CS 315B Project Proposal

Helmholtz Equation Wave Propagation with Multigrid and AMR

November 1st, 2022 Edmund Chen

The Helmholtz equation is a time-harmonic linear partial differential equation that, with Sommerfeld radiation conditions, can be seen to govern the behavior of wave propagation through waveguides. I will seek to develop a numerical simulation of this problem using a multigrid-accelerated quadratic finite element discretization on an unstructured mesh. This provides many interesting chances to work on accelerated parallel techniques, particularly dealing with the multigrid and adaptive mesh refinement steps of the process. For a starting point, in a *Numerical Solutions for PDEs* course I took at UC Berkeley [2], I implemented the problem described with linear finite elements in a serial fashion; the focus of my project here would be to parallelize this and further sophisticate the problem with multigrid, AMR, and quadratic elements.

Concretely, the equation formulation would be in a 2d waveguide domain with certain slits, for a given wave number k would be governed by the following equation and boundary conditions.

$$\begin{aligned} -\nabla^2 u - k^2 u &= f \quad \text{on } \Omega \\ \mathbf{n} \cdot \nabla u &= 0 \quad \text{on } \partial\Omega_{\text{wall}} \\ \mathbf{n} \cdot \nabla u + iku &= 0 \quad \text{on } \partial\Omega_{\text{out}} \\ \mathbf{n} \cdot \nabla u + iku &= 2ik \quad \text{on } \partial\Omega_{\text{in}} \end{aligned}$$

The mesh would be a well-formulated delaunay-based triangular mesh, with each element's circumcenter inside itself to ensure relatively normal elements. The discretization would use quadratic basis functions in a Galerkin finite element method. On this initial unstructured mesh, multigrid will be used to compute residuals with a coarser mesh. A simple adaptive mesh refinement can then be used to uniformly focus more resolution on certain parts of the solution that we seek more detail on. This can be done in a way that still allows multigrid to progress, for instance, by connecting the midpoints of all three sides of a triangular mesh element. These last two tasks, similar to those mentioned on the course website [1], would present many interesting ways to go about parallelization and load balancing in order to achieve a performant implementation. Towards that end, I will benchmark and document the performance of these techniques at scale.

[1] *CS 315B Parallel Programming*. URL: http://web.stanford.edu/class/cs315b/.

[2] *Numerical Solutions for Partial Differential Equations Spring 2019*. URL: https://math.berkeley.edu/courses/spring-2019-math-228b-001-lec.