Homework 1

CAAM 520, Spring 2019

Posted January 14, 2019. Due Jan 30, 2019 by 5pm.

1. Your solutions to the homework must be committed to your Github repository in a sub-directory HW01.
2. You are required to include a Makefile that creates an excecutable called "hw01" in that directory. All source code, header files, LaTeXfiles, Makefiles, etc. should be committed to your repository in the same sub-directory.
3. You may work in a group of one, two, three, or four students.
4. Use LaTeX to write and typeset your report (saved as "report.pdf" in your repository sub-directory). Document all your steps, including code snippets within your report.
5. If you are working in a team you may discuss the effort with the other member of your team. Otherwise, you may only consult the instructor or graders for verbal assistance. You are encouraged to use textbooks and internet resources. You must cite all resources used via footnotes or a bibliography.

Goal: to solve the matrix system $\boldsymbol{A}\boldsymbol{u} = \boldsymbol{b}$ resulting from an $(N+2) \times (N+2)$ 2D finite difference method for Laplace's equation $-\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) = f(x,y)$ on $[-1,1]^2$. At each point (x_i, y_j), derivatives are approximated by

$$\frac{\partial^2 u}{\partial x^2} \approx \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{h^2}, \qquad \frac{\partial^2 u}{\partial y^2} \approx \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{h^2}$$

where $h = 2/(N+1)$. Assuming zero boundary conditions $u(x,y) = 0$ reduces this to an $N \times N$ system for the interior nodes.

Assignment: Your task is to create a *serial* C/C++ code to compute the length $(N+2)$ solution $\boldsymbol{u}$ to the system $\boldsymbol{A}\boldsymbol{u} = \boldsymbol{b}$. Your code should solve for the unknowns $u_{i,j}$ at each node. Test your code using $f(x,y) = \sin(\pi x)\sin(\pi y)$ and initial guess $\boldsymbol{u} = \boldsymbol{1}$.

If working in a team, each member of the team will need to implement a different iterative solver from the following list to solve the linear system to find the unknowns (please specify which solver you implemented in your write-up). Possible linear solvers to choose from are:

- Weighted Jacobi: `https://en.wikipedia.org/wiki/Jacobi_method`. Use a weight of 1/2.
- Gauss-Seidel: `https://en.wikipedia.org/wiki/Gauss-Seidel_method`

- Successive over-relaxation (SOR) iteration: `https://en.wikipedia.org/wiki/Successive_over-relaxation`. You will have to determine what relaxation parameter to use.
- Conjugate gradient method: `https://en.wikipedia.org/wiki/Conjugate_gradient_method`.

You should converge each linear solver to a user-supplied tolerance of tol. Make the two parameters N and tol user inputs. The remaining details are up to you.

Testing and verification: For a small $N = 2$ (4 interior nodes) grid, compute the solution by hand and make sure that your solver yields the correct result. You may also wish to additionally verify your code's correctness by comparing your results to Matlab.

Results: have your program compute and print out the maximum error for $N = 10, 20, 30$. The exact solution is

$$u(x, y) = \frac{1}{2\pi^2} \sin(\pi x) \sin(\pi y).$$

Compare the number of iterations and number of operations that your chosen methods require to converge to a residual $\|\boldsymbol{b} - \boldsymbol{A}\boldsymbol{u}\| < 10^{-6}$ and document them in your report.

Scaling: Try running your code for $N \times N$ grids with $N = 10, 100, 1000, \ldots$. What size matrix can you reasonably run to? What are the limiting factors (memory, computational runtime, etc)? Provide timing results and estimate memory requirements.

Grading rubric:

- 33 points for documentation and code for applying $\boldsymbol{A}$ to $\boldsymbol{u}$.
- 33 points for verification of code using the test case, operation counts.
- 33 points for documenting the number of iterations, timings, and scaling results.