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.

1. Time step

The time step for each iteration is determined by the following equation:

Text

Description automatically generated

Where |λmax| is the max between the following:

A picture containing text, antenna

Description automatically generated

Δtc is found at each node, and the minimum value of Δtc is used to determine the next time step. This process of using the minimum value of tc ­across all nodes is called global time stepping.

Results

1. Comparison of Numerical Schemes used

The two numerical schemes used in this project are the point Jacobi and symmetric Gauss-Seidel. Both methods are explicit and used the discretization described in the prior section. The difference between these methods lie in the Gauss-Seidel having a forward and backward sweep where it updates its values twice in one loop, where as the Point Jacobi only uses one sweep, and updates its values from the prior iteration.

Below is a plot comparing the iterative convergence of the two methods.

1. The effect of the C(4) constant on discretization error

The C(4) constant can be seen used in the equation for artificial viscosity (reference that section for said equation). Changing the C(4) constant will effect your artificial viscosity and in tern your discretization error. Below is a plot of discretization error at y close to the top of the chamber and x from 0 – 0.01. 

As can be seen from the above plot, as you decrease the C(4) constant from 0.02 to 0.001 the overall discretization error decreases. There is large fluctuation in the DE at the higher values and odd-even decoupling can be seen. The sweet spot for the C(4) constant appears to be around 0.001. Once the C(4) constant is decreased too far, the odd even decoupling comes back.

1. The effect of the rkappa (κ) constant

The rkappa constant is a constant that shows up in the β2 time derivative preconditioning term. This constant can be seen in the equation given in the artificial viscosity section. Like the C(4) study, this too will be performed on the MMS solution with a Reynolds number of 100 on a 65x65 grid. rkappa can range from 0.001 to 0.9.

Below are two plots of the pressure, the left figure having an rkappa of 0.9 and the right figure having a rkappa of 0.001



The only difference that is apparent between these two cases, is that the run with the higher rkappa of 0.9 (left), was completed in slightly less iterations than that of the higher rkappa of 0.001 (right).

1. Comparison of the Symmetric Gauss Seidel to the manufactured solution case

Below are the results for a 65x65 SGS run for manufactured solution

**

**

As can be seen from above, the manufactured solutions match very closely with the manufactured solutions given in the project statement. More importantly the iterative residuals converge. Our discretization error norms are low as well (most falling in the magnitude of 10-4).

1. Lid driven cavity results

Below are the results of the LDC test case

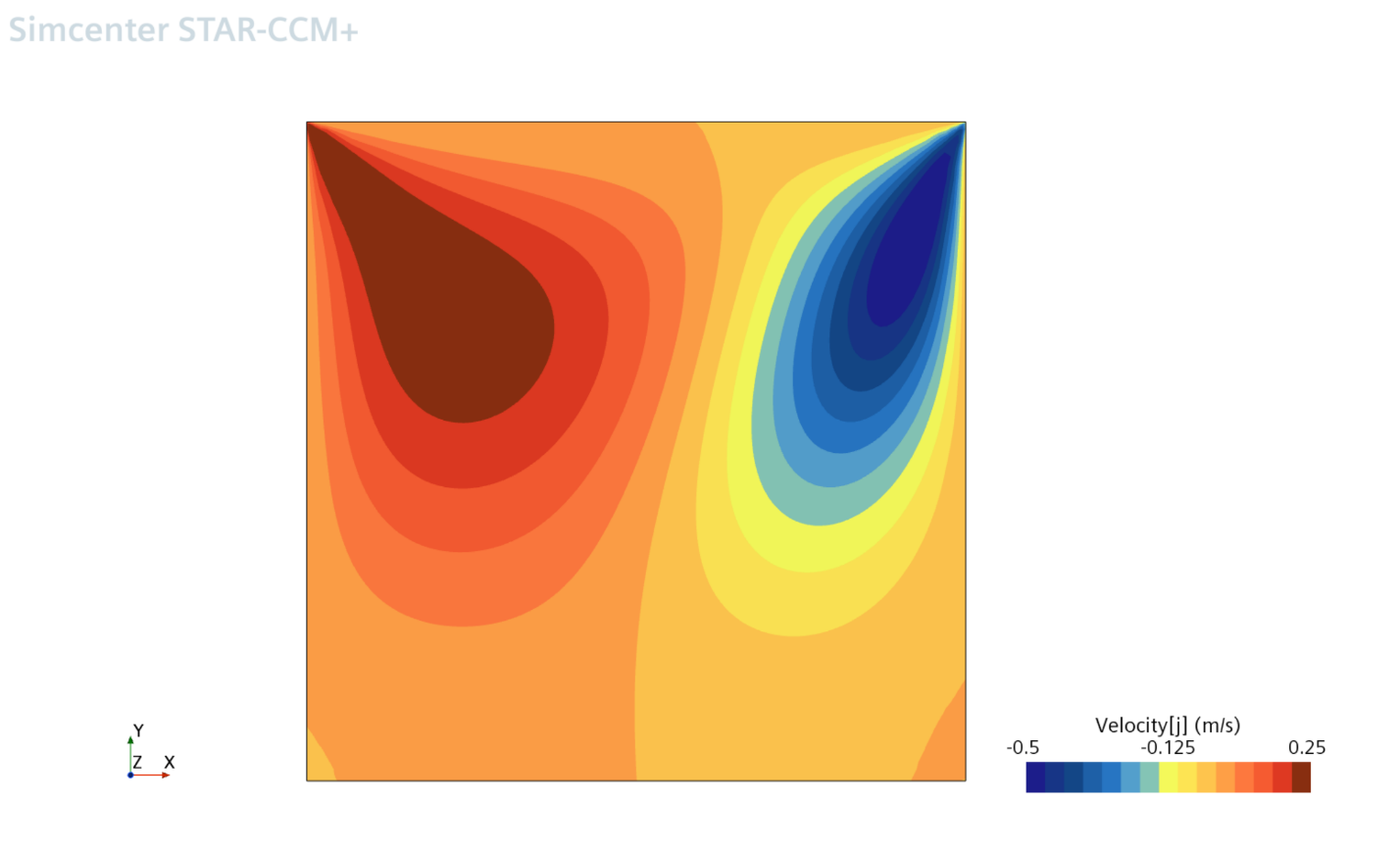
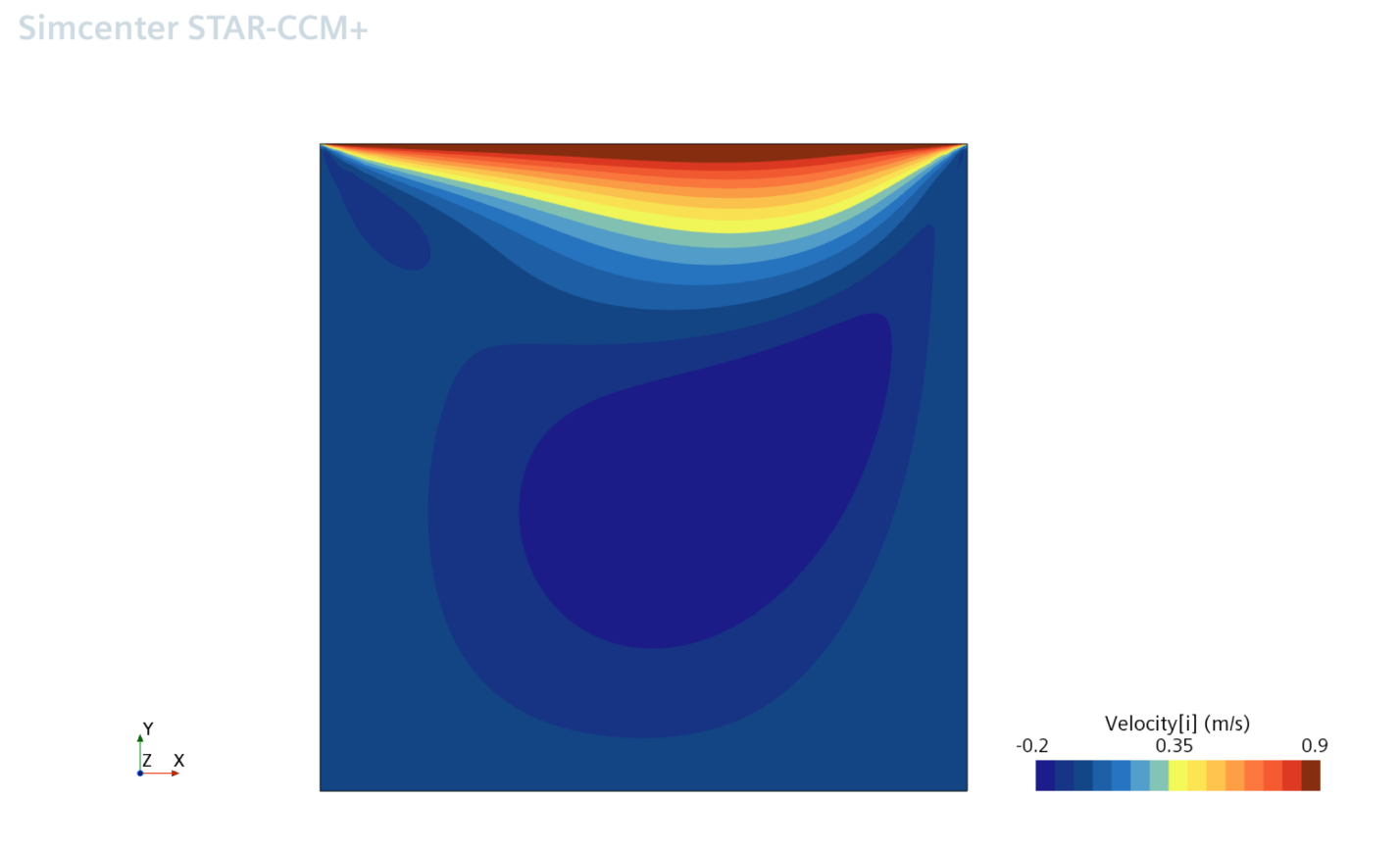
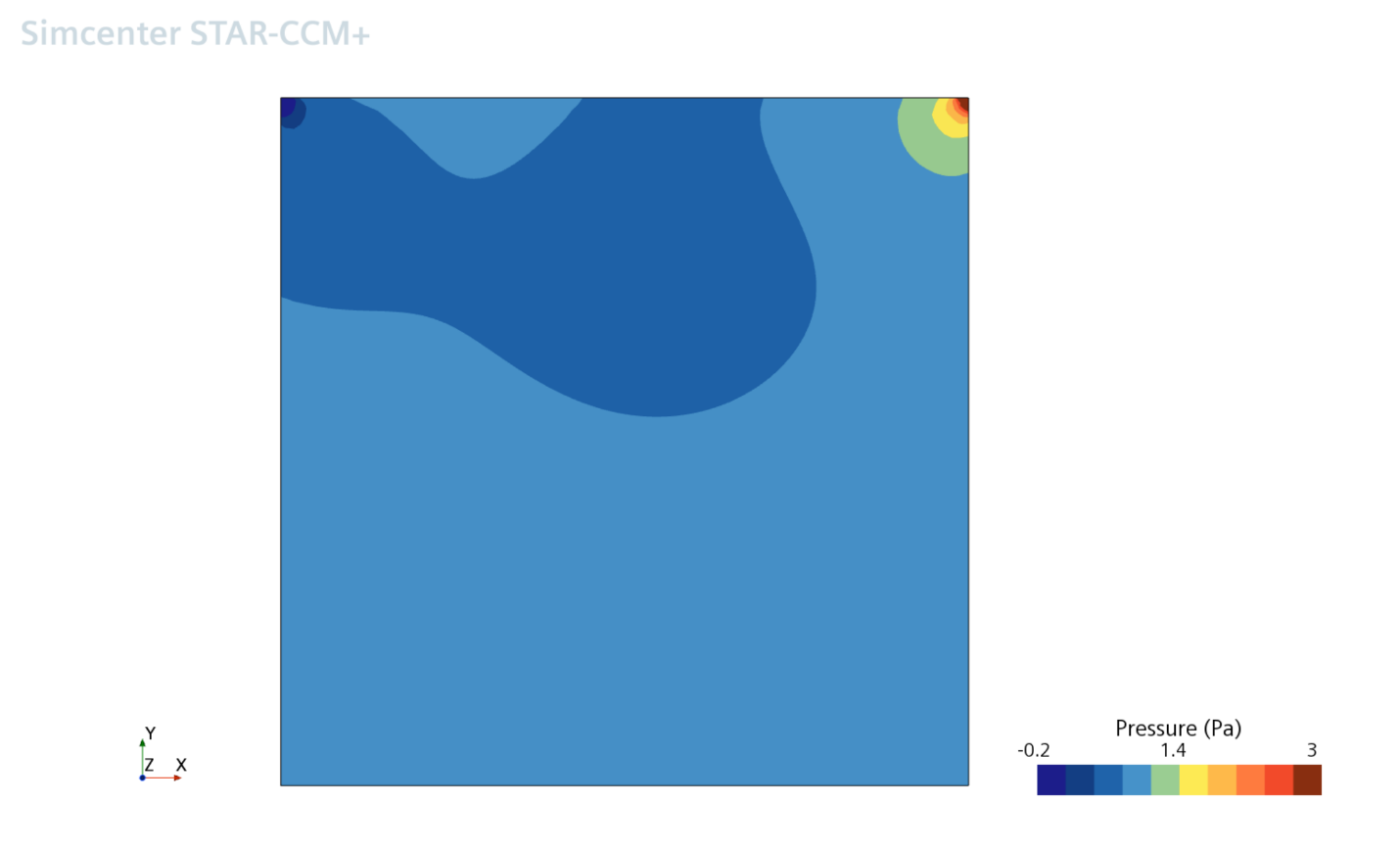








As can be seen from the results above, these compare closely to the results given in the project statement. These also match closely with the below results from StarCCM + run with the same initial and boundary conditions.



1. Higher Reynolds number cases

For these cases the residuals plot and the u velocity plot with stream lines are the only thing shown. However all of the plots can be generated from the raw .mat files in the git-repository. (link provided in apendix). Below are the results from the Higher Reynolds number cases

Re 500:





Re 1000:



Chart, diagram

Description automatically generated

In these higher Reynolds number cases you can start to see the flow separation off of the corners of the cavity. This is to be expected in these cases. For the Re of 1000 case I had to take the results from an un converged solution due to time constraints.

1. Remarks

Unfortunately, due to time limitations and computational resources, both the simulation runs, the code development, and the project write up were cut short and not all of the required runs and results could be show, talked about, or finished. A bug in the artificial viscosity was found only days before the due date. Fixing this bug dramatically changed results I was getting, Discretization errors and plots, and would make any results before this fix invalid. Even with that bug being found, the results that were found even after this bug fix had issues to do with another, unfound bug. Unfortunately, time ran out before this other bug could be fixed. This is the problem with going behind someone else and developing code. Even though it is a template, it might have been quicker to code from scratch. Not to mention only having half a semester to code and debug such a gargantuan piece of code, especially when bugs are still found last minute.

Nonetheless I have presented what I have done thus far, the code will be both pasted at the end of this document and attached as a .m file.