Advanced Intro to CFD final project report

*Alex Perron*

*12/12/22*

Abstract

The purpose of this project is to edit and debug a coded template to create a functioning 2D finite difference CFD code. The code uses both point Jacobi and symmetric Gauss-Seidel schemes to solve the 2D incompressible Navier-Stokes equations. To do this, these schemes make use of both time derivative preconditioning and artificial viscosity. The cases solved in this project are a manufactured solution and a lid driven cavity. Varying sizes of grids, CFL numbers and Re numbers are used.

Theory

1. Governing equations and Discretization

Text, letter

Description automatically generatedThe base governing equations used in this code are the 2D incompressible N-S equations. These equations can be seen below:

These equations already include both the time preconditioning element as well as the artificial viscosity. The source terms are also included for the manufactured solution case.

The above equations are discretized for a simple explicit method using central in space, forward in time. Below is the discretization for the continuity and x momentum equations.

A picture containing diagram

Description automatically generated

*Continuity discretization*

Graphical user interface, application, Word

Description automatically generated

*X-momentum discretization*

1. Boundary Conditions

The boundary conditions used for the Lid driven cavity were wall boundary conditions. For the bottom, left and right walls, both components of velocity (u,v) are set to zero and the pressure is linearly extrapolated from the inner 2 nodes. For example, the pressure on the bottom wall can be found using the following equation:

For the top wall, the pressure and y component of velocity (v) are treated the same as the other 3 walls, however now the x component of velocity is 1m/s. ie:

utop wall = uinf

1. Artificial viscosity

The artificial viscosity shows up in the Continuity equation as “S”. The expression for S is derived as:

A picture containing diagram

Description automatically generated

The C(4) constant in this case is a chosen constant that can very from 1/128 to 1/16. β2 is the time derivate preconditioning term and is found via the following equation:



Lastly, the 4th derivative of pressure is found using a simple finite difference discretization throughout the domain. For the interior nodes the discretization is a 2nd order 4th derivative central difference scheme as follows:

For the nodes one node away from a wall or corner, where i or j = 2 or Nodemax -1, the 4th derivative was found using a forwards or backwards difference (2nd order accurate) scheme. For example, on the left wall, the 4th derivative of pressure w.r.t x is found via the following equation:

The following equation keeps the same signs regardless if it is forwards or backwards difference. Whether using forwards or backwards difference depends on which side the pressure derivate was calculated on.