Computational Astrophysic
Homework 2

d07222009 Guan-Ming Su

April 28, 2020

Problem 1

(a) In this problem, we simulated the strong shock Riemann problem using Lax-Friedrichs, Lax-Wendroff, MUSCL-Hancock schemes, and GAMER, compared with the analytical solution. The simulated density rho, velocity v_x and pressure P are plotted in Fig. 1, 2, and 3. It is obvious that Lax-Friedrichs scheme is unable to depict the sharp transition between different wave region, due to the strong dissipation. Lax-Wendroff solved the dissipation problem; however, it introduces artifical high-k wave around the transition region. As for the MUSCL-Hancock scheme, we can see that it catches the shock wave quite successfully: density, velocity and pressure are all close to the analytical result. Nevertheless, the GAMER seems to out-perform MUSCL-Hancock: it consumes fewer points to resolve the shock wave front and the contact discontinuity, as shown in the zoom-in figures for each physical quanties around the contact discontinuity or shock region. This might due to its capability of changing gridding adaptively (For Lax-Wendroff, MUSCL-Hancock schemes, and GAMER, the simulated curves along with the analytical counter-parts are shifted for clearness).

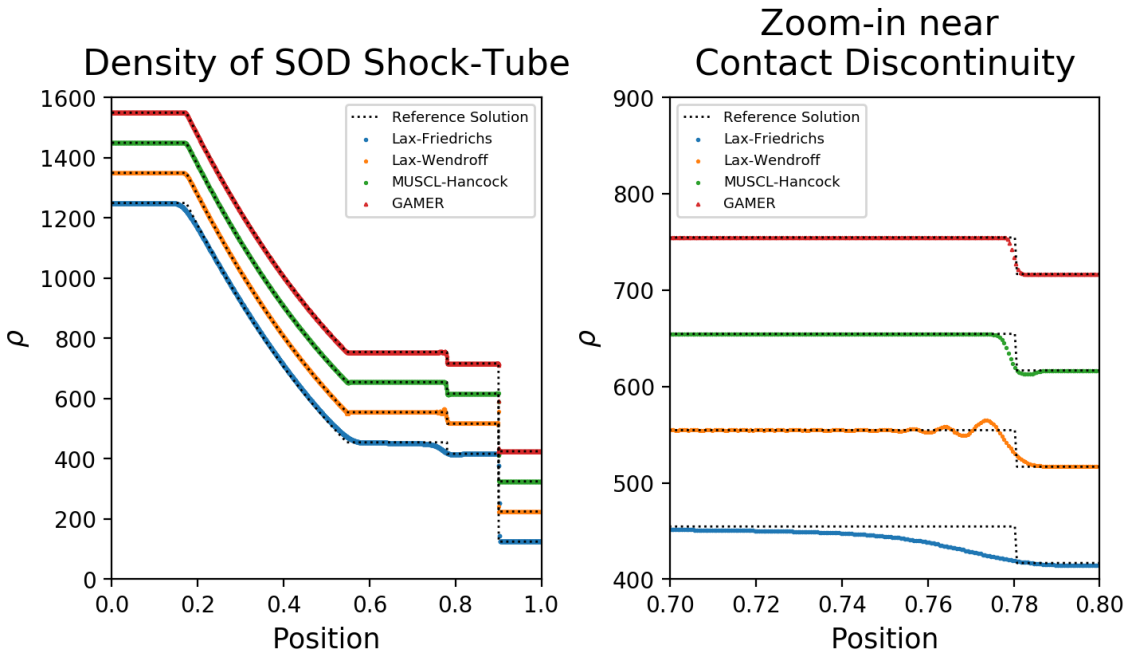

Figure 1: Density of Strong Shock

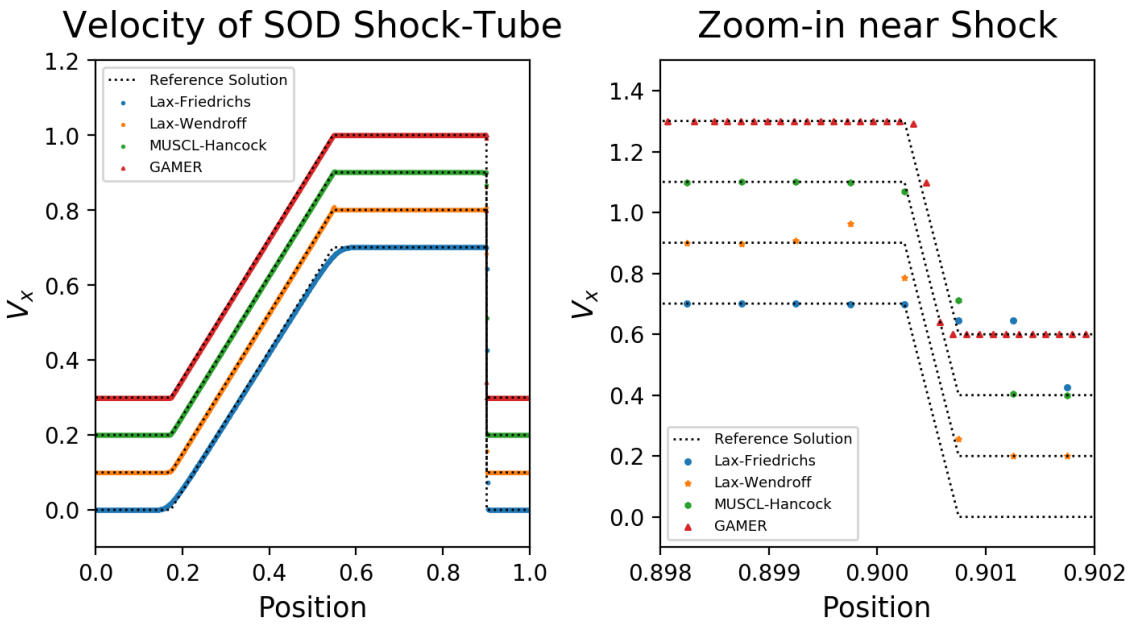

Figure 2: Velocity of Strong Shock

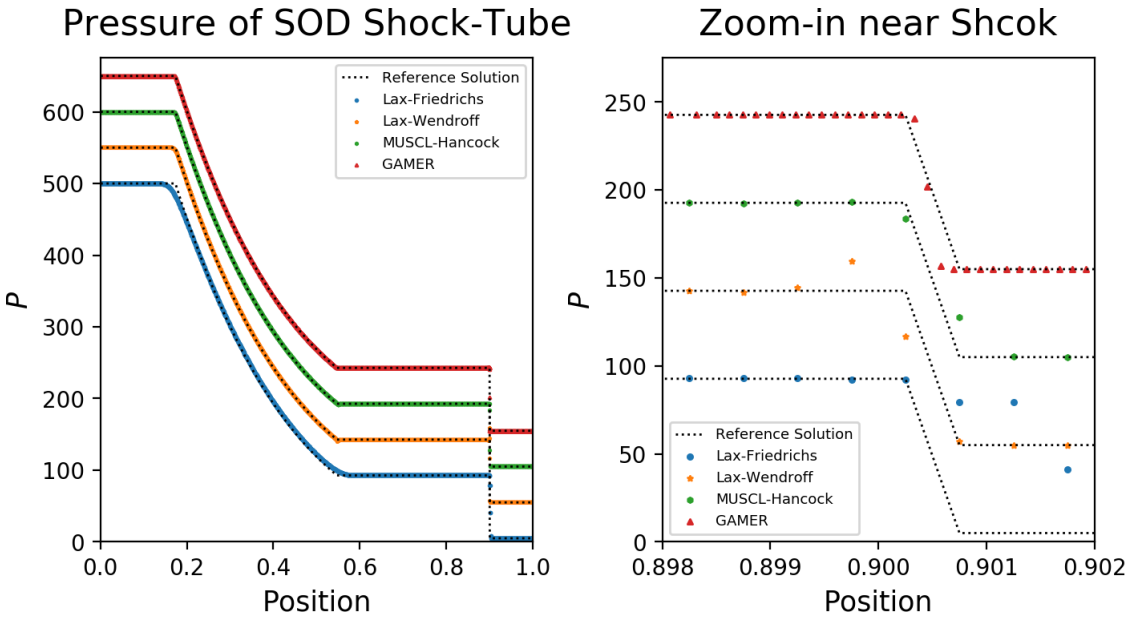

Figure 3: Pressure of Strong Shock

(b) In this problem, we tried to crash the MUSCL-Hancock scheme by using initial condition with large velocity, i.e., large kinetic energy, compared with the pressure. The code is "successfully" crashed when inital primitive quanties $\begin{bmatrix} \rho_L \\ V_{xL} \\ P_L \end{bmatrix} = \begin{bmatrix} 0.2 \\ 50.0 \\ 10.0 \end{bmatrix}$ and $\begin{bmatrix} \rho_L \\ V_{xL} \\ P_L \end{bmatrix} = \begin{bmatrix} 0.125 \\ -25.0 \\ 1.0 \end{bmatrix}$ are given, from $t = 0$ to $t = 0.05$. Error of negative pressure popped out soon after only a few updates, which might be caused by the relatively large kinetic energy compared with the internal energy, as mentioned above (We also tried $\begin{bmatrix} \rho_L \\ V_{xL} \\ P_L \end{bmatrix} = \begin{bmatrix} 0.2 \\ 50.0 \\ 50.0 \end{bmatrix}$ and $\begin{bmatrix} \rho_R \\ V_{xR} \\ P_R \end{bmatrix} = \begin{bmatrix} 0.125 \\ -25.0 \\ 10.0 \end{bmatrix}$, which did converge). Using dual energy probably can deal with this, however, we found that GAMER is not hampered by this issue as well. GAMER with dual energy scheme is also tried, which does not make much difference, suggesting that adaptive changed gridding itself is enough to tackle the problem caused by overwhelming kinetic energy.. GAMER with density refinement criteria is tested as well, with criteria shown in the table below (For GAMER with dual energy and with refinement criteria, the simulated value are shifted for clearness).

Level	0	1	2	3	4	5
Density	0.2	0.3	0.4	0.6	0.8	1.0

Table 1: Refinement Criteria for Riemann Problem (Input_Flag_Rho)

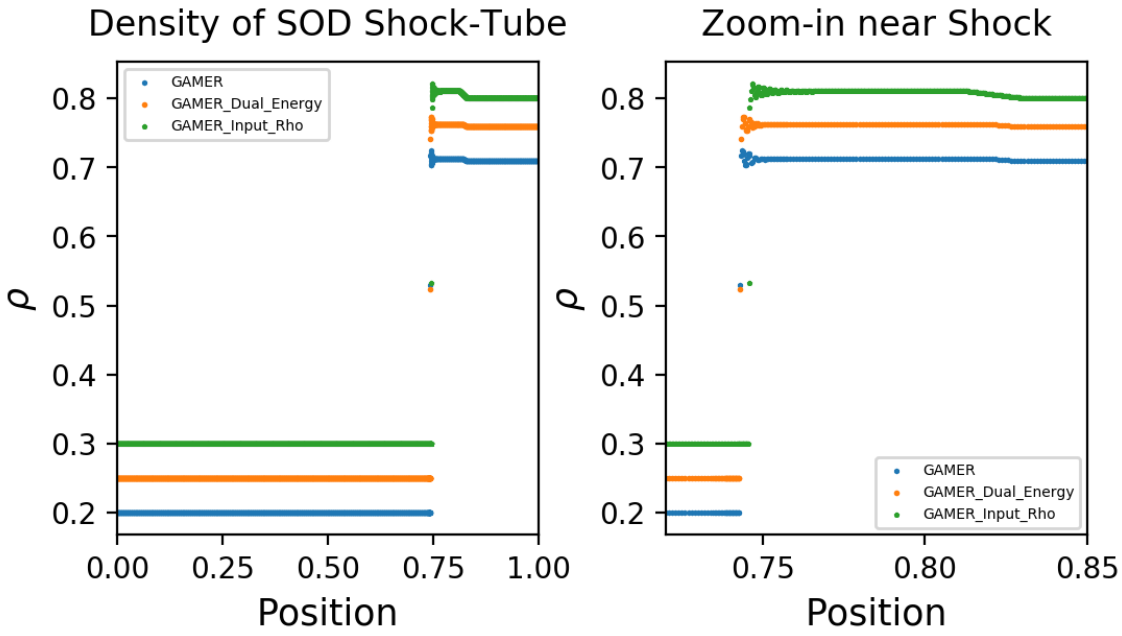

Figure 4: Density of Riemann Problem with Large Kinetic Energy

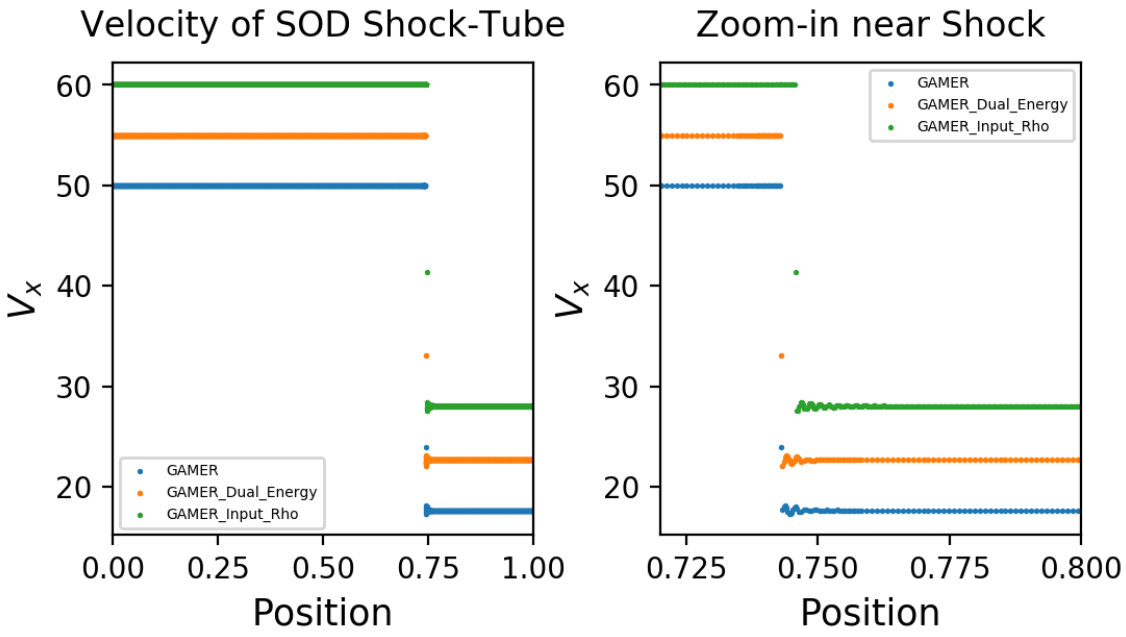

Figure 5: Velocity of Riemann Problem with Large Kinetic Energy

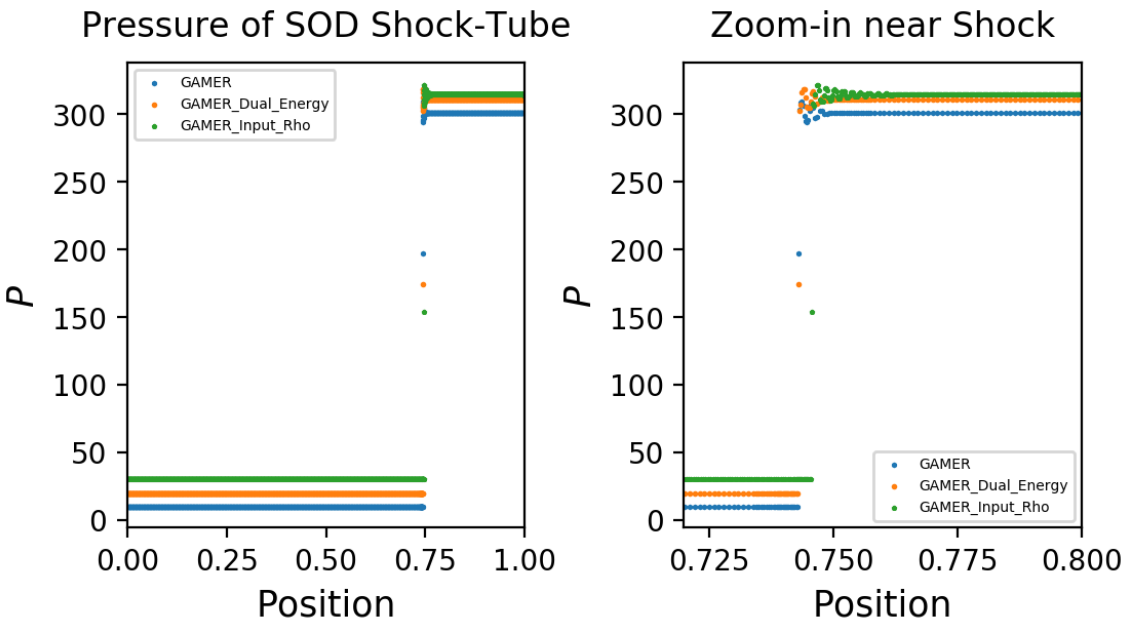

Figure 6: Pressure of Riemann Problem with Large Kinetic Energy

Problem 2

(a) Choosen AMR code: GAMER (GPU-accelerated Adaptive Mesh Refinement Code for Astrophysics)
(b) The Ryu and Jones Riemann problem (RJ2a), which is similar to Riemann problem but with fluid subjected to magnetic field, is selected as the GAMER test function for the following two reasons: (1) Since the magnetic field is considered here, the problem becomes more complicate with total 8 field components need to be solved. It is interesting to see how well GAMER can do. (2) A reference solution is provided in the GAMER's example archive, which can be used to validate the simulation result. This 3D problem is gridded in 16x16x16 base-level cells in $x,\ y,\ z$ direction, with refinement criteria specified in Input__Flag_Rho. The initial condition is given as: $\begin{bmatrix} \rho_L \\ V_{xL} \\ V_{yL} \\ V_{zL} \\ P_L \end{bmatrix} = \begin{bmatrix} 1.08 \\ 1.2 \\ 0.01 \\ 0.5 \\ 0.95 \end{bmatrix}$; $\begin{bmatrix} \rho_R \\ V_{xR} \\ V_{yR} \\ V_{zR} \\ P_R \end{bmatrix} = \begin{bmatrix} 1.0 \\ 0.0 \\ 0.0 \\ 0.0 \\ 1.0 \end{bmatrix}$ and $\begin{bmatrix} M_{xL} \\ M_{yL} \\ M_{zL} \end{bmatrix} = \begin{bmatrix} \frac{2}{\sqrt{4\pi}} \\ \frac{3.6}{\sqrt{4\pi}} \\ \frac{2}{\sqrt{4\pi}} \end{bmatrix}$; $\begin{bmatrix} M_{xR} \\ M_{yR} \\ M_{zR} \end{bmatrix} = \begin{bmatrix} \frac{2}{\sqrt{4\pi}} \\ \frac{4}{\sqrt{4\pi}} \\ \frac{2}{\sqrt{4\pi}} \end{bmatrix}$. Time evolves from 0 to 0.2 .
(c) The refinement criteria is specified in the Input_Flat_Rho, where the content is listed below:

Level	0	1	2	3	4	5
Density	1.0	1.1	1.2	1.3	1.4	1.45

Table 2: Refinement Criteria for RJ2a (Input_Flag_Rho)

While the compile parameter -DNLEVEL is set as 10 in the Makefile, we constrainted the available level by runtime parameter MAX_LEVEL, varying from 0 to 5, which affects the result quite significantly (shall be shown in (f)). Another criteria is set in the Input_Flag_Lohner, as listed below

Level	Threshold	Filter	Soften	MinDensity
0	0.50	0.01	0.00	0.00
1	0.50	0.01	0.00	0.00
2	0.50	0.01	0.00	0.00
3	0.50	0.01	0.00	0.00
4	0.50	0.01	0.00	0.00
5	0.50	0.01	0.00	0.00
6	0.50	0.01	0.00	0.00
7	0.50	0.01	0.00	0.00
8	0.50	0.01	0.00	0.00
9	0.50	0.01	0.00	0.00
10	0.50	0.01	0.00	0.00
11	0.50	0.01	0.00	0.00

Table 3: Refinement Criteria for RJ2a (Input_Flag_Lohner)

(d) The boundary condition is specified in the Input__Parameter, where the fluid boundary conditions in $x,\ y,\ z$ are all set to outflow.

(e) and (f) The result for density ρ, pressue P, velocity (V_x, V_y, V_z) and magnetic field (B_x, B_y, B_z) are listed below, with MAX_LEVEL increasing from 0 to 5. Though all simulation converges, we can clearly see that when the AMRlevel increases, the data points overlap more well with the reference curve. The best performance is given by MAX_LEVEL=4 and 5, whose simulation on ρ, P, (V_x, V_y, V_z) and (B_x, B_y) are quite close to the reference solution; however, the simulated B_z seems to deviates a little bit, especially around $x = 0.7$ to 0.9 region. The time consumed for each simulation with different MAX_LEVEL is also attached.

MAX_LEVEL	0	1	2	3	4	5
Time(s)	0.393366	0.606961	2.486324	23.490036	197.151280	2038.830045

Table 4: Time Consumed by RJ2a with Different MAX_LEVEL

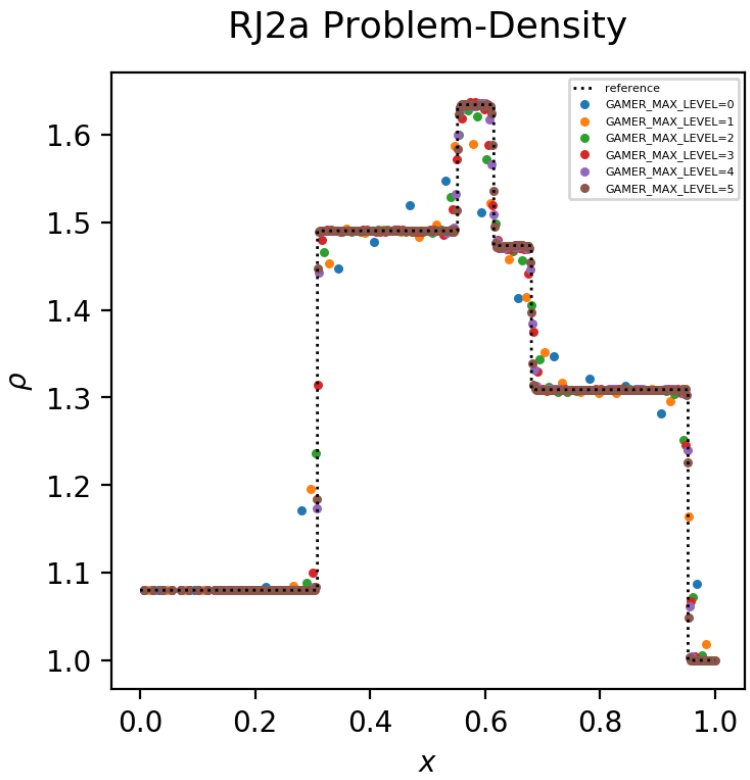

Figure 7: Density of RJ2a Problem

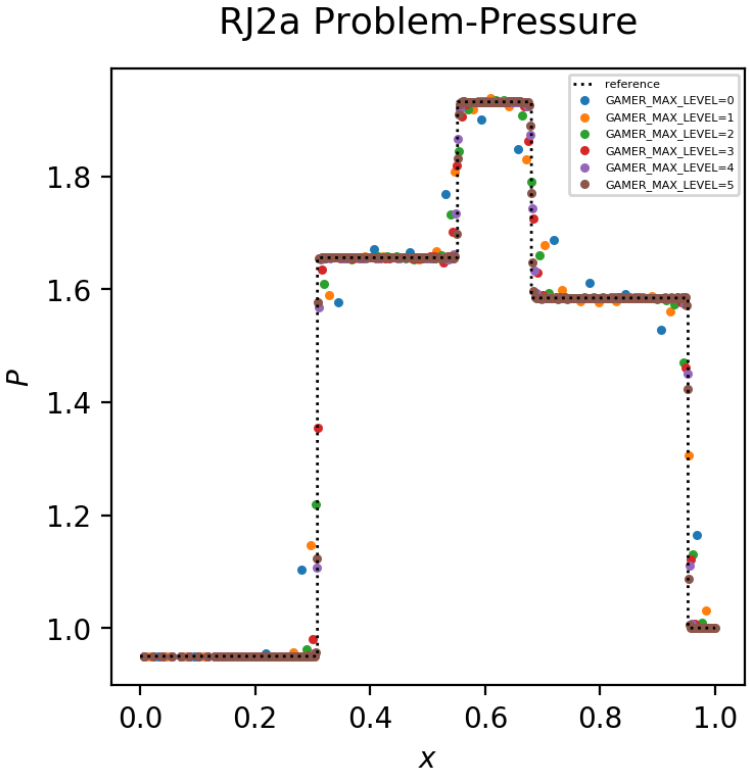

Figure 8: Pressure of RJ2a Problem

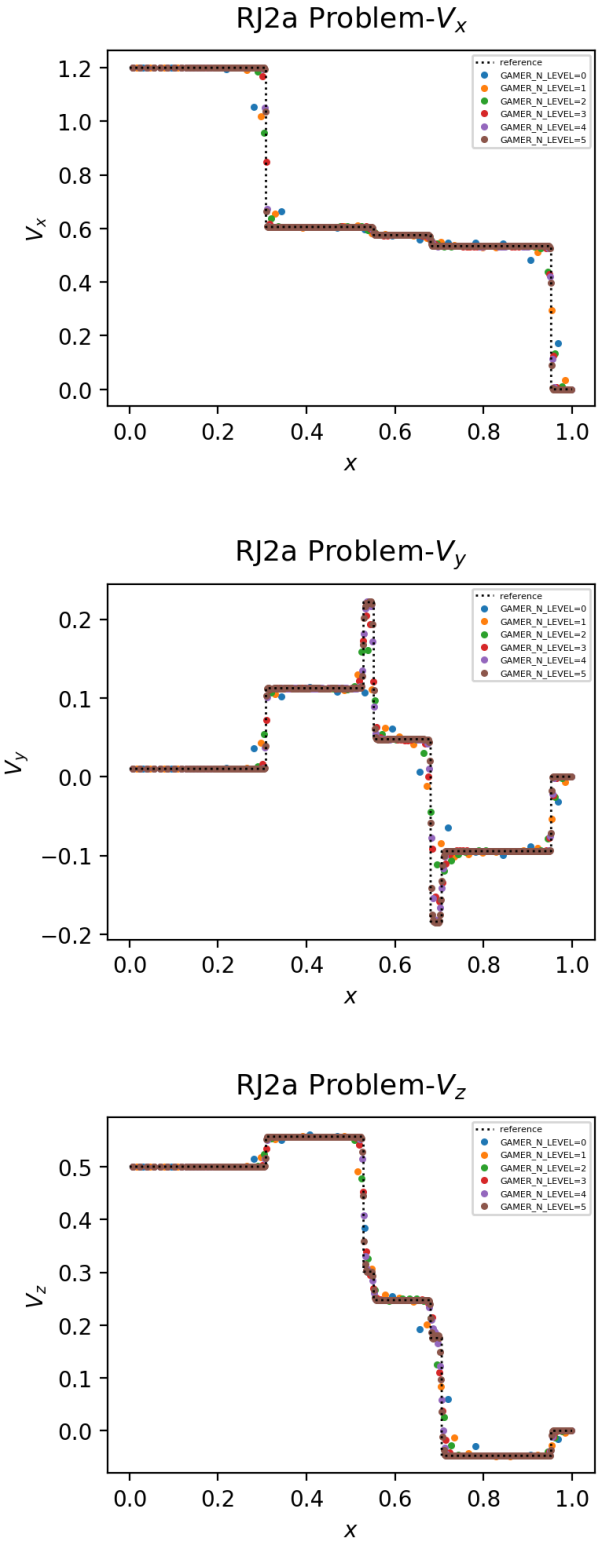

Figure 9: Velocity of RJ2a Problem

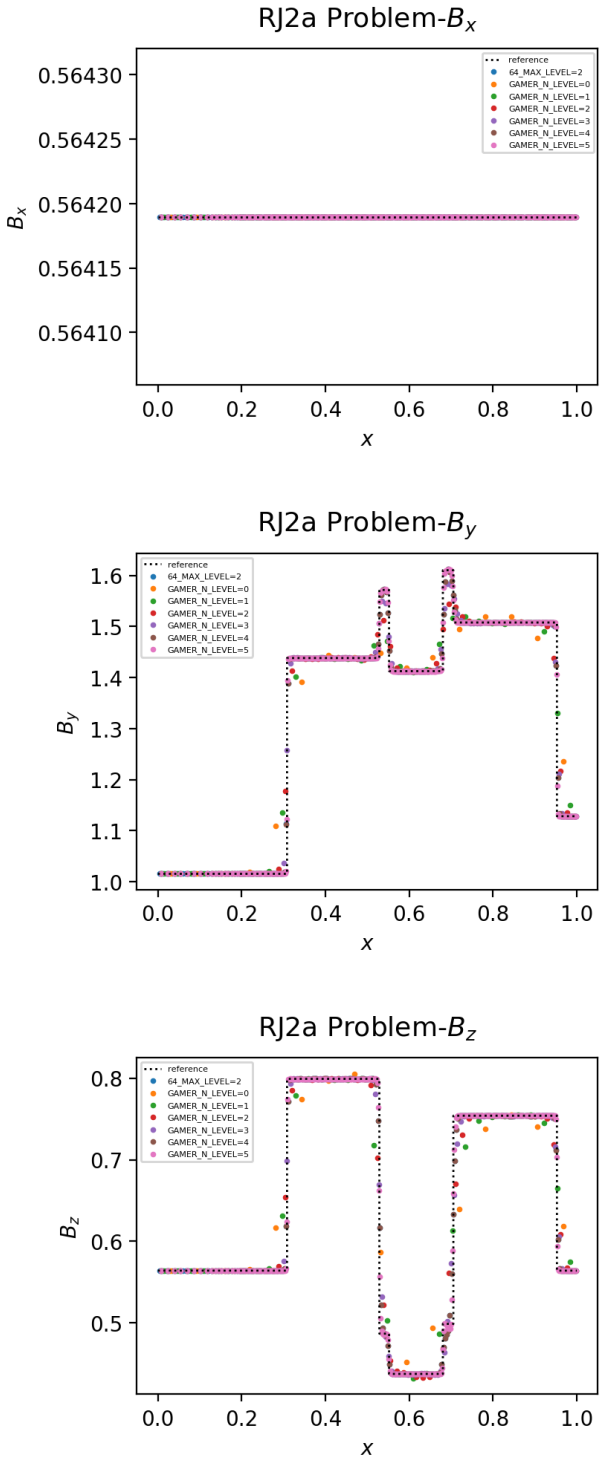

Figure 10: Magnetic Field of RJ2a Problem

Problem 3

Demonstrate that the CT(Constrainted Transport) update guarantees $\boldsymbol{\nabla \cdot B^{n+1} = \nabla \cdot B^n}$ in 3D.

From the relation between area-averaged magnetic field and time-and-line-averaged EMF:

$$\begin{cases} B^{n+1}_{x,i\pm\frac{1}{2},j,k} - B^{n}_{x,i\pm\frac{1}{2},j,k} = -\frac{\Delta t}{\Delta z}(\epsilon^{n+\frac{1}{2}}_{y,i\pm\frac{1}{2},j,k-\frac{1}{2}} - \epsilon^{n+\frac{1}{2}}_{y,i\pm\frac{1}{2},j,k+\frac{1}{2}}) - \frac{\Delta t}{\Delta y}(\epsilon^{n+\frac{1}{2}}_{z,i\pm\frac{1}{2},j+\frac{1}{2},k} - \epsilon^{n+\frac{1}{2}}_{z,i\pm\frac{1}{2},j-\frac{1}{2},k}) \\ B^{n+1}_{y,i,j\pm\frac{1}{2},k} - B^{n}_{y,i,j\pm\frac{1}{2},k} = -\frac{\Delta t}{\Delta x}(\epsilon^{n+\frac{1}{2}}_{z,i-\frac{1}{2},j\pm\frac{1}{2},k} - \epsilon^{n+\frac{1}{2}}_{z,i+\frac{1}{2},j\pm\frac{1}{2},k}) - \frac{\Delta t}{\Delta z}(\epsilon^{n+\frac{1}{2}}_{x,i,j\pm\frac{1}{2},k+\frac{1}{2}} - \epsilon^{n+\frac{1}{2}}_{x,i,j\pm\frac{1}{2},k-\frac{1}{2}}) \\ B^{n+1}_{z,i,j,k\pm\frac{1}{2}} - B^{n}_{z,i,j,k\pm\frac{1}{2}} = -\frac{\Delta t}{\Delta y}(\epsilon^{n+\frac{1}{2}}_{x,i,j-\frac{1}{2},k\pm\frac{1}{2}} - \epsilon^{n+\frac{1}{2}}_{x,i,j+\frac{1}{2},k\pm\frac{1}{2}}) - \frac{\Delta t}{\Delta x}(\epsilon^{n+\frac{1}{2}}_{y,i+\frac{1}{2},j,k\pm\frac{1}{2}} - \epsilon^{n+\frac{1}{2}}_{y,i-\frac{1}{2},j,k\pm\frac{1}{2}}) \end{cases} \tag{1}$$

along with the discretized form of:

$$\nabla \cdot B \Rightarrow \frac{1}{\Delta x}(B^n_{x,i+\frac{1}{2},j,k} - B^n_{x,i-\frac{1}{2},j,k}) + \frac{1}{\Delta y}(B^n_{y,i,j+\frac{1}{2},k} - B^n_{x,i,j-\frac{1}{2},k}) + \frac{1}{z}(B^n_{z,i,j,k+\frac{1}{2}} - B^n_{z,i,j,k-\frac{1}{2}}) \tag{2}$$

we have:

$$\begin{aligned}
\nabla \cdot B^{n+1} - \nabla \cdot B^n =& \frac{1}{\Delta x}(B^{n+1}_{x,i+\frac{1}{2},j,k} - B^n_{x,i+\frac{1}{2},j,k} - B^{n+1}_{x,i-\frac{1}{2},j,k} + B^n_{x,i-\frac{1}{2},j,k}) \\
&+ \frac{1}{\Delta y}(B^{n+1}_{y,i,j+\frac{1}{2},k} - B^n_{y,i,j+\frac{1}{2},k} - B^{n+1}_{y,i,j-\frac{1}{2},k} + B^n_{y,i,j-\frac{1}{2},k}) \\
&+ \frac{1}{\Delta z}(B^{n+1}_{z,i,j,k+\frac{1}{2}} - B^n_{z,i,j,k+\frac{1}{2}} - B^{n+1}_{z,i,j,k-\frac{1}{2}} + B^n_{z,i,j,k-\frac{1}{2}}) \\
&= -\frac{\Delta t}{\Delta x}(\frac{1}{\Delta z}(\epsilon^{n+\frac{1}{2}}_{y,i+\frac{1}{2},j,k-\frac{1}{2}} - \epsilon^{n+\frac{1}{2}}_{y,i+\frac{1}{2},j,k+\frac{1}{2}}) + \frac{1}{\Delta y}(\epsilon^{n+\frac{1}{2}}_{z,i+\frac{1}{2},j+\frac{1}{2},k} - \epsilon^{n+\frac{1}{2}}_{z,i+\frac{1}{2},j-\frac{1}{2},k}) \\
&\quad - \frac{1}{\Delta z}(\epsilon^{n+\frac{1}{2}}_{y,i-\frac{1}{2},j,k-\frac{1}{2}} - \epsilon^{n+\frac{1}{2}}_{y,i-\frac{1}{2},j,k+\frac{1}{2}}) - \frac{1}{\Delta y}(\epsilon^{n+\frac{1}{2}}_{z,i-\frac{1}{2},j+\frac{1}{2},k} - \epsilon^{n+\frac{1}{2}}_{z,i-\frac{1}{2},j-\frac{1}{2},k})) \\
&\quad - \frac{\Delta t}{\Delta y}(\frac{1}{\Delta x}(\epsilon^{n+\frac{1}{2}}_{z,i-\frac{1}{2},j+\frac{1}{2},k} - \epsilon^{n+\frac{1}{2}}_{z,i+\frac{1}{2},j+\frac{1}{2},k}) + \frac{1}{\Delta z}(\epsilon^{n+\frac{1}{2}}_{x,i,j+\frac{1}{2},k+\frac{1}{2}} - \epsilon^{n+\frac{1}{2}}_{x,i,j+\frac{1}{2},k-\frac{1}{2}}) \\
&\quad - \frac{1}{\Delta x}(\epsilon^{n+\frac{1}{2}}_{z,i-\frac{1}{2},j-\frac{1}{2},k} - \epsilon^{n+\frac{1}{2}}_{z,i+\frac{1}{2},j-\frac{1}{2},k}) - \frac{1}{\Delta z}(\epsilon^{n+\frac{1}{2}}_{x,i,j-\frac{1}{2},k+\frac{1}{2}} - \epsilon^{n+\frac{1}{2}}_{x,i,j-\frac{1}{2},k-\frac{1}{2}})) \\
&\quad - \frac{\Delta t}{\Delta z}(\frac{1}{\Delta y}(\epsilon^{n+\frac{1}{2}}_{x,i,j-\frac{1}{2},k+\frac{1}{2}} - \epsilon^{n+\frac{1}{2}}_{x,i,j+\frac{1}{2},k+\frac{1}{2}}) + \frac{1}{\Delta x}(\epsilon^{n+\frac{1}{2}}_{y,i+\frac{1}{2},j,k+\frac{1}{2}} - \epsilon^{n+\frac{1}{2}}_{y,i-\frac{1}{2},j,k+\frac{1}{2}}) \\
&\quad - \frac{1}{\Delta y}(\epsilon^{n+\frac{1}{2}}_{x,i,j-\frac{1}{2},k-\frac{1}{2}} - \epsilon^{n+\frac{1}{2}}_{x,i,j+\frac{1}{2},k-\frac{1}{2}}) - \frac{1}{\Delta x}(\epsilon^{n+\frac{1}{2}}_{y,i+\frac{1}{2},j,k-\frac{1}{2}} - \epsilon^{n+\frac{1}{2}}_{y,i-\frac{1}{2},j,k-\frac{1}{2}})) \\
&= -\frac{\Delta t}{\Delta z \Delta x}(\cancel{\epsilon^{n+\frac{1}{2}}_{y,i+\frac{1}{2},j,k-\frac{1}{2}}} - \cancel{\epsilon^{n+\frac{1}{2}}_{y,i+\frac{1}{2},j,k+\frac{1}{2}}} - \cancel{\epsilon^{n+\frac{1}{2}}_{y,i-\frac{1}{2},j,k-\frac{1}{2}}} + \cancel{\epsilon^{n+\frac{1}{2}}_{y,i-\frac{1}{2},j,k+\frac{1}{2}}} \\
&\qquad + \cancel{\epsilon^{n+\frac{1}{2}}_{y,i+\frac{1}{2},j,k+\frac{1}{2}}} - \cancel{\epsilon^{n+\frac{1}{2}}_{y,i-\frac{1}{2},j,k+\frac{1}{2}}} - \cancel{\epsilon^{n+\frac{1}{2}}_{y,i+\frac{1}{2},j,k-\frac{1}{2}}} + \cancel{\epsilon^{n+\frac{1}{2}}_{y,i-\frac{1}{2},j,k-\frac{1}{2}}}) \\
&\quad - \frac{\Delta t}{\Delta x \Delta y}(\cancel{\epsilon^{n+\frac{1}{2}}_{z,i+\frac{1}{2},j+\frac{1}{2},k}} - \cancel{\epsilon^{n+\frac{1}{2}}_{z,i+\frac{1}{2},j-\frac{1}{2},k}} - \cancel{\epsilon^{n+\frac{1}{2}}_{z,i-\frac{1}{2},j+\frac{1}{2},k}} + \cancel{\epsilon^{n+\frac{1}{2}}_{z,i-\frac{1}{2},j-\frac{1}{2},k}} \\
&\qquad + \cancel{\epsilon^{n+\frac{1}{2}}_{z,i-\frac{1}{2},j+\frac{1}{2},k}} - \cancel{\epsilon^{n+\frac{1}{2}}_{z,i+\frac{1}{2},j+\frac{1}{2},k}} - \cancel{\epsilon^{n+\frac{1}{2}}_{z,i-\frac{1}{2},j-\frac{1}{2},k}} + \cancel{\epsilon^{n+\frac{1}{2}}_{z,i+\frac{1}{2},j-\frac{1}{2},k}}) \\
&\quad - \frac{\Delta t}{\Delta y \Delta z}(\cancel{\epsilon^{n+\frac{1}{2}}_{x,i,j-\frac{1}{2},k+\frac{1}{2}}} - \cancel{\epsilon^{n+\frac{1}{2}}_{x,i,j+\frac{1}{2},k+\frac{1}{2}}} - \cancel{\epsilon^{n+\frac{1}{2}}_{x,i,j-\frac{1}{2},k-\frac{1}{2}}} + \cancel{\epsilon^{n+\frac{1}{2}}_{x,i,j+\frac{1}{2},k-\frac{1}{2}}} \\
&\qquad + \cancel{\epsilon^{n+\frac{1}{2}}_{x,i,j+\frac{1}{2},k+\frac{1}{2}}} - \cancel{\epsilon^{n+\frac{1}{2}}_{x,i,j+\frac{1}{2},k-\frac{1}{2}}} - \cancel{\epsilon^{n+\frac{1}{2}}_{x,i,j-\frac{1}{2},k+\frac{1}{2}}} + \cancel{\epsilon^{n+\frac{1}{2}}_{x,i,j-\frac{1}{2},k-\frac{1}{2}}}) \\
&= 0
\end{aligned} \tag{3}$$

which guarantees that $\nabla \cdot B$ is conserved for each updating, but one still need to choose the proper initial condition to realize $\nabla \cdot B = 0$.