

CAAM 520: COMPUTATIONAL SCIENCE II

HOMEWORK 1.

WEI WU

1. INTRODUCTION

In this project, via finite difference method we solved a 2D Laplace equation $-(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}) = \sin(\pi x)\sin(\pi y)$ with Dirichlet boundary conditions $u(x, y) = 0$. We solved the problem on a $(N+2) \times (N+2)$ meshgrid on $[-1, 1]^2$, with $h = 2/(N+1)$. We implemented a solver via Weighted Jacobi method for the resulting linear system. The Weighted Jacobi method is referenced from here: https://en.wikipedia.org/wiki/Jacobi_method.

Note that by assuming zero boundary conditions, the problem is reduced to solving an $N \times N$ system for the interior nodes.

1.1. **Subsection.** This is my first subsection.

2. TESTING AND VERIFICATION

For a small $N = 2$ grid, we could hand compute the solution by solving the following system:

$$(1) \quad \frac{3^2}{2^2} \begin{bmatrix} 4 & -1 & -1 & 0 \\ -1 & 4 & 0 & -1 \\ -1 & 0 & 4 & -1 \\ 0 & -1 & -1 & 4 \end{bmatrix} \begin{bmatrix} u_5 \\ u_6 \\ u_9 \\ u_{10} \end{bmatrix} = \begin{bmatrix} 0.75 \\ -0.75 \\ -0.75 \\ 0.75 \end{bmatrix}$$

This gives us the solution

$$(2) \quad \begin{bmatrix} u_5 \\ u_6 \\ u_9 \\ u_{10} \end{bmatrix} = \begin{bmatrix} 0.0556 \\ -0.0556 \\ -0.0556 \\ 0.0556 \end{bmatrix}$$

This corresponds to the solution produced by our solver. See u_N=2.dat for the produced solution.

3. RESULTS

For a residual $< 1e-6$.

When $N = 10$, the max error is 0.001372. It takes 524 iterations and about 7×524 many operations to converge. Wall time = 0.005255 seconds.

When $N = 20$, the max error is 0.000379. It takes 1818 iterations and about 7×1818 many operations to converge. Wall time = 0.033477 seconds

When $N = 30$, the max error is 0.000175. It takes 3845 iterations and about 7×3845 many operations to converge. Wall time = 0.109254 seconds

4. SCALING

Since I stored all of u, u_{true}, b as static arrays of doubles, it takes about $3 \times (N + 2)^2 \times 8$ bytes of memory.

For $N = 10$, it takes 0.005255 seconds for the solver to converge. Estimated memory usage is 384 bytes.

For $N = 100$, it takes 9.026760 seconds for the solver to converge. Estimated memory usage is about 0.2496 Megabytes.

When I tried $N = 1000$, the program gave me segmentation fault. This is because the memory usage is about 3×8 Megabytes, which is more than C's stack size. The limiting factor is hence the memory.

I then tried $N = 500$. It takes 5009.520565 seconds with 791297 many iterations. The limiting factor is hence run time.

My conclusion is my code can run reasonably well for N equals to a couple hundreds.