain boundary are labeled interior blocks. All blocks are set to have an $f = 0$, with the exception of four of the interior blocks which have $f = 1$ and are referred to as source blocks. The challenge in this particular problem is to determine the position of the 4 source blocks given the distribution of the normal flux $\frac{\partial\phi}{\partial n} = g(x,y)$ on the boundary of the domain.

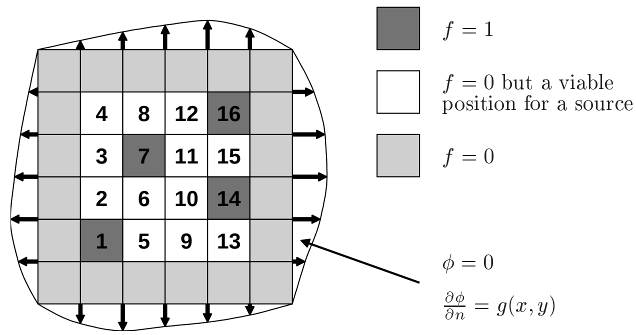


Figure 1: Pictorial explanation of the problem statement and $6 \times 6$ blocks domain. Here the source blocks are 1, 7, 14, and 16.

## Questions

In order to solve the Poisson equation, we overlay a computational grid on the domain. For this problem, a $25 \times 25$ node computational grid is used unless otherwise specified. This would mean that a source block would correspond to a $5 \times 5$ set of nodes being set to $f = 1$.

**1.** (16 pts) Write down Jacobi and Gauss-Seidel iteration schemes to solve Poisson's equation (1).

**2.** (32 pts) Code the Jacobi and Gauss-Seidel schemes. Allow variable specification of the "fineness" of the computational grid (*e.g.*, a computational grid of $7 \times 7$ nodes, $13 \times 13$ nodes, $25 \times 25$ nodes, *etc.*), while also allowing specification of the source blocks in the RHS vector.

For simplicity, assume that all nodes of source blocks, including the block boundaries, have $f = 1$. For two adjacent source blocks, $f = 1$ for the points on the common boundary also.

**a.** Implement a relaxation scheme for both Jacobi and Gauss-Seidel methods. Describe the equation that you use, if you haven't done so already.

**b.** Show the convergence of your Jacobi and Gauss-Seidel solvers for the configuration of source blocks (1,7,14,16), with several relaxation factors. Find the optimal relaxation factor for the Gauss-Seidel method to within $\pm 0.1$.

**c.** A sample solution for this source configuration of (1,7,14,16) is provided in Table 1(a). Verify that your solver produces a similar output. Show either your matrix results of the normal fluxes or an image of your results.

*Hint: For later problems it will be convenient to construct a function that takes as input 4 numbers corresponding to the test blocks (B1, B2, B3, B4* $\in$ *{1,...,16}), and returns the normal derivative around the perimeter as an output.*

**3.** (32 pts) Implement a multigrid routine that will perform a two-grid method. Explain the restriction and prolongation method you choose to implement. Try relaxation factors of $1/2$ and $4/5$. Remember how you are expected to plot your error values from the previous projects.

**a.** Plot and compare the convergence of the multigrid routine using either Gauss-Seidel, Jacobi, or both, with the various other methods. Which method is best? Which relaxation factor for multigrid is better, $1/2$ or $4/5$?

**b.** Implement multigrid routines where $\nu_1 = \nu_2 = 1$ and the number of iterations on the coarse grid is either $\nu_c = 2, 4$, or exact. How do the routines compare? What can we conclude from this?

4. (20 pts) Using the best method (based on convergence rate) and the normal flux distribution $\frac{\partial\phi}{\partial n} = g(x, y)$ given in Table 1(b), determine the unknown positions of the four source blocks. You do not need to find an elaborate/efficient way to do it; you can just try all the possible combinations. Pictorially show where the sources are located.

5. **BONUS**: (Possible +10 pts[1]) Write a generalized V-cycle multigrid routine which allows multiple grid refinements. Show convergence for 2, 3, and 4 grid refinement V-cycles. How do they compare to the other iterative routines?

## Numerical Values of Normal Flux Distribution

**(a) (1,7,14,16) Configuration**

| Point | Boundary | | | |
|---|---|---|---|---|
| | Left | Right | Bottom | Top |
| **1** | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| **2** | 0.0122 | 0.0072 | 0.0122 | 0.0054 |
| **3** | 0.0244 | 0.0145 | 0.0244 | 0.0109 |
| **4** | 0.0361 | 0.0222 | 0.0360 | 0.0163 |
| **5** | 0.0466 | 0.0303 | 0.0463 | 0.0217 |
| **6** | 0.0552 | 0.0391 | 0.0548 | 0.0270 |
| **7** | 0.0612 | 0.0483 | 0.0605 | 0.0323 |
| **8** | 0.0643 | 0.0576 | 0.0633 | 0.0373 |
| **9** | 0.0649 | 0.0662 | 0.0635 | 0.0420 |
| **10** | 0.0636 | 0.0734 | 0.0618 | 0.0464 |
| **11** | 0.0611 | 0.0784 | 0.0590 | 0.0504 |
| **12** | 0.0583 | 0.0811 | 0.0561 | 0.0542 |
| **13** | 0.0553 | 0.0819 | 0.0533 | 0.0578 |
| **14** | 0.0522 | 0.0814 | 0.0507 | 0.0612 |
| **15** | 0.0489 | 0.0803 | 0.0483 | 0.0645 |
| **16** | 0.0452 | 0.0790 | 0.0459 | 0.0673 |
| **17** | 0.0411 | 0.0771 | 0.0431 | 0.0689 |
| **18** | 0.0367 | 0.0740 | 0.0399 | 0.0684 |
| **19** | 0.0319 | 0.0687 | 0.0360 | 0.0652 |
| **20** | 0.0268 | 0.0610 | 0.0315 | 0.0589 |
| **21** | 0.0216 | 0.0509 | 0.0262 | 0.0498 |
| **22** | 0.0162 | 0.0392 | 0.0202 | 0.0387 |
| **23** | 0.0108 | 0.0264 | 0.0138 | 0.0262 |
| **24** | 0.0054 | 0.0132 | 0.0070 | 0.0132 |
| **25** | 0.0000 | 0.0000 | 0.0000 | 0.0000 |

**(b) Unknown Configuration**

| Point | Boundary | | | |
|---|---|---|---|---|
| | Left | Right | Bottom | Top |
| **1** | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| **2** | 0.0125 | 0.0061 | 0.0124 | 0.0135 |
| **3** | 0.0250 | 0.0123 | 0.0247 | 0.0268 |
| **4** | 0.0371 | 0.0188 | 0.0363 | 0.0393 |
| **5** | 0.0483 | 0.0257 | 0.0466 | 0.0502 |
| **6** | 0.0580 | 0.0330 | 0.0547 | 0.0587 |
| **7** | 0.0655 | 0.0406 | 0.0599 | 0.0639 |
| **8** | 0.0709 | 0.0481 | 0.0620 | 0.0657 |
| **9** | 0.0744 | 0.0546 | 0.0614 | 0.0644 |
| **10** | 0.0772 | 0.0593 | 0.0589 | 0.0609 |
| **11** | 0.0798 | 0.0615 | 0.0554 | 0.0560 |
| **12** | 0.0828 | 0.0609 | 0.0516 | 0.0508 |
| **13** | 0.0859 | 0.0577 | 0.0482 | 0.0458 |
| **14** | 0.0885 | 0.0527 | 0.0451 | 0.0411 |
| **15** | 0.0899 | 0.0468 | 0.0424 | 0.0368 |
| **16** | 0.0894 | 0.0407 | 0.0398 | 0.0329 |
| **17** | 0.0868 | 0.0349 | 0.0372 | 0.0292 |
| **18** | 0.0819 | 0.0296 | 0.0342 | 0.0256 |
| **19** | 0.0747 | 0.0246 | 0.0308 | 0.0221 |
| **20** | 0.0652 | 0.0201 | 0.0268 | 0.0186 |
| **21** | 0.0538 | 0.0158 | 0.0223 | 0.0150 |
| **22** | 0.0410 | 0.0117 | 0.0172 | 0.0113 |
| **23** | 0.0274 | 0.0077 | 0.0117 | 0.0076 |
| **24** | 0.0137 | 0.0039 | 0.0059 | 0.0038 |
| **25** | 0.0000 | 0.0000 | 0.0000 | 0.0000 |

Table 1: Tabulated results of the normal flux at the boundaries: (a) (1,7,14,16) configuration, and (b) unknown configuration for which the source locations are to be determined. The point numbers correspond to the grid points in a $25 \times 25$ node grid, for increasing $y$-values (left and right boundaries), and for increasing $x$-values (bottom and top boundaries).

[1] Applied up to a max of 100% on this project; additional bonus points are not carried over to future assignments.