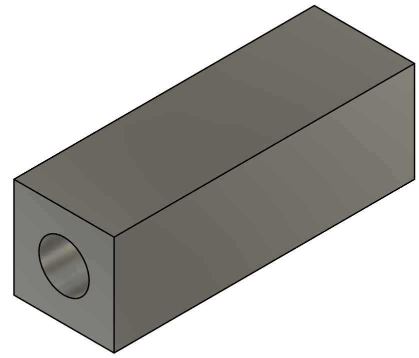
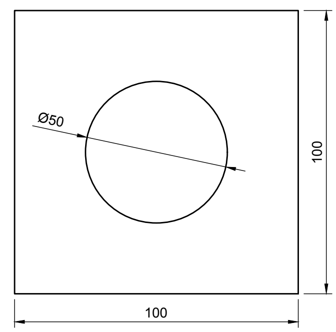
Scientific Computing

Assignment 6: Relaxation Methods for solving Laplace's Equation





The figure at right shows a rather unusually-shaped pipe, with a cross-section that's circular inside and square on the outside. The inner hole is 50 mm in diameter, the outer square is 100 mm on a side, and we assume the pipe is infinitely long. We're interested in calculating the temperature everywhere in the metal wall of the pipe. The square outside surface of the pipe is at room temperature (20°C); the round inside surface carries hot water with a constant temperature of 100°C

The equation for the steady-state temperature in a conducting material is:

$$.\nabla^2 T = 0$$

or in this 2-d situation,

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} = 0 \qquad (1)$$

Subject to the boundary conditions

$T = 20$ when $x = 0$, $x = 100$, $y = 0$, $y = 100$
$T = 100$ when $(x\text{-}50)^2 + (y\text{-}50)^2 < (25)^2$

Solve this problem via Jacobi and Gauss-Seidel relaxation. That is, define temperature on a grid of points:

$$T_{ij} = T(x = i\Delta x, y = j\Delta y)$$

and set each point equal to the average of its neighbors:

$$T_{ij} = \frac{T_{i-1,j} + T_{i+1,j} + T_{i,j-1}, T_{i,j+1}}{4} \qquad (2)$$

(This formula assumes $\Delta x = \Delta y$, but that's a good choice for this problem.) Repeat this process until the error is acceptably small (see below).

- For Jacobi relaxation, when computing new T_{ij}'s, always the use the "old" values for the values on the right hand side that were obtained in the previous relaxation step. It's best to do this by having two separate arrays, one for T_{old}, one for T_{new}.
- For Gauss-Seidel relaxation, use the new values computed in the current relaxation step whenever possible. It's best to store all your T's in a single array, overwriting the old elements as you go along.

Remember to ensure T = 20 for all points on the outer edge of the square domain, and T = 100 for all points within the central circle. It will be tempting to just set *Δx* = 1 mm for this problem, but make sure your code will work for any choice of grid spacing.

Error Tolerance

When should we say our solution is good enough? Define the "residual" as the degree to which equation (1) is not satisfied:

$$r = \frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2}$$

or in finite difference form,

$$r_{i,j} = \frac{T_{i-1,j} + T_{i+1,j} + T_{i,j-1} + T_{i,j+1} - 4T_{i,j}}{\Delta x^2} \qquad (3)$$

Relaxation steps should continue until *all* $r_{i,j}$ are acceptably small: in our case, we want **abs($r_{i,j}$)< 0.002** for all i,j within the empty space between conductors (but not within the central circle!) Calculate the residuals in a *separate i,j loop* from your relaxation method (eq. 2), or you'll just get zero by definition.

Graphics

It's useful to plot the solution as a 3-dimensional surface. To do this, use

surf(x,y,T)

where x and y are a list of x- and y-values. You may also use "surfl", which is similar, but shades the surface as if it were illuminated by a light source.

Timing

We care a lot in this assignment about how much time is required to find a solution. MATLAB includes two functions called **`tic`** and **`toc`**: these start and end a "stopwatch". You use them like this:

```
tic; myexcitingprogram; toc
```

"**`toc`**" reports the elapsed time since the last "**`tic`**". You want to type this out on one line separated by semicolons, because any time you spend typing is included!

Assignment

Find the temperature in the walls of the pipe using both the Jacobi and the Gauss-Seidel method, on a 20x20 grid. Which requires fewer relaxation steps to complete? (If you get the same for both, you've made a mistake: read the descriptions of the two methods carefully.)

Repeat, using Gauss-Seidel only, using a 50x50, 100x100, and 200x200 grid. For each of these, measure the time needed to find a solution.

How does the processing time depend on the number of rows or columns of the grid? Is it proportional to the number of rows? Proportional to the square? Cube? Fourth power? By extrapolation, make a prediction of the time that would be required to solve on a 1000x1000 grid.