ASEN-5331/ MCEN-5228 Computational Fluid Dynamics: Unstructured Grid

August 27, 2019

1 Instructor

Kenneth Jansen: AERO 363, x4359, email jansenke@colorado.edu.
Lecture hours: T/TH 10-11:15 AERO N240
Office hours: Wednesdays 10-11:30 + TBD
Web-page: fluid.colorado.edu/~kjansen/CFD19
Final exam: TBD

2 Course Outline

Course focuses on unstructured grid computational approaches to solve the Navier-Stokes equations which govern fluid flow in most engineering applications. Course assumes a basic knowledge of the solution of partial differential equations with numerical methods and therefore directly attacks the obstacles to applying these methods to the Navier-Stokes equations. Specifically, issues concerning implementation of finite volume methods (FVM) and finite element methods (FEM) will be discussed but only FEM methods will be covered in detail. These issues include: the discrete formulation, non-linear equation iterator(steady)/marcher(time-accurate), linear equation formation, boundary condition prescription and linear equation solution.

3 Prerequisites

Course assumes a basic knowledge of fluid dynamics, calculus, continuum mechanics, and numerical methods used to solve PDE's though all needed tools are introduced in class. Very little out of notes reading is required for students who have had a fundamental of Finite Element course. Students without prerequisite have done well in the course but not without significant additional effort.

4 Reference Books

The following text books are good references for this class (First one is highly recommended, second good for math background, third only if more detail on FVM methods desired).

1. T. J. R. Hughes, **The Finite Element Method**, Dover.
2. Claes Johnson, **Numerical Solution of Partial Differential Equations by the Finite Element Method**, Cambridge Press, 1987.
3. J.D. Anderson, **Computational fluid dynamics : the basics with applications**, McGraw-Hill, 1995.

5 Grade Criteria

1. Homework 25%
2. Spot Note Review Quizzes 10%
3. Midterm 25% (February 20th in class)
4. Final exam or Final Project 40%

Spot note review quizzes will be given at the start of some lectures to test synthesis of concepts covered in prior lectures. Homework will be focused on preparation to do a final project but experience has shown that this route is not for everyone since it requires first learning a fairly complicated CFD code, understanding it well enough to add a significant capability, and finally, coding that capability and demonstrating its correct implementation. Two code options will be provided. One is in FORTRAN 90 with some C and C++. The other is in C and makes heavy use of finite element libraries from libCEED which in turn makes heavy use of PETSc. Knowledge (or strong will to acquire knowledge) of at least one language and significant computer science skills are required (as well as willingness to learn others). Everyone will be required to take the Midterm. As the course progresses students interested in doing the project should consult the instructor on the feasibility of their project and whether they are making rapid enough progress in that area to continue or if they should consider switching to the Final Exam track.

6 Project

In place of the final, students are allowed to do a project. Students are encouraged to steer their project to be supportive of their research. They are further encouraged to start with an existing code rather than write one "from scratch".

We will be browsing the code in class from the following OpenGrok link

http://fluid.colorado.edu/source/xref/phasta-cfdhw

To prevent the inevitable procrastination that undermines such projects 4 deliverables are required for grading:

1. A proposed project where you choose a method (and a code) and a major feature to add to it. This could be:

   (a) a new linear algebra solver,

   (b) new boundary conditions,

   (c) different order difference stencil or basis functions,

   (d) additional physics equations like a scalar transport or turbulence model,

   (e) error estimation and adaptivity,

   (f) a new time integrator,

   (g) a different memory management scheme,

   (h) other

   I am expecting a detailed plan of action as well as a proposed validation problem. (DUE Friday October 11)

2. Detailed description of chosen base code explaining:

   (a) formulation,

   (b) input,

   (c) pre-processing,

   (d) non-linear equation iterator(steady)/marcher(time-accurate),

   (e) linear equation formation,

   (f) boundary condition prescription,

   (g) linear equation solution,

   (h) output,

   (i) electronic submission of chosen base code (only required if using a code other than one provided by me).

This document should contain not only theory but a clear outline of implementation including description of data structures, overall responsibilities of EACH and EVERY subroutine AND further detail for those routines where modifications are expected. A detailed plan of modification should also be given at this time. (DUE before FALL Break)

3. Final report updating previous reports and detailing:
   (a) modifications made to the base code (i.e. student contribution),
   (b) verification studies,
   (c) proposed future work based on these results,
   (d) discussion of requirements to complete same project using alternative methods (i.e. using a method other than the one selected),
   (e) reconsideration of advantages of chosen approach,
   (f) an appendix which gives detailed, step-by-step description of how to run the code including any pre- and post-processing software tools developed,
   (g) electronic submission of all source code (solver and any custom pre- and post-processing software tools).

   (DUE on last day of class)
4. 15 minute presentation of results (in the last 2-3 class periods).

The first half of the course lectures will be devoted the presentation of implementation details of the FEM method (with some discussion of FVM). This will gradually taper off with increasing discussion of project-specific issues as the semester progresses.

Students are encouraged to discuss preliminary drafts of all of the above work with each other AND with me to insure that complete understanding of expectations is achieved (and desired grade is attained). That said, all work should be your own (NO CLONED CODE DESCRIPTIONS). Group projects are possible but a case must be made that the project is too big for an individual to complete. If this is the case, roughly twice the amount of work/additions are expected making such plans advisable only for groups that are comfortable linking their fate together.

Note carefully that I have asked for electronic submission of the base code and your final project code. In addition to your written report, I will be looking at your actual code. I have coded CFD for many years and have a very good understanding of the work involved in various projects. By this I mean to warn you that I am quite good at discerning how much effort was put into this class and your grade will reflect this. During the last few offerings of the class, students broke cleanly into 3 groups: 1) those that worked hard and ended up completing a manageable project, 2) those that took on a bit too significant of a project but worked hard and made good progress (final project was not a completely working code), and 3) those that put in a low amount of effort and made little progress. The first two groups got A's, the third group's grades included B's, C's and I's (which ended up turning into F's in some cases). Because of past problems, no incomplete grades will be given this time. Students not prepared to put significant effort against a project should either audit the class or go the exam route (these are not the easiest 3 credits). I define significant effort as at least 12 (focused/productive) hours a week. Those taking the class without the prerequisites should expect to spend an additional two to three hours each week backfilling information.

Warnings aside, those students that took the class seriously gained not only knowledge but valuable experience (a few even got papers out of their work). Projects from past include: New linear equations solvers (2, one used PETS, one AZTEC), new boundary conditions (3), new basis functions (2), added turbulence model (5), added fluid structure interaction (2), added computation of acoustic flux for noise calculation (1), added new element topology (pyramid).