

# Weekly Report 1

Damien

November 10, 2023

## 1 Progress

This week I worked on implementing the results in problem 5.1.1 in Kao et al. I made good progress, but the results are still not satisfactory. I describe what I did in detail to help figure out where the problem lies.

### 1.1 My Understanding

I outline here my understanding of sections 2.1.2 and 2.1.3 in Kao et al. that discuss the essential details necessary for implementing problem 5.1.1. The WENO method produces approximations for the fluxes on the cell boundaries. In my implementation I use the third order WENO method. We work on a staggered grid similar to Figure 1.

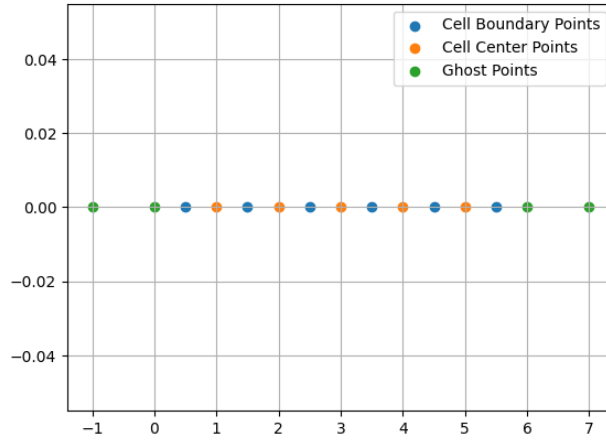


Figure 1: This figure shows how the grid cells are distributed for the third order WENO scheme. The numbers on the x-axis denote the indices of the points. Since this scheme is third order we have two ghost cells on either side.

The forward fluxes are given by

$$\hat{f}_{j+\frac{1}{2}} = w_0 \hat{f}_{j+\frac{1}{2}}^{(0)} + w_1 \hat{f}_{j+\frac{1}{2}}^{(1)} = w_0 \left( \frac{1}{2} f_{j+1} + \frac{1}{2} f_j \right) + w_1 \left( \frac{3}{2} f_j - \frac{1}{2} f_{j-1} \right). \quad (1)$$

Using this, the forward Lax-Friedrichs WENO flux is

$$\hat{\hat{f}}_{j+\frac{1}{2}} = \hat{f}_{j+\frac{1}{2}} + \frac{\sigma}{2} (u_{j+1} - u_j).$$

Putting this all together we get that the forward Lax-Friedrichs WENO sweep is

$$\begin{aligned} \frac{(\hat{f}_{j+\frac{1}{2}} - \frac{\sigma}{2}(u_{j+1} - u_j)) - (\hat{f}_{j-\frac{1}{2}} - \frac{\sigma}{2}(u_j - u_{j-1}))}{\Delta x} &= s(u_j, x_j) \rightarrow \\ \hat{f}_{j+\frac{1}{2}} - \hat{f}_{j-\frac{1}{2}} - \frac{\sigma}{2}(u_{j+1} - u_j) + \frac{\sigma}{2}(u_j - u_{j-1}) &= \Delta x s(u_j, x_j) \rightarrow \\ \sigma(u_j - \frac{1}{2}(u_{j+1} + u_{j-1})) &= \Delta x s(u_j, x_j) - (\hat{f}_{j+\frac{1}{2}} - \hat{f}_{j-\frac{1}{2}}) \rightarrow \\ u_j &= \frac{1}{2}(u_{j+1} + u_{j-1}) + \frac{1}{\sigma}(\Delta x s(u_j, x_j) - (\hat{f}_{j+\frac{1}{2}} - \hat{f}_{j-\frac{1}{2}})). \end{aligned}$$

Filling in the values of  $\hat{f}_{j+\frac{1}{2}}$  and  $\hat{f}_{j-\frac{1}{2}}$  gives

$$u_j = \frac{1}{2}(u_{j+1} + u_{j-1}) + \frac{1}{\sigma}(\Delta x s(u_j, x_j) - ((\hat{f}_{j+\frac{1}{2}} + \frac{\sigma}{2}(u_{j+1} - u_j)) - (\hat{f}_{j-\frac{1}{2}} + \frac{\sigma}{2}(u_j - u_{j-1}))).$$

Filling in the time steps gives

$$u_j^{n+1} = \frac{1}{2}(u_{j+1}^n + u_{j-1}^n) + \frac{1}{\sigma}(\Delta x s(u_j^n, x_j) - ((\hat{f}_{j+\frac{1}{2}} + \frac{\sigma}{2}(u_{j+1}^n - u_j^n)) - (\hat{f}_{j-\frac{1}{2}} + \frac{\sigma}{2}(u_j^n - u_{j-1}^n)))).$$

Note that if we replaced the  $u_j^{n+1}$  on the left-hand side with  $u_j^n$  we would simply have equality. This method is a non-linear analogue to a simple iteration method. Now, let's look at the backward iteration. The backwards fluxes are given by

$$\hat{f}_{j+\frac{1}{2}} = w_0 \hat{f}_{j+\frac{1}{2}}^{(0)} + w_1 \hat{f}_{j+\frac{1}{2}}^{(1)} = w_0 \left( \frac{1}{2} f_j + \frac{1}{2} f_{j+1} \right) + w_1 \left( \frac{3}{2} f_{j+1} - \frac{1}{2} f_{j+2} \right). \quad (2)$$

Using this, the backwards Lax-Friedrichs WENO flux is

$$\hat{f}_{j+\frac{1}{2}} = \hat{f}_{j+\frac{1}{2}} + \frac{\sigma}{2}(u_{j+1} - u_j).$$

We use this to get the backwards Lax-Friedrichs WENO sweep.

$$u_j = \frac{1}{2}(u_{j+1} + u_{j-1}) + \frac{1}{\sigma}(\Delta x s(u_j, x_j) - (\hat{f}_{j+\frac{1}{2}} - \hat{f}_{j-\frac{1}{2}})).$$

Filling in the values of  $\hat{f}_{j+\frac{1}{2}}$  and  $\hat{f}_{j-\frac{1}{2}}$  gives

$$u_j = \frac{1}{2}(u_{j+1} + u_{j-1}) + \frac{1}{\sigma}(\Delta x s(u_j, x_j) - ((\hat{f}_{j+\frac{1}{2}} + \frac{\sigma}{2}(u_{j+1} - u_j)) - (\hat{f}_{j-\frac{1}{2}} + \frac{\sigma}{2}(u_j - u_{j-1}))).$$

Filling in the time steps gives

$$u_j^{n+1} = \frac{1}{2}(u_{j+1}^n + u_{j-1}^n) + \frac{1}{\sigma}(\Delta x s(u_j^n, x_j) - ((\hat{f}_{j+\frac{1}{2}} + \frac{\sigma}{2}(u_{j+1}^n - u_j^n)) - (\hat{f}_{j-\frac{1}{2}} + \frac{\sigma}{2}(u_j^n - u_{j-1}^n)))).$$

Specifically for problem 5.1.1, I set the values on the ghost cells to be zero to enforce the dirichlet boundary conditions.

## What I Implemented

I was a bit confused at first, so the method I originally implemented was the the following. I know this is not the method that they are proposing in the paper, but bear with me briefly. Let  $\alpha = \max_u |f'(u)|$ . The forward sweep is

$$u_j^{n+1} = u_j^n - \frac{\Delta t}{\Delta x}(\hat{f}_{j+\frac{1}{2}}^+ - \hat{f}_{j-\frac{1}{2}}^+) + \Delta t s(u_j^n, x_j)$$

where

$$\hat{f}_{j+\frac{1}{2}}^+ = \frac{1}{2}(\hat{f}_{j+\frac{1}{2}} + \alpha u)$$

and  $\hat{f}_{j+\frac{1}{2}}$  is defined using Equation 1. The backwards sweep is

$$u_j^{n+1} = u_j^n - \frac{\Delta t}{\Delta x}(\hat{f}_{j+\frac{1}{2}}^- - \hat{f}_{j-\frac{1}{2}}^-) + \Delta t s(u_j^n, x_j)$$

where

$$\hat{f}_{j+\frac{1}{2}}^- = \frac{1}{2}(\hat{f}_{j+\frac{1}{2}} + \alpha u)$$

and  $\hat{f}_{j+\frac{1}{2}}$  is defined using Equation 2. I applied this method to problem 5.1.1 and got the following plots. This is obviously not the desired method, but it shows that I am on the right track. I am not sure if the

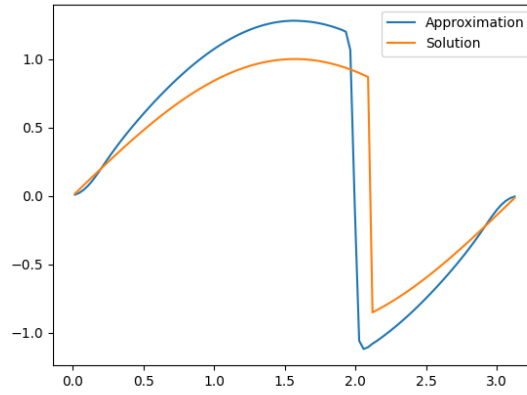


Figure 2: Problem 5.1.1 with initial condition  $u(x, 0) = \beta \sin(x)$  where  $\beta = \frac{1}{2}$ .

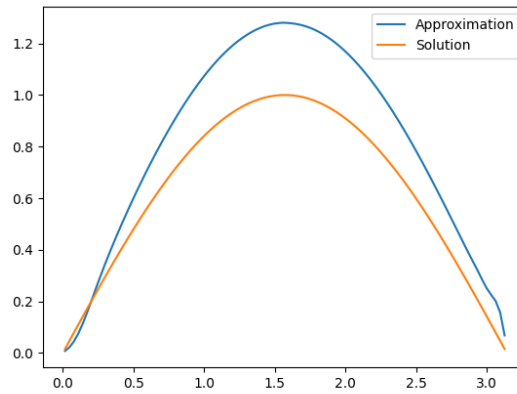


Figure 3: Problem 5.1.1 with initial condition  $u(x, 0) = \beta \sin(x)$  where  $\beta = 2$ .

difference between the true solution and the approximation is due to numerical issues or whether there is a problem with my implementation.

## Fixing My Mistake

Once I realized this was not the method I was supposed to implement I tried to adjust the code I had written to match the Lax-Friedrichs WENO sweeping method outlined in section 2.1.3. However, the method does not currently run. There is probably some error baked into the implementation that I am missing so I am going to implement the method from scratch next week. However, I understand the paper much better now so hopefully it will work this time.

## Miscellaneous

I have begun re-reading Finite Volume Methods by Randy LeVeque to solidify my understanding of numerical methods for hyperbolic equations.

## 2 To Do

Re-implement the method from section 2.1.3 to solve 5.1.1.