Numerical Study of the High-order WENO Scheme

K. W. Wong

I. Introduction

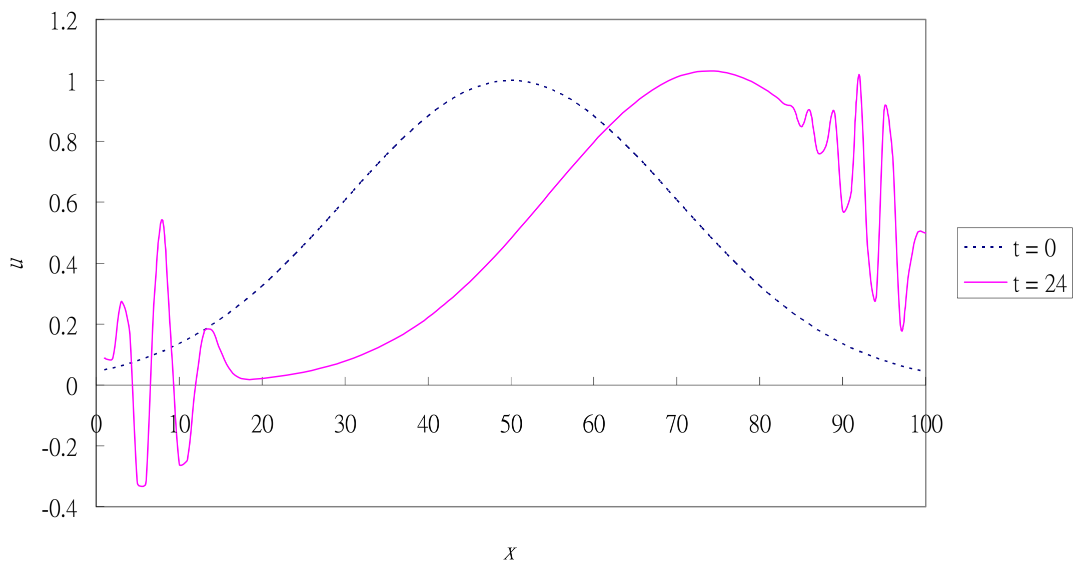
The purpose of my Physics Project in this semester is to study WENO (Weighted Essentially Non-Oscillatory) Scheme, which is a high order accurate finite difference or finite volume scheme designed for problems with piecewise smooth solutions containing discontinuities. WENO Scheme has been quite successful in computational fluid dynamics and other applications, especially for problems containing shocks. Studying this scheme in this semester is a preparation for Physics Project in the next semester, in which I will develop a program using this scheme to solve the spherical symmetric fluid dynamics in a star. Before I discuss the WENO Scheme in this report, I will briefly discuss some basic finite-differencing methods.

II. 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 $t = 24$.

Fig 1. Solving Linear Advection Equation Using Finite Difference Approximation



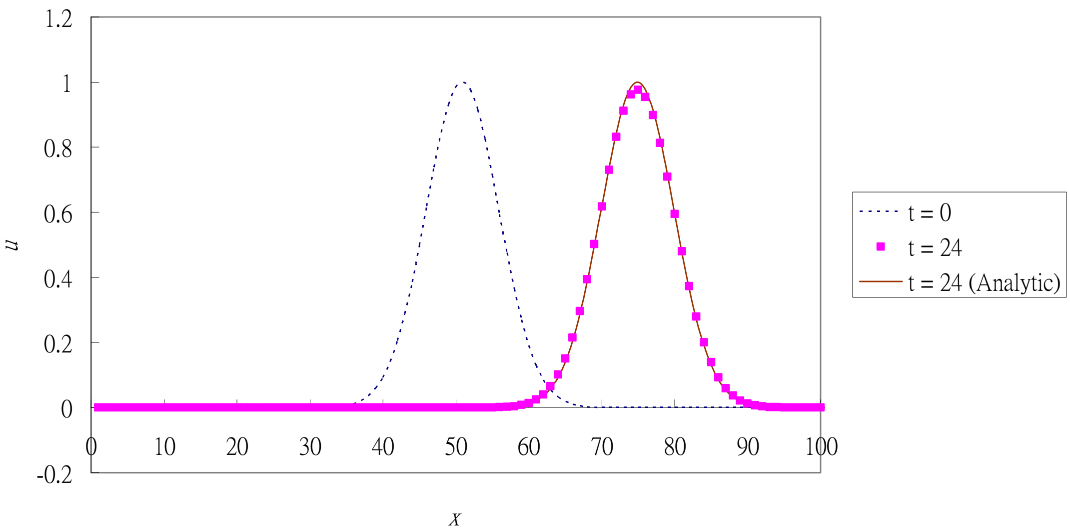
The curve for $t = 24$ should move to the right by 24 units (because I have set $a = 1$) but it turns out that there are some oscillations, which indicate the scheme is unstable.

III. 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.

IV. 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.

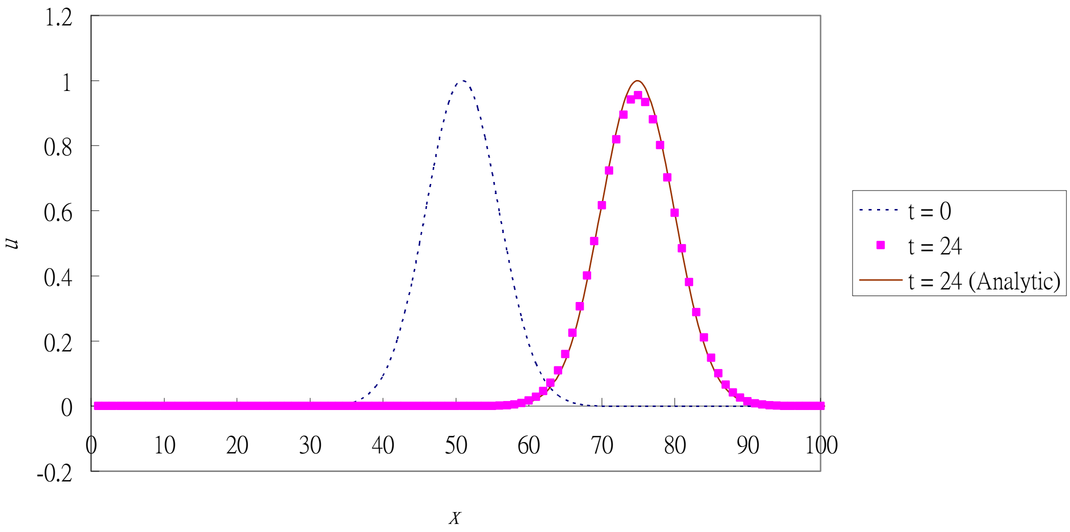
First, write u_t as a subject of the equation, then $u_t = L(u)$. For linear advection equation, $L(u) = -au_x$. Now, u^{n+1} can be calculated by using u^n:

$$
\begin{aligned}
u^{(1)} &= u^n + \Delta t L(u^n) \\
u^{(2)} &= \frac{3}{4}u^n + \frac{1}{4}u^{(1)} + \frac{1}{4}\Delta t L(u^{(1)}) . \\
u^{n+1} &= \frac{1}{3}u^n + \frac{2}{3}u^{(2)} + \frac{2}{3}\Delta t L(u^{(2)})
\end{aligned}
$$

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.

V. 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

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

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

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

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

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

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

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

$\varepsilon = 10^{-6}$ 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

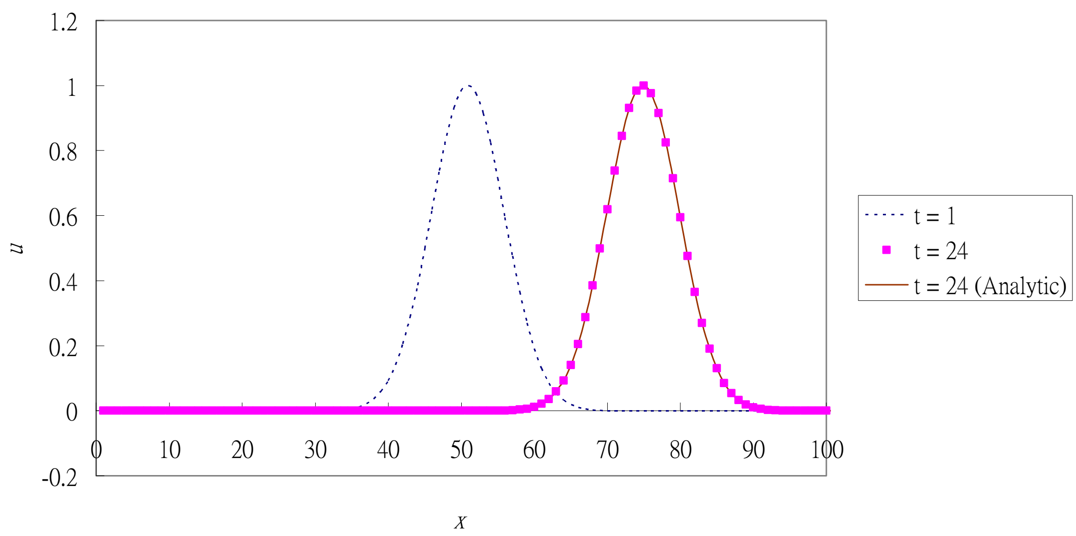
$$\tilde{\omega}_r = \frac{\tilde{\alpha}_r}{\sum_{s=0}^{2} \tilde{\alpha}_s} ,$$

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

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

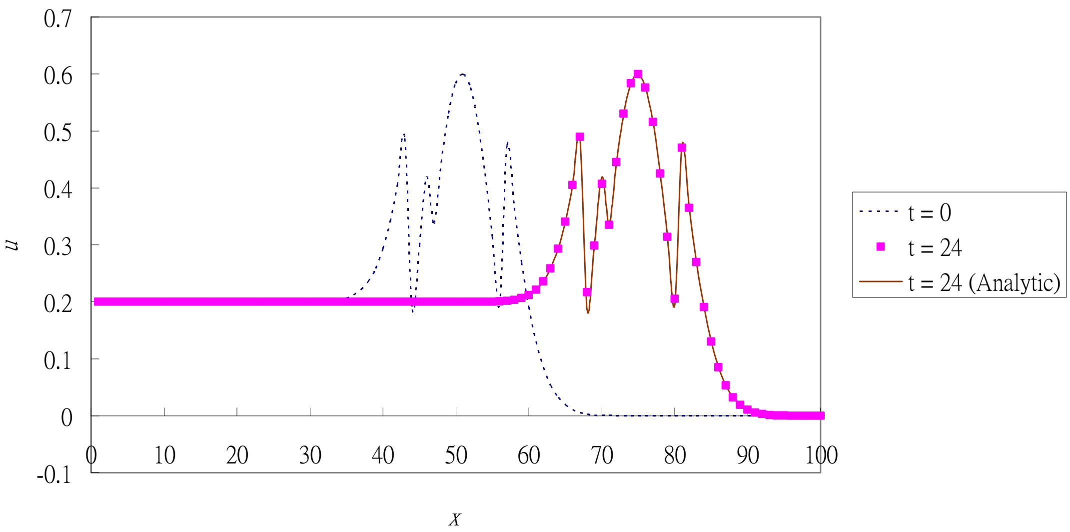
When I apply WENO Scheme together with Runge-Kutta Time Discretization to solve linear advection equation, the result is unimaginably accurate! Fig 4. shows the values of u at different x for $t = 0$ and $t = 24$.

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)



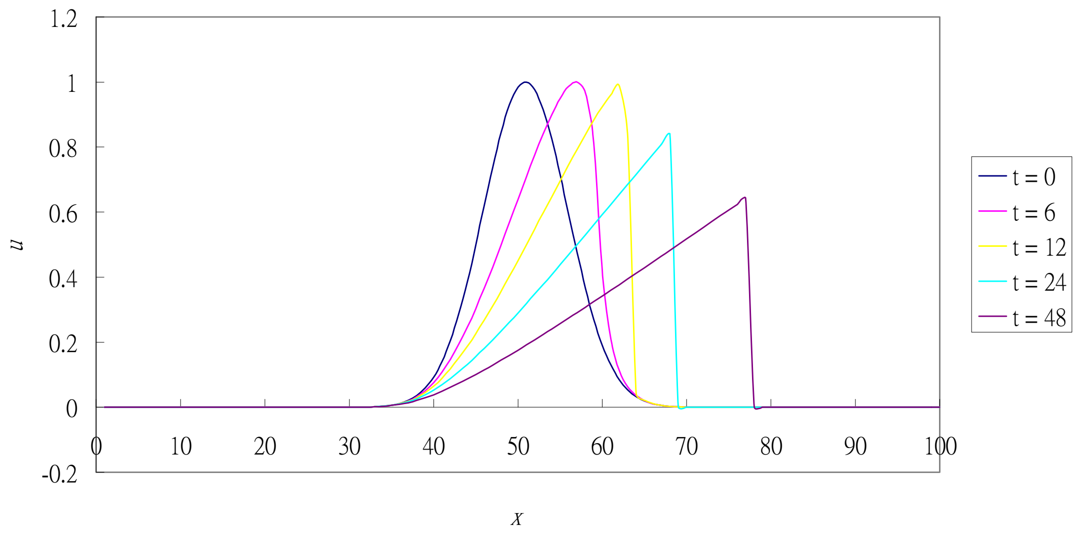
Originally, the initial data is a Gaussian. Then I moved the curve up by 0.2 units for $x \le 46$ and moved the curve down by 0.4 units for $44 \le x \le 56$. This made three jumps to the initial condition. At $t = 24$, the broken Gaussian should move to the right by 24 units (because I have set $a =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

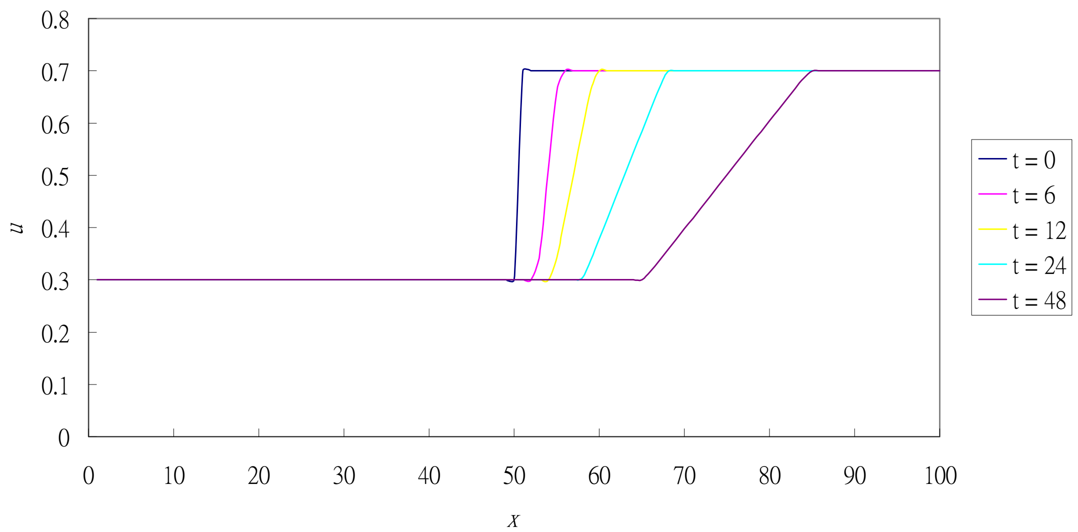


At $t=0$, for $x \geq 51$, the values of u on the left are larger that that of the right. The wave on the left goes to the right with a speed faster than that of the right. Therefore, a shock should be produced.

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)

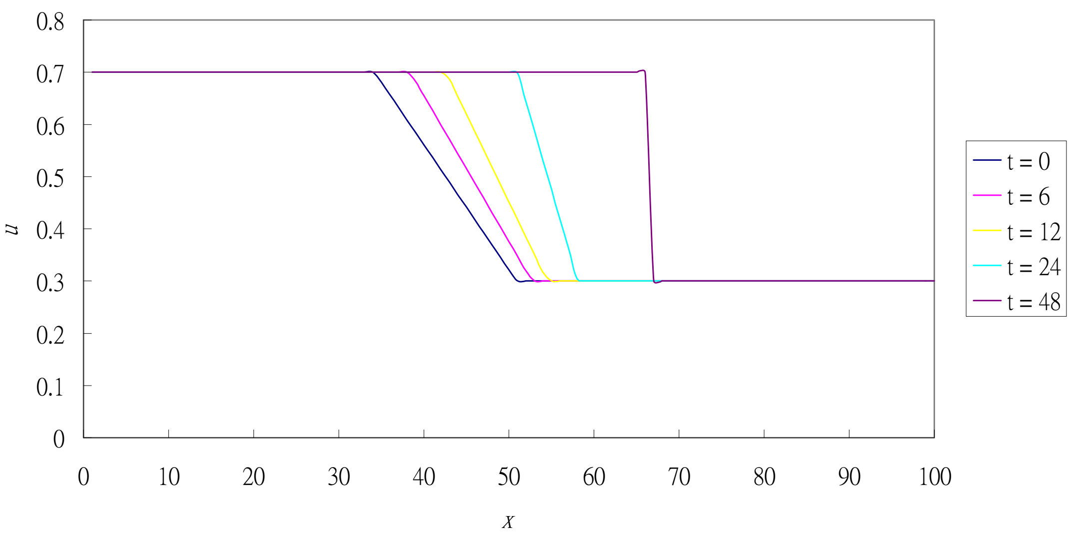


The value of u on the right is larger than that on the left. The speed of the wave on the right goes to the right with a speed larger than that of the left. Therefore, a rarefaction shock is produced in between.

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 $x \leq 33$ and $u = 0.3$ for $x \geq 51$, 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)

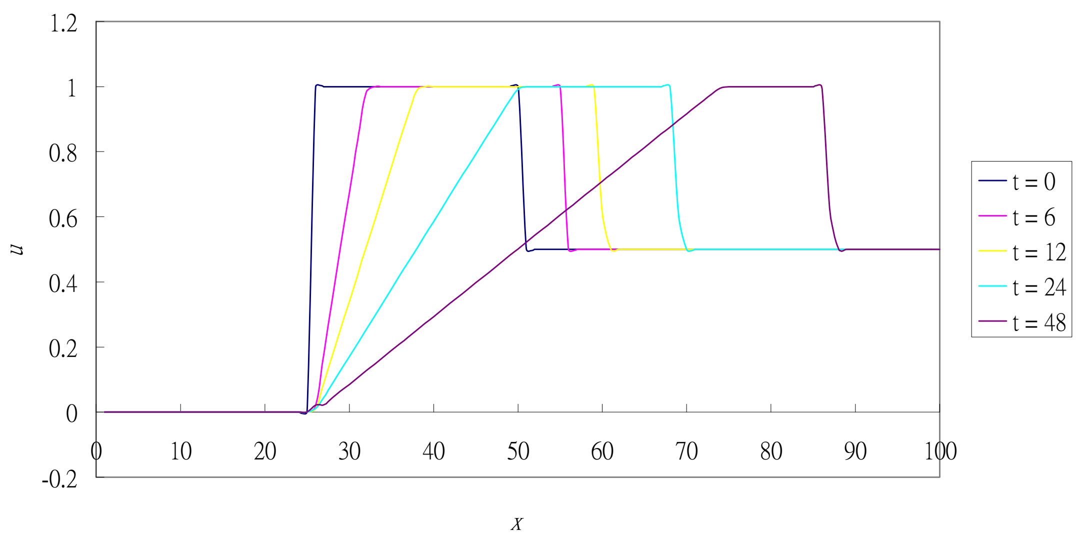


The values of u on the left of the rarefaction are larger than that of the right of the rarefaction. The wave on the left of the rarefaction goes to the right with a speed faster than that of the right of the rarefaction. Therefore, the width of the rarefaction contracts and become a shock at last.

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 $x \leq 25$, $u = 1$ for $26 \leq x \leq 50$ and $u = 0.5$ for $x \geq 51$. 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.

VI. 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:

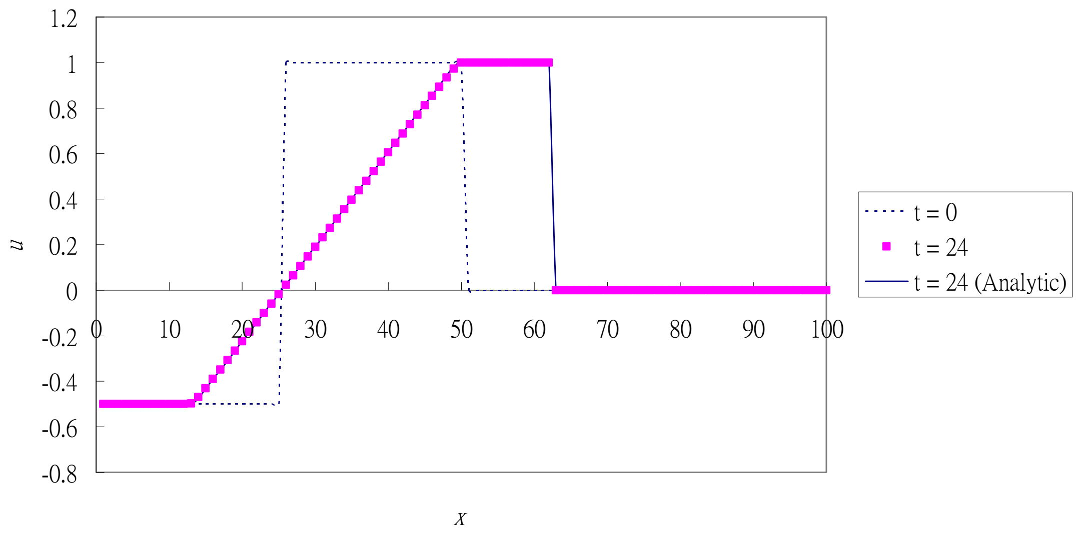
$$f^{\pm}(u) = \frac{1}{2}\left[f(u) \pm \alpha u\right],$$

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.

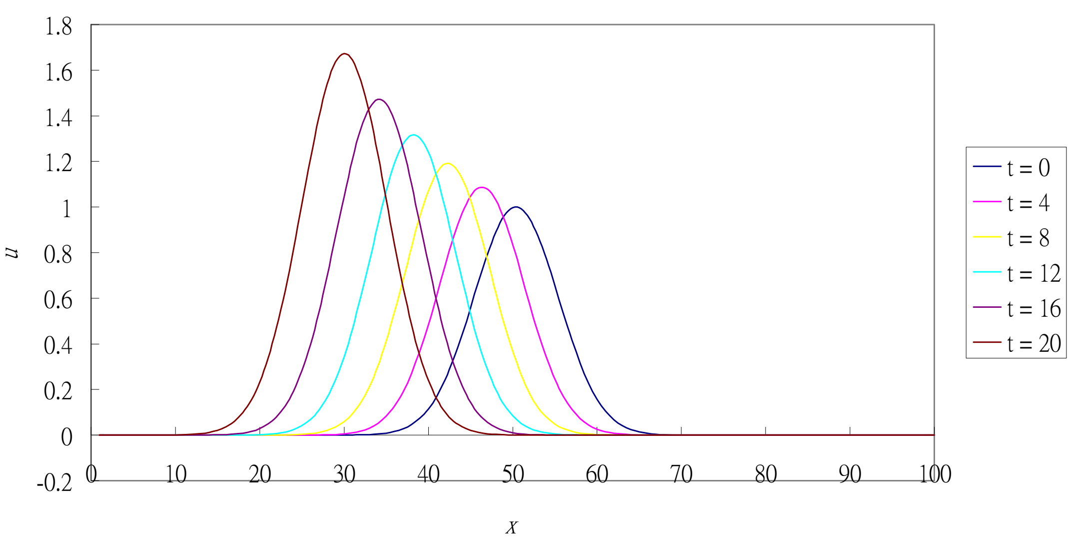
Here, I also plotted the analytical result for $t = 24$ so that I can compare the numerical result with the analytical result to see how accurate the scheme is. The program works very nice! The numerical result is extremely closed to the analytical result. WENO Scheme is stable and accurate even with the existence of shocks and rarefaction shocks.

VII. 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.

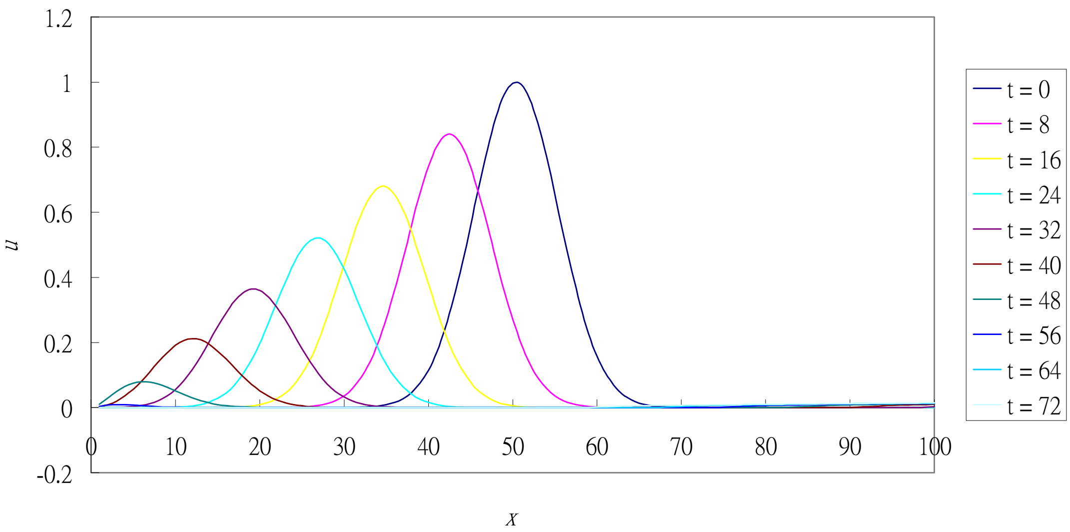
Fig 11. Solving Linear Advection Equation with Source Term u / x
Using WENO Scheme with Runge-Kutta Time Discretization (dx = 0.1, dt = 0.05)



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.

VIII. What's Next?

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. 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 and $\alpha = 2$.

For the Physics Project in the next semester, I will use the WENO Scheme together with the Runge-Kutta Time Discretization to develop a program to solve the spherical symmetric fluid dynamics. With the equation of state and adding the effect of gravity, the spherical symmetric fluid dynamics in a star can be simulated.

IX. Reference

1. R. J. LeVeque, “Numerical methods for conservation laws”, 2^{nd} ed. (Birkhauser Basel 1992)
2. 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)
3. E. F. Toro, “Riemann solvers and numerical methods for fluid dynamics: a practical introduction”, 2^{nd} ed. (Springer 1999)