# Problem Set 2

Relativistic Shock Tube Problem

due date: 2nd of February 2024

# 1D Relativistic Shock Tube

In this problem set, we follow Martí & Müller[1] and solve the 1D relativistic shock tube problem in flat spacetime, which is one of the standard numerical tests in special relativistic hydrodynamics. In Minkowski spacetime, the evolution equations in conservative form can be written in vector form as

$$\frac{\partial}{\partial t}\mathbf{U} + \frac{\partial}{\partial x^j}\mathbf{F}^j = 0. \tag{1}$$

In the finite volume formulation, the quantity $u_i^n$ represents the cell average of $u$ centered in $x_i$

$$u_i^n = \frac{1}{\Delta x}\int_{x_{i-1/2}}^{x_{i+1/2}} u(x, t^n)dx, \tag{2}$$

where $x_{i\pm1/2} = x_i \pm \frac{1}{2}\Delta x$ the locations of cell boundary. Hence we can write the equations as

$$\partial_t u_i = -\frac{F_{i+1/2} - F_{i-1/2}}{\Delta x}, \tag{3}$$

where the fluxes $F_{i\pm1/2}$ are evaluated at the cell boundaries such that the total quantity $\partial_t \int_{x_1}^{x_N} u(x)dx = \partial_t \left(\Delta x \sum_{i=1}^{N} u_i\right) = -(F_{N+1/2} - F_{1/2})$ depends only on the flux through the boundaries of the domain.

In our example, the state vector $\mathbf{U} = (D, S_i, \tau)$ is defined by

$$\begin{aligned} D &= \rho W, \\ S_i &= \rho h W^2 v_i, \\ \tau &= \rho h W^2 - p - D, \end{aligned} \tag{4}$$

and the flux vectors $\mathbf{F}^j = \left(F_D^j, F_{S_i}^j, F_\tau^j\right)$ are given by

$$\begin{aligned} F_D^j &= D v^j, \\ F_{S_i}^j &= S_i v^j + p\delta_i^j, \\ F_\tau^j &= (\tau + p)\, v^j, \end{aligned} \tag{5}$$

where $\rho$ and $p$ are rest mass density and pressure respectively, $h = 1 + \epsilon + P/\rho$ is the specific enthalpy, $\epsilon$ is the specific internal energy, $v^i = u^i/u^0$ are the components of the three-velocity of the fluid, and $W = u^0 = \frac{1}{\sqrt{1-v^i v_i}}$ is the Lorentz factor.

The system is closed by an equation of state (EOS) given in form $p = P(\rho, \epsilon)$. The characteristic velocities for this system is given by

$$\begin{aligned} \lambda_0^i &= v^i, \\ \lambda_\pm^i &= \frac{(1-c_s^2)v^i \pm c_s\sqrt{(1-v^2)[1-v^2c_s^2 - (1-c_s^2)(v^i)^2]}}{1-v^2c_s^2}, \end{aligned} \tag{6}$$

[1] https://link.springer.com/article/10.12942/lrr-2003-7

where $c_s$ is the sound speed defined by

$$c_s^2 = \left(\frac{\partial P}{\partial \rho} + \frac{P}{\rho^2}\frac{\partial P}{\partial \epsilon}\right)\frac{1}{h}. \tag{7}$$

Throughout this problem set, we use an ideal-gas EOS

$$p = (\Gamma - 1)\rho\epsilon, \tag{8}$$

with constant adiabatic index $\Gamma = 5/3$. In the hydrodynamic evolution, obtaining the conservative variables from primitive variables is straightforward. However, recovering primitive variables from the conservative quantities is non-trivial and need to solve numerically in the relativistic case.

## A Primitive recovery

Here, we first try out the primitive recovery scheme from Thierfelder et al.[2], which was also discussed during the lecture. With an initial guess of pressure $p_*$ (usually from the previous time step), one can estimate the primitive variables from the conservative variables and $p_*$ as

$$\hat{v}_i(p_*) = \frac{S_i}{\tau + D + p_*}, \tag{9}$$

$$\hat{W}(p_*) = \frac{1}{\sqrt{1 - \hat{v}^i\hat{v}_i}}, \tag{10}$$

$$\hat{\rho}(p_*) = \frac{D}{\hat{W}}, \tag{11}$$

$$\hat{\epsilon}(p_*) = \frac{\tau + D\left(1 - \hat{W}\right) + p_*\left(1 - \hat{W}^2\right)}{D\hat{W}}. \tag{12}$$

I here use a hat symbol for the estimated variables to distinguish from the actual primitive variables. Thus, the pressure can be determined by finding the root of the master function using the EOS

$$f(p_*) := p_* - P\left(\hat{\rho}(p_*), \hat{\epsilon}(p_*)\right). \tag{13}$$

We can then use the Newton-Raphson method to iterate until the root is found

$$p_*^{\texttt{new}} = p_*^{\texttt{old}} - \frac{f(p_*)}{f'(p_*)}. \tag{14}$$

**(10 marks)** Obtain the form of $f'(p_*)$ in terms of conservative variables $(D, S_i, \tau)$, $p_*$, estimated variables $(\hat{\rho}, \hat{\epsilon}, \hat{v}_i, \hat{W})$ and EOS derivatives $\frac{\partial P}{\partial \rho}$, $\frac{\partial P}{\partial \epsilon}$.

**(10 marks)** Write a short code for the primitive recovery. Use the ideal-gas EOS

[2] https://arxiv.org/pdf/1104.4751.pdf

$p = (\Gamma - 1)\rho\epsilon$ and $\Gamma = 5/3$. Find the corresponding $(D, S_x, \tau)$ for $(\rho, p, v^x) = (1, 0, 0)$, $(10, 40/3, 0)$ and $(0.1, 1, 0.99)$, and then try to recover the primitive variables from those using your script.

The Newton-Raphson method works fine with analytic EOS, but it might give inaccurate results with tabulated EOS because it requires EOS derivatives $\frac{\partial P}{\partial \rho}$, $\frac{\partial P}{\partial \epsilon}$. Here, we examine another popular and robust primitive recovery method introduced by Galeazzi et al.[3] (for a more advanced algorithm in GRMHD, it is presented in Kastaun et al.[4]). We first define the following rescaled variables

$$S := \sqrt{S^i S_i} \quad , \quad r := \frac{S}{D} \quad , \quad q := \frac{\tau}{D} \quad , \quad k := \frac{S}{\tau + D}. \tag{15}$$

Then, we can estimate the primitive variables in terms of $z$

$$\begin{aligned}
\hat{W}(z) &= \sqrt{1+z^2} \quad &, \quad \hat{\epsilon}(z) &= \hat{W} q - zr + \frac{z^2}{1+\hat{W}}, \\
\hat{\rho}(z) &= \frac{D}{\hat{W}} \quad &, \quad \hat{p}(z) &= P(\hat{\rho}, \hat{\epsilon}), \\
\hat{a}(z) &= \frac{\hat{p}}{\hat{\rho}(1+\hat{\epsilon})} \quad &, \quad \hat{h}(z) &= (1+\hat{\epsilon})(1+\hat{a}) .
\end{aligned} \tag{16}$$

The bounds of $z$ is given by

$$z_- := \frac{k/2}{\sqrt{1 - k^2/4}}, \quad z_+ := \frac{k}{\sqrt{1-k^2}}. \tag{17}$$

In the interval $[z_-, z_+]$, we can use any bracketing method (e.g. bisection method or Regula falsi method) to solve

$$f(z) := z - \frac{r}{\hat{h}} = 0. \tag{18}$$

With the root $z_0$ obtained, we can work out the primitive variables

$$\begin{aligned}
(\rho, p, \epsilon, W) &= (\hat{\rho}(z_0), \hat{p}(z_0), \hat{\epsilon}(z_0), \hat{W}(z_0)), \\
v_i &= \frac{S_i / D}{\hat{h}(z_0)\hat{W}(z_0)}.
\end{aligned} \tag{19}$$

**(20 marks)** Write a short code for this method. You can use the bisection method for the rootfinding. Test your script with the same testing cases as the previous Newton-Raphson method.

In this problem set we are restricted to only flat spacetime, but both the Newton-Raphson method and the bracketing method can be easily generalized to general relativistic case,

[3] https://arxiv.org/pdf/1306.4953.pdf
[4] https://arxiv.org/pdf/2005.01821.pdf

# B Riemann solver

Now we are ready to perform the 1D relativistic shock tube test numerically. We consider computational domain covers the region $0 \le x \le 1$ with the initial condition as

$$(\rho, p, v^x) = \begin{cases} (10, 40/3, 0) & \text{if } x < 0.5, \\ (1, 0, 0) & \text{if } x > 0.5. \end{cases} \tag{20}$$

### HLL solver

For simplicity, we consider the Hartan, Lax and van Leer (HLL) solver which approximates the solution of original Riemann problems with a single intermediate state

$$\mathbf{u}^{HLL} = \begin{cases} \mathbf{u}_L & \text{for } 0 < \lambda_L, \\ \mathbf{u}_* & \text{for } \lambda_L \le 0 \le \lambda_R, \\ \mathbf{u}_R & \text{for } 0 > \lambda_R, \end{cases} \tag{21}$$

where $\lambda_L$ and $\lambda_R$ are the minimum and maximum of the characteristic speeds respectively

$$\begin{aligned} \lambda_L &:= \min(\lambda_-(\mathbf{u}_L), \lambda_-(\mathbf{u}_R)), \\ \lambda_R &:= \max(\lambda_+(\mathbf{u}_L), \lambda_+(\mathbf{u}_R)). \end{aligned} \tag{22}$$

The intermediate state $\mathbf{u}_*$ is given by

$$\mathbf{u}_* = \frac{\lambda_R \mathbf{u}_R - \lambda_L \mathbf{u}_L + \mathbf{F}(\mathbf{u}_L) - \mathbf{F}(\mathbf{u}_R)}{\lambda_R - \lambda_L}, \tag{23}$$

which gives the numerical flux

$$\mathbf{F}_* = \frac{\lambda_R \mathbf{F}(\mathbf{u}_L) - \lambda_L \mathbf{F}(\mathbf{u}_R) + \lambda_L \lambda_R (\mathbf{u}_R - \mathbf{u}_L)}{\lambda_R - \lambda_L}. \tag{24}$$

Thus, the final numerical flux is given by

$$\mathbf{F}^{HLL} = \begin{cases} \mathbf{F}_L & \text{for } 0 < \lambda_L, \\ \mathbf{F}_* & \text{for } \lambda_L \le 0 \le \lambda_R, \\ \mathbf{F}_R & \text{for } 0 > \lambda_R. \end{cases} \tag{25}$$

Write a program to solve the 1D shock tube problem numerically up to $t = 0.4$ with HLL solver and different reconstruction schemes, namely Minmod, PPM, and WENO.

1. **(20 Marks) Minmod.** We can improve the accuracy by using a piecewise-linear reconstruction of $u_i(x)$ at each cell

$$u_i(x) = u_i + \sigma_i \frac{x - x_i}{\Delta x}, \quad \text{for } x_{i-1/2} \le x \le x_{i+1/2}, \tag{26}$$

where $\sigma_i$ is the slope of the linear reconstruction. One of the linear reconstruction is the *minmod* slope-limiter

$$\sigma_i := \mathrm{minmod}\left(u_i - u_{i-1}, u_{i+1} - u_i\right), \tag{27}$$

where

$$\mathrm{minmod}\left(\alpha, \beta\right) := \begin{cases} \alpha & \text{if } |\alpha| < |\beta| \text{ and } \alpha\beta > 0, \\ \beta & \text{if } |\alpha| > |\beta| \text{ and } \alpha\beta > 0, \\ 0 & \text{if } \alpha\beta \le 0. \end{cases} \tag{28}$$

At cell boundary $x_{i+1/2}$, the left and right states are given by

$$\begin{aligned} u_L &= u_i + \frac{1}{2}\sigma_i \\ u_R &= u_{i+1} - \frac{1}{2}\sigma_{i+1} \end{aligned} \tag{29}$$

2. **(20 Marks) Piecewise parabolic method (PPM).** One could obtain a higher order method by using parabolic reconstruction

$$u_i(x) = u_{i,-} + \xi\left(u_{i,+} - u_{i,-} + u_{6,i}\left(1 - \xi\right)\right), \quad \text{for } x_{i-1/2} \le x \le x_{i+1/2}, \tag{30}$$

where $\xi = \left(x - x_{i-1/2}\right)/\Delta x$, $u_{6,i} = 6u_i - 3\left(u_{i,+} + u_{i,-}\right)$, and $u_{i,\pm}$ is obtained by the following procedure

(a) Initialize $u_{i,\pm}$ as

$$u_{i,\pm} = \frac{1}{2}\left(u_i + u_{i\pm1}\right) \pm \frac{1}{6}\delta u_i \mp \frac{1}{6}\delta u_{i\pm1}, \tag{31}$$

where $\delta u_i = \mathrm{minmod}\left(u_{i+1} - u_{i-1}, 2(u_{i+1} - u_i), 2(u_i - u_{i-1})\right)$ is evaluated using monotonised central-difference limiter (MC) given by

$$\mathrm{minmod}\left(a, b, c\right) = \begin{cases} \min(a, b, c) & \text{, if } a, b, c > 0, \\ \max(a, b, c) & \text{, if } a, b, c < 0, \\ 0 & \text{, otherwise.} \end{cases} \tag{32}$$

In smooth regions away from extrema, this leads to the formula

$$u_{i,\pm} = \frac{7}{12}\left(u_i + u_{i\pm1}\right) - \frac{1}{12}\left(u_{i\mp1} + u_{i\pm2}\right). \tag{33}$$

(b) If $\left(u_{i,+} - u_i\right)\left(u_{i,-} - u_i\right) > 0$, then we set

$$u_{i,+} = u_{i,-} = u_i. \tag{34}$$

Otherwise, if one of $\left|u_{i,\pm} - u_i\right| \ge 2\left|u_{i,\mp} - u_i\right|$ then for that choice of $\pm = +, -$, we set

$$u_{i,\pm} = u_i - 2\left(u_{i,\mp} - u_i\right). \tag{35}$$

One drawback of such reconstruction is that the monotonicity constraints in 34 reduces the truncation error at extrema to $\mathcal{O}(\Delta x)$.

3. **(20 Marks) Weighted essentially non-oscillatory (WENO).** The WENO schemes make use all locally smooth subset stencils through convex combination to approximate the values on interface. Here, we consider the fifth-order WENO-JS scheme. The left and right states at $x_{i+1/2}$ are given by

$$\begin{aligned} u_L &= \omega_{L,1} u_{L,1} + \omega_{L,2} u_{L,2} + \omega_{L,3} u_{L,3}, \\ u_R &= \omega_{R,1} u_{R,1} + \omega_{R,2} u_{R,2} + \omega_{R,3} u_{R,3}, \end{aligned} \tag{36}$$

where $u_{L/R,1/2/3}$ are the three possible third order fluxes given by

$$\begin{aligned} u_{L,1} &= \frac{1}{3}u_{i-2} - \frac{7}{6}u_{i-1} + \frac{11}{6}u_i \quad , \quad u_{R,1} = \frac{1}{3}u_{i+3} - \frac{7}{6}u_{i+2} + \frac{11}{6}u_{i+1}, \\ u_{L,2} &= -\frac{1}{6}u_{i-1} + \frac{5}{6}u_i + \frac{1}{3}u_{i+1} \quad , \quad u_{R,2} = -\frac{1}{6}u_{i+2} + \frac{5}{6}u_{i+1} + \frac{1}{3}u_i, \\ u_{L,3} &= \frac{1}{3}u_i + \frac{5}{6}u_{i+1} - \frac{1}{6}u_{i+2} \quad , \quad u_{R,3} = \frac{1}{3}u_{i+1} + \frac{5}{6}u_i - \frac{1}{6}u_{i-1}. \end{aligned} \tag{37}$$

By considering the optimal order (fifth order) of $u_{L/R}$, one can obtain the optimal weights

$$\omega^{\circ}_{L,1} = \omega^{\circ}_{R,1} = \frac{1}{10}, \qquad \omega^{\circ}_{L,2} = \omega^{\circ}_{R,2} = \frac{6}{10}, \qquad \omega^{\circ}_{L,3} = \omega^{\circ}_{R,3} = \frac{3}{10}. \tag{38}$$

To achieve optimal accuracy in smooth region while keeping essentially non-oscillatory property near discontinuities, the non-linear weights are defined as

$$\omega_{L/R,r} = \frac{\alpha_{L/R,r}}{\sum_{i=1}^{3} \alpha_{L/R,r}}, \quad \alpha_{L/R,r} = \frac{\omega^{\circ}_{L/R,r}}{\left(\epsilon + IS_{L/R,r}\right)^p}, \quad r = 1, 2, 3, \tag{39}$$

where typically $\epsilon = 10^{-6}$ and $p = 2$, and $IS_{L/R,r}$ is the smoothness indicator given by

$$\begin{aligned} IS_{L,1} &= \frac{13}{12}\left(u_{i-2} - 2u_{i-1} + u_i\right)^2 + \frac{1}{4}\left(u_{i-2} - 4u_{i-1} + 3u_i\right)^2 \\ IS_{L,2} &= \frac{13}{12}\left(u_{i-1} - 2u_i + u_{i+1}\right)^2 + \frac{1}{4}\left(u_{i-1} - u_{i+1}\right)^2 \\ IS_{L,3} &= \frac{13}{12}\left(u_i - 2u_{i+1} + u_{i+2}\right)^2 + \frac{1}{4}\left(3u_i - 4u_{i+1} + u_{i+2}\right)^2 \\ IS_{R,1} &= \frac{13}{12}\left(u_{i+3} - 2u_{i+2} + u_{i+1}\right)^2 + \frac{1}{4}\left(u_{i+3} - 4u_{i+2} + 3u_{i+1}\right)^2 \\ IS_{R,2} &= \frac{13}{12}\left(u_{i+2} - 2u_{i+1} + u_{i-1}\right)^2 + \frac{1}{4}\left(u_{i+2} - u_i\right)^2 \\ IS_{R,3} &= \frac{13}{12}\left(u_{i+1} - 2u_i + u_{i-1}\right)^2 + \frac{1}{4}\left(3u_{i+1} - 4u_i + u_{i-1}\right)^2 \end{aligned} \tag{40}$$

*Hint:* It is better to reconstruct $(\rho, \epsilon, u^i)$ at the cell interfaces and then convert it to $p$ and conservative variables. Here, $u^i = Wv^i$ is used for reconstruction instead of $v^i$ to ensure that $v^i$ will never exceed 1 at the interfaces. Lorentz factor $W = \sqrt{1 + u^i u_i}$ and $v^i = u^i/W$ can be easily obtained afterwards.

# C Parallelization

Finally, it is time to use again different parallelization schemes to speed up your code. Please use

1. **(20 Marks)** Open MP, and
2. **(30 Marks)** MPI

Show that your parallelized code gives the same result as the serial version and measure the performance of your code by performing a strong scaling test (i.e. fix your problem size and measure the execution time with different number of processors). You may need to use a large number of grid points for the test.