Final Project Reprort

Paper : An improved weighted essentially non-oscillatory schem for hyperbolic conservation laws

Department of Computational Science & Technology JUNG HO IN hoyin@snu.ac.kr

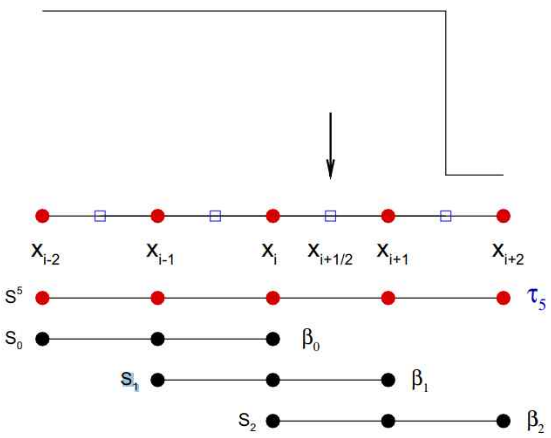
1. Introduction

Although significiant success of classical WENO(WENO-JS) using a dynamic set of stencils with nonlinear convex combination, the classical choice of smoothness indicators in generated weights failed to recover the maximum order of the scheme at points of the solutions where the first or higher derivatives of the flux function vanish. Other article [2] suggesting new method named mapped WENO(WENO-M) delineated this phenomena as “weakness“ in the smooth extrema or near critical points, archiving to recovering the optimal order of convergence at critical points of a smooth function and presented sharper results close to discontinuities. In this article, an improved version of the classical fifth-order WENO scheme(WENO-Z) was introduced having good performance and lower computational cost comparing WENO-M.

2. Theoretical background

1) classical WENO(WENO-JS)

The classical fifth-order WENO scheme uses a 5-points stencil, which is subdiveded into 3-points stencils. The fifth-order polynomial approximation is built through the convex combination of the interpolated values $\hat{f}^k(x_{i+\frac{1}{2}})$, in which $f^k(x)$is the third degree polynomial defined in each one of the stencils S_k.



$\hat{f}_{i\pm\frac{1}{2}} = \sum_{k=0}^{2}\omega_k\hat{f}^k(x_{i\pm\frac{1}{2}})$, where the weights are defiend as $\omega_k = \frac{\alpha_k}{\sum_{l=0}^{2}\alpha_l}$, $\alpha_k = \frac{d_k}{(\beta_k+\epsilon)^p}$.

The coeeficiendts $d_0 = \frac{3}{10}$, $d_1 = \frac{3}{5}$, $d_2 = \frac{1}{10}$ are called the ideal weights, and $p = 2$, $\epsilon = 10^{-6}$

The smoothness indicators β_k measure the regularity of the kth polynomial approximation at the stencil S_k and are given by $\beta_k = \sum_{l=1}^{2}\triangle x^{2l-1}\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}}\left(\frac{d^l}{dx^l}\hat{f}^k(x)\right)^2 dx$

2) The new WENO scheme(WENO-Z)

The enhancements of WENO-Z come from the larger weights that it assigns to discontinuos stencils. Contrary to common belief, the strategy should be to augment the influence of the stencil containing the discontinuity as much as possible, without destroying the essentially non-oscillatory behavior. The novel idea is to use the whole 5-points stencil S^5 to devise a new smoothness indicator of higher order than the classical smoothnes indicators β_k.

Denote $\tau_5 = |\beta_0 - \beta_2|$, and define new smoothness indicators β_z^k as

$\beta_k^z = \left(\frac{\beta_k+\epsilon}{\beta_k+\tau_5+\epsilon}\right)$, $k = 0, 1, 2$, where $\epsilon = 10^{-40}$

and the new WENO weights w_k^z as

$$\omega_k^z = \frac{\alpha_k^z}{\sum_{l=0}^{2}\alpha_l^z}\ \alpha_k^z = \frac{d_k}{\beta_k^z} = d_k\left(1+\frac{\tau_5}{\beta_k+\epsilon}\right),\ k = 0, 1, 2$$

3. Experimental (Reproduction)

1) 1D Euler equation

In this section, several numerical experiments are executed with the on dimensional system of the Euler equations for gas dynamics in strong conservation form :

$Q_t + F_x = 0$ where $\boldsymbol{Q} = (\rho, \rho u, E)^T$, $\boldsymbol{F} = (\rho u, \rho u^2 + P, (E+P)u)^T$,

$P = (\gamma-1)\left(E - \frac{1}{2}\rho u^2\right)$, $\gamma = 1.4$

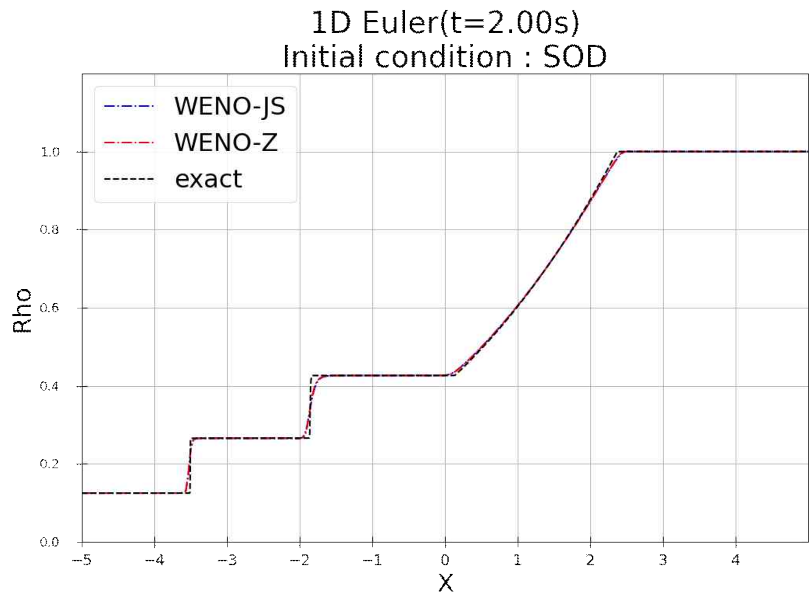
The eigenvalues and eighenvectors are obtained via the linearized Riemann solver of Roe and the first-order Lax-Friedrichs flux is used as the low order building block for the high-order reconstruction step of the WENO schemes. After projecting the fluxes on the characteristic fields via the left eigenvectors, the high-order WENO reconstruction step is applied to obtain the high-order approximation at the cell boundaries, which are projected back into the physical space via the right eigenvectors.

All reproduction was executed under same condition $CFL=0.4$, $N=300$ (for blastwave problem, $N=400$, reflective boundary condition), and exact solution was obtained $CFL=0.4$, $N=4000$ with free stream inflow and outflow boundary condition

2) Riemann initial value problems

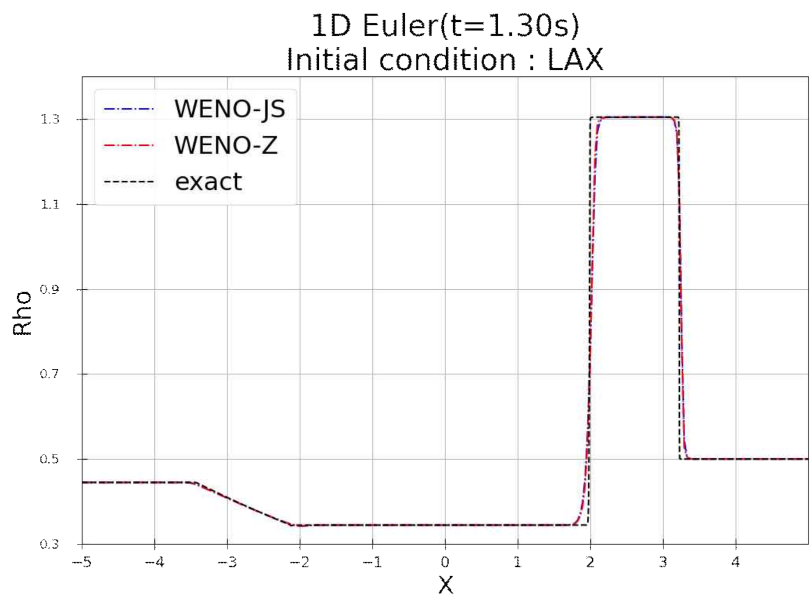
i) SOD Problem

$$(\rho, U, P)=\begin{cases}(0.125 & 0 & 0.1) & -5 \leq x<0 \\ (0.1 & 0 & 1) & 0 \leq x<5\end{cases} \quad \text{and the finial time is } t=2$$



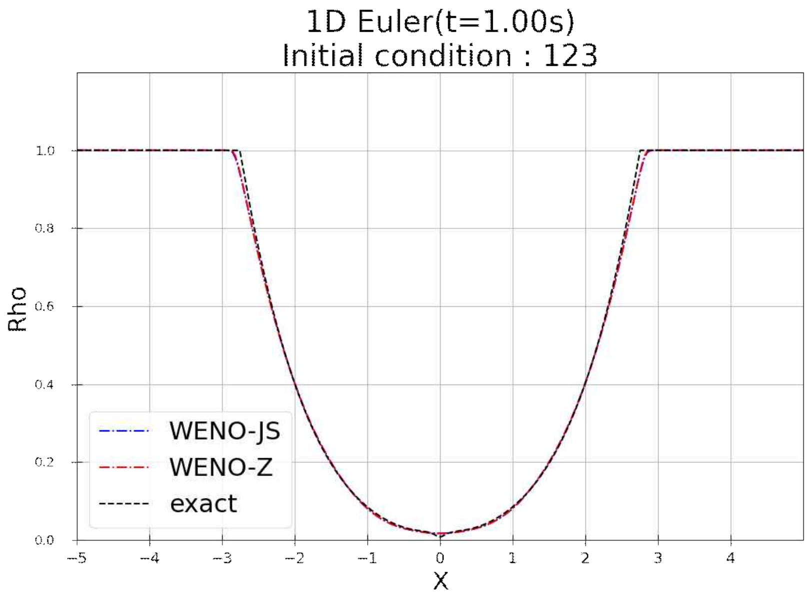
ii) Lax Problem

$$(\rho, U, P)=\begin{cases}(0.445 & 0.698 & 3.528) & -5 \leq x<0 \\ (0.5 & 0 & 0.571) & 0 \leq x \leq 5\end{cases} \quad \text{and the final time is } t=1.3$$



iii) 123 Problem

$(\rho, U, P) = \begin{cases} (1 \quad -2 \quad 0.4) & -5 \leq x < 0 \\ (1 \quad 2 \quad 0.4) & 0 \leq x \leq 5 \end{cases}$ and the final time is $t = 1$

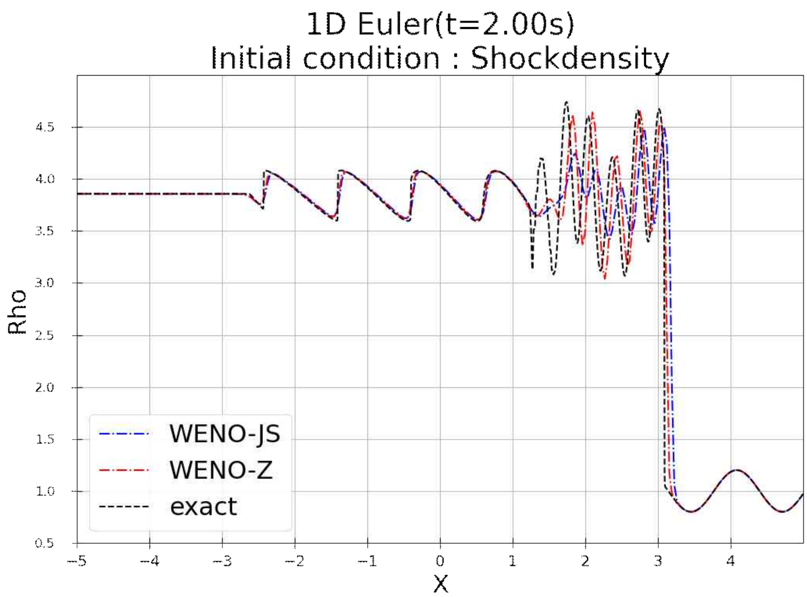


3) Shock-density wave interaction

Consider Mach 3 shock-entropy wave interaction, specified by the following initial conditions.

$$(\rho, U, P) = \begin{cases} (3.857143 \quad 2.629369 \quad \frac{31}{3}) & -5 \leq x < -4 \\ (1 + 0.2\sin(kx) \quad 0 \quad 1) & -4 \leq x \leq 5 \end{cases}$$

with $k = 5$. and the final time $t = 2$.

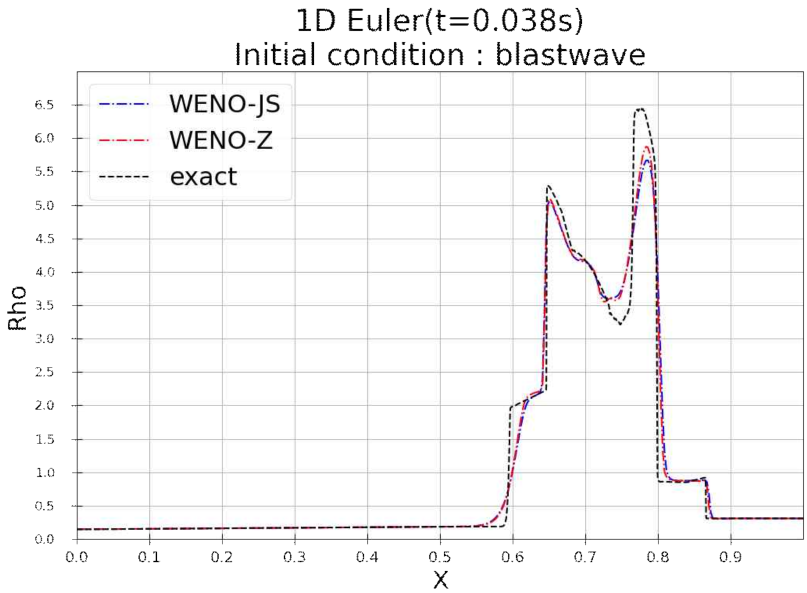


4) Interacting blastwaves

The one dimensinal blastwaves interaction problem which its initial pressure gradients generate two density shock waves that collide and interact later in time.

$$(\rho, U, P) = \begin{cases} (1 \quad 0 \quad 1000) & 0 \le x < 0.1 \\ (1 \quad 0 \quad 0.01) & 0.1 \le x < 0.9 \\ (1 \quad 0 \quad 100) & 0.9 \le x \le 1.0 \end{cases}$$

and the final time $t = 0.038$



4. Code (Python)

This code refer to [4] as baseline about classical WENO for 1D Euler equation. I modified [4] for several initial values and WENO-Z and fixed some bugs.

1) Entire structure of program

main.py	run on the linux terminal as "time python main.py"
euler_1d_weno.py	imported to main.py, including serveral equations about WENO, Euler etc.
plot_graph.ipynb	plot graph of result of WENO-Z, WENO-JS call *.csv of /result_csv
<dir> result_csv	automatically made by 'main.py'. save the result of simulation as *.csv file
<dir> result_graph	automatically made by 'plot_graph.ipynb' save the plot result as *.png file

2) Code analysis

File name	Works	Detail
main.py	Initialize	Initialize all array and list for calculate. And setup initial condition and boudary condition
	Running on terminal	Run on the linux terminal as "time python main.py" Initial condition will be displayed and start iteration
	Main iteration	Calculate wave speed and execute WENO protocol
	Save result	Make directory to save result. And save simulation result as *.csv consisting of x-rho and time data
euler_1d_weno.py	Initial condition	Setup detailed initial condition, including constants and physical state variables of Euler equation, $q = [\rho, \rho u, E]^T$. E is obtained by P, ρ, γ, u
	Physical flux	Define and calculate physical flux terms f from q
	Wave speed	Calculate wave speed from q
	Update ghost points	Calculate q of ghost points around boundary. It would depends on boundary condition.
	project_to_char	Get left eigenvectors of A and project solution/flux into characteristic space for each point in stencil
	numerical flux at characteristic space	Loop through each $x_{i+\frac{1}{2}}$ point on grid, and compute the $i+\frac{1}{2}$ points flux from WENO-JS or WENO-Z
	WENO-JS, WENO Z	Function which computes a 5th-order WENO reconstruction of the numerical flux at location $x_{i+\frac{1}{2}}$.
	Update RHS	Compute the R matrix at every half point flux location. And update the RHS values of grids from characteristic form of numerical flux.
plot_graph.ipynb	Environment	It works on Python Jupyter Notebook to deal with dataframe and to make figure of final result.
	Read csv	read *.csv data from (/result_csv) made in main.py, and turn into numpy array to plot figure.
	Make figure	Make plot following initial condition and save the plot into the directory (/result_graph)

5. Conclusions

Based on the results alone, the final equations to obtain weights and smoothness indicators of WENO-JS and WENO-Z are not much different. τ_5 - the absolute diffrence between β_0, β_2- was just added to define new smoothness indicator β_z^k. As shown above results, the slight change made considerable gap between two schemes. The results of WENO-Z can represent numerical solution more than ones of WENO-JS in addition to convege faster at shocklet.. Moreover, contrary to common belief, the strategy allocating larger weights to discontinuous stencils might hard to devise for anyone who research this subject. It means that the authors of this article have amazing intuition.

Although reproducing WENO-Z scheme via Python was terrific experience for me, some obstacles existed because of the paper. At first, the Lax problem in this article, pressure of lower section was introduced as 0.3528. However, the result of this initial condition is not same with figure in article. Furthermore, other paper solved this problem by setting the pressure 3.5278 or 3.528. For this reason, I should fixed my correct and fine code for nothing and search other paper[4] to find right initial value to handle this problem. Secondly, some boundary conditions are not given for 1D Euler equation, except interacting blastwaves. For interacting blastwaves, the reflective boundary condition was commented. I set the boundary condition as inflow-outflow boundary condition(non-reflecting condition), and I could get same result with paper.

When I obtain the exact solution for 1D shock-tube problems, I tried to compute the analytical solution. However, this work is totally different from calculating WENO method and the functions to get exact solution in the baseline does not work well. And then, I obtained exact solution by setting numerous grids, similar with the article(N=4000). Different point is that WENO-M method was used as exact solution in this paper, I switched it to WENO-Z to reuse my own code. While the results of exact solution between the paper and mine was almost equal, the figure in Shock-density problem was slightly different; The shocklet occured earlier than ones of WENO-M.

Because of my poor skills and knowledge, I could not reproduce WENO-M method for 1D Euler equations neither WENO-Z for 2D Euler equations. Based on this project, I will gradually revise my code to cover several method using WENO-M and covering 2D Euler equation.

6. Reference

[1] G.S. Jiang, C.W. Shu, Efficient implementation of weighted ENO schemes, J. Journal of Computational Physics, 1996, 126:202-228

[2] A.K. Henrick, T.D. Aslam, J.M. Powers, Mapped weighted essentially non-oscillatory schemes: achieving optimal order near critical points, J. Journal of Computational Physics, 2005, 207:542-567

[3] Borges R, Carmona M, Costa B, Don W S, An improved weighted essentially non-oscillatory scheme for hyperbolic conservation laws, J. Journal of Computational Physics, 2008, 227:3191-3211.

[4] Lin Fu. A very-high-order TENO scheme for all-speed gas dynamics and turbulence, J. Journal of Computational Physics, 2019, 244:117-131

[5] B.Sullivan, S. R. Murthy, University of Illinois (2018), https://github.com/btsllvn2/weno-1d-euler