Computing at Scale
Assignment 2 (group project)

Implementation	Monday, April 14th at 10:00pm
Submit on Github	

In this assignment, you will be working in groups to implement part of a parallel finite element solver for the 2D Poisson equation using Kokkos.

1. Getting started: You should create a GitHub project repository and add each team member and me as a collaborator. Create a README file that outlines your team members roles and responsibilities. All code needs to be added to the repository through a pull request and and must be reviewed by your team members before being merged. You may want to designate a project maintainer who will be responsible for merging the pull requests. Additionally, you should outline your teams's code style guide and installation instructions. It would be beneficial to get your CI/CD pipeline in place ASAP. You must use CMake for your build system and either Catch2 or Google Test for unit testing.

   Note: Every team member must demonstrate usage of Kokkos.

   Groups as determined by random sampling.

   - **Group 1:** Scott Blender, Bibek Shreshtha, Kairvi Lodhiya
   - **Group 2:** Abhiyan Paudel, Zachary Knowlan, Fuad Hasan
   - **Group 3:** Jay Gaiardelli, Ickbum Kim, Mikel Gica

2. In this assignment, we will only perform mesh loading and assembly. It is not much additional work to apply boundary conditions and solve the system, but you will not be required to do that for this assignment.

   (a) Mesh reader: you will be provided the 2D mesh in gmsh file format (`https://gmsh.info/doc/texinfo/gmsh.html#MSH-file-format`) with all triangular or all quadrilateral linear elements, but you will not know which one until you read the file (at runtime).

   (b) Element stiffness matrices: for each element you will need to form the stiffness matrix and load vector given by:

   $$K_e = \int_e \kappa N_{A,x} N_{B,x} d\Omega_e \tag{1}$$

   $$f_e = \int_e N_A f d\Omega_e \tag{2}$$

   where $N_{A,x}$ and $N_{B,x}$ are the shape function derivatives, and κ is the material property. The first order Lagrange shape functions in parametric coordinates are given by:

   $$N_1 = \frac{1}{4}(1-\xi)(1-\eta), \quad N_2 = \frac{1}{4}(1+\xi)(1-\eta), \quad N_3 = \frac{1}{4}(1+\xi)(1+\eta), \quad N_4 = \frac{1}{4}(1-\xi)(1+\eta) \tag{3}$$

   for quadrilateral elements, and

   $$N_1 = 1 - \xi - \eta, \quad N_2 = \xi, \quad N_3 = \eta \tag{4}$$

   where xi and η are the barycentric coordinates. Remember, your integration rules will be given in terms of of the parametric coordinates, so you will need to use the determinant of the Jacobian to transform to physical coordinates.

   (c) Implement a sparse matrix class to store the global stiffness matrix. You should implement this class using the compressed sparse row (CSR), or compressed sparse column (CSC) storage format. Remember, your storage should be on the GPU.

(d) Implement a vector class to store the global load vector.

(e) Implement the matrix vector product for your sparse matrix and vector classes that works on the GPU.

(f) Implement the assembly process. For each element, you will sum the contributions of the element stiffness and load vectors into the global ones. You must have a test that checks that this matrix is correct.

3. Once you have implemented the solver, you will need to characterize the performance.

   (a) Make plots of the runtime vs. number of elements for serial, openmp, and cuda.

   (b) Plot the speedup compared to the serial version as you increase the number of elements. Speedup is defined as $S = T_{serial}/T_{parallel}$.

   (c) Plot the parallel efficiency for the OpenMP backend, which is defined as $E = S/p$, where p is the number of threads.

4. Documentation it is expected that your code is documented. This includes documentation on each function, class, etc. As well as a README with a description of what you implemented, and how to build/run it.

5. Your implementation should include automated unit tests and integration tests that are run on GitHub Actions for every pull request.

6. Code review needs to be performed as you and your team mates are working on the project.