CSE 380: Project Proposal

Christopher Cameron

After careful consideration and discussion with my advisor I have decided to program a modified version of the project assignment example from Fall 2013. My research is almost purely experimental in nature apart from a few small codes implementing reduced order modeling for rotorcraft aerodynamics and structural dynamics. Since my research presents no obvious project I feel that the assignment from Fall 2013 will give me an opportunity to apply the widest range of tools presented in the course.

I will write a program to solve the steady state heat equation with (optionally) spatially varying conductivity given in equation 1 using Galerkin finite element approximations. The program will work for one and two dimensional problems with Dirichlet boundary conditions. The program will automatically generate uniform meshes for rectangular domains using triangular elements with sizing controlled by config file arguments. If time permits the option to import a non-uniform mesh from file may be implemented. Lagrangian basis functions will be used with first and second order shape functions for both one and two dimensional problems (possibly higher order elements in one dimension). Integration will be performed on master elements using gaussian quadrature. Options will be provided to iteratively solve the resulting linear system using either Jacobi or Gauss-Seidel methods.

$$-k(x,y)\nabla^2 T(x,y) = q(x,y) \tag{1}$$

The code will be written in C++ and use the header only unit testing framework library, catch. A makefile based build system will be used. The MASA library will be used to implement a verification mode which computes the l_2 error norm of a manufactured solution from MASA. The MASA library will also be necessary to evaluate the forcing function for the manufactured solution. The program will run with an input file which accepts arguments as outlined in the Fall 2013 document:

1. Choice of 1D or 2D analysis
2. Mesh sizing (domain dimensions, number of elements in x and y)
3. Shape function order
4. Solver choice
5. Flag for verification mode using MASA manufactured solution

6. Flag for controlling verbosity of standard output during program execution

If time permits and the option to import a non-uniform mesh is implemented then there will be an additional argument pointing to a mesh input file. Final documentation will include convergence and performance timing measurements for the available combinations of elements, shape functions, and solvers. Additional profiling will identify the contributions of various code components to the overall solution time including initialization, matrix assembly and linear system solution.