# Homework 2

CAAM 520, Spring 2019

Posted February 4, 2019. Due February 18, 2019 by 5pm.

1. Your solutions to the homework must be committed to your Github repository in a sub-directory HW02.
2. You are required to include a Makefile that creates an excecutable called "hw02" 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 base your code off of the solution code for Homework 1 or use your own code.
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. You may only consult the instructor for assistance, but are encouraged to use textbooks and internet resources. Cite all external resources used via footnotes or a bibliography.

**Assignment:** Your task is to create a *parallel* C/C++ code to compute the solution $\boldsymbol{u}$ to the system $\boldsymbol{A}\boldsymbol{u} = \boldsymbol{b}$ arising from the finite difference discretization of Poisson's equation. You may use any of the iterative solvers from Homework 1. You can test your code on your own machine, and do not have to use NOTS for this assignment.

Requirements for your code are that you

1. do not store a full copy of the unknowns and/or the system matrix on any single process,
2. implement your algorithm using MPI (the choice of which function calls to use is up to you).

**Documentation:** Your report should also include the following:

1. diagrams describing your partition of the grid,
2. descriptions of your algorithm using pseudocode,
3. descriptions of debugging strategies used.

Test your program by running on $n = 1, 2, \ldots, 8$ processes. Your code should achieve the same answer and maximum error as your serial code for all $n$. You may or may not observe the same number of iterations depending on your algorithm. Explain any differences you observe.

**Bonus (5 points):** Compile and run your code on NOTS. Determine the parallel efficiency of your program for $n = 1, \ldots, 8$ and several problem sizes. Comment on the expected strong and weak scalability of your solver based on these results.