SECTION 1: Introduction

Introduction to Computational Fluid Dynamics

Why CFD?

Fluid flow governed by Navier-Stokes equations:

Set of nonlinear, coupled, partial differential equations

Analytical: very difficult to solve; limited number of solutions available

Experimental: can be costly and time consuming; under certain conditions, perhaps not even feasible

Numerical: can be cost effective, particularly as inexpensive workstations have become available; can isolate physics (i.e., potential flow solutions); however, confidence in results can be an issue

Experimental and CFD approaches should be considered complementary rather than competing techniques

Applications of CFD

Aerodynamics – aircraft lift and drag calculations, sailboat sail interactions, automobile drag calculations, etc.

Hydrodynamics – boat and ship drag; pitch and roll motions; keel design; propeller analysis, etc.

Pipes and fittings – loss coefficients

Turbomachinery

Automotive engines

Environmental modelling of rivers, streams, lakes

Weather prediction

Biological fluid dynamics – flow through blood vessels and organs

Wind turbine design, placement, and analysis

Chemical process engineering

Electronic packaging and cooling

Heating and air conditioning applications

Rapid design optimization

In fact, it might be said that most major industrial companies now utilize CFD in the design/optimization process.

Main Steps to a CFD Solution

0) Pre-processing – generate a model from a CAD or solid modelling package. Typical file formats would include .obj or .stl

1) Create a mesh with proper distribution of mesh points or finite volume cells. This is often the most time consuming and important step.

2) Discretize and solve the governing equations with appropriate boundary conditions. This is what we will concentrate on in this course.

3) Post-processing – usually use a commercial or open source package to aid in this process. Typical would be the open source visualization code ParaView. Construct and interpret streamlines, contour and line plots, velocity vector plots, etc. I will give examples of how to use ParaView as we proceed.

Scope of this Course

CFD has come a long way since the 1970’s and is now a mature field.

Consequently large numbers of both general and specialized techniques exist to solve the Navier-Stokes equations or simplifications thereof.

The most common technique used by the largest commercial solvers is the finite volume method on unstructured meshes with a collocated variable arrangement. Also the approach taken by the open source code OpenFoam.

For this course, we will limit our efforts to the finite volume method to solve the 2D, incompressible Navier-Stokes equations over a structured, Cartesian mesh using a staggered grid.

We will start slowly, first concentrating on developing the technique for diffusion only problems.

We will then move on to a pure (linear) convection problem.

We will then go over the SIMPLE procedure which we will use to write our own code the *Navier-Stokes* equations. The difficulty increases significantly with this step, as the process now involves the solution of a set of coupled, nonlinear partial differential equations.

We will use our Navier-Stokes code to solve the "driven cavity" problem, and in addition, the developing flow in a straight channel.

We will then look at the implementation of different boundary conditions.

We will also extend the code to somewhat more complicated geometries by blocking out regions of the flow.

Finally, we will briefly discuss the topics of CFD errors and uncertainty.

I have uploaded three codes that you may download which are written in Fortran95. You can download a free Fortran95 compiler for Windows from the following sites:

1) g95 compiler:
http://www.g95.org/index.shtml
or
https://www.fortran.com/the-fortran-company-homepage/whats-new/g95-windows-download/

2) gfortran compiler:
https://gcc.gnu.org/wiki/GFortranBinaries

3) gfortran compiler (site which I like best for ease of installation including openmp support):
https://sourceforge.net/projects/tdm-gcc/

as you install from tdm-gcc, when you get to the components to install, be sure to open the gcc "tree" to add fortran and openmp