FINITE ELEMENT METHOD IN SCIENTIFIC COMPUTING: A SHORT COURSE

1.0 Overview

The finite element method is a conceptual framework for the numerical approximation of solutions of partial differential equations (PDE). It is based on the reformulation of the PDE in weak form, and then uses a piecewise polynomial approximation of the exact solution. To this end, it subdivides the domain on which the PDE is posed (i.e., in which you want to describe the physical process) into a "mesh" of "finite elements" – i.e., "cells" of finite size (as opposed to infinitesimally small). On each of these cells, one then approximates the solution by a polynomial. The result is that we can describe the solution by the finitely many expansion coefficients of these polynomials. Because finitely many unknowns can be stored and solved for in finite time, this method is now amenable to computation. In this short course, the students will be introduced to some of the ideas of how this conversion from an unsolvable, infinite dimensional problem (the PDE) to a solvable, finite dimensional problem (the finite element approximation) works for some typical equations, and how all of this is represented on a computer. The focus of this course will be on the practical side of the finite element method and it will emphasize the modern algorithms and approaches for its implementation using deal.II as an example.

An important part of this course is to show students how finite element methods look like in implementation practice today. For this, the deal.II library will be used (see http://www.dealii.org/). deal.II is a C++ software package that provides building blocks from which one can assemble finite element solvers in much the same way as one writes MATLAB programs: it offers a wide variety of data structures and algorithms for everything one typically needs in finite element codes, but how these are put together is left to the user – just like MATLAB offers matrix and vector data structures along with things like singular value decompositions, Fourier transforms, etc., but it is left to the user what to do with all of this.

deal.II is a project that is today also the largest and most widely used open source finite element software. It has grown to more than one million lines of C++, has thousands of users, and a developer community that contributes hundreds of patches every month. It also has excellent documentation, and the course will show students how to navigate it and in particular walk them through some of the many tutorial programs that explain how deal.II can be applied to particular partial differential equations in any domain of interest.

The short course will also have plenty of time for the students to play with deal.II: explore what happens if one changes the equation, the boundary values, the domain, the right hand side; how best to visualize the solution.

2.0 Objectives

The primary objectives of the course are as follows:

i) Train students on writing modern finite element codes,
ii) Train students on using several modern software tools that will help them with writing finite element codes,
iii) Describe the approach to building a finite element approximation for whatever problem one is interested in,
iv) Demonstrate deal.II as a general purpose finite element library to solve partial differential equations.

3.0 Course details

Dates: April 29 – May 03, 2024 (5 days): 12 lecture hours and 9 tutorial hours
Lecture schedule

Monday, April 29, 2024

Lecture 1: 1 hrs:
- Course overview; why we use software libraries
- An introduction to the finite element method, part 1

Lecture 2: 1 hrs:
- An introduction to the finite element method, part 2

Lecture 3: 1 hrs:
- A brief introduction to deal.II

Tutorial 1: 1 hrs:
- Lab activity: Working on the command line;
- Lab activity: Getting started with installing deal.II

Lecture 4: 1 hrs:
- Generating meshes in deal.II

Tuesday, April 30, 2024

Lecture 5: 1 hrs:
- Allocating degrees of freedom in deal.II

Tutorial 2:1 hrs:
- Executing step-2 tutorial program in deal.II and visualize the DoF locations and sparsity patterns for a given mesh

Lecture 6: 1 hrs:
- Solving Laplace's equation using the Finite Element Method

Tutorial 3: 1 hrs:
- Executing step-3 tutorial program in deal.II to solve Laplace's equation and visualize solutions in Paraview

Wednesday, May 1, 2024

Lecture 7: 1 hrs:
- Using templates in C++
- Applying templates to solve Laplace's equation in deal.II in a dimension independent way

Tutorial 4: 1 hrs:
- Executing the step-4 tutorial program in deal.II to solve Laplace's equation in both two-dimensional and three-dimensional domains

Lecture 8: 1 hrs:
- Generating adaptively refined meshes, treatment of hanging nodes in the finite element method, and constraint enforcement

Tutorial 5: 1 hrs:
- Executing the step-6 tutorial program in deal.II to solve Laplace's equation using adaptively refined meshes

Thursday, May 2, 2024

Lecture 9: 1 hrs:
- Solving problems with more than one solution variable (“vector valued problems”) with linear elasticity as an example

Tutorial 6: 1 hrs:
- Executing the step-8 tutorial program in deal.II to solve the linear elasticity equation, set up and run patch tests, and visualize the results

Lecture 10: 1 hrs:
- Solving the mixed Laplace problem
- Using the block structured solvers

Tutorial 7: 1 hrs:
- Executing the step-20 tutorial program in deal.II to learn the use of block structured solvers for the mixed Laplace problem

Friday, May 3, 2024

Lecture 11: 1 hrs:
- Discussing best programming practices
  Using linear solvers and preconditioners

Tutorial 8: 1 hrs:
- Experimenting with different linear solvers and preconditioners and examining their influence on code performance

Lecture 12: 1 hrs:
- Beyond computing: workflows in scientific computing and software development

Tutorial 9: 1 hrs:
- Wrap-up lab activities and final exam

4.0 Who can attend

- Researchers and practicing engineers in civil, mechanical, and aerospace engineering, engineers and researchers from government organizations including R&D laboratories.
- Students at all levels (B. Tech. ($7^{th}/8^{th}$ semester)/M. Tech./MS/PhD) or Faculty from reputed academic institutions and technical institutions.