Fluid Dynamics

Jonah Sachs, Eliza Rocks, T Thomas

November 2023

1 Introduction

In this project, we hoped to solve Euler's equations for an ideal fluid in two dimensions through the Finite Volume Method.

2 Implementation

2.1 Finite Volume Method

The most basic idea of the finite volume method is that we have primitive and conserved variables that relate to each other through a certain set of equations. At each time step we then calculate the flux, which is the flow rate of the variable, in the x and y direction of every cell using the primitive variables.

This implementation isn't very stable and so we can use the steps below to improve upon it so that it can be 2nd order in both space and time.

1. Compute Δt using the CFL condition
2. Take a trial step of $0.5\Delta t$ for all primitive variables. Enforce the periodic boundary condition for the ghost cells.
3. Extrapolate primitive variables to left and right cell interfaces in the x and y directions. Enforce the periodic boundary condition to the left face extrapolated variables.
4. Calculate the fluxes of conserved variables using the extrapolated primitive variables. Enforce the periodic boundary condition to the fluxes.
5. Calculate the new values of the conserved variables using their fluxes. Enforce the periodic boundary condition for the conserved variables.
6. Update the primitive variables using the conserved variables. Enforce the periodic boundary condition for the primitive variables.

Our code to perform these steps is stored in fluid_solver_d.h.

2.2 Converting between Conserved and Primitive Values

Throughout the code we use both conserved quantities as well as primitive values. Our primitive values include density: ρ, velocity: v_x and v_y broken into the x and y components respectively, and pressure: P. The conserved quantities include mass, the momentum in the x and y directions, and energy. To convert between the quantities the following equations were used:

$$mass = \rho \tag{1}$$

$$mom_x = v_x\rho \tag{2}$$

$$mom_y = v_y\rho \tag{3}$$

$$U = \rho(v_x^2 + v_y^2) + \frac{P}{(\gamma - 1)\rho} \tag{4}$$

To covert from the conserved to the primitive values (1-4) were solved for their respective conserved values.

2.3 Finding Δt

Δt is determined by a CFL condition, which helps keep the model stable. The CFL condition can be realized as:

$$v_{max}\Delta t \leq Cmin(\Delta x, \Delta y) \tag{5}$$

In this case solving the condition leads to Δt being equal to some factor C multiplied by the minimum value between the change in x and the change in y and then divided by the maximum signal speed, v_{max}. The signal speed is defined as

$$v = \sqrt{\gamma\frac{P}{\rho}} + \sqrt{v_x^2 + v_y^2} \tag{6}$$

Unless otherwise specified we use a CFL condition of 0.4.

2.4 Primitive Variable trial step

Using the calculated Δt , we then update our primitive variables by taking a half-step forward using the convective form of Euler's equations for an ideal fluid in 2D. For example the convection form of Euler's equation for the primitive variable P looks like

$$\frac{\partial P}{\partial t} + v_x\frac{\partial P}{\partial x} + v_y\frac{\partial P}{\partial y} = -\gamma P(\frac{\partial v_x}{\partial x} + \frac{\partial v_y}{\partial y}) \tag{7}$$

This in turn can be written to define the change in P

$$\partial P = -\partial t(\gamma P(\frac{\partial v_x}{\partial x} + \frac{\partial v_y}{\partial y}) + v_x\frac{\partial P}{\partial x} + v_y\frac{\partial P}{\partial y}) \tag{8}$$

To implement this into code, temporary arrays for the primitive variables are filled with new values that take the variable at each index and add the change. The partial derivatives are defined as the following in the x (8) and y (9) direction where Q is the quantity

$$x : \frac{Q_{i+1,j} - Q_{i-1,j}}{2\Delta x} \tag{9}$$

$$y : \frac{Q_{i,j+1} - Q_{i,j-1}}{2\Delta y} \tag{10}$$

Plugging these estimate values to find the change in the primitive variables as shown in 7, and multiplying by 0.5 allows us to take a half step.

2.5 Computing the Fluxes

Next, we calculate the fluxes which requires an additional step in which we extrapolate our primitive variables to their left and right sides in both the x and y directions. The extrapolation estimates the values on these left and right faces by either subtracting (for the left face) or adding (for the right face) the estimate for the partial derivative of the variable as shown in equations (8-9) and multiplying it by $\frac{dx}{2}$. With these extrapolated values, we now can calculate the fluxes.

Using the extrapolated values we then figure out the maximum signal speed, shown in equation 6, in the x and y directions between the left and right faces for every index location which then are used in the equation to calculate the fluxes.

The conserved quantities are calculated using the same equations (1-4) but with the extrapolated values. The left values are calculated at the current indexes using the Right face extrapolated values while the right values are calculated using the Left face extrapolated values to one index to the right in either the x or y direction depending on the flux being calculated. The extrapolated flux values are calculated using the conserved forms of Euler's equations in 2D. For example the conserved form for Energy, U, is:

$$\frac{\partial U}{\partial t} + \frac{\partial}{\partial x}((U + P)v_x) + \frac{\partial}{\partial y}((U + P)v_y) = 0 \tag{11}$$

For example the flux in the x direction on the right face would equal:

$$Flux^{Rx}_{energy} = ((U^{Lx}_{ij} + P^{Lx}_{i+1j})v^{Lx}_{x,i+1j}) \tag{12}$$

We subsequently did this for all conserved quantities on both of their left and right faces in the x and y directions.

The final flux values for each conserved quantity can be calculated using the following equation:

$$F = \frac{1}{2}(F_L + F_R) - \frac{v_{max}}{2}(Q_R - Q_L) \tag{13}$$

We then update the fluxes on the periodic boundary.

2.6 Updating Conserved Variables

Finally, we update the conserved quantities using the following where Q is the quantity:

$$Q_{ij}^{n+1} = Q_{ij}^{n} - (F_{Q,ij}^{x} - F_{Q,i-1j}^{x})\frac{\Delta t}{\Delta x} - (F_{Q,ij}^{y} - F_{Q,ij-1}^{y})\frac{\Delta t}{\Delta y} \tag{14}$$

To finish we convert the conserved variables to their primitive forms and output the density values to graph.

3 Results

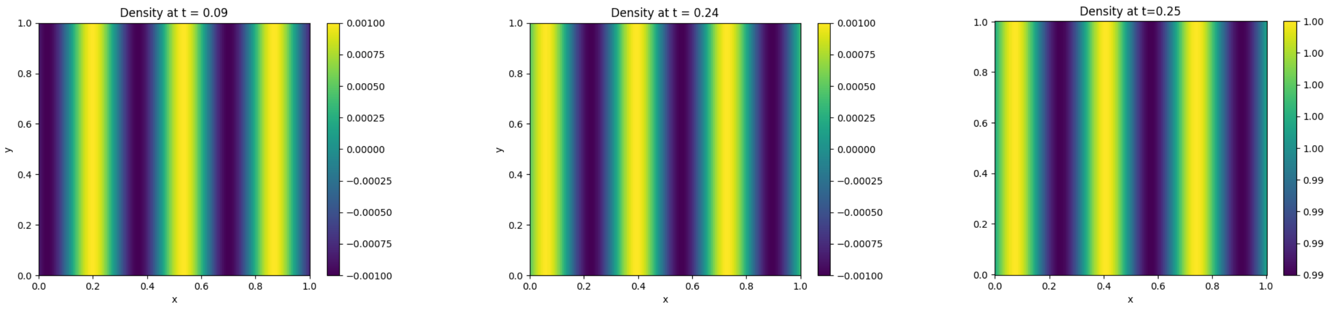
3.1 Test Case: Sound Waves in a Fluid

In order to first test our implementation of the finite volume method we were given code in sound_wave.cpp to simulate a sound wave traveling through a perfect fluid with a velocity v_x shown in the equation below.

$$v_x = \frac{\sqrt{\gamma P_0/\rho_0}\rho_1}{\rho_0} \tag{15}$$

where $P_0 = 1$, $\rho_0 = 1$, and $\rho_1 = 10^{-3} sin(6\pi x)$. As this project only focuses on ideal gas our adiabatic index, γ throughout is $\frac{5}{3}$.

Below are some images from the movie we created at different time frames. The full movie can be found under output.mp4.



(a) Sound wave at t=0.09

(b) Sound wave at t=0.24

(c) Sound wave at t=0.25

We can estimate the velocity of the sound wave using the images above. At t=0.09 we see a local density maxima centered at x=0.2. By following that same maxima as it moves in the movie we can see that it is almost centered at x=0.4 at t=0.24. Therefore we can estimate the wave velocity to be about 1.333. If we assume that t=0.25 is closer to centered at x=0.4 we estimate the wave velocity to be 1.25. We can reasonably assume that the velocity is between 1.25 and 1.333. This matches the theory calculation of $c = \sqrt{\gamma P_0/\rho_0}$ which in our case, $c = \sqrt{5/3 * 1/1} = 1.291$ which is between the expected 1.25 and 1.333.

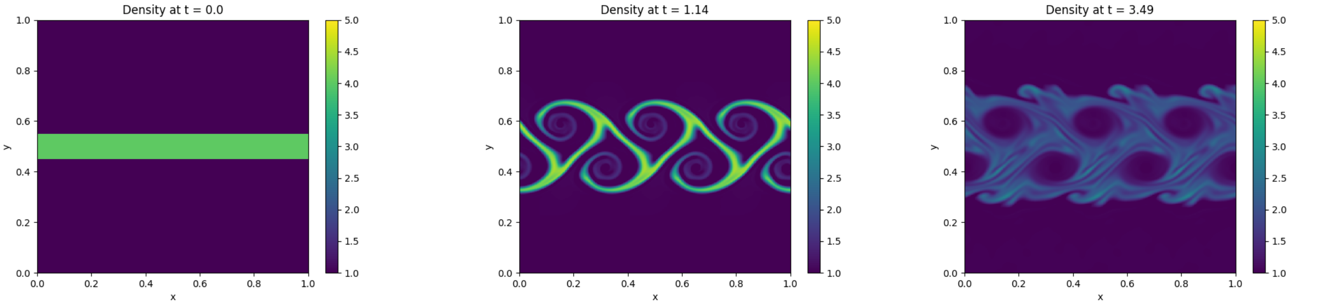
3.2 Kelvin Helmholtz Instability

The Kelvin Helmholtz instability is a particular case of fluid motion where there is a boundary layer between the flow of the fluid going two different velocities. We implemented this instability within the file kelvin_helmholtz.cpp. The parameters relate to each other as follows where y is the boundary layer:

$$a \leq y \leq b : \rho = \rho_d, v_x = v_0 \tag{16}$$

$$y < a \text{ or } y > b : \rho = \rho_0, v_x = -v_0 \tag{17}$$

Below are images from the movie we created with the parameters $a = 0.45, b = 0.55, \rho_d = 4, \rho_0 = 0, v_0 = 0.5$ and a resolution of 514x514. The full movie can be found under kelvin.mp4.



(a) KH Instability at t=0

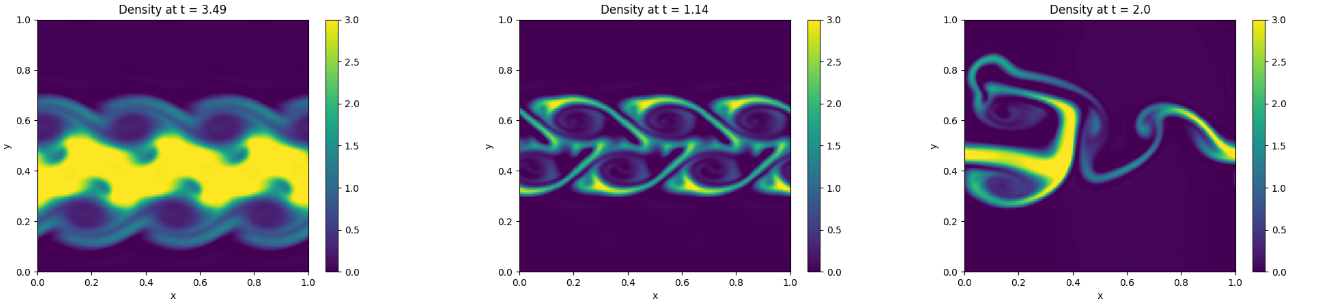
(b) KH Instability at t=1.14

(c) KH Instability at t=3.49

As seen in the images we are successfully able to observe the evolution of the KH instability and the slow dissipation of the density as it forms vertices. The high resolution example in particular allows us to get a good image of the very small structures within and around the vertices, although it is computationally costly.

3.2.1 Variance with Initial Conditions

In order to further understand KH instability and the different parameters that go into it we varied certain initial conditions and observed what happened during each.



(a) KH Instability at t=3.49 with an a=0.25 and b=0.55.

(b) KH Instability at t=1.14 with a $v_0 = 0.75$.

(c) KH Instability at t = 2.0 with a set k=π.

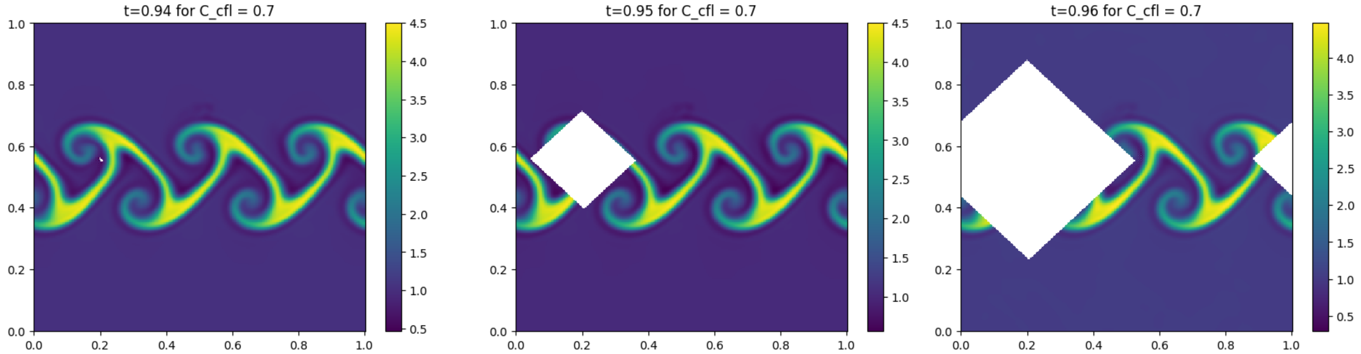
We found that, rather intuitively, when you increase the size of the layer of dense fluid it takes longer for the varying velocities and density to dissipate and so you get an extended continuation of the KH instability behavior.

Additionally, if we increase the velocity in the x direction of the fluid we observe much quicker evolution. This makes sense as there's a larger overall difference in velocity causing more extreme behavior and as the fluid is colliding more often. The full movie for this can be found under kelvinVaryV.mp4.

When we changed k we verified that the multiple of π input correlated with the number of vertices produced. For a single vertex this resulted in a very interesting plot and evolution as the vertex eventually broke off and a new boundary between the varying velocity fluids was produced. At this new boundary you can see what looks like another singular vertex being formed. The full movie for this can be found under kelvinVaryk.mp4.

3.2.2 Variance with CFL Condition

Going back to equation 5 we are reminded of the arbitrary value 0.4 that we set for C in using the CFL condition. After trying a value of C = 0.7 the Kelvin Helmholtz scheme became unstable, therefore we know that C is between 0.4 and 0.7, After trying a value of C = 0.6 the scheme successfully ran from t=0 to t=2. Therefore we can now pinpoint that C is between 0.6 and 0.7. The scheme runs successfully from t=0 to t=0.94 for a C value of 0.7.



(a) Graph when cfl constant is 0.7 at t=0.94

(b) Graph when cfl constant is 0.7 at t=0.95

(c) Graph when cfl constant is 0.7 at t=0.96

In our graph the NAN values are just white space. The instability begins at t=0.94 when $C_{cfl} = 0.7$ and then increases from there. With lower C_{cfl} the instability appears sooner and with higher values the instability appears after more time. This makes sense when looking at equation (5) and the dependency on time and velocity.

4 Conclusion

Through the finite volume method, we can reasonably simulate fluid dynamics. Our resulting images, movies and calculated values correspond to the expected and theorized results within a reasonable range of error considering some values, such as the partial derivatives, were computationally estimated.