Weekly Report

Damien Beecroft

February 23, 2024

1 Introduction

My update this week is on the shorter side. I was able to implement the time-stepping scheme with operator splitting. The solutions that this method produced were very similar to the results that I saw in the Lax-Friedrichs fast sweeping method.

2 Detailed Progress Overview

2.1 Time-Stepping Scheme for the Normal Shock Problem

Impplementing the time-stepping scheme for the normal shock problem is the best way to determine where the issue in my implementation of the Lax-Friedrichs fast sweeping method for the Boltzmann equation is coming from. The time dependent equation we are trying to solve is

$$\partial_t f + v\partial_x f = Q^+(f,f) - C\rho f.$$

Suppose $v > 0$. We use the following time-stepping discretization with operator splitting:

$$\frac{f_i^* - f_i^{(l)}}{\Delta t} + v\frac{f_i^{(l)} - f_{i-1}^{(l)}}{\Delta x} = 0$$
$$\frac{f_i^{(l+1)} - f_i^*}{\Delta t} = Q^+(f^*, f^*) - C\rho_i^* f_i^*.$$

Isolating the relevant terms we get the following update rule:

$$f_i^* = f_i^{(l)} - \frac{v\Delta t}{\Delta x}(f_i^{(l)} - f_{i-1}^{(l)}) = \left(1 - \frac{v\Delta t}{\Delta x}\right) f_i^{(l)} + \frac{v\Delta t}{\Delta x} f_{i-1}^{(l)}$$
$$f_i^{(l+1)} = f_i^* + \Delta t(Q^+(f^*, f^*) - C\rho_i^* f_i^*) = (1 - C\Delta t\rho_i^*)f_i^* + \Delta t Q^+(f^*, f^*).$$

There is a subtlety to what has been written above. If we want to apply the global operator $Q^+(\cdot,\cdot)$ to f^*, then we need all of f^*. Therefore, the entire intermediate field f^* must be determined before starting to compute $f^{(l+1)}$.

2.2 Results

The results from the solution achieved with this method are similar to the results I got using the Lax-Friedrichs fast sweeping method. What I mean by this is that the density function begins to blow up with a large hump for $x < 0$.

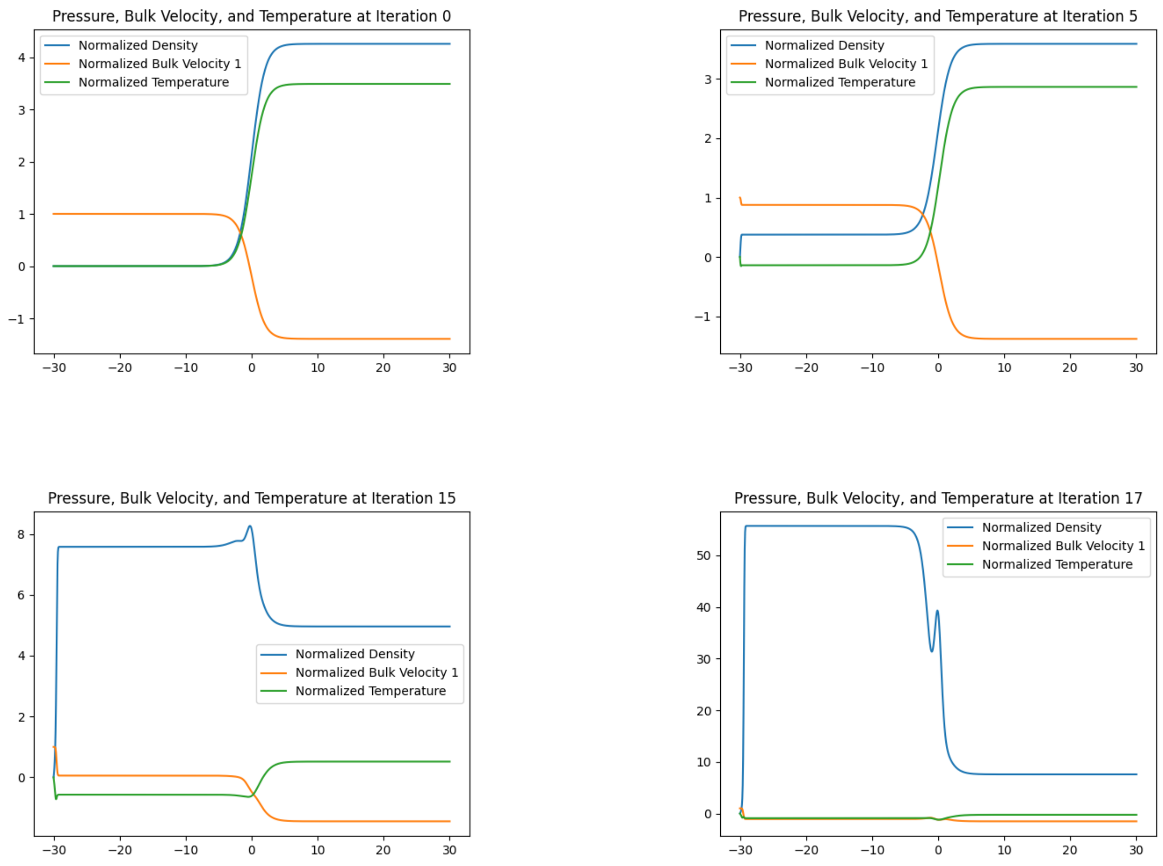

Figure 1: These are the results from the time-stepping method. We see that the density function begins to grow without bound as time marches forward. This is similar to what occurs in the Lax-Friedrichs fast sweeping scheme that I implemented

I also had time to perform one sanity check this week. I removed the collision operator from the equation in order to see whether or not the solution simply advected to the right as it should. In other words, I modeled

$$\partial_t f + v\partial_x f = 0.$$

I achieved exactly what I expected for this equation. The initial conditions simply moved to the left. This confirms what I had suspected: I am handling the collision operator incorrectly.

3 To Do For Next Week

It is clear that the collision operator is the source of my problems. I plan to revisit the implementation of the collision operator and determine why it misbehaving. I mentioned in the previous update that I altered Jingwei's code as seen below.

```
def Qplus(f, N, R, L, Ntheta):
    """
    Carleman spectral method for the classical Boltzmann collision operator
    2D Maxwell molecule
    N # of Fourier modes: f(N,N), Q(N,N)
    theta: mid-point rule
    """
    temp = np.concatenate((np.arange(0,N//2),np.arange(-N//2,0,1)))
    l1 = np.array([[row]*N for row in temp])
    l2 = l1.T

    FTf = fft2(f)

    QG = np.zeros((N, N), dtype=np.complex_)
    bb = np.zeros((N, N), dtype=np.complex_)

    wtheta = np.pi / Ntheta
    theta = np.arange(wtheta / 2, np.pi, wtheta)
    sig1 = np.cos(theta)
    sig2 = np.sin(theta)

    for q in range(Ntheta):
        aa1 = alpha2(l1 * sig1[q] + l2 * sig2[q], R, L)
        aa2 = alpha2(np.sqrt(l1**2 + l2**2 - (l1 * sig1[q] + l2 * sig2[q])**2), R, L)

        QG += 2 * wtheta * ifft2(aa1 * FTf) * ifft2(aa2 * FTf)
        bb += 2 * wtheta * aa1 * aa2

    # Original code #################
    # QL = f * ifft2(bb * FTf)
    # Q = np.real(QG - QL)
    ################################

    # Adjusted code #################
    Q = np.real(QG)
    ################################

    return Q
```

This is an inuitive way to change the method, but I have not actually done any analysis to ensure that this is correct thinng to do.