Simulating Spherical Symmetric Fluid Dynamics using the High-order WENO Scheme

K. W. Wong

I. Table of Content

II. Introduction

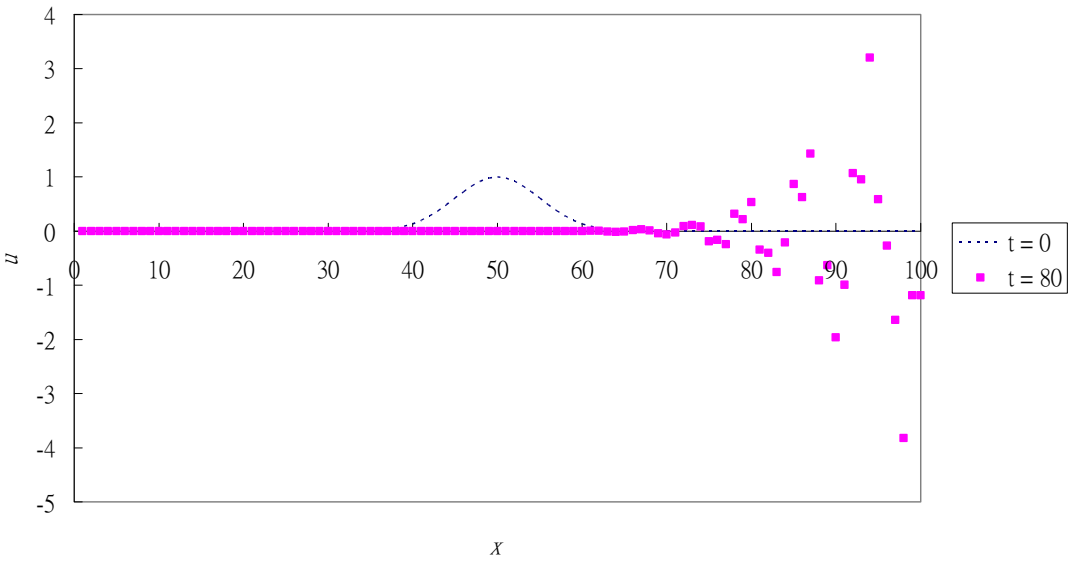
The purpose of my Physics Project is to use WENO (Weighted Essentially Non-Oscillatory) Scheme, which is a high order accurate finite difference or finite volume scheme designed for problems with piecewise smooth solutions containing discontinuities, to simulate spherical symmetric fluid dynamics. WENO schemes are based on ENO scheme, which were first introduced by Harten, Osher, Engquist, and Chakravarthy in the form of cell averages. The key idea of ENO schemes is to use the smoothest stencil among several candidates to approximate the fluxes at the cell boundaries to high order accuracy, and at the same time to avoid spurious oscillations near shocks. This program is ready to be modified to solve the spherical symmetric fluid dynamics in a star. Before I discuss the fluid dynamics simulated by WENO Scheme in this report, I will briefly discuss some basic schemes and study some toy models using the WENO Scheme.

III. Finite Difference Approximation

Consider the linear advection equation $u_t + au_x = 0$ in the domain $[0,L]\times[0,T]$ on the $x-t$ plane. Divide $[0,L]$ by $\frac{L}{\Delta x}+1$ grid points which are equally spaced. For u at $x = i\Delta x$ and $t = n\Delta t$, it is denoted as u_i^n. Now, $u_t = \frac{\partial u}{\partial t} \approx \frac{u_i^{n+1} - u_i^n}{\Delta t}$, and $u_x = \frac{\partial u}{\partial x} \approx \frac{u_{i+1}^n - u_{i-1}^n}{2\Delta x}$. The linear advection equation can be approximated by $u_i^{n+1} = u_i^n - \frac{1}{2}c\left(u_{i+1}^n - u_{i-1}^n\right)$, where $c = \frac{a}{\Delta x/\Delta t}$ is a dimensionless quantity known as the Courant number. The values of u at $t = (n+1)\Delta t$ can be found by the values of u at an earlier time of $t = n\Delta t$. However, it is disappointing that the scheme above is totally useless because it is unconditionally unstable.

I have tried to use Finite Difference Approximation to solve the linear advection equation with $a = 1$ for Gaussian initial condition. Fig 1 shows the values of u at different x for $t = 0$ and $t = 80$.

Fig 1. Solving Linear Advection Equation Using Finite Difference Approximation



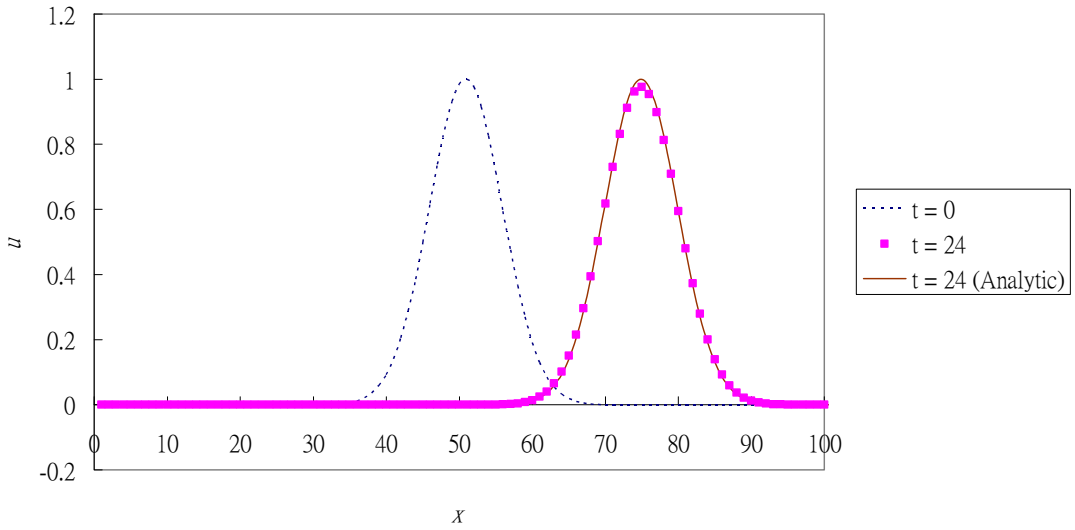
The curve for $t = 80$ should move to the right by 80 units (because I have set $a = 1$), that is the values of u for all x should be 0. But it turns out that there are some oscillations, which indicate the scheme is unstable.

IV. First Order Upwind Scheme

In Finite Difference Approximation, we approximated u_x by $\frac{u_{i+1}^n - u_{i-1}^n}{2\Delta x}$, which should have second order of accuracy. However, if we approximate u_x by $\frac{u_i^n - u_{i-1}^n}{\Delta x}$, which has first order of accuracy, the scheme is stable for $a > 0$ and $0 \le c \le 1$. The linear advection equation can then be approximated by $u_i^{n+1} = u_i^n - c\left(u_i^n - u_{i-1}^n\right)$.

I have tried to use First Order Upwind Scheme to solve linear advection equation for Gaussian initial condition. Fig 2 shows the values of u at different x for $t = 0$ and $t = 24$.

Fig 2. Solving Linear Advection Equation
Using First Order Upwind Scheme with c = 0.5 (dx = 0.1, dt = 0.05)



The shape of the Gaussian is preserved, but it seems to be flattened. The numerical data points of u at x in a range of $72 \le x \le 78$ are lower than the analytical results, while in ranges of $60 \le x \le 68$ and $82 \le x \le 90$ are little bit higher than the analytical results. The scheme is stable, but not quite accurate. This might be due to the first order of accuracies in both space and time domains.

V. Runge-Kutta Time Discretization

In the last test, I have used First Order Upwind Scheme together with First Order Time Discretization. The result is not quiet accurate because both space and time domains have only first order of accuracies. I wondered if this can be improved by replacing the First Order Time Discretization by Runge-Kutta Time Discretization, which has third order of accuracy.

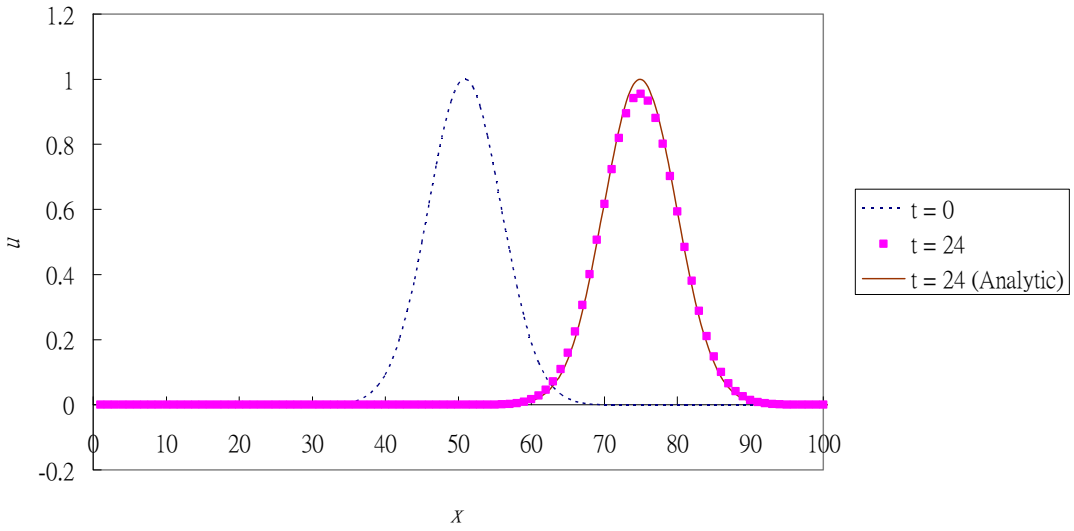
First, write u_t as a subject of the equation, then $u_t = L(u)$. For linear advection equation, $L(u) = -au_x$. Now, u^{n+1} can be calculated by using u^n:

$$u^{(1)} = u^n + \Delta t L\left(u^n\right)$$
$$u^{(2)} = \frac{3}{4}u^n + \frac{1}{4}u^{(1)} + \frac{1}{4}\Delta t L\left(u^{(1)}\right).$$
$$u^{n+1} = \frac{1}{3}u^n + \frac{2}{3}u^{(2)} + \frac{2}{3}\Delta t L\left(u^{(2)}\right)$$

However, when I tried to solve the linear advection equation for Gaussian initial condition using First Order Upwind Scheme together with Runge-Kutta Time

Discretization, the result seems not to be improved. Fig 3 shows the values of u at different x for $t = 0$ and $t = 24$.

Fig 3. Solving Linear Advection Equation Using First Order Upwind Scheme and Runge-Kutta time discretization (dx = 0.1, dt = 0.05)



The curve for $t = 24$ is again flattened. The numerical data points of u at x in a range of $72 \le x \le 78$ are again lower than the analytical results, while in ranges of $60 \le x \le 68$ and $82 \le x \le 90$ are again little bit higher than the analytical results. This might due to the fact that the error is dominated by the error in spatial dimension, which has first order of accuracy.

VI. WENO Scheme

To improve, a scheme of higher order of accuracy for spatial discretization must be used. I selected WENO (Weighted Essentially Non-Oscillatory) Scheme with two reasons. First, it has fifth order of accuracy. Second, it is easier to code for solving the spherical symmetric fluid dynamics in a star in the next semester.

Consider one dimensional conservation law $u_t + f(u)_x = 0$ ($f(u) = au$ for linear advection equation). First, write u_t as a subject of the equation so that we can use the Runge-Kutta Time Discretization. In doing so, we have $L(u) = -f(u)_x$. Next we approximate $f(u)_x$ by $\frac{1}{\Delta x}\left(\hat{f}_{i+\frac{1}{2}} - \hat{f}_{i-\frac{1}{2}} \right)$, where $\hat{f}_{i+\frac{1}{2}}$ is called the numerical flux. The numerical fluxes can be found by the following algorithm.

First, calculate the Roe speed $\bar{a}_{i+\frac{1}{2}} \equiv \dfrac{f(u_{i+1}) - f(u_i)}{u_{i+1} - u_i}$ for all i.

If $\bar{a}_{i+\frac{1}{2}} \geq 0$ for all i, then

$$\hat{f}_{i+\frac{1}{2}} = v^{-}_{i+\frac{1}{2}} = \sum_{r=0}^{2} \omega_r v^{(r)}_{i+\frac{1}{2}} ,$$

where

$$v^{(r)}_{i+\frac{1}{2}} = \sum_{j=0}^{2} c_{rj} \bar{v}_{i-r+j} ,$$

$$\bar{v}_i = f(u_i),$$

$$c_{rj} = \sum_{m=j+1}^{3} \frac{\sum_{\substack{l=0 \\ l\neq m}}^{3} \prod_{\substack{q=0 \\ q\neq m,l}}^{3} (r-q+1)}{\prod_{\substack{l=0 \\ l\neq m}}^{3} (m-l)},$$

$$\omega_r = \frac{\alpha_r}{\sum_{s=0}^{2} \alpha_s},$$

$$\alpha_r = \frac{d_r}{(\varepsilon + \beta_r)^2},$$

$$\begin{cases} d_0 = 3/10 \\ d_1 = 3/5 \\ d_2 = 1/10 \end{cases},$$

$$\begin{cases} \beta_0 = \dfrac{13}{12}(\bar{v}_i - 2\bar{v}_{i+1} + \bar{v}_{i+2})^2 + \dfrac{1}{4}(3\bar{v}_i - 4\bar{v}_{i+1} + \bar{v}_{i+2})^2 \\ \beta_1 = \dfrac{13}{12}(\bar{v}_{i-1} - 2\bar{v}_i + \bar{v}_{i+1})^2 + \dfrac{1}{4}(\bar{v}_{i-1} - \bar{v}_{i+1})^2 \\ \beta_2 = \dfrac{13}{12}(\bar{v}_{i-2} - 2\bar{v}_{i-1} + \bar{v}_i)^2 + \dfrac{1}{4}(\bar{v}_{i-2} - 4\bar{v}_{i-1} + 3\bar{v}_i)^2 \end{cases} \quad \text{and}$$

$\varepsilon = 10^{-6}$ to avoid the denominator of α_r to become zero.

If $\bar{a}_{i+\frac{1}{2}} < 0$ for all i, then

$$\hat{f}_{i+\frac{1}{2}} = v^{+}_{i+\frac{1}{2}} = \sum_{r=0}^{2} \tilde{\omega}_r v^{(r)}_{i-\frac{1}{2}} ,$$

where

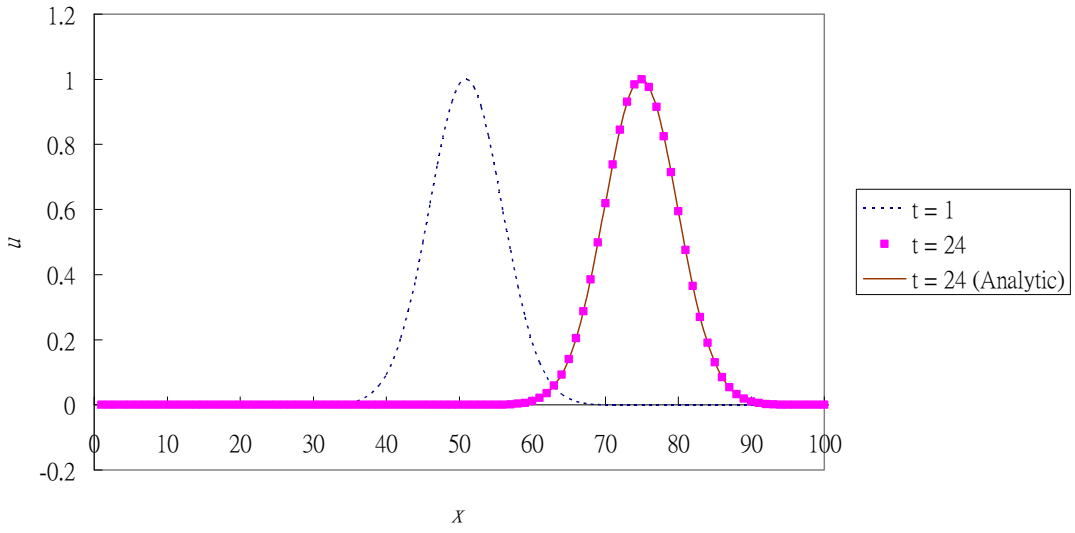
$$\tilde{\omega}_r = \frac{\tilde{\alpha}_r}{\sum_{s=0}^{2} \tilde{\alpha}_s},$$

$$\tilde{\alpha}_r = \frac{\tilde{d}_r}{(\varepsilon + \beta_r)^2}$$ and

$$\tilde{d}_r = d_{2-r}.$$

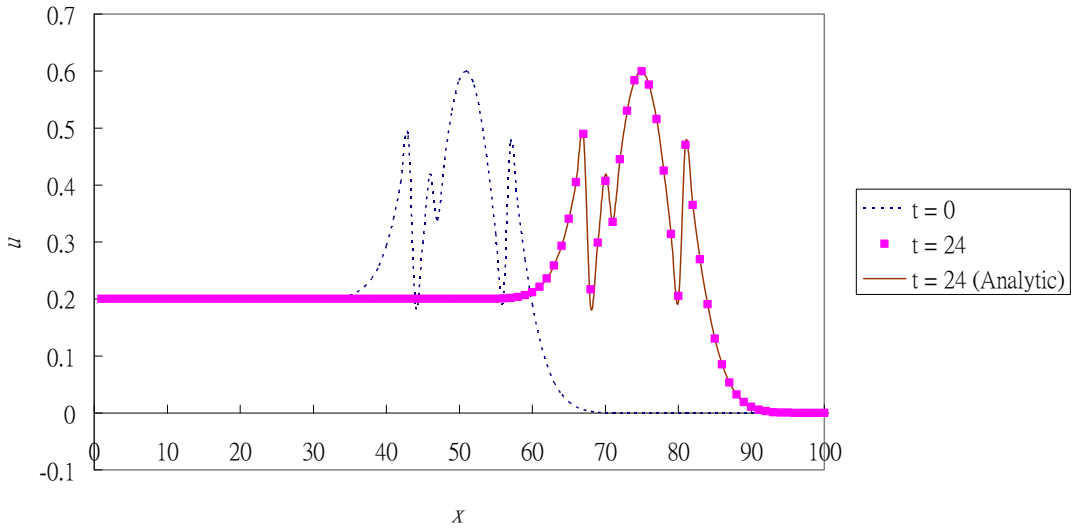
When I apply WENO Scheme together with Runge-Kutta Time Discretization to solve linear advection equation, the result is unimaginably accurate! Fig 4 shows the values of u at different x for $t = 0$ and $t = 24$.

Fig 4. Solving Linear Advection Equation Using WENO Scheme with Runge-Kutta Time Discretization (dx = 0.1, dt = 0.05)



The numerical data points are placed right at the curve of the analytical results. When I looked at the data file, I found the numerical data is exactly the same as the analytical result! I wondered if this is the case only for the smooth initial condition. Therefore I introduce some sharp jumps to the initial condition. The numerical result is no longer exactly the same as the analytical one, but it is still very accurate! Fig 5 shows the values of u at different x for $t = 0$ and $t = 24$, with a broken Gaussian initial condition.

Fig 5. Solving Linear Advection Equation Using WENO Scheme
with Runge-Kutta Time Discretization (dx = 0.1, dt = 0.05)



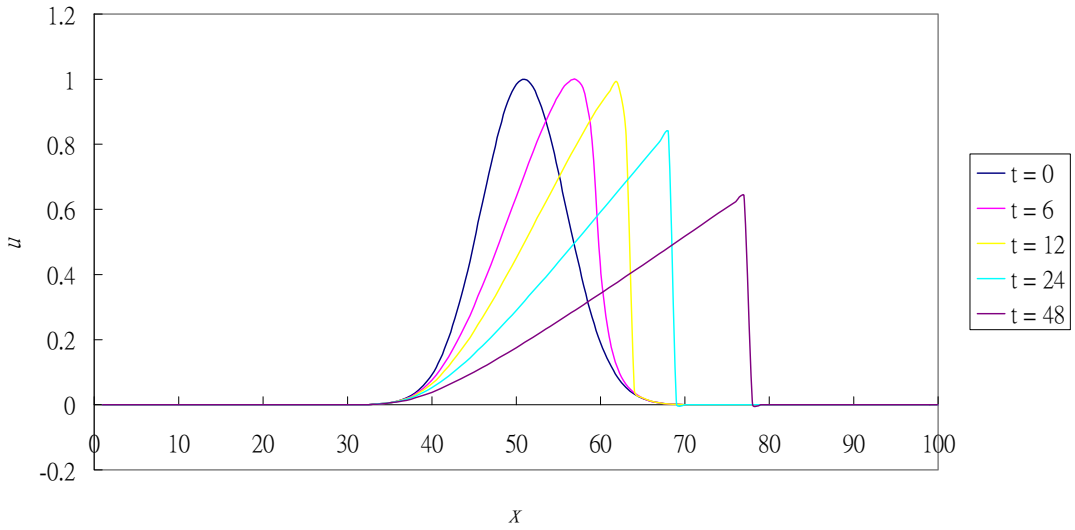
Originally, the initial data is a Gaussian. Then I moved the curve up by 0.2 units for $x \le 46$ and moved the curve down by 0.4 units for $44 \le x \le 56$. This made three jumps to the initial condition. At $t = 24$, the broken Gaussian should move to the right by 24 units (because I have set $a = 1$), and the numerical data are quite closed to the analytical results. The scheme is stable and accurate!

Now, it is the time to consider non-linear equation. The simplest one is the inviscid Burgers equation $u_t + uu_x = 0$. In conservation-law form, the equation reads $u_t + f(u)_x = 0$, with $f(u) = \frac{u^2}{2}$.

For inviscid Burgers equation, the curve is no long move to the right with the same speed at different position and different time. Comparing the inviscid Burgers equation with the linear advection equation, we will know the speed of the curve at a position x is the value of u at that position x at that particular time t. The speed of the curve at different x and different t will change. Therefore, the shape of the curve will change. Shocks may be produced in this equation.

I have tried to use WENO Scheme together with the Runge-Kutta Time Discretization to solve the inviscid Burgers equation with Gaussian initial condition. Fig 6 shows the values of u at different x for $t = 0$, $t = 6$, $t = 12$, $t = 24$ and $t = 48$.

Fig 6. Solving Inviscid Burgers' Equation Using WENO Scheme
with Runge-Kutta Time Discretization (dx = 0.1, dt = 0.05)

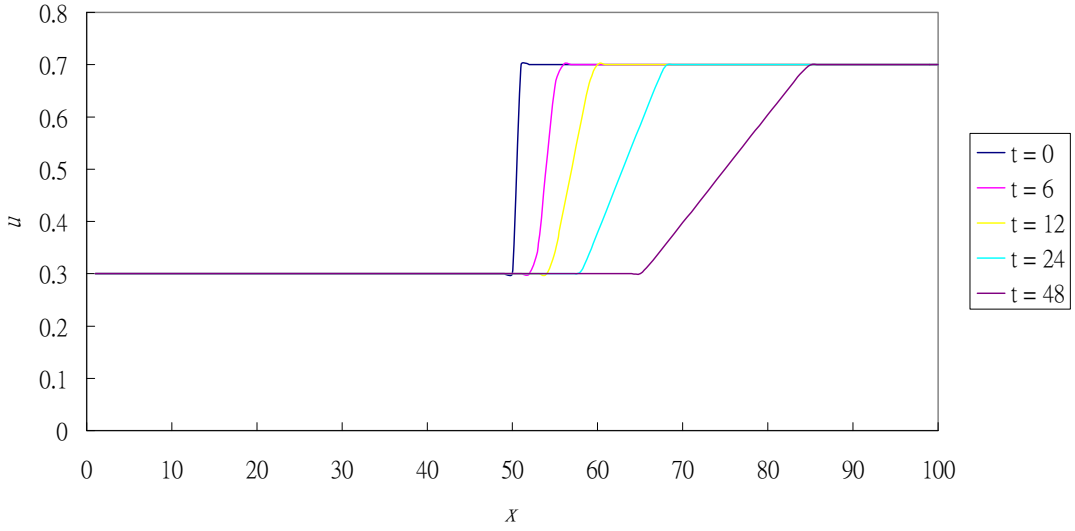


At $t = 0$, for $x \geq 51$, the values of u on the left are larger that that of the right. The wave on the left goes to the right with a speed faster than that of the right. Therefore, a shock should be produced.

It can be seen that a shock wave is produced without oscillations using WENO Scheme. The scheme is stable in producing shocks.

Then I have tried to solve the inviscid Burgers equation using WENO Scheme with another initial condition, Riemann initial condition. For $x \leq 50$, I have set $u = 0.3$; for $x \geq 51$, I have set $u = 0.7$. Fig 7 shows the values of u at different x for $t = 0$, $t = 6$, $t = 12$, $t = 24$ and $t = 48$.

Fig 7. Solving Inviscid Burgers' Equation Using WENO Scheme
with Runge-Kutta Time Discretization (dx = 0.1, dt = 0.05)

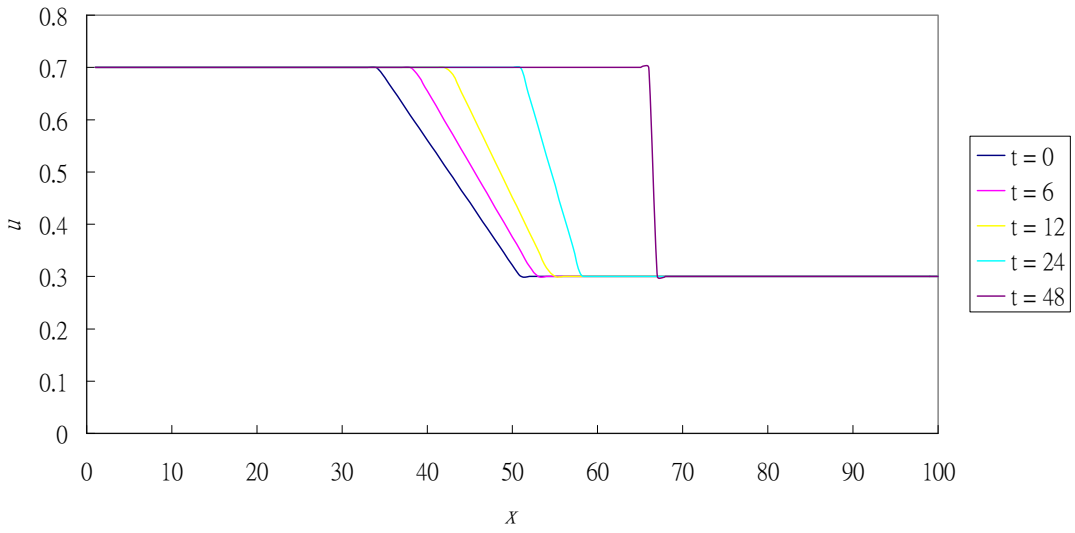


The value of u on the right is larger than that on the left. The speed of the wave on the right goes to the right with a speed larger than that of the left. Therefore, a rarefaction shock is produced in between.

It can be seen that a rarefaction shock is produced without oscillations using WENO Scheme. The scheme is also stable in producing rarefaction shocks.

Next, I have tried to see whether the program is able to produce shocks from rarefaction shocks. I have set $u = 0.7$ for $x \le 33$ and $u = 0.3$ for $x \ge 51$, and made a rarefaction in between. Fig 8 shows the values of u at different x for $t = 0$, $t = 6$, $t = 12$, $t = 24$ and $t = 48$.

Fig 8. Solving Inviscid Burgers' Equation Using WENO Scheme with Runge-Kutta Time Discretization (dx = 0.1, dt = 0.05)

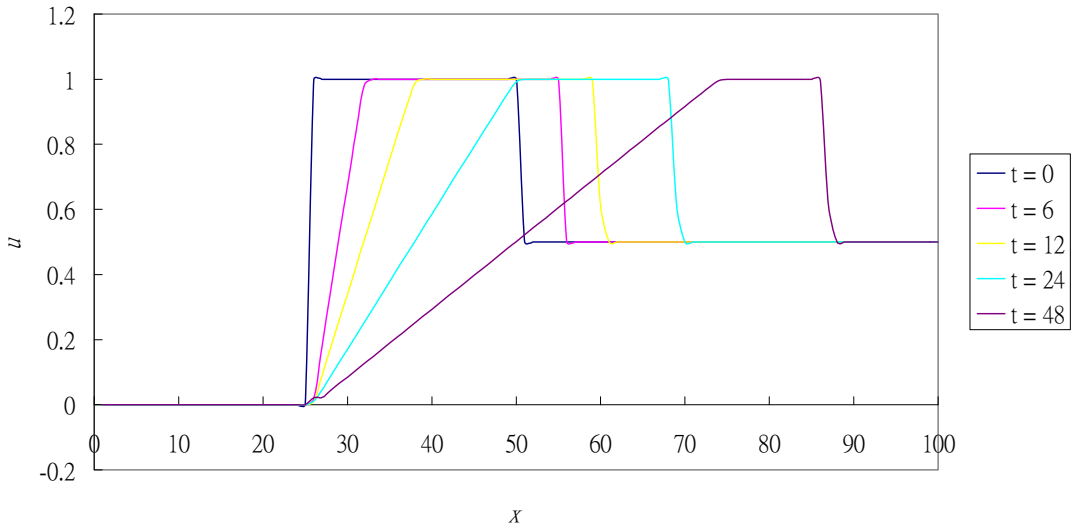


The values of u on the left of the rarefaction are larger than that of the right of the rarefaction. The wave on the left of the rarefaction goes to the right with a speed faster than that of the right of the rarefaction. Therefore, the width of the rarefaction contracts and become a shock at last.

Shock is produced from the rarefaction shock without oscillations using WENO Scheme. The scheme is stable in producing shocks from rarefaction shocks.

Then, I have tried to produce shock and rarefaction shock in the same case to see whether the program work here. I have set $u = 0$ for $x \le 25$, $u = 1$ for $26 \le x \le 50$ and $u = 0.5$ for $x \ge 51$. Fig 9 shows the values of u at different x for $t = 0$, $t = 6$, $t = 12$, $t = 24$ and $t = 48$.

Fig 9. Solving Inviscid Burgers' Equation Using WENO Scheme
with Runge-Kutta Time Discretization (dx = 0.1, dt = 0.05)



The values of u on the left are smaller than that of the middle, and that of the middle are larger than that on the right. The wave in the middle goes to the right with a speed faster than that of both of the left and right. Therefore a rarefaction shock should be produced between the left and the middle, while a shock should be produced between the middle and the right.
The program is successful to produce shock and rarefaction shock here. The scheme is stable for producing shock and rarefaction shock in the same case.

VII. Flux Splitting

In the WENO Scheme, it requires $f'(u)$ having the same sign for the whole spatial domain for the program to work. For the linear advection equation, $f(u) = au$, the requirement is automatically satisfied because a is a constant. For the inviscid Burgers equation, $f(u) = \frac{u^2}{2}$, the requirement is u having the same sign for all x. In all the previous tests, the requirements are fulfilled. The programs work. However, in the reality, it is not always the case that $f'(u)$ has the same sign for all x. It is necessary to develop a program for general cases.
One of the methods is to split the flux into two parts:

$$f(u) = f^{+}(u) + f^{-}(u),$$

where

$$\frac{df^{+}(u)}{du} \geq 0, \qquad \frac{df^{-}(u)}{du} \leq 0.$$

Here, the positive and negative fluxes $f^{\pm}(u)$ should have as many derivatives as the order of the scheme, i.e. fifth. This unfortunately rules out many popular flux splittings. The simplest smooth splitting is the Lax-Friedrichs splitting:

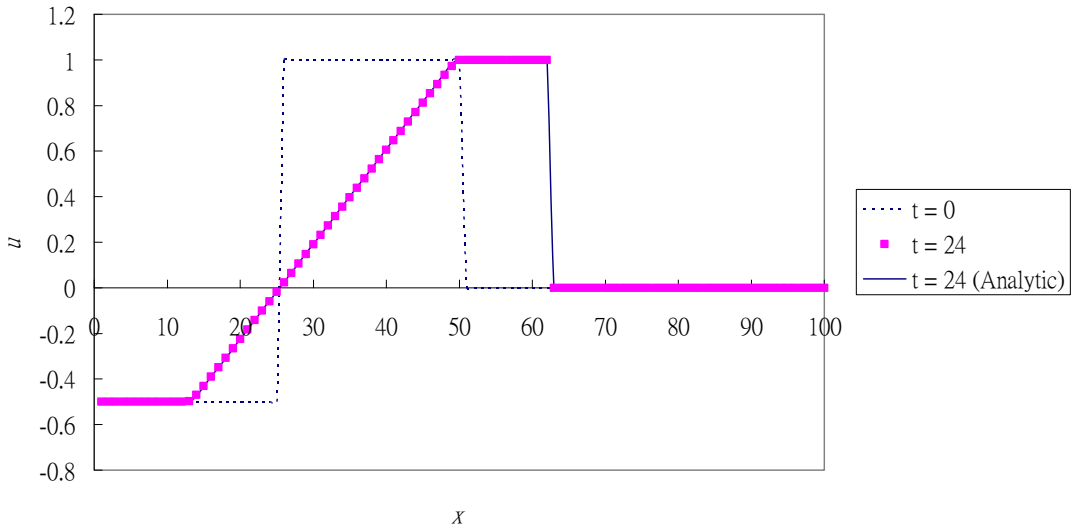
$$f^{\pm}(u)=\frac{1}{2}[f(u)\pm \alpha u],$$

where α is taken as $\alpha=\max_u|f'(u)|$ over the relevant range of u.

With the positive and negative fluxes, we can apply the WENO reconstruction procedures to each of them separately to obtain two numerical fluxes $\hat{f}^{+}_{i+\frac{1}{2}}$ and $\hat{f}^{-}_{i+\frac{1}{2}}$, and then sum them to get the numerical flux $\hat{f}_{i+\frac{1}{2}}$.

I have used this method to solve the inviscid Burgers equation with the Riemann initial condition. I have set $u=-0.5$ for $x\le 25$, $u=1$ for $26\le x\le 50$ and $u=0$ for $x\ge 51$. Now, $f'(u)=u$ is no longer single sign for all x. Without flux splitting, WENO Scheme cannot be applied. Fig 10 shows the values of u at different x for $t=0$ and $t=24$ using WENO Scheme with flux splitting.

Fig 10. Solving Inviscid Burgers' Equation Using WENO Scheme with Runge-Kutta Time Discretization (dx = 0.1, dt = 0.05)



The values of u on the left are negative and that in the middle are positive. The wave on the left moves to the left while that of the middle moves to the right, creating a rarefaction in between.

Here, I also plotted the analytical result for $t=24$ so that I can compare the

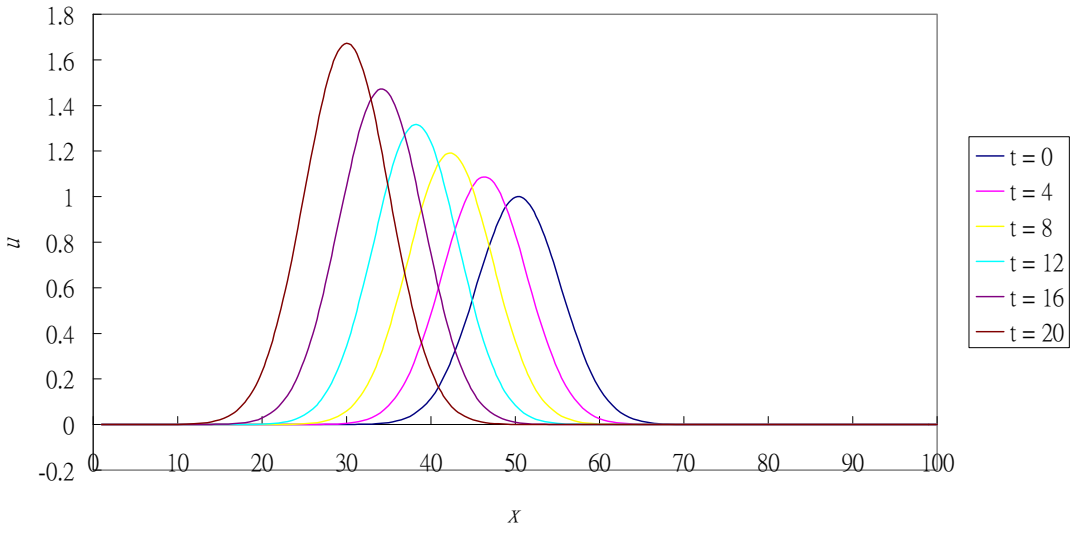
numerical result with the analytical result to see how accurate the scheme is. The program works very nice! The numerical result is extremely closed to the analytical result. WENO Scheme is stable and accurate even with the existence of shocks and rarefaction shocks.

VIII. Source Term

For the linear advection equation $u_t + au_x = 0$, the right hand side is zero. If it is not zero, say u/x, then the equation contains a source term. It is very important to know how to write a program to solve the equations with source terms because the equations for the spherical symmetric fluid dynamics contain source terms.

I have used the WENO Scheme to solve the linear advection equation with a source term u/x, with $a = -1$ (since the source will be larger for small x, I made the wave to go to the left). I have also set the boundary condition that $u(x = 0) = 0$ to avoid the diverging source at $x = 0$. Fig 11 shows the values of u at different x for various time.

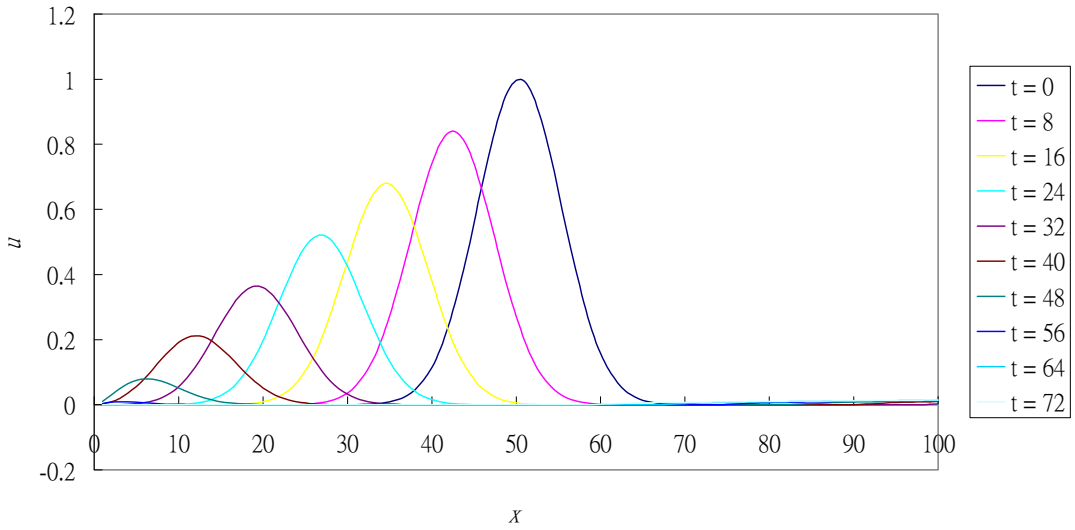
Fig 11. Solving Linear Advection Equation with Source Term u / x
Using WENO Scheme with Runge-Kutta Time Discretization (dx = 0.1, dt = 0.05)



Both u and x are positive, the source is positive. The peak is getting higher and higher. The increasing rate of the Gaussian is also increasing because the source is larger for larger u and smaller x.

Now, let's consider the equation with the source term $-u/x$. The source is negative. The peak of the Gaussian should decrease. Again, I have also set the boundary condition that $u(x = 0) = 0$ to avoid the diverging source at $x = 0$. Fig 12 shows the values of u at different x for various time.

Fig 12. Solving Linear Advection Equation with Source Term - u / x
Using WENO Scheme with Runge-Kutta Time Discretization (dx = 0.1, dt = 0.05)



The source is negative, the peak of the Gaussian decreases, as predicted.

IX. One-dimensional Fluid Dynamics in Cartesian Coordinates

Now it is ready to develop a program to solve the fluid dynamics. The equations for fluid dynamics in three dimensions are:

$$\begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ \tau \end{bmatrix}_t + \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uw \\ u(\tau + p) \end{bmatrix}_x + \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho vw \\ v(\tau + p) \end{bmatrix}_y + \begin{bmatrix} \rho w \\ \rho uw \\ \rho vw \\ \rho w^2 + p \\ w(\tau + p) \end{bmatrix}_z = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix},$$

where u, v and w are velocities in x, y and z directions, respectively, and τ is the total energy density, defined by

$$\tau = \rho\varepsilon + \frac{1}{2}\rho\mathbf{v}^2,$$

in which ε is the internal energy per unit mass of the fluid and $\mathbf{v}$ is the velocity vector of the fluid element defined by

$$\mathbf{v} = u\hat{\mathbf{x}} + v\hat{\mathbf{y}} + w\hat{\mathbf{z}}.$$

Consider the fluid dynamics in x direction only, the equations become:

$$\begin{bmatrix} \rho \\ \rho u \\ \tau \end{bmatrix}_t + \begin{bmatrix} \rho u \\ \rho u^2 + p \\ u(\tau + p) \end{bmatrix}_x = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix},$$

where u is the velocity in x direction and $\tau = \rho\varepsilon + \frac{1}{2}\rho u^2$.

To complete the program, we need the equation of state. Here I have used a polytropic equation of state $p = k\rho^\gamma$ with $\varepsilon = \frac{k}{\gamma - 1}\rho^{\gamma-1}$, which gives:

$$p = (\gamma - 1)\rho\varepsilon .$$

In order to test the program, I have performed five Riemann problem tests. The initial data of those tests are in the format:

$$\rho = \begin{cases} \rho_L & 0 \le x \le x_0 \\ \rho_R & x_0 < x \le 1 \end{cases}, \quad u = \begin{cases} u_L & 0 \le x \le x_0 \\ u_R & x_0 < x \le 1 \end{cases}, \quad p = \begin{cases} p_L & 0 \le x \le x_0 \\ p_R & x_0 < x \le 1 \end{cases}.$$

The initial data are:

Test	ρ_L	u_L	p_L	x_0	ρ_R	u_R	p_R
1	1.0	0.75	1.0	0.3	0.125	0.0	0.1
2	1.0	-2.0	0.4	0.5	1.0	2.0	0.4
3	1.0	0.0	1000.0	0.5	1.0	0.0	0.01
4	5.99924	19.5975	460.894	0.4	5.99242	-6.19633	46.0950
5	1.0	-19.59745	1000.0	0.8	1.0	-19.59745	0.01

In those tests, I have set $\gamma = 1.4$ and the boundaries are free.

My results will be compared with the results in [4] which use other schemes.

Test 1

At $t = 0.2$, the density, velocity, pressure and internal energy versus position calculated using WENO Scheme are plotted in Fig 13, 14, 15 and 16, respectively.

Fig 13. Density vs Position (Test 1)

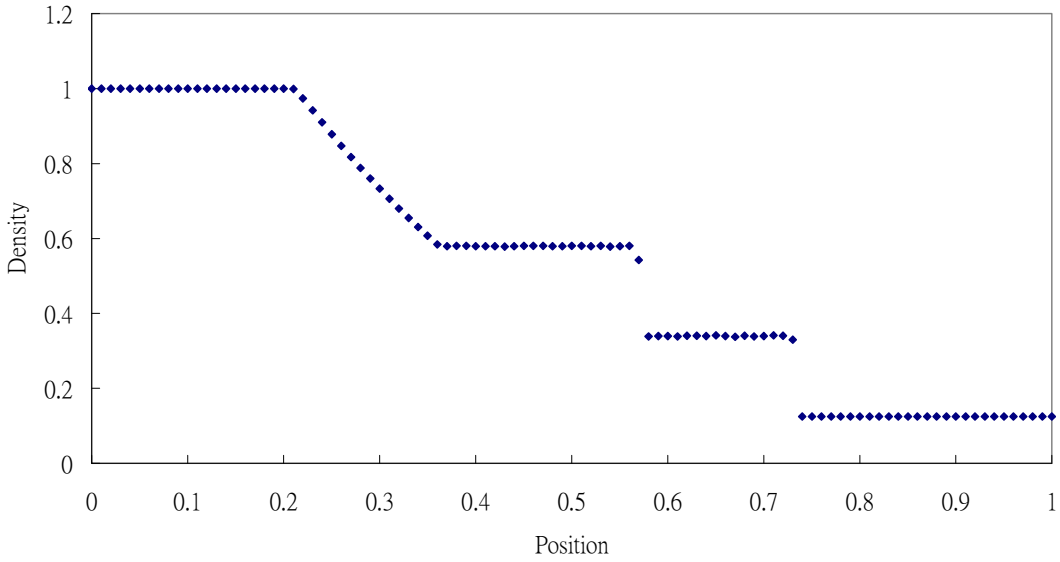


Fig 14. Velocity vs Position (Test 1)

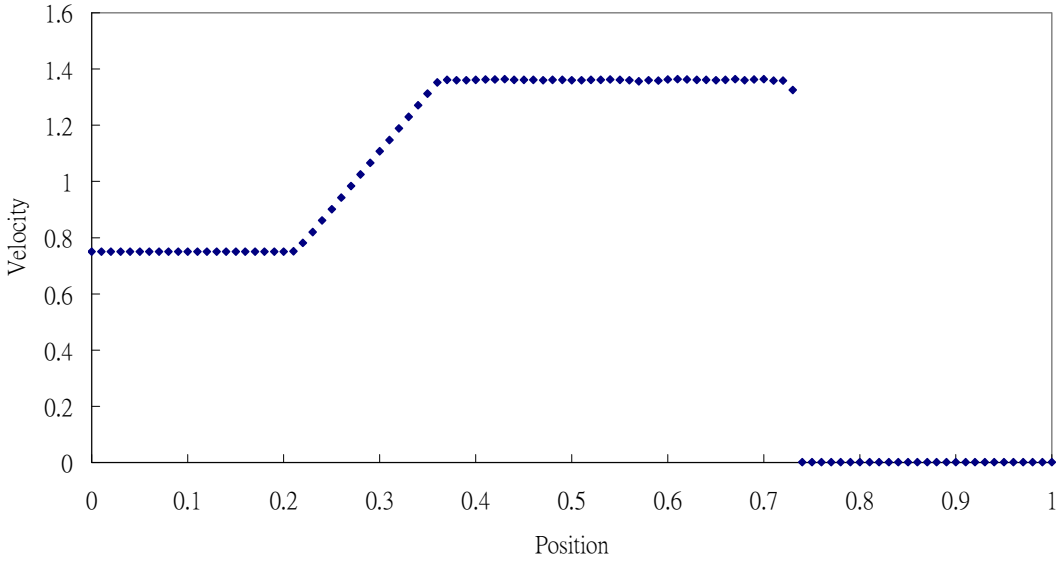


Fig 15. Pressure vs Position (Test 1)

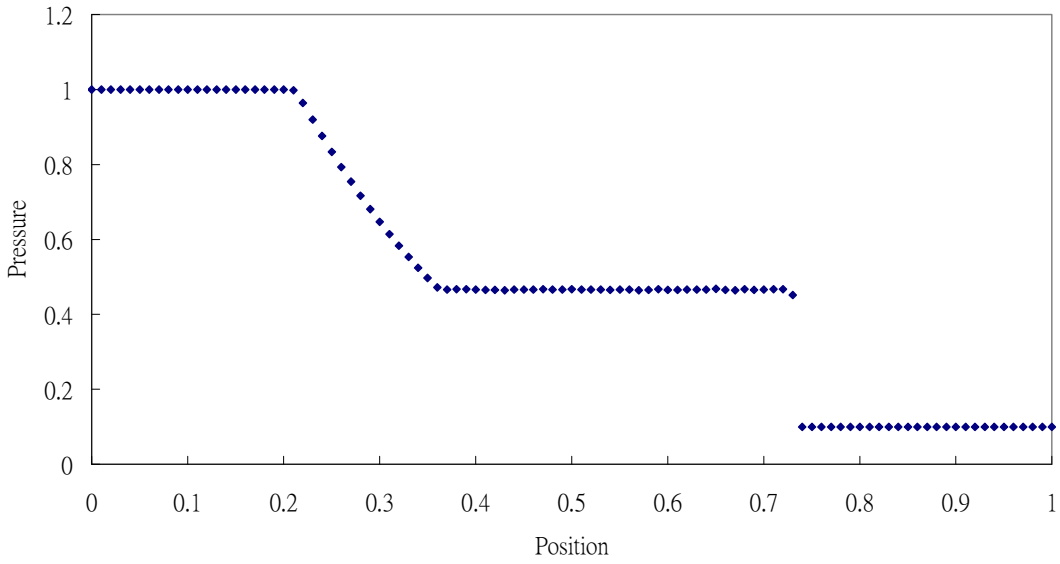
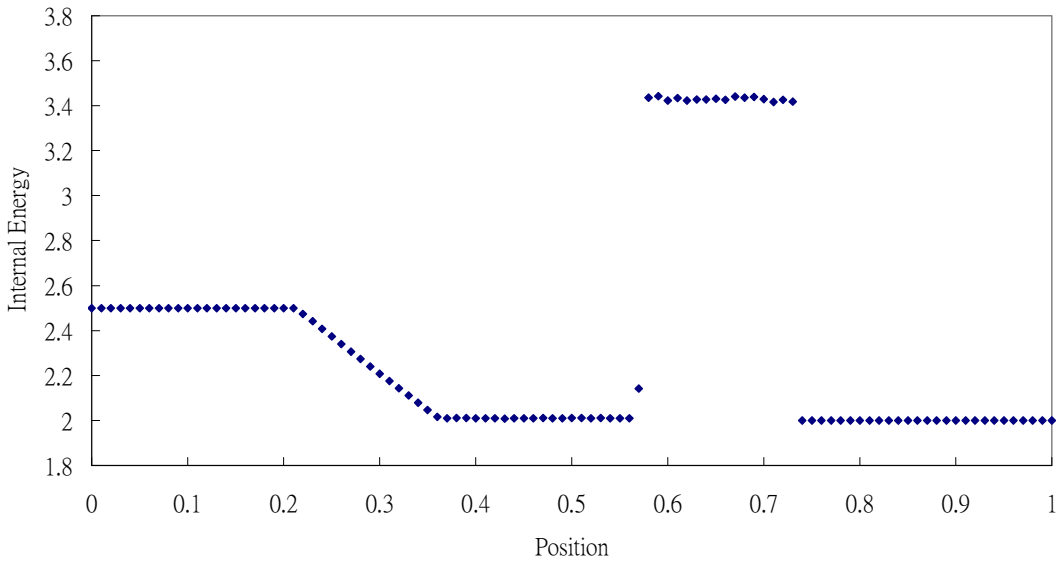
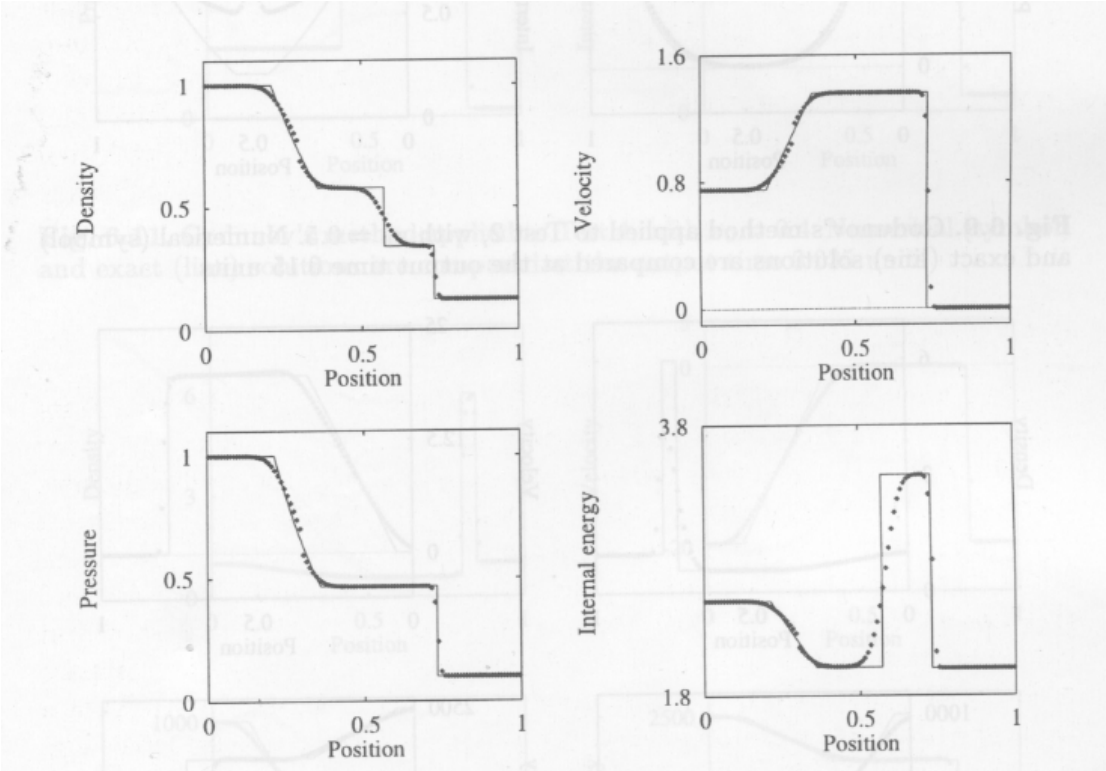


Fig 16. Internal Energy vs Position (Test 1)



The results for density, velocity, pressure and internal energy versus position calculated using the second-order Richtmyer (or two-step Lax-Wendraff) method are plotted in Fig 17.

Fig 17. Results for Test 1 Using Second-order Richtmyer Method (Picture adopted in [4])



The dots are the numerical results using the second-order Richtmyer method and the solid lines are the analytical results using the program provided in NUMERICA, a library of source codes.

In the density profile in Fig 17, the numerical data is broken at $x \approx 0.3$. The second-order Richtmyer method fails but WENO Scheme is successful to produce continuous data (Fig 13), which is the case for analytical result. At $x \approx 0.6$, there is a sharp jump in density for analytical result. The second-order Richtmyer method fails to simulate this jump while WENO Scheme successes.

In the velocity profile in Fig 17, the numerical data for the second-order Richtmyer method fails to produce smooth data at $x \approx 0.3$ but the WENO Scheme is successful (Fig 14), which is the case for analytical result.

In the pressure profile in Fig 17, the numerical data for the second-order Richtmyer method fails to produce continuous data at $x \approx 0.3$ while the analytical data should be continuous. WENO Scheme is able to produce this continuous data (Fig 15).

In the internal energy profile in Fig 17, again the numerical data for the second-order Richtmyer method fails to produce the smooth data at $x \approx 0.3$ while it should be smooth. It also cannot simulate the energy at $0.5 < x < 0.7$ while it should be a sharp increase at $x \approx 0.6$. However, WENO Scheme is

successful to produce continuous data at $x \approx 0.3$, just like the analytical result, and it can simulate the sharp increase in internal energy at $x \approx 0.6$.

Test 2

At $t = 0.15$, the density, velocity, pressure and internal energy versus position calculated using WENO Scheme are plotted in Fig 18, 19, 20 and 21, respectively.

Fig 18. Density vs Position (Test 2)

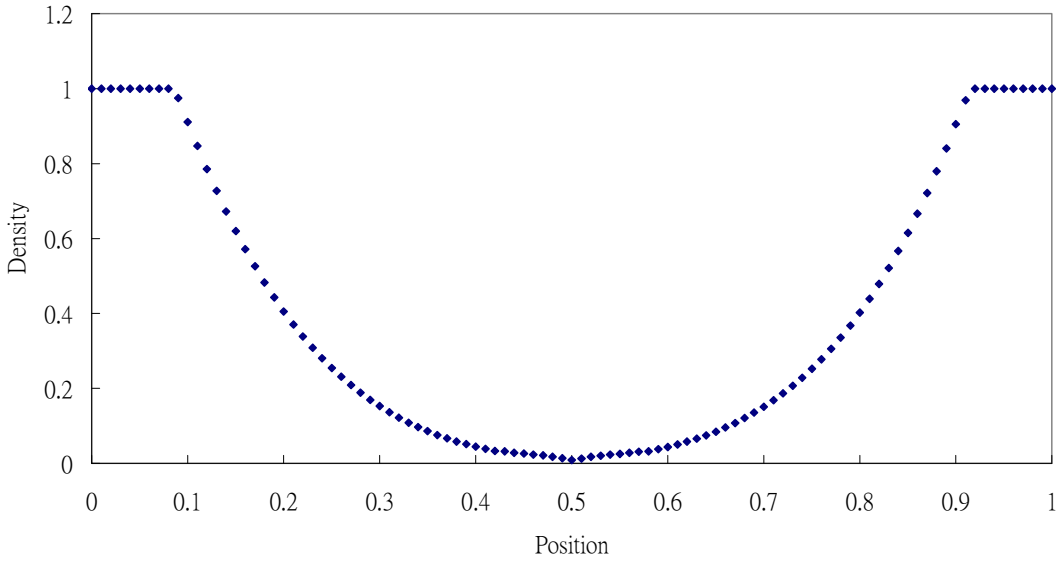


Fig 19. Velocity vs Position (Test 2)

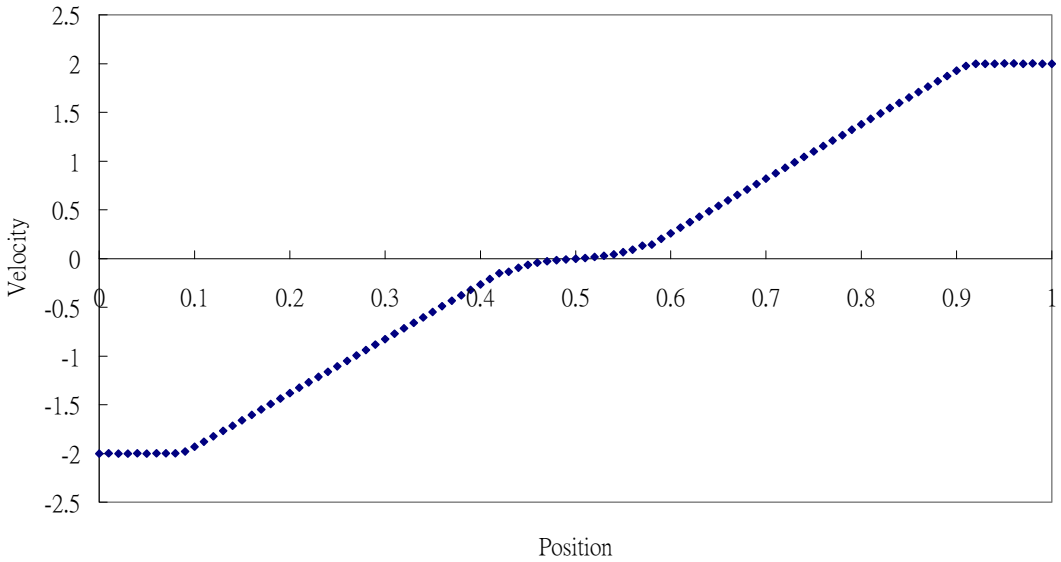


Fig 20. Pressure vs Position (Test 2)

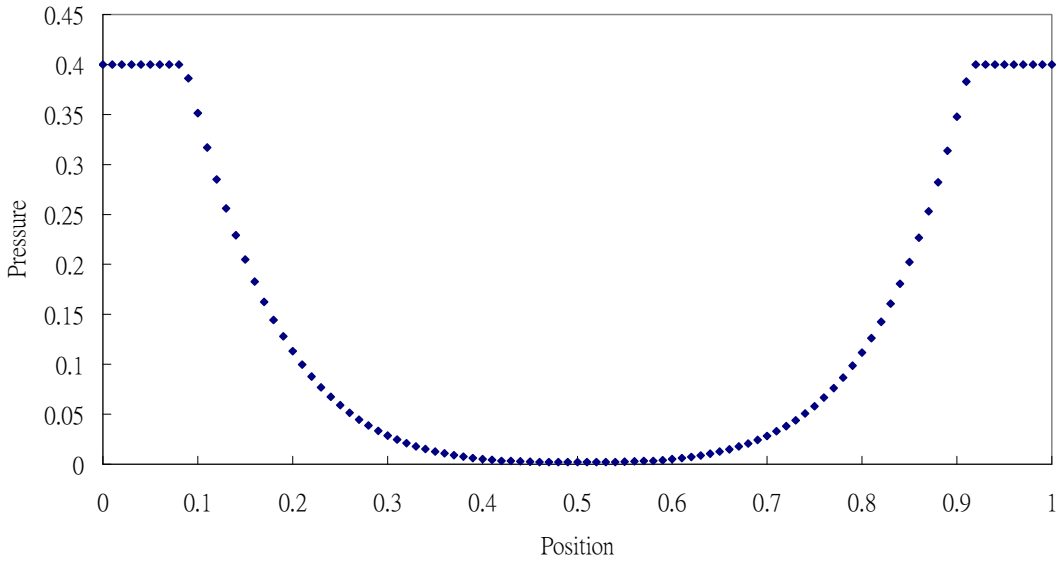
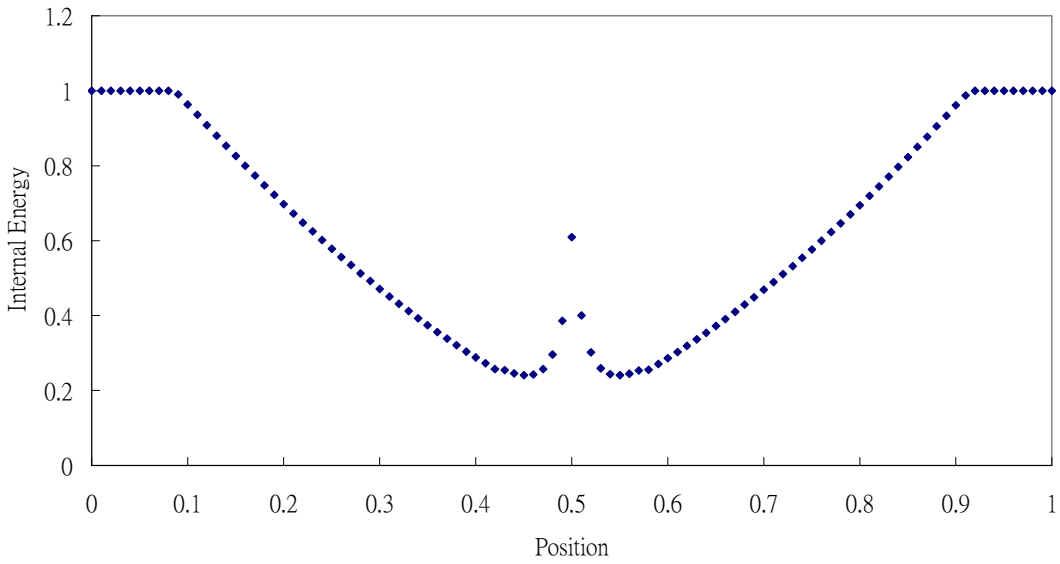
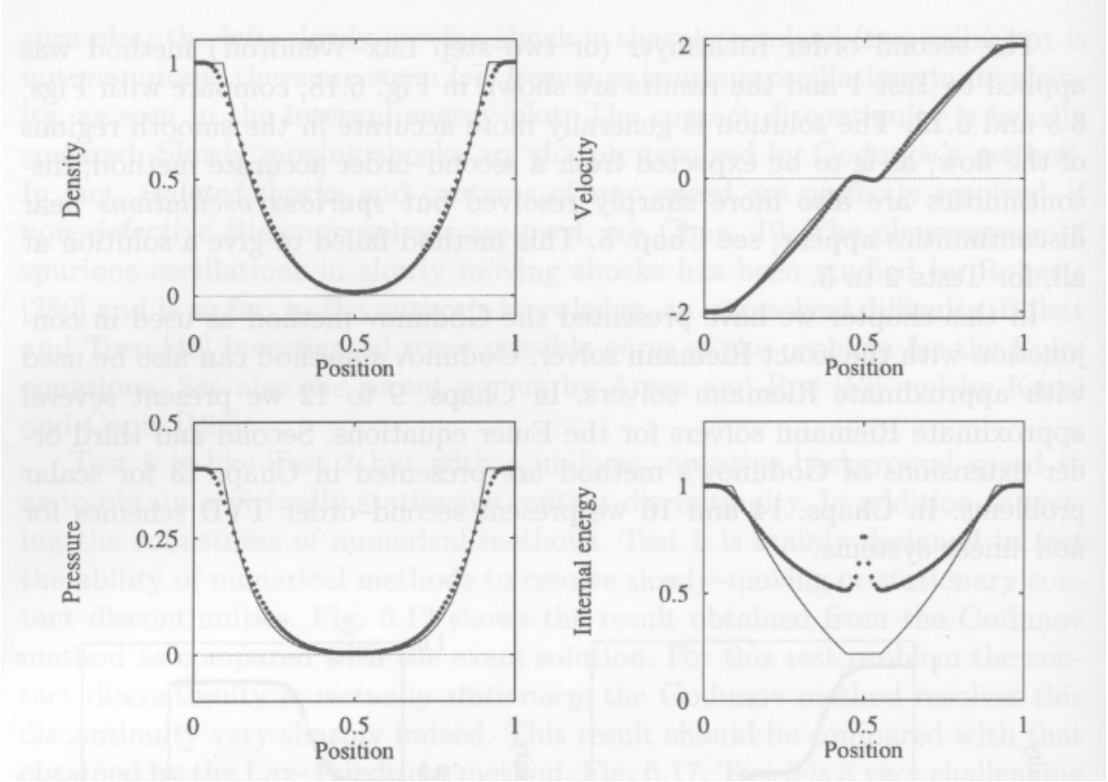


Fig 21. Internal Energy vs Position (Test 2)



The results for density, velocity, pressure and internal energy versus position calculated using the second-order Richtmyer (or two-step Lax-Wendraff) method are plotted in Fig 22.

Fig 22. Results for Test 2 Using Second-order Richtmyer Method (Picture adopted in [4])



In the density and pressure profiles in Fig 22, the numerical data for the second-order Richtmyer method cannot produce sharp changes at $x = 0.1$ and $x = 0.9$, while WENO Scheme is able to do them (Fig 18, Fig 20) and there should be sharp changes.

In the velocity profile in Fig 22, the analytical data for velocity is slightly larger than 0 at position slightly less than 0.5 and is slightly smaller than 0 at position slightly larger than 0.5. The second-order Richtmyer method is able to simulate this but WENO Scheme cannot (Fig 19).

In the internal energy profile in Fig 22, the second-order Richtmyer method is not able to simulate the internal energy for $0.2 < x < 0.8$, almost the whole domain, while WENO Scheme cannot simulate for $0.45 < x < 0.55$ (Fig 21), 10% of the domain. The internal energy calculated numerically jumps up at $x = 0.5$. But before jumping up and after jumping up, WENO Scheme can still get the value of 0.2 for the internal energy while the second-order Richtmyer method fails to do so. In the program, the evolve variable is the total energy density τ. The internal energy density is then obtained by $\varepsilon = \tau/\rho - u^2/2$. From the density profiles (Fig 18, 22), the density is nearly 0 at position 0.5. The numerical value of the internal energy will easily become a much larger value than the analytical one.

Test 3

At $t = 0.012$, the density, velocity, pressure and internal energy versus position calculated using WENO Scheme are plotted in Fig 23, 24, 25 and 26, respectively.

Fig 23. Density vs Position (Test 3)

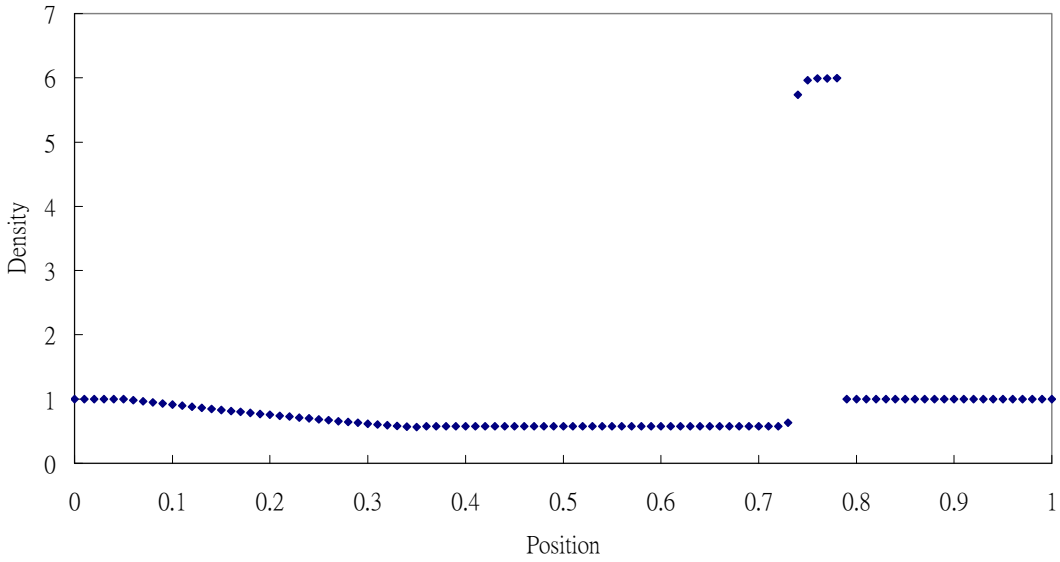


Fig 24. Velocity vs Position (Test 3)

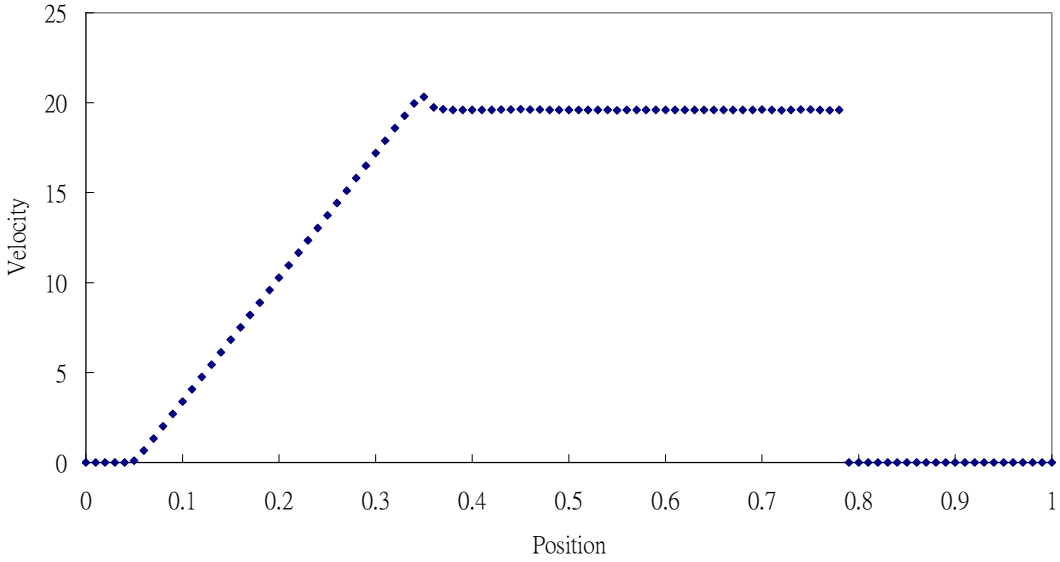


Fig 25. Pressure vs Position (Test 3)

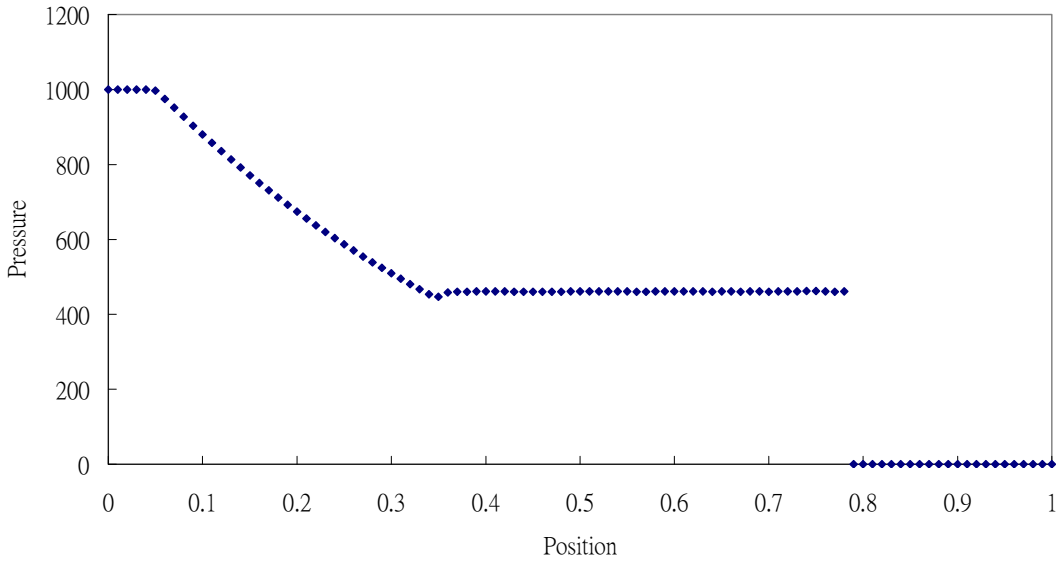
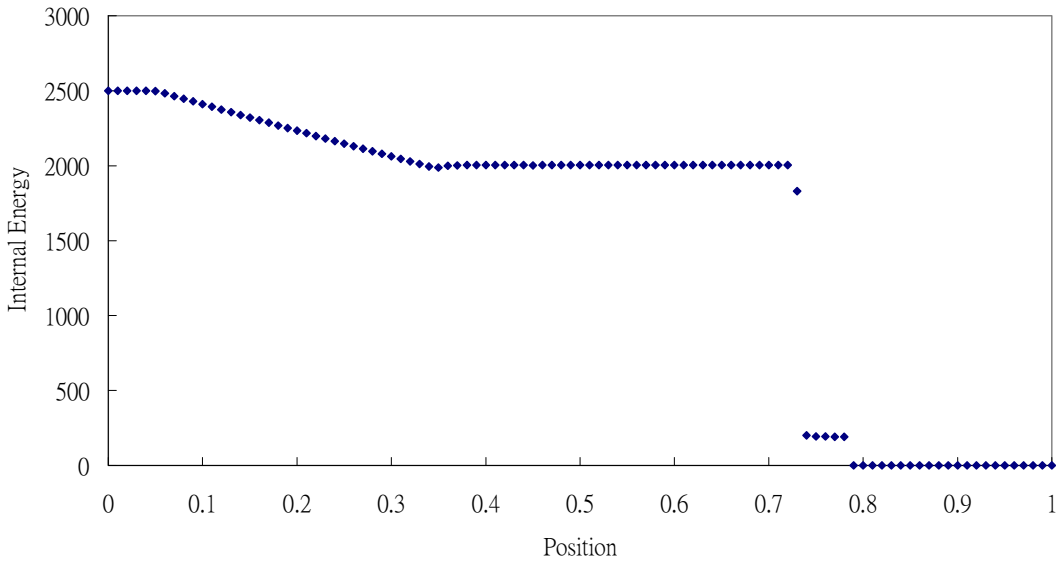
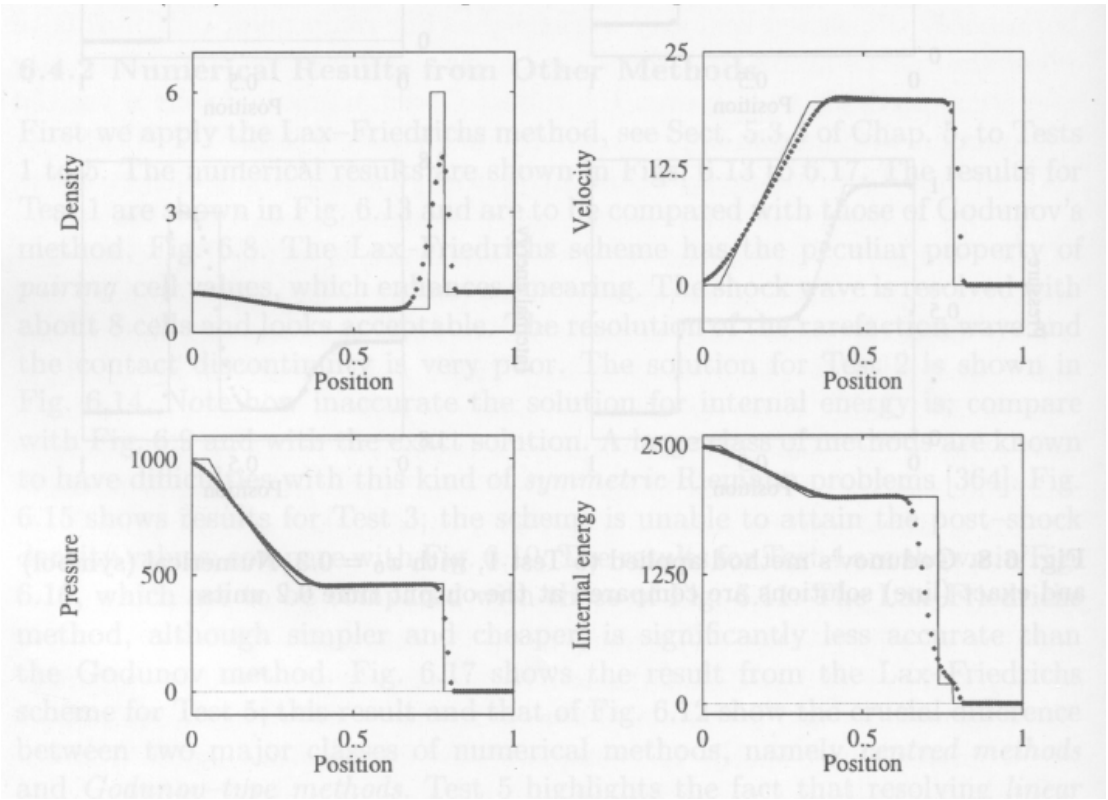


Fig 26. Internal Energy vs Position (Test 3)



The results for density, velocity, pressure and internal energy versus position calculated using the second-order Richtmyer (or two-step Lax-Wendraff) method are plotted in Fig 27.

Fig 27. Results for Test 3 Using Second-order Richtmyer Method (Picture adopted in [4])



In the density profile in Fig 27, the second-order Richtmyer method is not able to simulate the density for $0.7 < x < 0.8$, while the WENO Scheme is able to do so (Fig 23). The analytical value of the density for $0.75 < x < 0.80$ is 6. The second-order Richtmyer method can only get 4.5, while the WENO Scheme can get 6 (Fig 23).

In the velocity, pressure and the internal energy profiles in Fig 27, the second-order Richtmyer method cannot produce sharp changes at $x = 0.05$ and $x = 0.35$ while the WENO Scheme can produce these sharp changes (Fig 24, Fig 25, Fig 26), which is the case for analytical result.

In the internal energy profile in Fig 27, the second-order Richtmyer method cannot produce sharp drop at $x = 0.73$ and cannot produce the small flat region for $0.74 \le x \le 0.78$ while the WENO Scheme can able to produce those. Those are the cases for analytical result.

Test 4

At $t = 0.035$, the density, velocity, pressure and internal energy versus position calculated using WENO Scheme are plotted in Fig 28, 29, 30 and 31, respectively.

Fig 28. Density vs Position (Test 4)

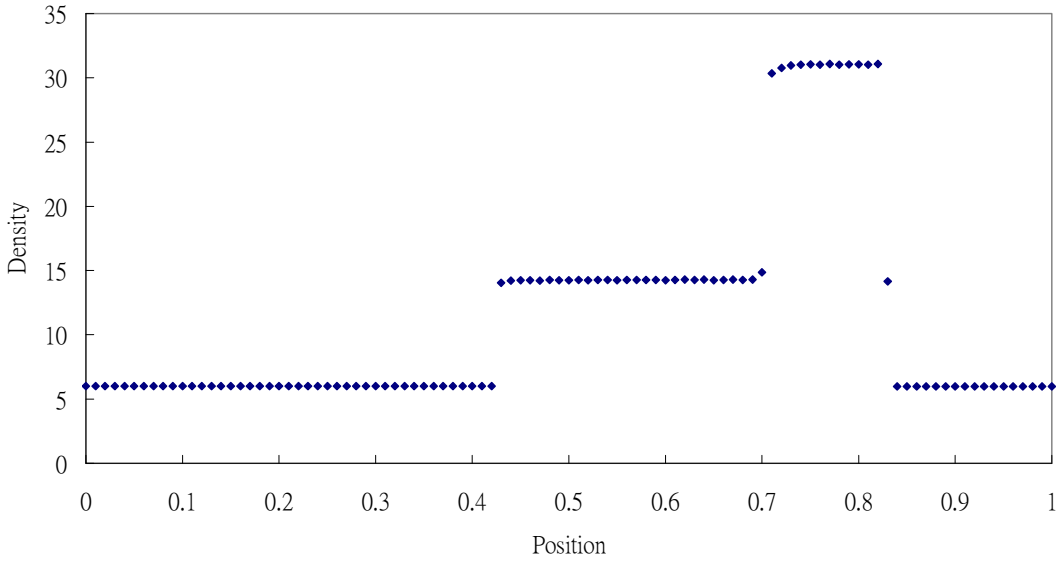


Fig 29. Velocity vs Position (Test 4)

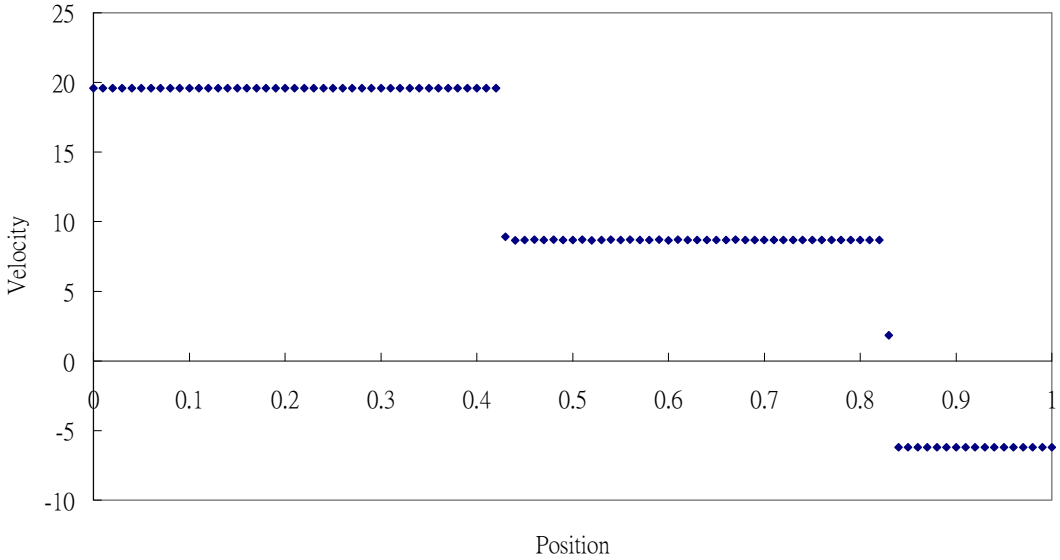


Fig 30. Pressure vs Position (Test 4)

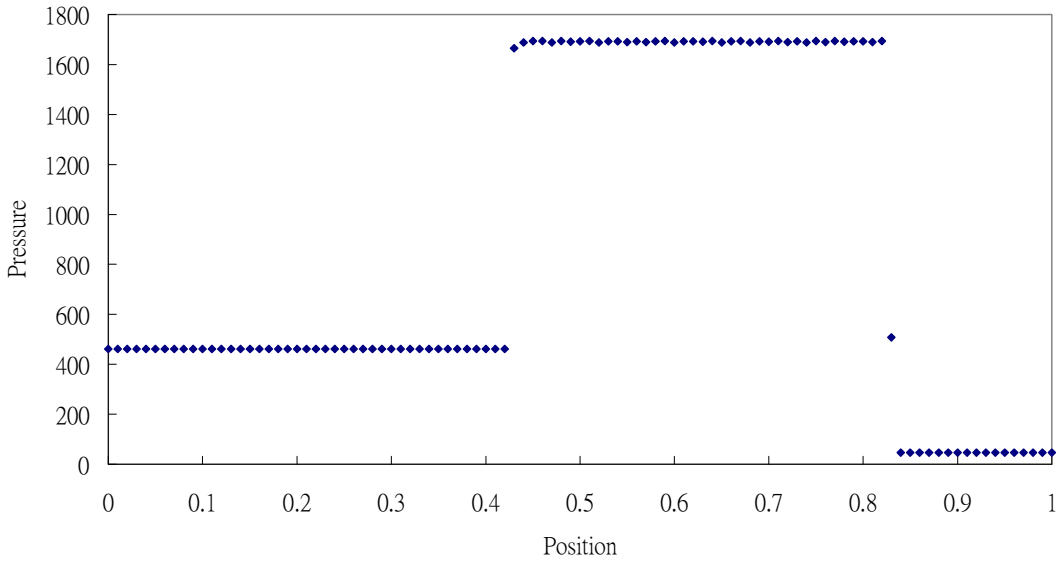
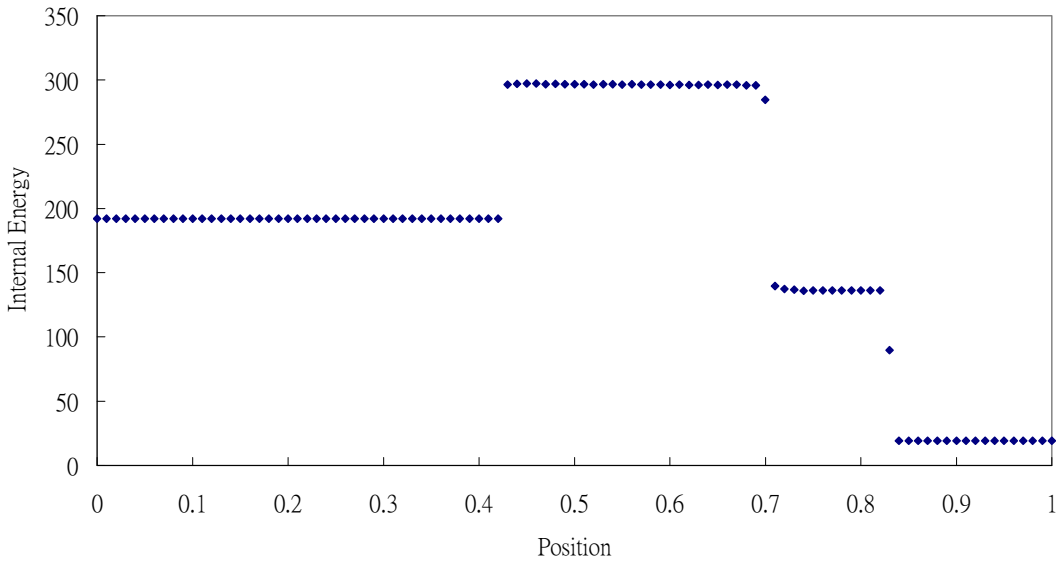
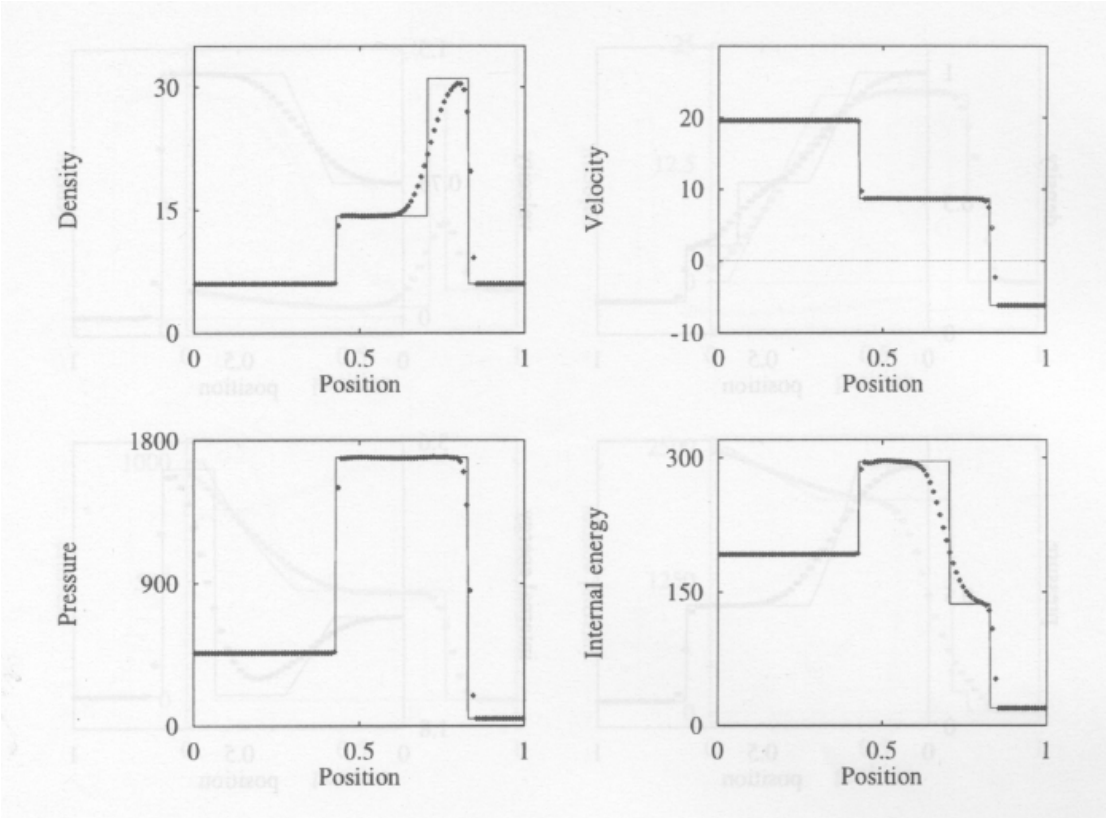


Fig 31. Internal Energy vs Position (Test 4)



The results for density, velocity, pressure and internal energy versus position calculated using the second-order Richtmyer (or two-step Lax-Wendraff) method are plotted in Fig 32.

Fig 32. Results for Test 4 Using Second-order Richtmyer Method (Picture adopted in [4])



In the density profile of Fig 32, the second-order Richtmyer method cannot produce correct values for $0.6 < x < 0.8$ and the sharp jump at $x = 0.7$, while the WENO Scheme is success to do so (Fig 28).

For the velocity and the pressure profiles, both the second-order Richtmyer method (Fig 32) and the WENO Scheme (Fig 29, 30) are able to produce correct results.

In the internal energy profile of Fig 32, the second-order Richtmyer method cannot produce correct values for $0.6 < x < 0.8$ and the sharp drop at $x = 0.7$, while the WENO Scheme is success to do so (Fig 28).

Test 5

At $t = 0.012$, the density, velocity, pressure and internal energy versus position calculated using WENO Scheme are plotted in Fig 33, 34, 35 and 36, respectively.

Fig 33. Density vs Position (Test 5)

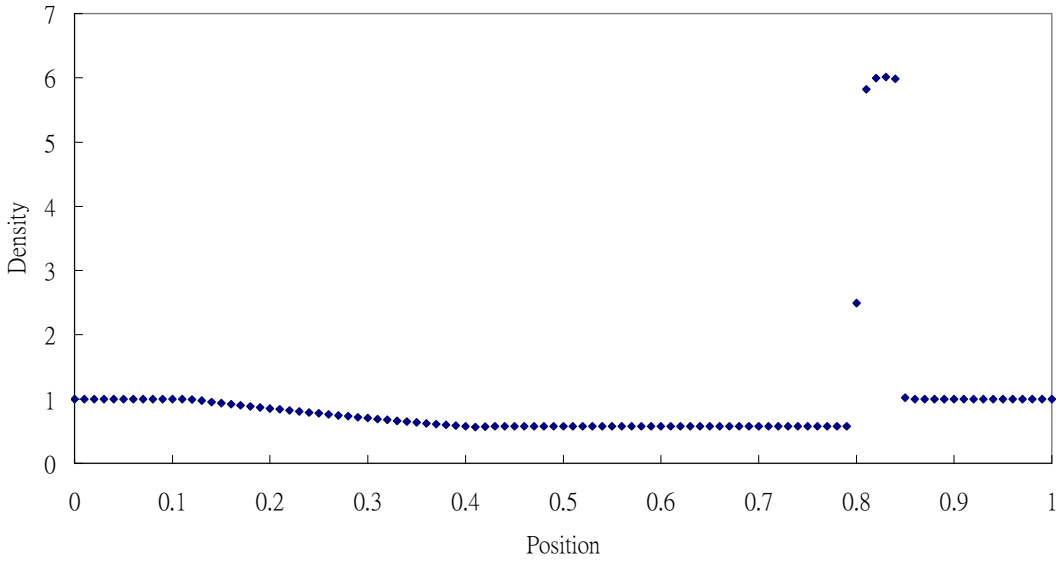


Fig 34. Velocity vs Position (Test 5)

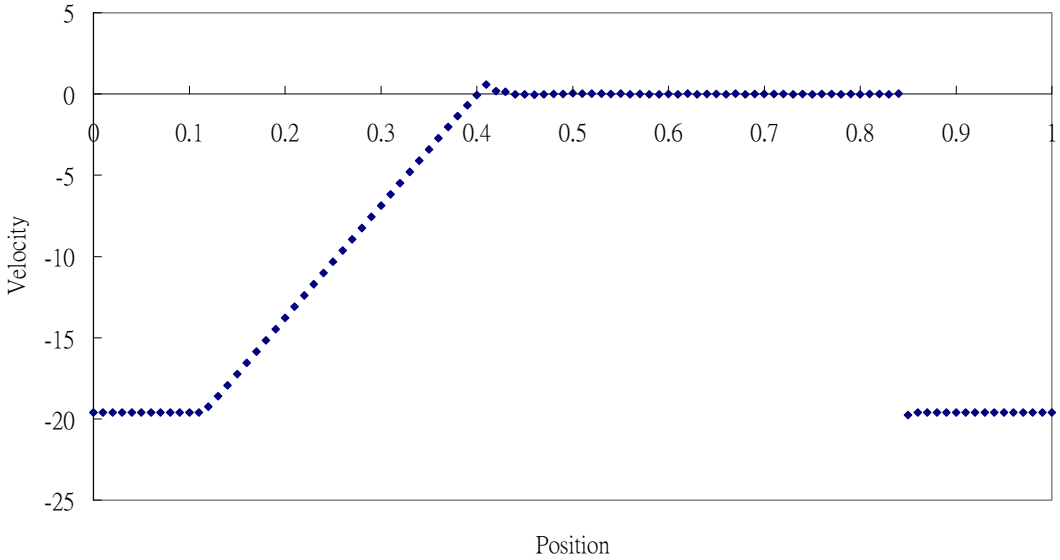


Fig 35. Pressure vs Position (Test 5)

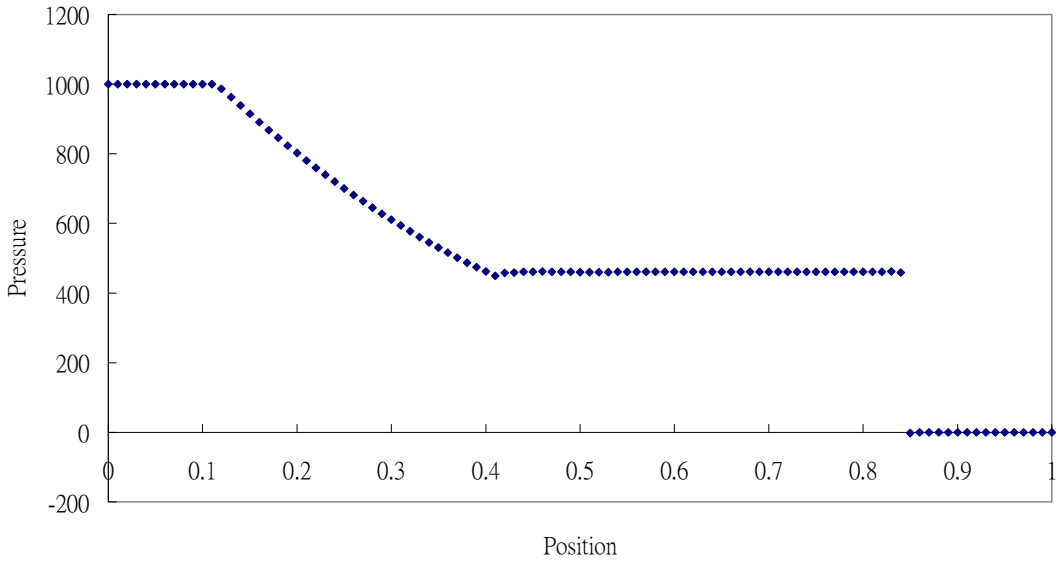
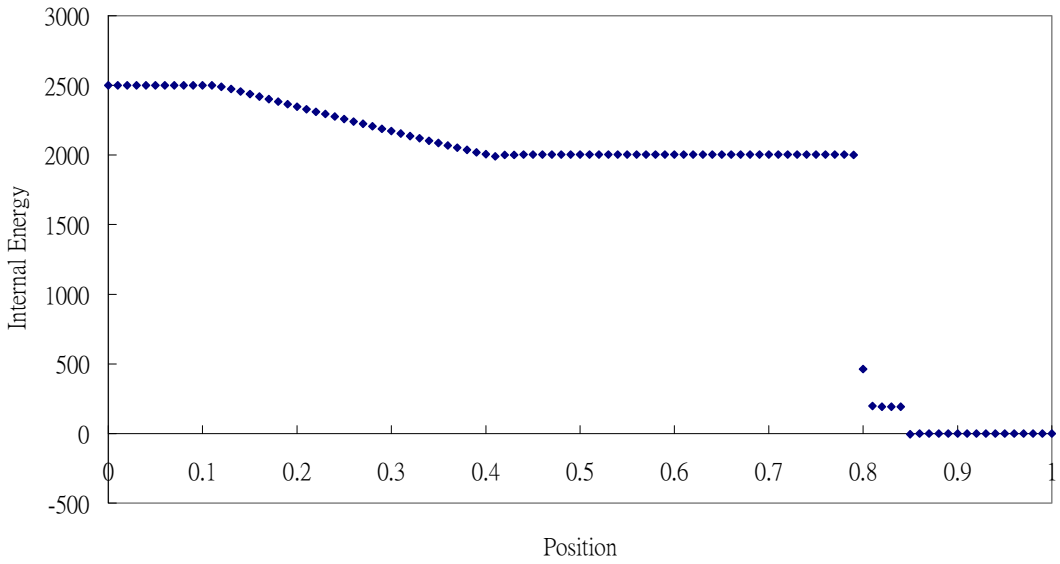
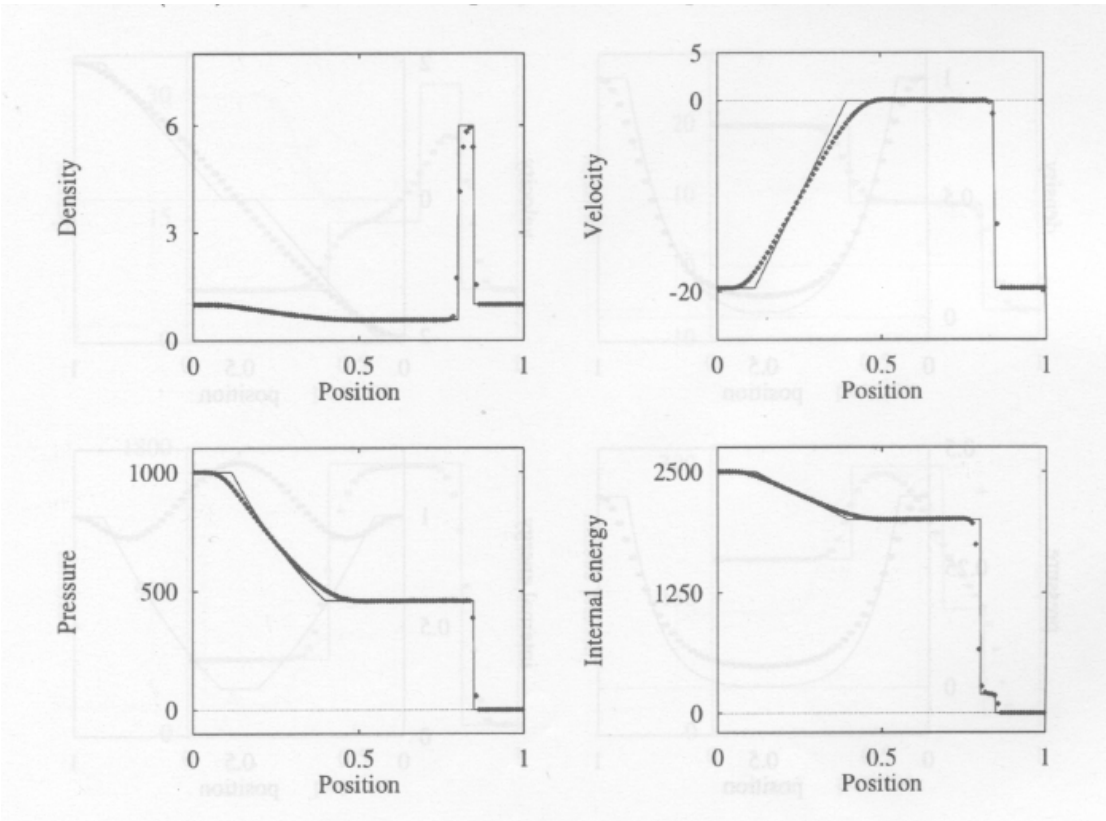


Fig 36. Internal Energy vs Position (Test 5)



The results for density, velocity, pressure and internal energy versus position calculated using the second-order Richtmyer (or two-step Lax-Wendraff) method are plotted in Fig 37.

Fig 37. Results for Test 5 Using Second-order Richtmyer Method (Picture adopted in [4])



Actually, Test 5 is like Test 3 with negative uniform background speed.

In the density profile in Fig 37, the second-order Richtmyer method cannot simulate the sharp jump at $x = 0.8$, while the WENO Scheme can do it better (Fig 33).

In the velocity, pressure and the internal energy profile in Fig 37, the second-order Richtmyer method cannot produce sharp changes at $x = 0.1$ and $x = 0.4$, while the WENO Scheme can produce those (Fig 34, 35, 36), which are the cases for analytical result.

X. Spherical Symmetric Fluid Dynamics

For spherical symmetric case, the equations can be written as:

$$\begin{bmatrix} \rho \\ \rho v \\ \tau \end{bmatrix}_t + \begin{bmatrix} \rho v \\ \rho v^2 + p \\ v(\tau + p) \end{bmatrix}_r = -\frac{\alpha}{r} \begin{bmatrix} \rho v \\ \rho v^2 \\ v(\tau + p) \end{bmatrix},$$

where v is the radial velocity, $\alpha = 2$ and $\tau = \rho\varepsilon + \frac{1}{2}\rho v^2$.

The equations are the same as that in Cartesian Coordinates, except that there are source terms.

Again, I have used the polytropic equation of state $p = k\rho^{\gamma}$ with $\varepsilon = \frac{k}{\gamma - 1}\rho^{\gamma - 1}$, which gives:

$$p = (\gamma - 1)\rho\varepsilon .$$

In order to test the program whether there are problems for $r \to 0$ as there is $1/r$ source term for the equations, I have performed a diffusion test.

Diffusion Test

Initially, I have a constant density and constant pressure spherical gas bubble in a vacuum, with smooth drops of density and pressure at the boundary of the bubble. Then I let it diffuses.

The density, velocity, pressure and the internal energy versus position (radius) are plotted in Fig 38, 39, 40. and 41, respectively.

Fig 38. Density vs Position (dx = 0.001)

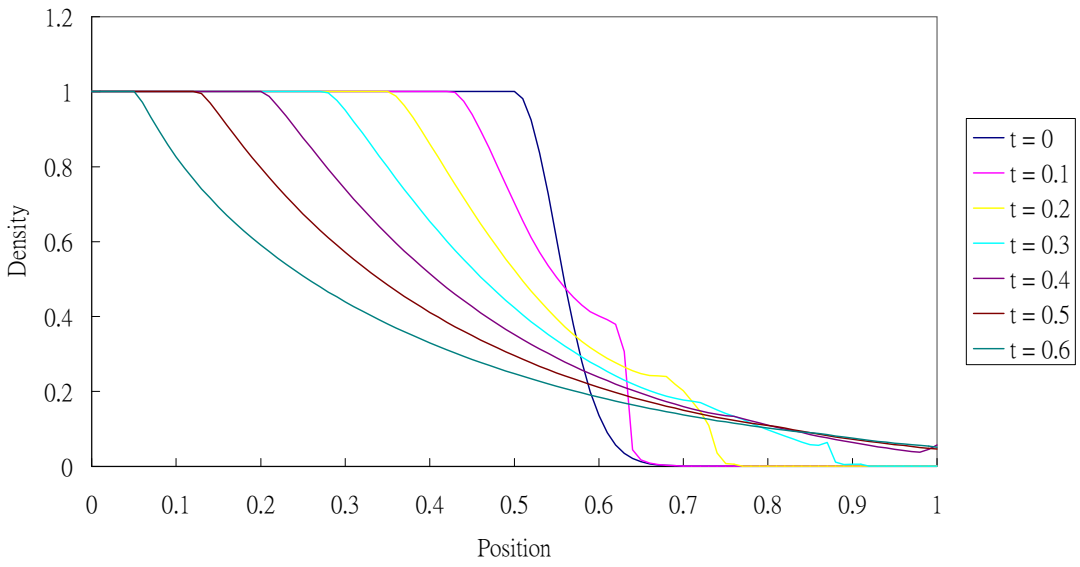


Fig 39. Velocity vs Position (dx = 0.001)

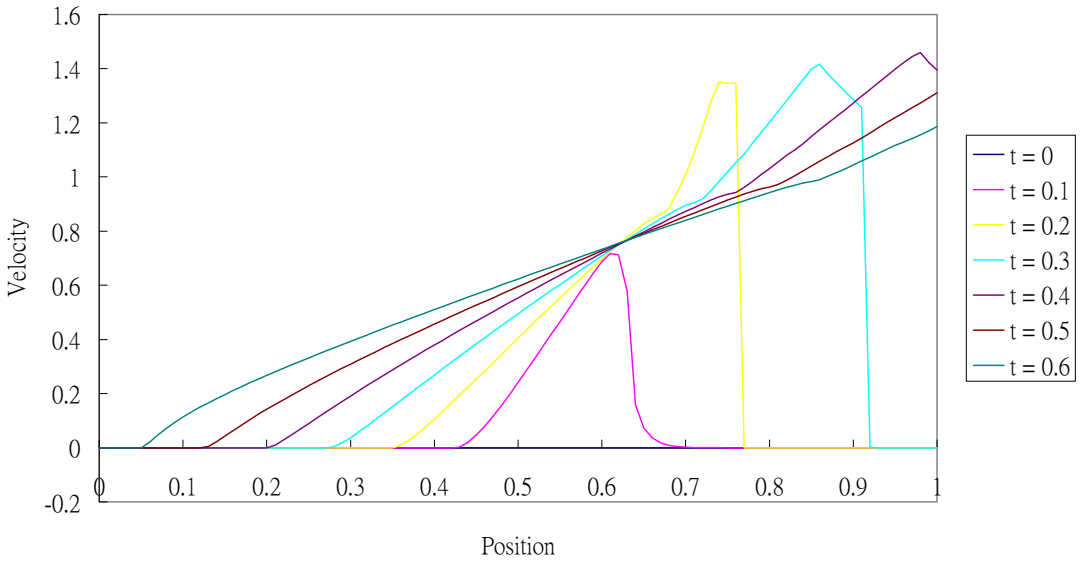


Fig 40. Pressure vs Position (dx = 0.001)

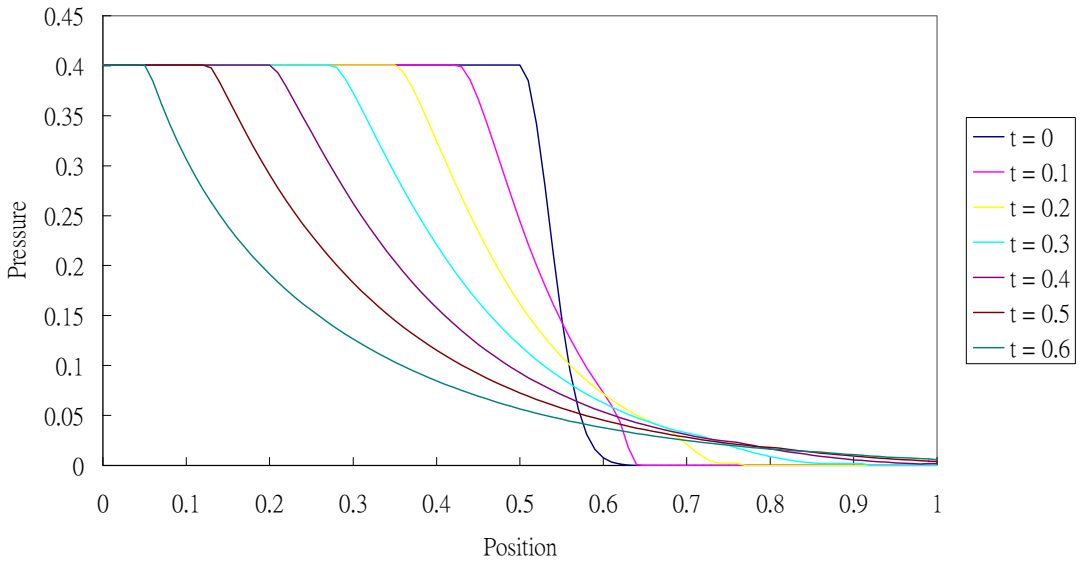
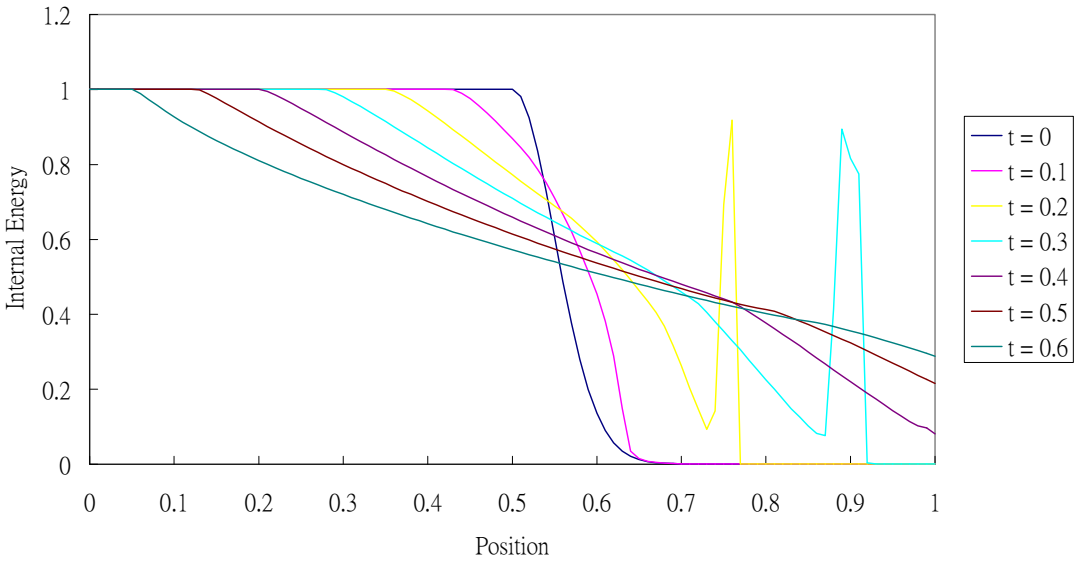


Fig 41. Internal Energy vs Position (dx = 0.001)



As the bubble diffuses, the density, pressure and the internal energy at the centre keeps constant rather than blows up or drops down (Fig 38, 40, 41).
At $t = 0.2$, the gas front is at $x \approx 0.75$ and at $t = 0.3$, the gas front is at $x \approx 0.90$ (Fig 38). The gas at the gas front is being compressed, making its temperature to rise. Therefore, the internal energy at the gas front is much higher compared with the region around it (Fig 41).

As the program is successful even for $r \to 0$ region, I also perform the five Riemann problem tests for the spherical symmetric case. The initial conditions and the value of γ are the same as that in the Cartesian Coordinates, but the boundary at $r = 0$ is reflecting rather than free.
The initial data of those tests are in the format:

$$\rho = \begin{cases} \rho_L & 0 \le r \le r_0 \\ \rho_R & r_0 < r \le 1 \end{cases}, \quad u = \begin{cases} u_L & 0 \le r \le r_0 \\ u_R & r_0 < r \le 1 \end{cases}, \quad p = \begin{cases} p_L & 0 \le r \le r_0 \\ p_R & r_0 < r \le 1 \end{cases}.$$

The initial data are:

Test	ρ_L	u_L	p_L	r_0	ρ_R	u_R	p_R
1	1.0	0.75	1.0	0.3	0.125	0.0	0.1
2	1.0	-2.0	0.4	0.5	1.0	2.0	0.4
3	1.0	0.0	1000.0	0.5	1.0	0.0	0.01
4	5.99924	19.5975	460.894	0.4	5.99242	-6.19633	46.0950
5	1.0	-19.59745	1000.0	0.8	1.0	-19.59745	0.01

Test 1

The density, velocity, pressure and internal energy versus position calculated using WENO Scheme are plotted in Fig 42, 43, 44 and 45, respectively.

Fig 42. Density vs Position (Test 1, Spherical)

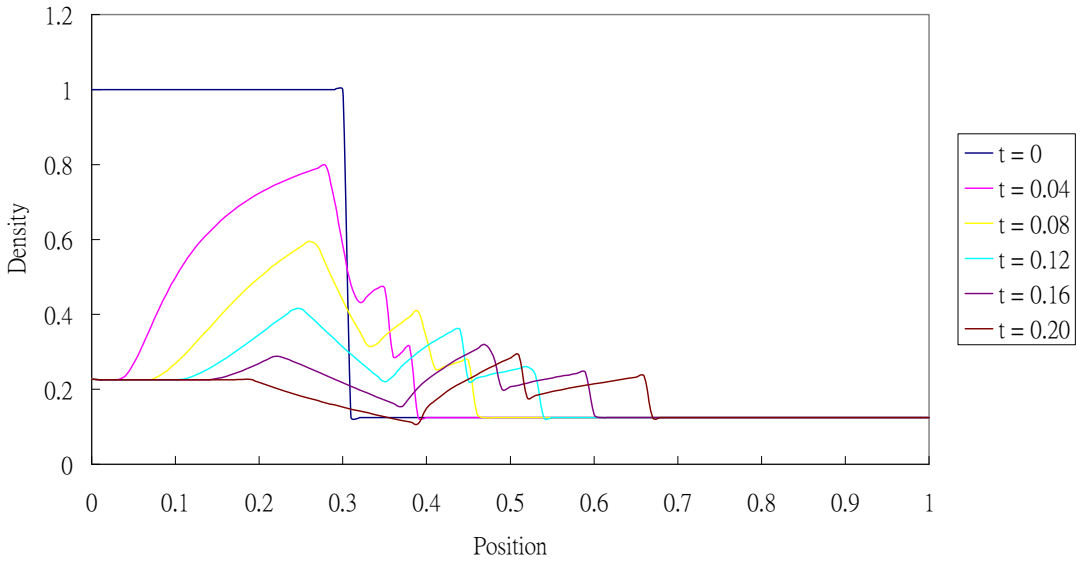


Fig 43. Velocity vs Position (Test 1, Spherical)

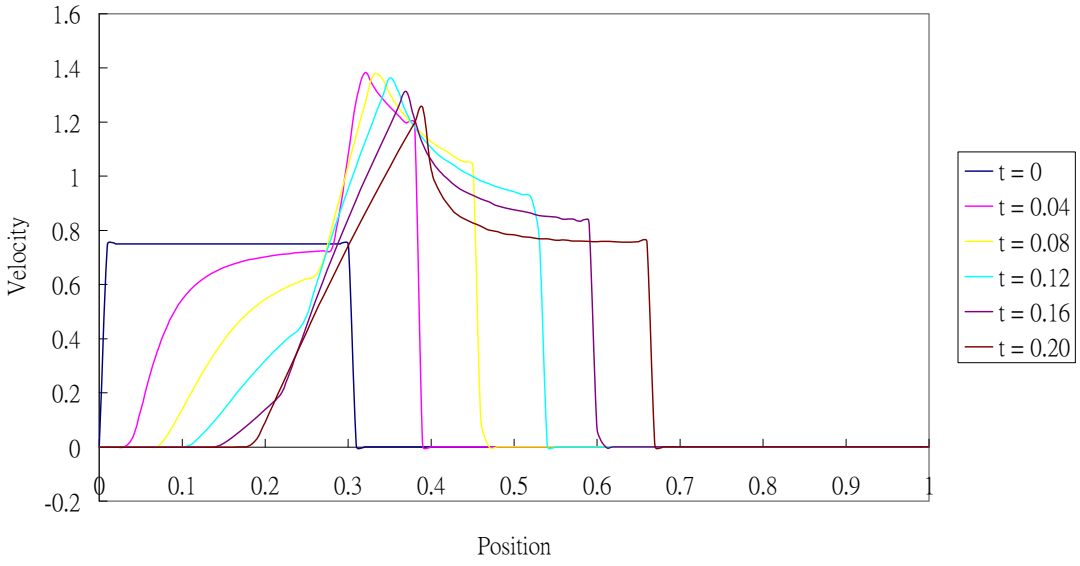


Fig 44. Pressure vs Position (Test 1, Spherical)

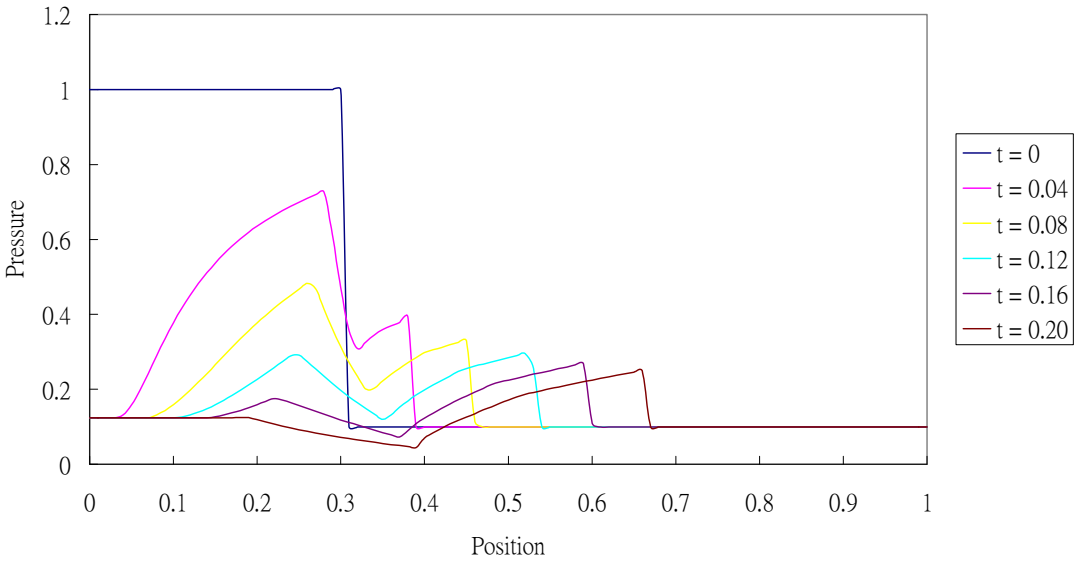
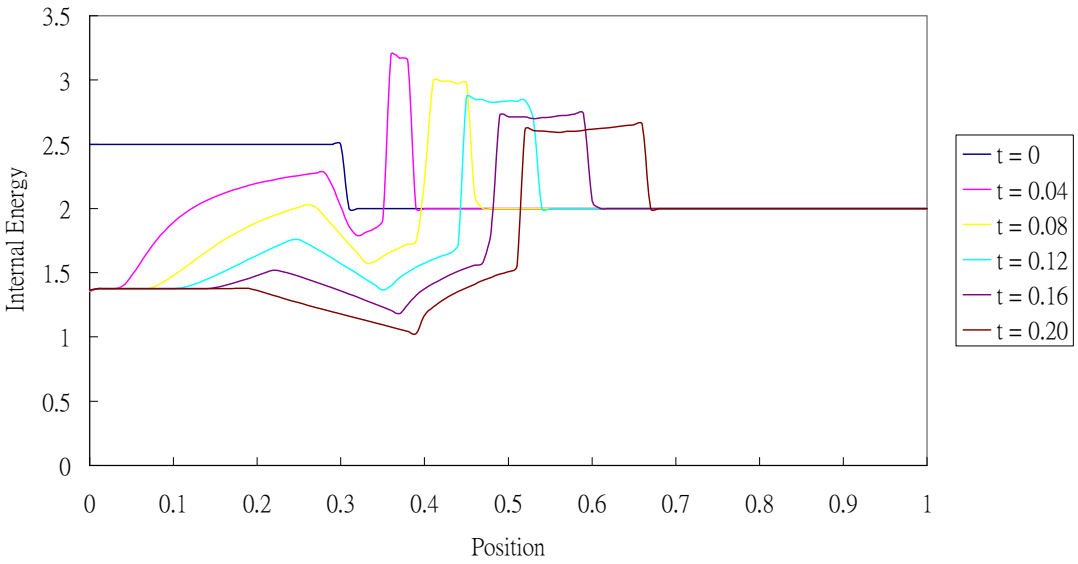


Fig 45. Internal Energy vs Positoin (Test 1, Spherical)



Initially, it is more condense in the core and the core is expanding. As it evolves, shocks are created at the interface between the core and the envelope (Fig 42). Pressure at the core decreases and pressure at the shocks are higher than the region around them (Fig 44). Due to the compression at the shocks, the temperature and thus the internal energy rise at the shocks (Fig 45).

Test 2

The density, velocity, pressure and internal energy versus position calculated using WENO Scheme are plotted in Fig 46, 47, 48 and 49, respectively.

Fig 46. Density vs Position (Test 2, Spherical)

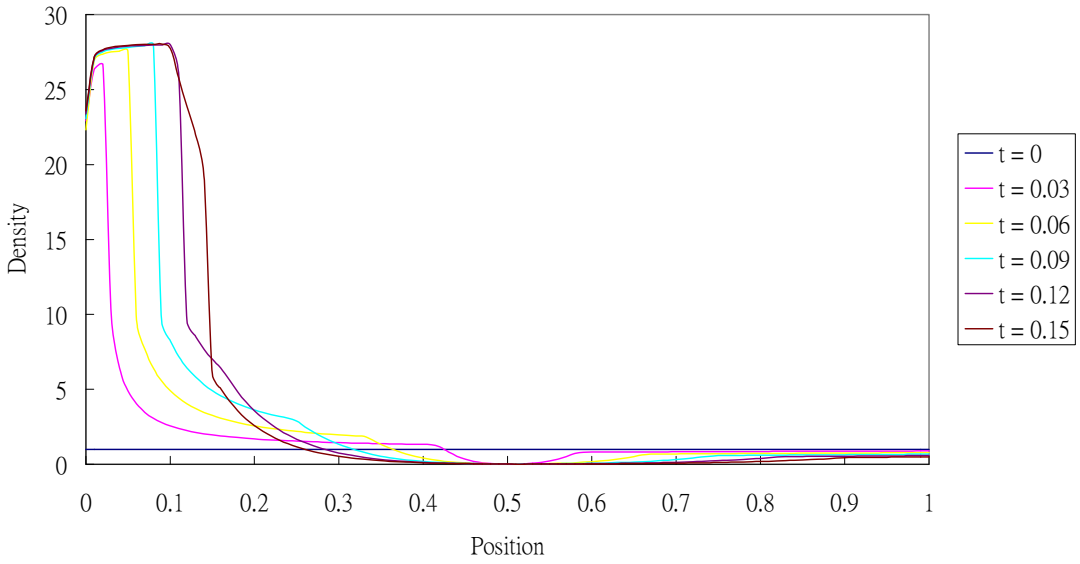


Fig 47. Velocity vs Position (Test 2, Spherical)

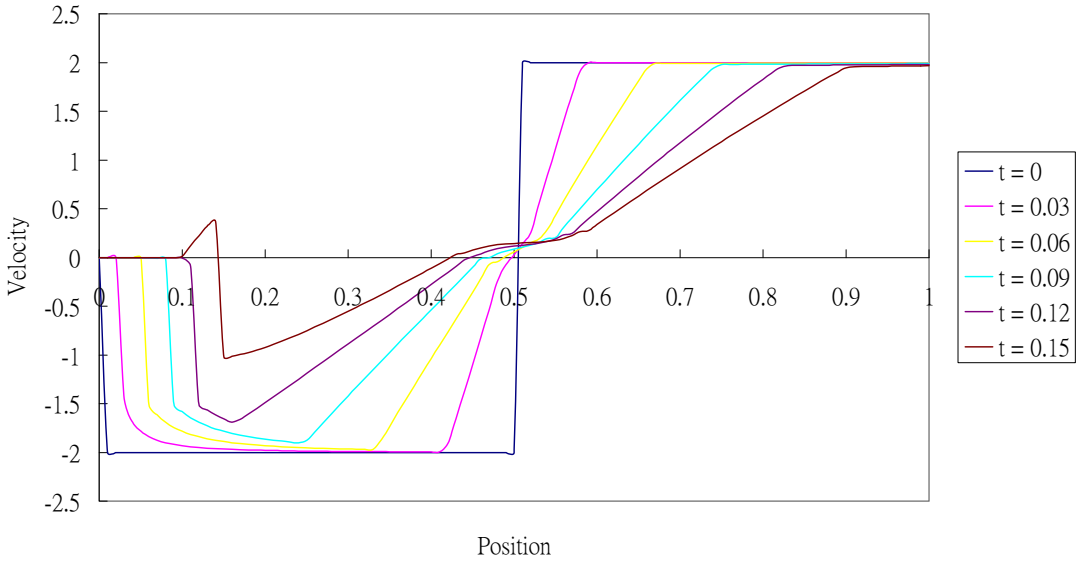


Fig 48. Pressure vs Position (Test 2, Spherical)

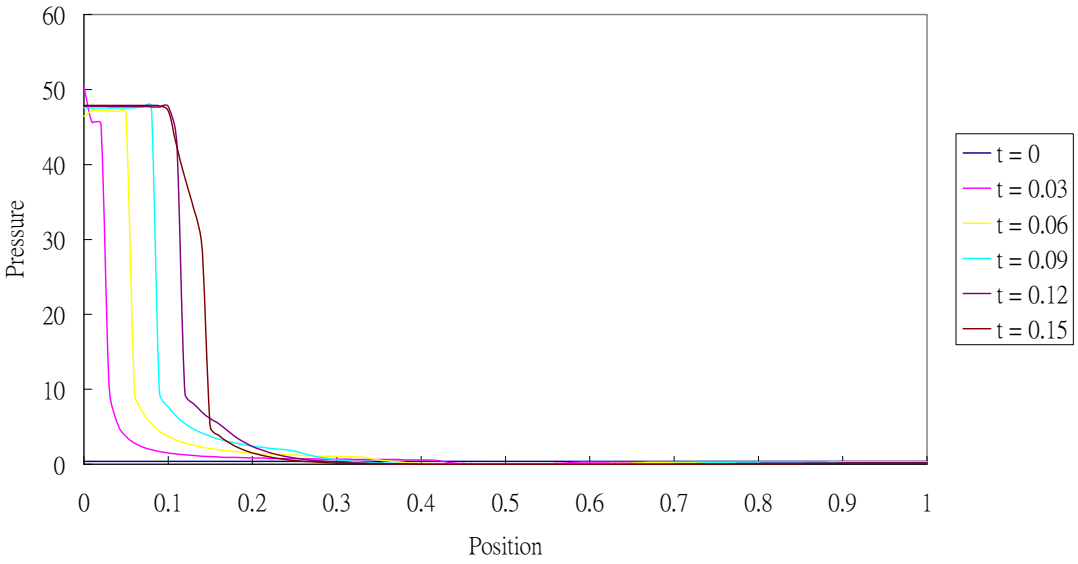
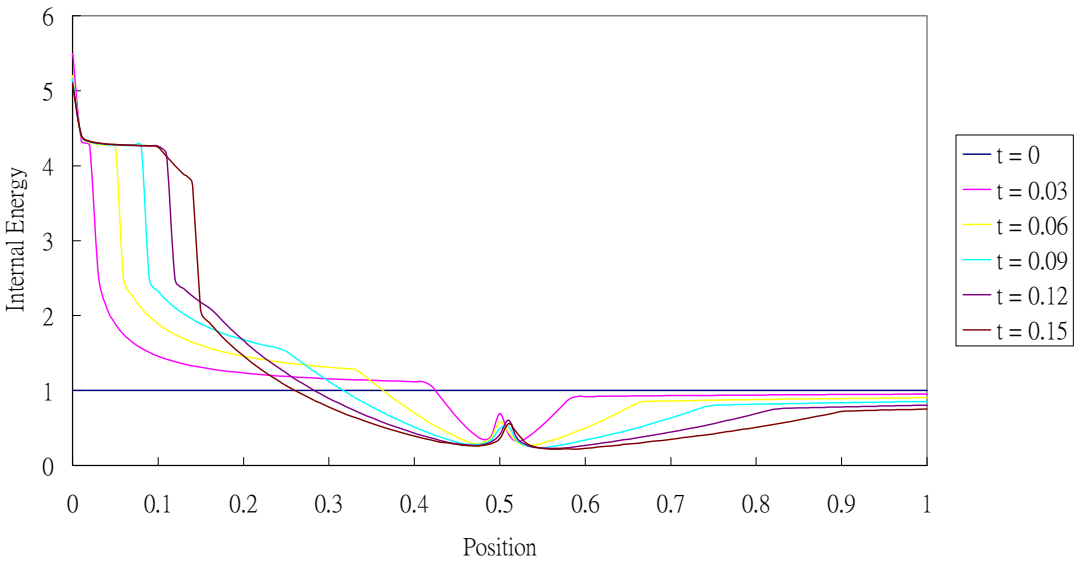


Fig 49. Internal Energy vs Position (Test 2, Spherical)



Initially, the whole space has uniform density and pressure, but the core is collapsing and the envelope is expanding. As it evolves, the core is compressed to a high density and pressure (Fig 46, 48) and this core is growing is size. This compressed core has high pressure so that it will not compress any more. The velocity of this compressed core is zero (Fig 47) The temperature of this compressed core is large so that the internal energy is large (Fig 49).

Test 3

The density, velocity, pressure and internal energy versus position calculated using WENO Scheme are plotted in Fig 50, 51, 52 and 53, respectively.

Fig 50. Density vs Position (Test 3, Spherical)

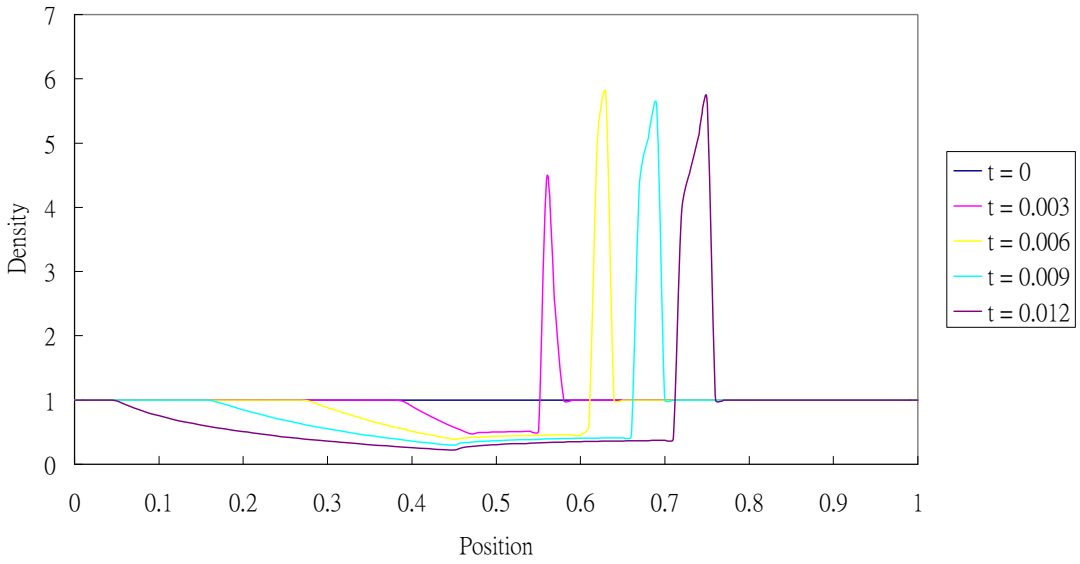


Fig 51. Velocity vs Position (Test 3, Spherical)

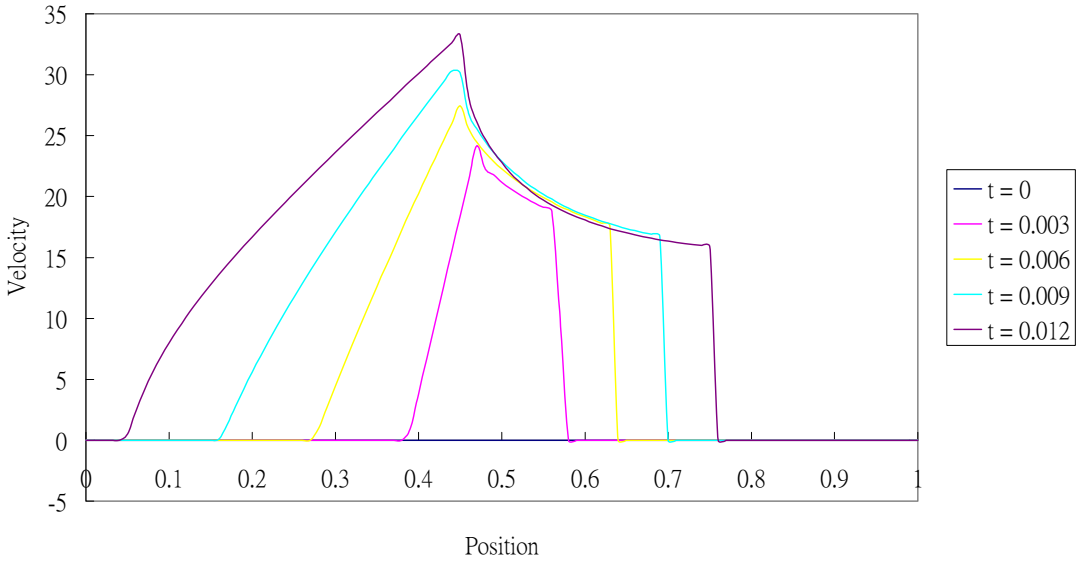


Fig 52. Pressure vs Position (Test 3, Spherical)

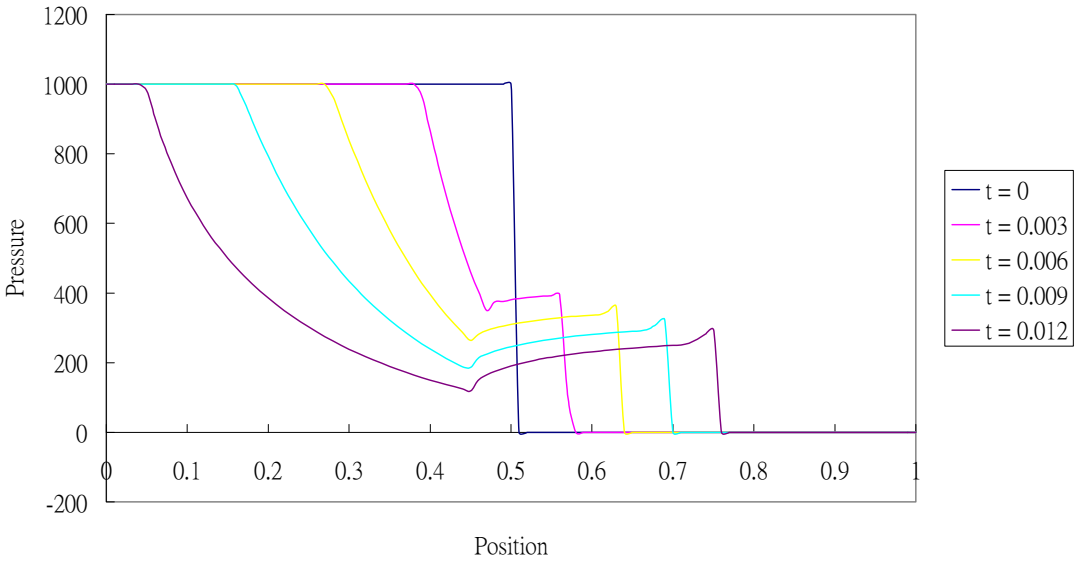
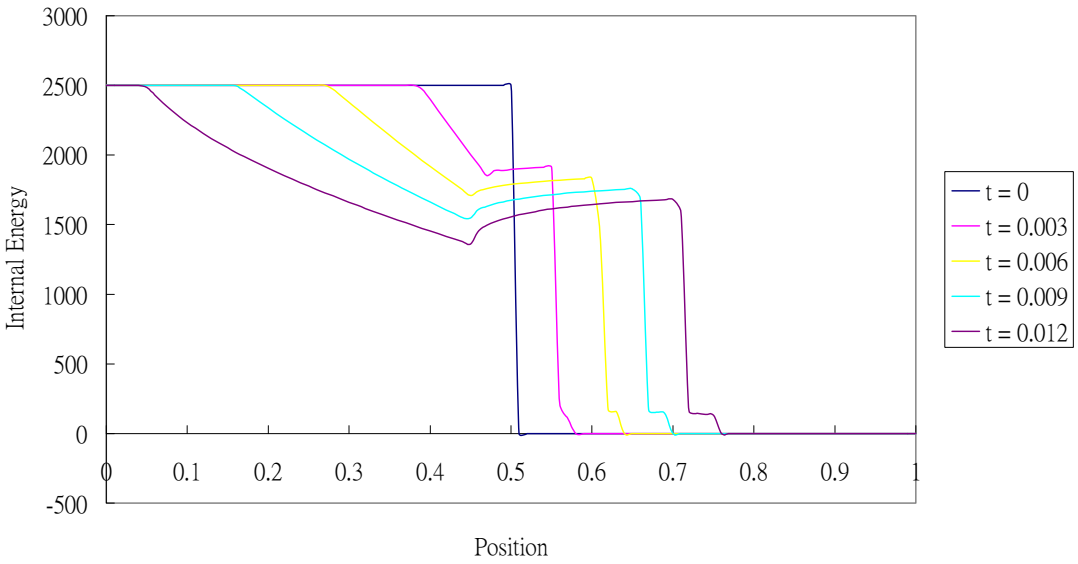


Fig 53. Internal Energy vs Position (Test 3, Spherical)



Initially, there is a gas of uniform density distributed in the whole space, but the pressure for the core is 10^5 larger than the envelope. As it explodes, shocks are created at the interface between the core and the envelope (Fig 50). The pressure and the internal energy of the core start to decrease and that of the envelope start to increase (Fig 52, 53).

Test 4

The density, velocity, pressure and internal energy versus position calculated using WENO Scheme are plotted in Fig 54, 55, 56 and 57, respectively.

Fig 54. Density vs Position (Test 4, Spherical)

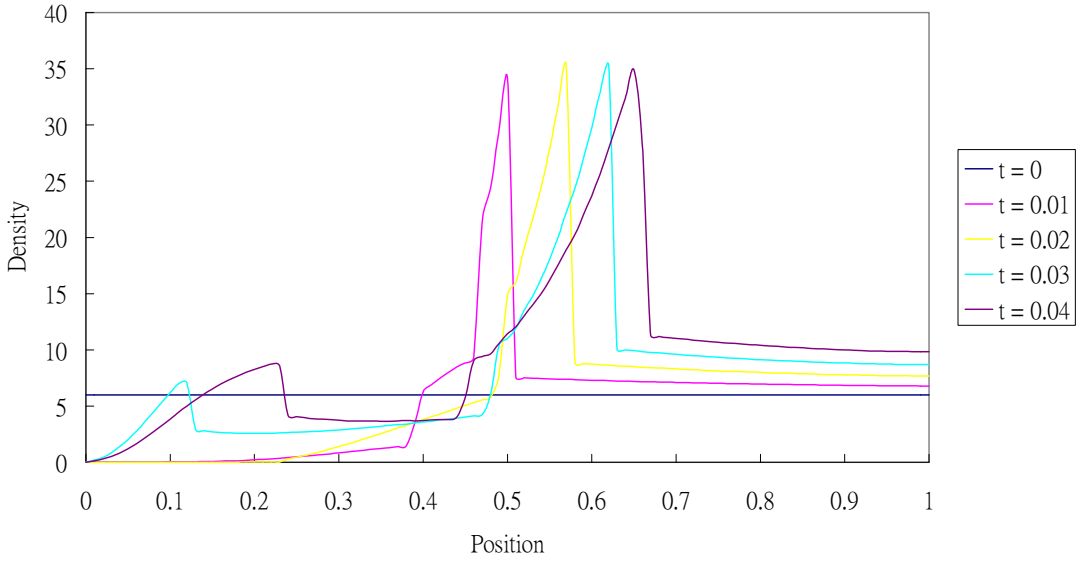


Fig 55. Velocity vs Position (Test 4, Spherical)

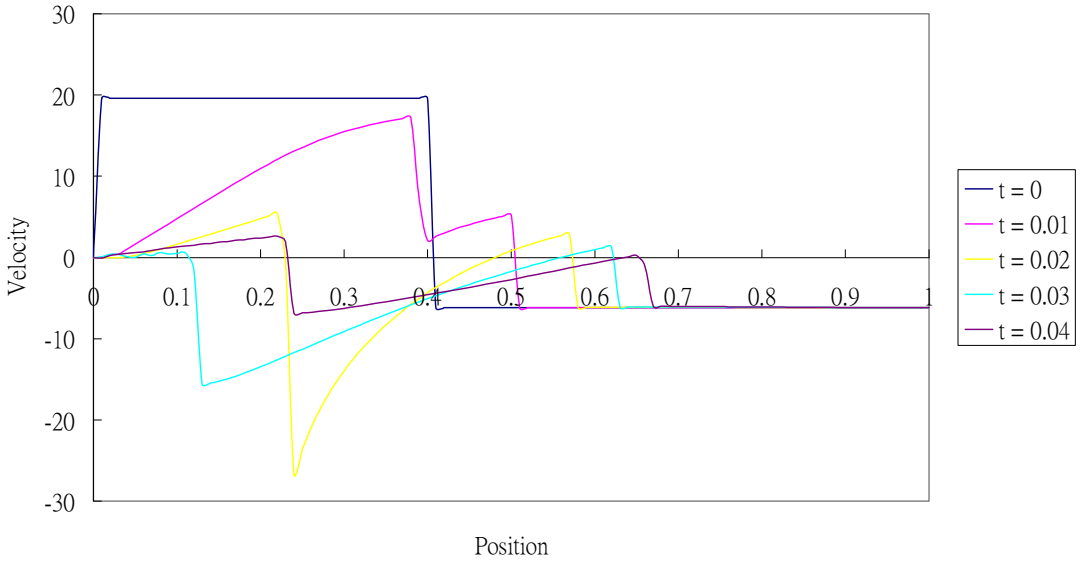


Fig 56. Pressure vs Position (Test 4, Spherical)

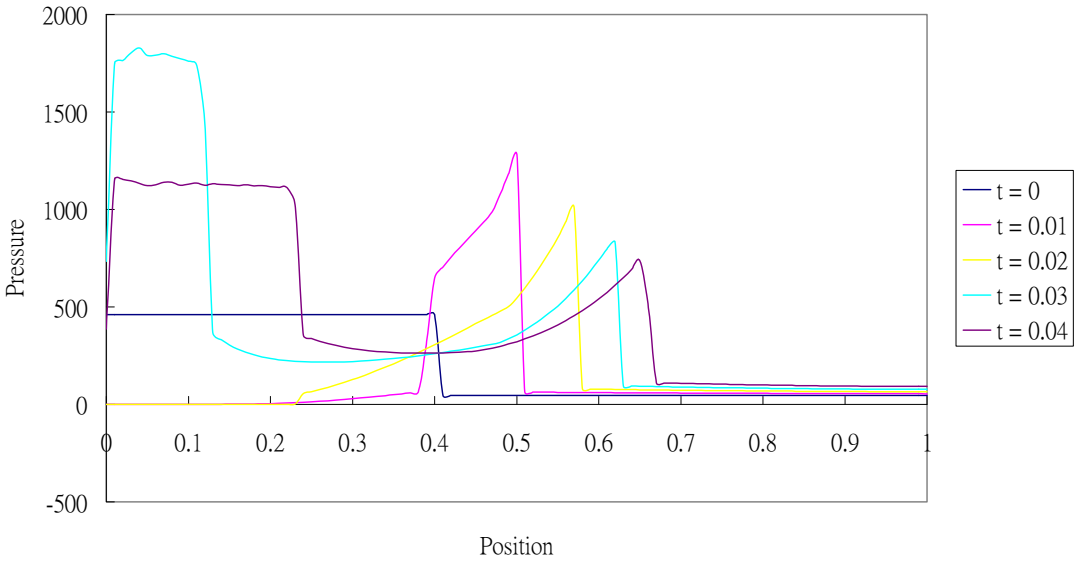
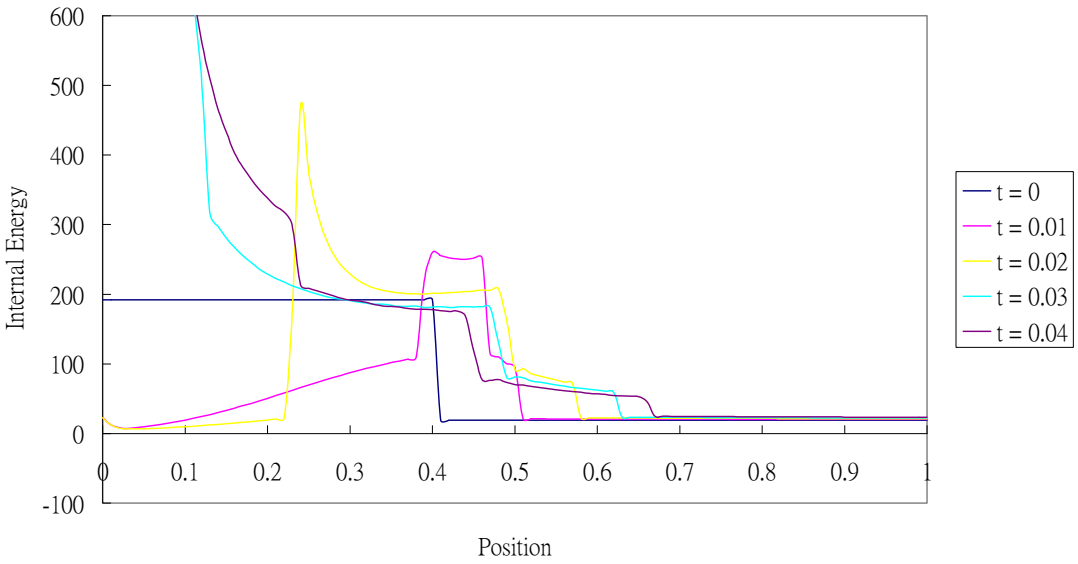


Fig 57. Internal Energy vs Position (Test 4, Spherical)



Initially, the densities for the core and the envelope are nearly the same, but the core is expanding and the envelope is contracting, creating shocks at the interface and move outside (Fig 54). Then the core becomes nearly vacuum and some mass contract back to the core (Fig 54, 55) making the pressure at the core increases again (Fig 56). And then the mass rebounds back at the centre (Fig 54, 55, 56) In the program, the evolve variable is the total energy density τ. The internal energy density is then obtained by $\varepsilon = \tau/\rho - v^2/2$. From the density profiles (Fig 54), the density is nearly 0 near the centre.

The numerical value of the internal energy near the centre will easily mistakenly blow up (Fig 57).

Test 5

The density, velocity, pressure and internal energy versus position calculated using WENO Scheme are plotted in Fig 58, 59, 60 and 61, respectively.

Fig 58. Density vs Position (Test 5, Spherical)

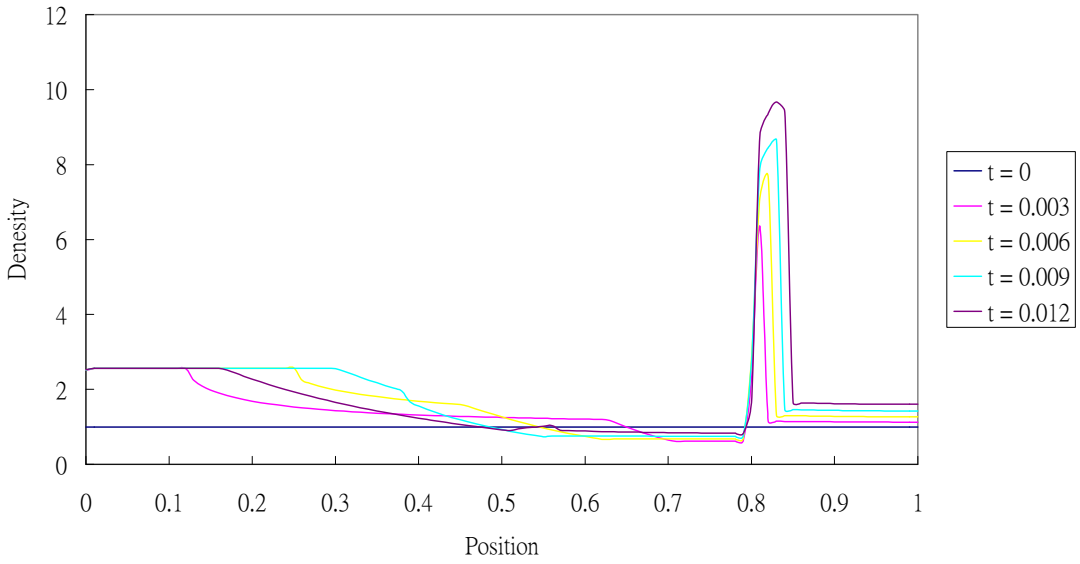


Fig 59. Velocity vs Position (Test 5, Spherical)

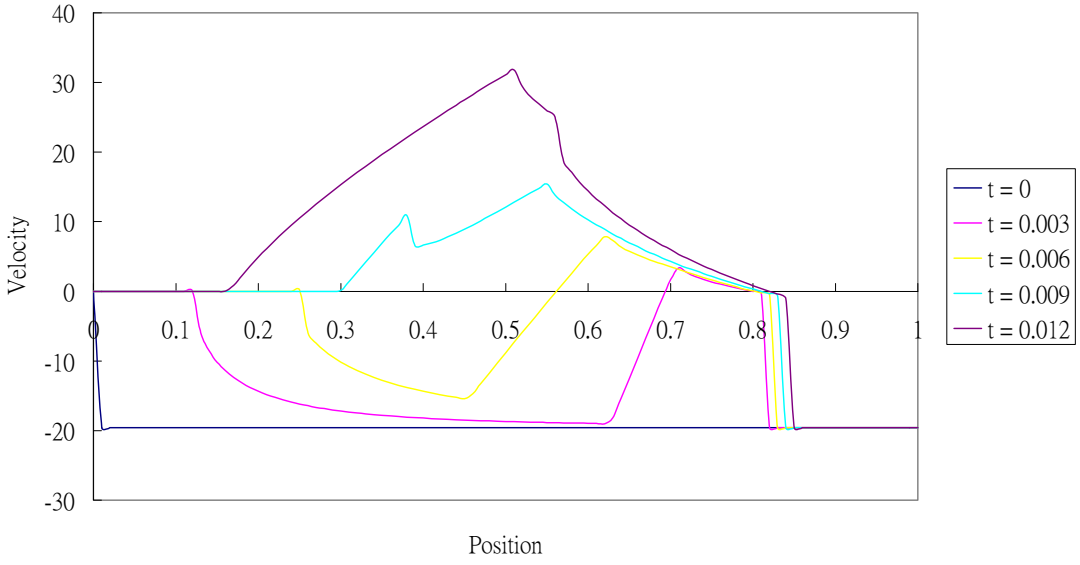


Fig 60. Pressure vs Position (Test 5, Spherical)

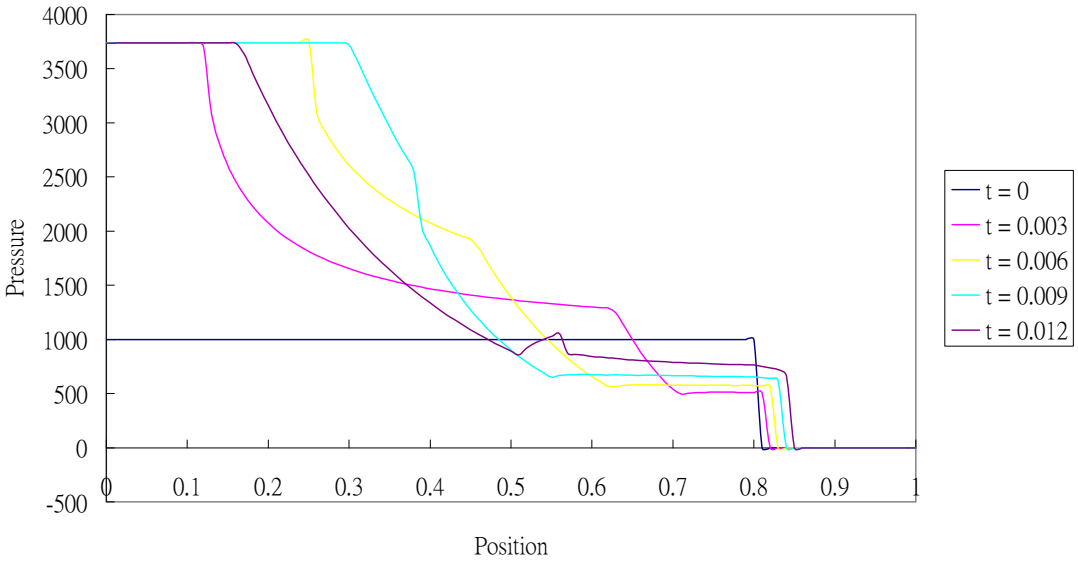
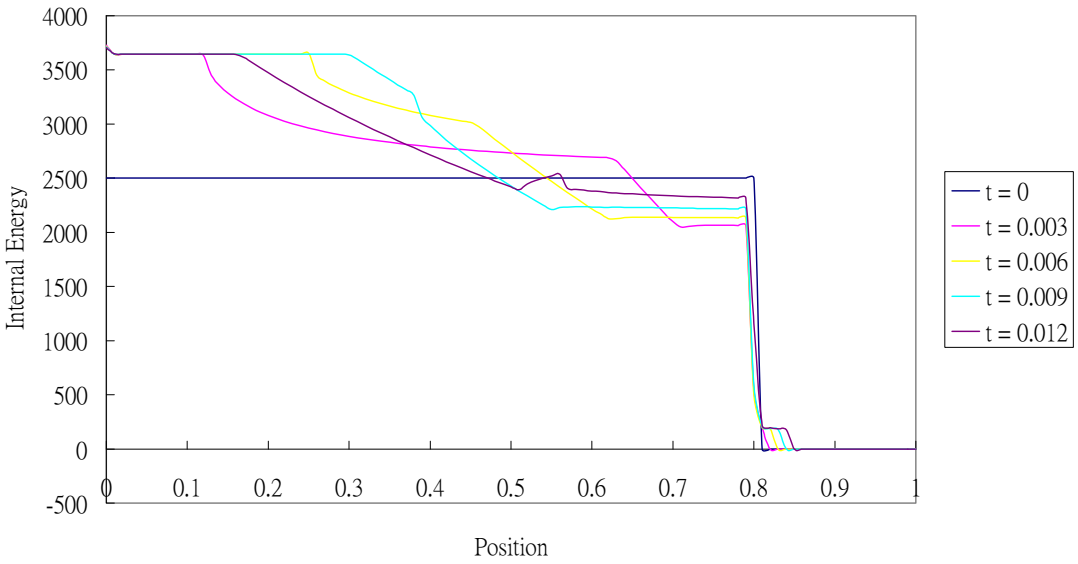


Fig 61. Internal Energy vs Position (Test 5, Spherical)



This test is like Test 3, except the size of the core is larger and the whole gas is contracting. Similar to Test 3, shocks are created at the interface between the core and the envelope (Fig 58). As the whole gas is contracting, the density, pressure and the internal energy of the inner core increase (Fig 58, 60, 61). The high pressure of the inner core will stop the contraction at that inner core, making the velocity at the inner core is zero (Fig 59). Then the inner core starts to expand (Fig 58, 59, 60, 61).

Sedov Explosion Problem

Now, let me use the program to simulate an explosion problem, Sedov Explosion Problem (Sedov 1959). Initially, a unit of energy is put into a small region with radius $r = 0.01$. Then the pressure in this region can be calculated by $p = \frac{3(\gamma - 1)\varepsilon}{(\alpha + 2)\pi r^{\alpha+1}} = 95492.96$. The density is set to $\rho = 1$ throughout the whole space and the pressure is set to $p = 10^{-3}$, except that explosion region $r = 0.01$. Here, I have used the WENO Scheme with $dx = 10^{-3}$ to do the simulation. The program with this resolution requires one week computational time to calculate the density, velocity, pressure and internal energy versus position at $t = 0.05$ using one CPU with $3.00\,\mathrm{GHz}$. Due to the limitation of time and computational equipment, we cannot run the program with higher resolution. The density, velocity, pressure and internal energy versus position at $t = 0.05$ are plotted in Fig 62, 63, 64 and 65, respectively.

Fig 62. Density vs Position (t = 0.05, dx = 0.001)

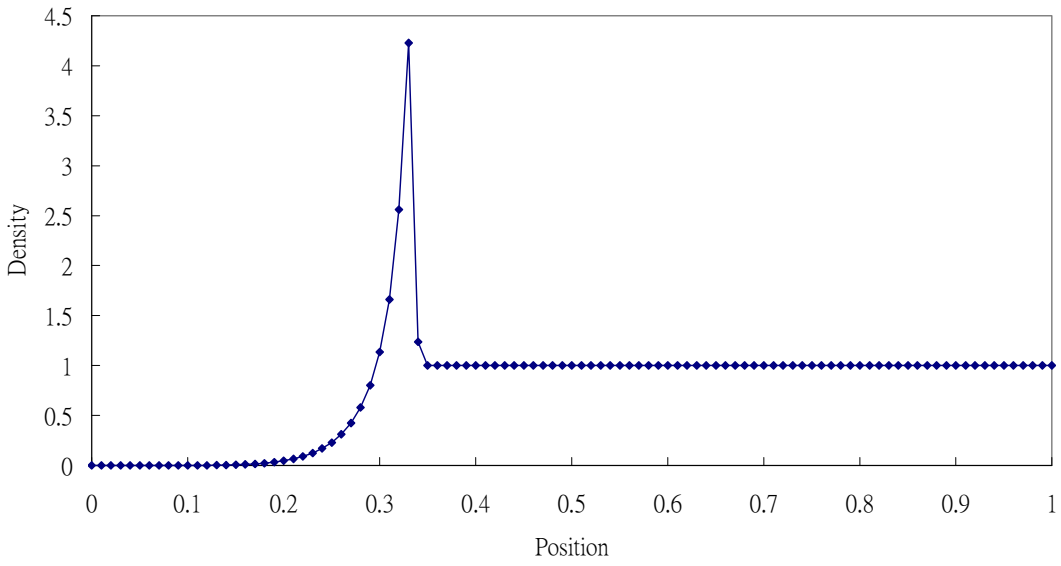


Fig 63. Velocity vs Position (t = 0.05, dx = 0.001)

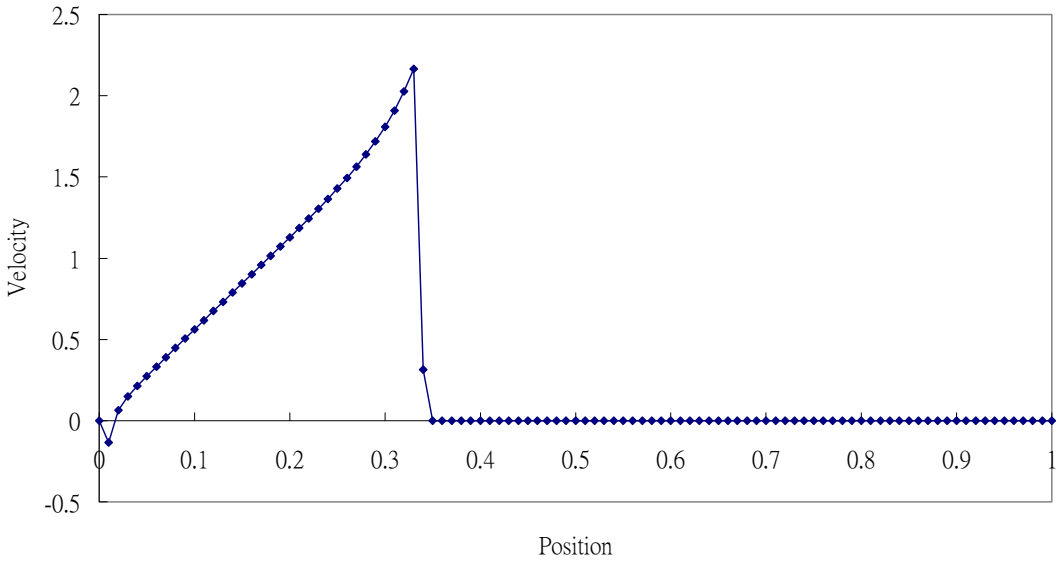


Fig 64. Pressure vs Position (t = 0.05, dx = 0.001)

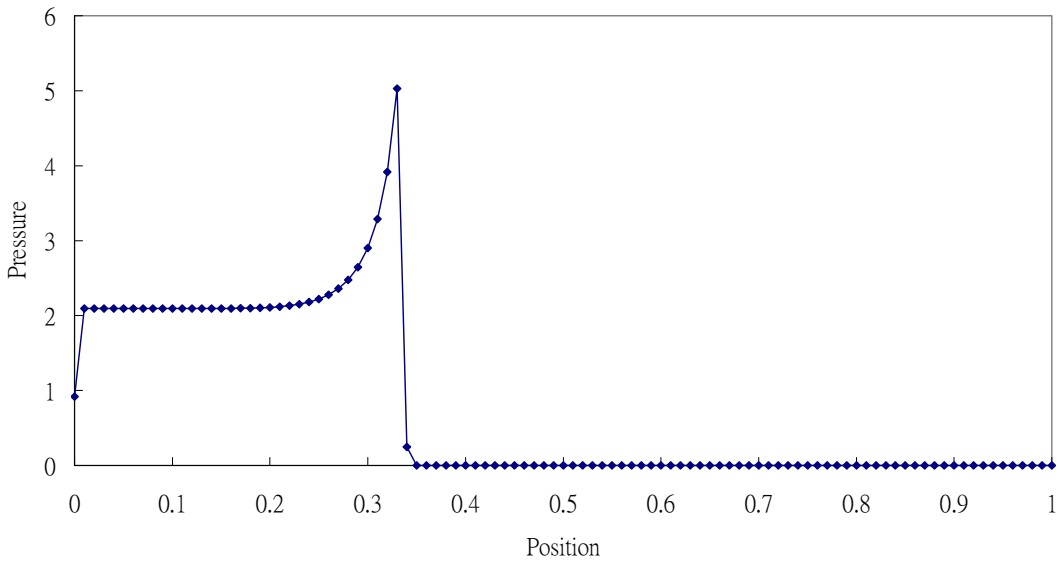
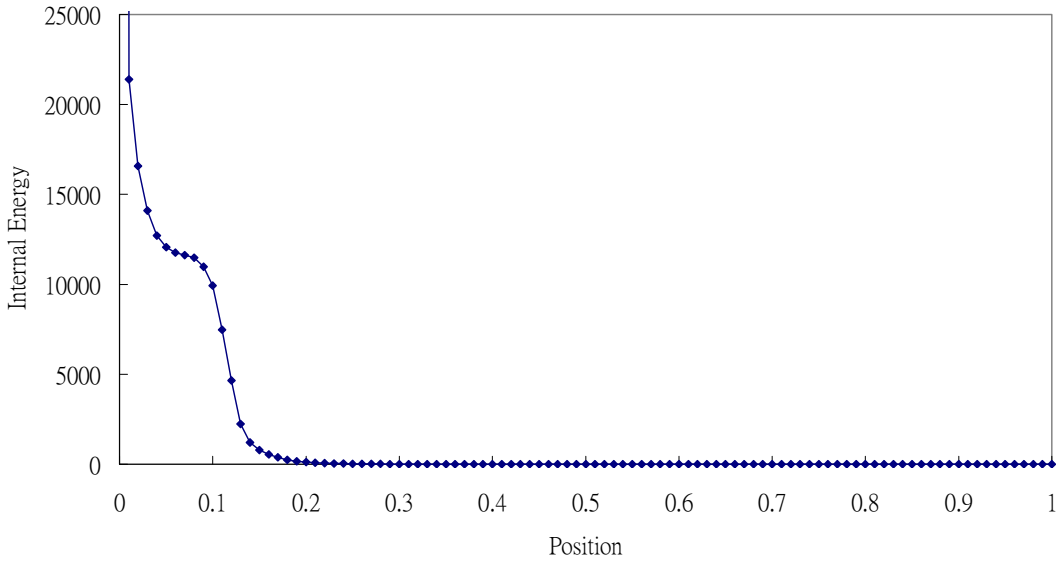
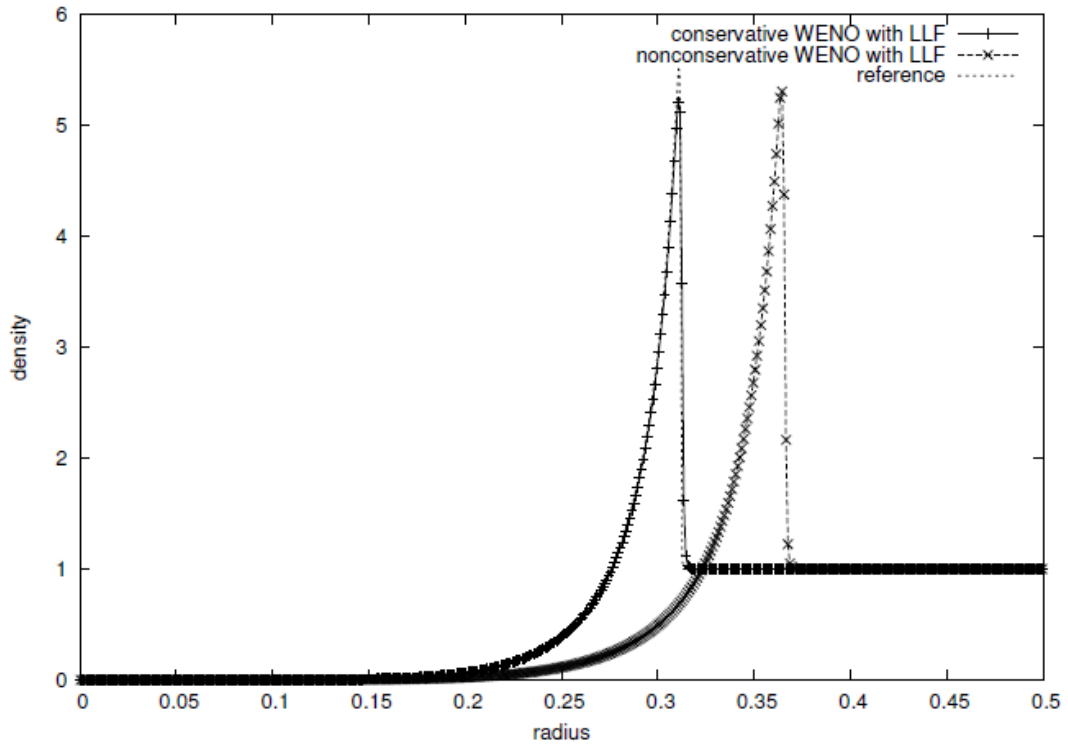


Fig 65. Internal Energy vs Position (t = 0.05, dx = 0.001)



The WENO Scheme I have used here is called non-conservative WENO Scheme in [2]. The density profile at $t = 0.05$ calculated by other WENO Schemes with higher resolution and by the second order HLLC Riemann solve with very fine grids (Reference) in [2] is shown in Fig 66.

Fig 66. Density Profiles for Other WENO Schemes and 2nd Order HLLC Riemann Solve with very Fine Grids (Reference) (Picture adopted in [2])



The conservative WENO Scheme requires nine times more expensive the computational time compared with the non-conservative one. In Fig 62, I get a shock like that in Fig 66. However, due to the low resolution, the peak of the shock in Fig 62 is smaller than that in Fig 66. Initially, the internal energy is unitary. Therefore, the internal energy profile in Fig 65 is not correct. It is because in the program, the evolve variable is the total energy density τ. The internal energy density is then obtained by $\varepsilon = \tau/\rho - v^2/2$. From the density profiles (Fig 62), the density is nearly 0 for $r \le 0.15$. The numerical value of the internal energy at that region will easily blow up (Fig 65).

XI. Conclusion

In this Physics Project, I have used WENO Scheme (fifth order in accuracy) with Runge-Kutta Time Discretization (third order in accuracy) to write program to solve linear advection equation, inviscid Burgers equation, fluid dynamics in one-dimensional Cartesian Coordinates and spherical symmetric fluid dynamics in Spherical Coordinates. In all the tests, the scheme shows its stability, accuracy and its non-oscillatory property near the shocks. The program can simulate the spherical symmetric fluid dynamics in a star if the effect of gravity is added. The equations are:

$$\begin{bmatrix} \rho \\ \rho v \\ \tau \end{bmatrix}_t + \begin{bmatrix} \rho v \\ \rho v^2 + p \\ v(\tau + p) \end{bmatrix}_r = -\frac{\alpha}{r}\begin{bmatrix} \rho v \\ \rho v^2 \\ v(\tau + p) \end{bmatrix} - \begin{bmatrix} 0 \\ \rho \Phi_r \\ \rho v \Phi_r \end{bmatrix},$$

where v is the radial velocity, $\alpha = 2$, $\tau = \rho\varepsilon + \frac{1}{2}\rho v^2$ and $\nabla^2 \Phi = 4\pi G\rho$.

With the equation of state $p = p(\rho, \varepsilon)$, the spherical symmetric fluid dynamics in a star can be simulated.

XII. Reference

1. R. J. LeVeque, “Numerical methods for conservation laws”, 2nd ed. (Birkhauser Basel 1992)
2. S.-T. Li, “WENO Schemes for Cylindrical and Spherical Geometry”, Los Alamos Report LA-UR-03-8922
3. C.-W. Shu, in “Lecture notes in computational science and engineering: high-order methods for computational physics”, ed. T. J. Barth & H. Deconinck (Springer 1999)
4. E. F. Toro, “Riemann solvers and numerical methods for fluid dynamics: a practical introduction”, 2nd ed. (Springer 1999)