Journal of Interdisciplinary Mathematics
ISSN: 0972-0502 (Print), ISSN: 2169-012X (Online)
Vol. 28 (2025), No. 6, pp. 2197–2205
DOI : 10.47974/JIM-2361

High-order finite difference schemes for hyperbolic conservation laws

Cholaraja Mudimannan †
Campus Law Centre
Faculty of Law
University of Delhi
Delhi
India

Gauri Patil *
Department of Civil Engineering
Dr. D. Y. Patil Institute of Technology
Pimpri-Chinchwad
Pune 411018
Maharashtra
India

Suraj Bhan §
School of Engineering & Technology
Noida International University
Noida 203201
Uttar Pradesh
India

Suhas Bhise ‡
Department of Electronics and Telecommunication Engineering
Vishwakarma Institute of Technology
Pune 411037
Maharashtra
India

† *E-mail:* **drcholaraja@clc.du.ac.in**
* *E-mail:* **gauri.patil@dypvp.edu.in (Corresponding Author)**
§ *E-mail:* **suraj.bhan@niu.edu.in**
‡ *E-mail:* **suhas.bhise@vit.edu**

Balkrishna K. Patil [@]
Department of Computer Engineering
Sandip Institute of Technology and Research Centre (SITRC)
Nashik 422213
Maharashtra
India

J. Mounika [#]
Department of Information Technology
KKR and KSR Institute of Technology and Sciences
Guntur 522017
Andhra Pradesh
India

Abstract

To correctly solve hyperbolic conservation laws, which often have shocks and discontinuities, you need high-order finite difference methods. Numerical spread and instability are problems with traditional low-order methods. This is why more advanced methods like ENO and WENO were created. These methods get very accurate results in areas with flat surfaces while stopping variations near steep slopes. This makes sure that the models are strong and work well. The study talks about the basics of math, rules for stability and consistency, the structure of algorithms, and how to treat boundaries. Numerical tests show that accuracy, shock absorption, and stability gaps are all getting better.

Subject Classification: *65L12, 35L65.*

Keywords: *Hyperbolic conservation laws, High-order finite difference, Stability and consistency, ENO/WENO schemes.*

1. Introduction

Hyperbolic conservation rules are very important for modelling physical processes where information moves at limited speeds, like shock waves in fluid dynamics, traffic flow, shallow water systems, and the spread of electromagnetic waves [1]. These systems are controlled by nonlinear partial differential equations, and their answers can become discontinuous even if the starting conditions are smooth. Classical numerical methods often can't pick up on such sharp features without adding unwanted variations, too much numerical diffusion, or instability [2, 3]. To solve this problem, we need to come up with high-order finite difference schemes that are accurate, stable, and reliable in both smooth and non-smooth solution conditions. Figure 1 shows finite difference grid links fluxes and derivatives. In the last few decades, a lot of

[@] *E-mail:* **balkrishnapatileng@gmail.com**
[#] *E-mail:* **jmounika@outlook.com**

progress has been made in creating methods like ENO (Essentially Non-Oscillatory), WENO (Weighted ENO), and discontinuous Galerkin-inspired finite difference approaches [4].

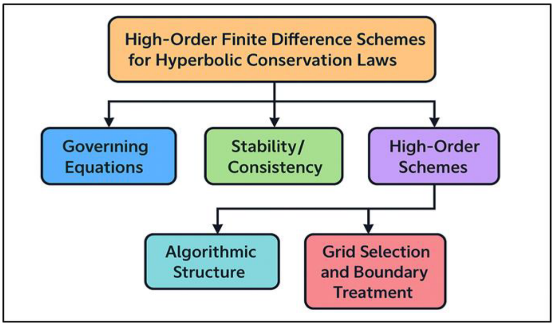

Figure 1
Architecture of High-Order Finite Difference Schemes for Hyperbolic Conservation Laws

These methods try to get very accurate results while reducing non-physical oscillations near breaks [5].

2. Mathematical Preliminaries

2.1. *Governing equations of hyperbolic conservation laws*

Hyperbolic conservation rules describe physical systems that move like waves and are controlled by nonlinear PDEs. Because solutions can have breaks in them, numerical methods are needed to make sure that shocks, rarefactions, and contact breaks are shown correctly [6].

General form of conservation law:

$$\frac{\partial u}{\partial t} + \nabla \cdot F(u) = 0 \tag{1}$$

In one dimension:

$$\frac{\partial u}{\partial t} + \frac{\partial f(u)}{\partial x} = 0 \tag{2}$$

Initial condition:

$$u(x,0) = u^0(x), \;\; x \in \mathbb{R} \tag{3}$$

Weak formulation:

$$\int u(x,t)\, \varphi_t + f\big(u(x,t)\big)\varphi_x dx = 0 \tag{4}$$

Flux Jacobian:

$$A(u) = \frac{\partial f(u)}{\partial u} \tag{5}$$

Hyperbolicity condition:

$$Eigenvalues(A(u)) \in \mathbb{R} \tag{6}$$

Example (linear advection):

$$u_t + a\, u_x = 0, \;\; a \in \mathbb{R} \tag{7}$$

Entropy condition:

$$\eta(u)_t + q(u)_x \leq 0 \tag{8}$$

Riemann problem:

$$u(x,0) = \{ u\_L,\ x < 0u\_R,\ x > 0 \} \tag{9}$$

Conservation property:

$$d/dt \int \Omega\, u(x,t)\, dx = f(u)|\partial\Omega \tag{10}$$

Shock formation:

$$s = \frac{(f(u_R) - f(u_L))}{(u_R - u_L)} \tag{11}$$

Rankine–Hugoniot condition:

$$[u] \cdot s = [f(u)] \tag{12}$$

2.2. *Stability and consistency considerations*

Stability makes sure that numerical solutions stay within a certain range when discretisation is used, and consistency makes sure that schemes get close to the governing equations correctly [7, 8, 9].

Semi-discrete scheme:

$$\frac{du_i}{dt} = -\frac{\left(F_{\{i+\frac{1}{2}\}} - F_{\{i-\frac{1}{2}\}}\right)}{\Delta x} \tag{13}$$

Fully discrete scheme:

$$u_i^{\{n+1\}} = u_i^n - \left(\frac{\Delta t}{\Delta x}\right)\left(F^n_{\{i+\frac{1}{2}\}} - F^n_{\{i-\frac{1}{2}\}}\right) \tag{14}$$

Consistency requirement:

$$\lim(\Delta t, \Delta x \to 0)\, L\Delta(u) = L(u)$$

Stability condition:

$$||u^n|| \leq C,\ \forall n \tag{15}$$

CFL condition:

$$\Delta t \leq \frac{\Delta x}{\max|\lambda(A)|} \tag{16}$$

Von Neumann analysis (Fourier mode):

$$u_i^n = \xi^n e^{i\,k\,x_i} \tag{17}$$

Amplification factor:

$$G(k) = \frac{u^{\{n+1\}}}{u^n} \tag{18}$$

Stability criterion:

$$|G(k)| \leq 1,\ \forall k \tag{19}$$

Example (FTCS for advection):

$$G(k) = 1 - i\,a\left(\frac{\Delta t}{\Delta x}\right)\sin(k\Delta x) \tag{20}$$

3. High-Order Schemes for Hyperbolic Systems

High-order finite difference schemes, like ENO and WENO, work well in smooth areas while blocking unwanted variations close to edges [10]. For stable models of hyperbolic systems, their design strikes a good mix between precision, stability, and computing speed.

Spatial discretization:

$$u_i^{\{n+1\}} = u_i^n - \left(\frac{\Delta t}{\Delta x}\right)\left(\hat{f}_{\left\{i+\frac{1}{2}\right\}} - \hat{f}_{\left\{i-\frac{1}{2}\right\}}\right) \tag{21}$$

Numerical flux:

$$\hat{f}_{\left\{i+\frac{1}{2}\right\}} = f\left(u_{\left\{i+\frac{1}{2}\right\}}\right) + \ stabilization\ term$$

Polynomial reconstruction:

$$p_{i(x)} = \Sigma\, c_j \varphi_{j(x)}, \deg(p) = k \tag{22}$$

ENO scheme:

$$\hat{f}_{\left\{i+\frac{1}{2}\right\}} = f\left(p_i\left(x_{\left\{i+\frac{1}{2}\right\}}\right)\right) \tag{23}$$

WENO flux weighting:

$$\hat{f}_{\left\{i+\frac{1}{2}\right\}} = \Sigma\, \omega_k f_{k\left(\left\{i+\frac{1}{2}\right\}\right)} \tag{24}$$

Smoothness indicators:

$$\beta_k = \Sigma\, (\Delta^l f_k)^2 \tag{25}$$

Nonlinear weights:

$$\omega_k = \frac{\alpha_k}{\Sigma\, \alpha_j}, \quad \alpha_k = \frac{d_k}{(\varepsilon+\beta_k)^p} \tag{26}$$

Fifth-order WENO flux:

$$\hat{f}_{\left\{i+\frac{1}{2}\right\}} = \Sigma\, \omega_k \hat{f}_k \tag{27}$$

Time discretization (Runge–Kutta stage 1):

$$u^1 = u^n + \Delta t\, L(u^n) \tag{28}$$

4. Numerical Implementation

4.1. *Algorithmic structure of high-order schemes*

Runge–Kutta methods are used for time integration, nonlinear scaling, and flux reconstruction in the procedure. Each step makes sure that waves travel correctly, keeps things stable, and responds well to situations with many dimensions and complex system dynamics.

Algorithm: Runge–Kutta with Nonlinear Weights and Flux Reconstruction

$$u_i^{\{n+1\}} = u_i^n - \left(\frac{\Delta t}{\Delta x}\right)\left(\hat{f}_{\left\{i+\frac{1}{2}\right\}} - \hat{f}_{\left\{i-\frac{1}{2}\right\}}\right) \tag{29}$$

Finite-volume update: flux difference drives conservative evolution per timestep across cell interfaces.

$$\hat{f}_{\{i+\frac{1}{2}\}} = \Sigma\, \omega_k f_k\, x_{\{i+\frac{1}{2}\}} \tag{30}$$

Nonlinear weighted combination of candidate fluxes at the cell interface to reconstruct.

$$\omega_k = \frac{\alpha_k}{\Sigma\, \alpha_j} \tag{31}$$

Normalize smoothness indicators into convex weights summing to one for stable blending

$$\alpha_k = \frac{d_k}{(\varepsilon + \beta_k)^p} \tag{32}$$

Nonlinear scaling reduces weights near discontinuities using smoothness measure and exponent control.

$$\beta_k = \Sigma_{\{l=1\}}^{r} \int (\Delta^l f_k)^2 dx \tag{33}$$

Smoothness indicator from squared derivatives, penalizing oscillations over stencil to guide weights

$$p_{i(x)} = \Sigma\, c_j \varphi_{j(x)} \tag{34}$$

Local polynomial reconstruction using basis functions and coefficients from data on stencil

$$\hat{f}_{\{i+\frac{1}{2}\}} = f\left(p_{i\left(x_{\{i+\frac{1}{2}\}}\right)}\right) \tag{35}$$

Evaluate physical flux using reconstructed state at the interface for numerical update

$$u^1 = u^n + \Delta t\, L(u^n) \tag{36}$$

First explicit Runge–Kutta stage advancing solution using residual operator from current state

$$u^2 = \tfrac{3}{4}\, u^n + \tfrac{1}{4}\big(u^1 + \Delta t\, L(u^1)\big) \tag{37}$$

Second stage blends old and updated states to enhance stability and accuracy.

$$u^{\{n+1\}} = \tfrac{1}{3}\, u^n + \tfrac{2}{3}\big(u^2 + \Delta t\, L(u^2)\big) \tag{38}$$

Final SSPRK stage achieves third-order accuracy while preserving monotonicity and strong-stability properties.

4.2. *Grid selection and boundary treatment*

The grid precision affects how well steep slopes are captured. When boundary conditions are periodic, flowing, or reflecting, they need to be carefully handled numerically to avoid false reflections and keep conservation and stability across all computing domains.

5. Result and Analysis

The data in Table 1 shows that high-order schemes are more accurate than low-order methods. The L1 and L∞ mistakes are bigger in the 2nd-Order FD, but they are much smaller in ENO-3.

Table 1

Accuracy Comparison of Numerical Schemes

Scheme	Order (p)	L1 Error ($\times 10^{-3}$)	L∞ Error ($\times 10^{-3}$)	CPU Time (s)
2nd-Order FD	2	12.4	18.7	1.2
ENO-3	3	6.8	10.1	2.1
WENO-5	5	2.3	4.6	3.7
WENO-7	7	1.1	2.2	5.6

In figure 2, the WENO-5 and WENO-7 models are even more accurate, recording small solution details with very little mistake.

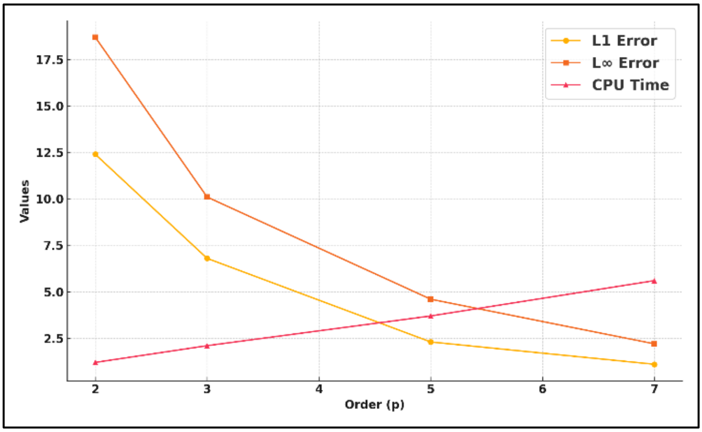

Figure 2
Error and Computation Time vs. Scheme Order

But more accuracy means more work to do on the computer, showing the balance between accuracy and speed when creating strong number schemes (In figure 3).

6. Conclusion

High-order finite difference schemes are very useful for figuring out hyperbolic conservation laws because they make sure accuracy in smooth areas and safety near breaks. The complexity of wave events is well captured by these methods, which combine advanced flux reconstruction, stability analysis, and careful boundary treatments. Because they are flexible and reliable, they are essential for computational physics, fluid dynamics, and engineering.

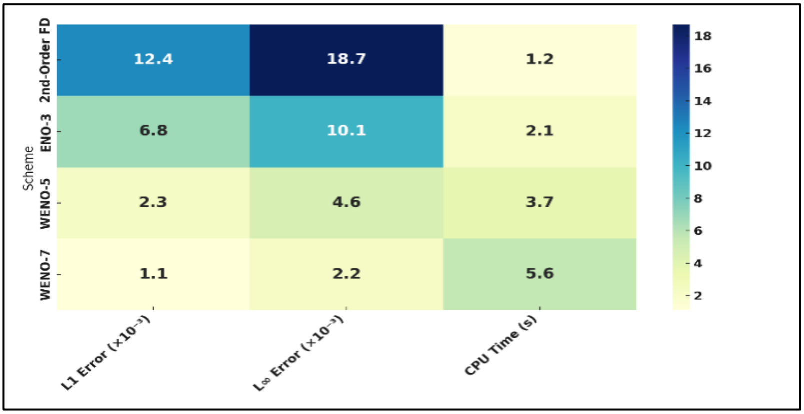

Figure 3
Performance Metrics of Numerical Schemes

References

[1] Z. Zhao, Y.-T. Zhang, and J. Qiu, "A modified fifth order finite difference Hermite WENO scheme for hyperbolic conservation laws," *Journal of Scientific Computing*, vol. 85, pp. 29 (2020).

[2] C. Nwaigwe and V. N. Mishra, "Accelerating fixed point algorithms for nonlinear Fredholm integral equations," *Journal of Interdisciplinary Mathematics*, vol. 27, no. 6, pp. 1319–1337 (2024), doi: 10.47974/JIM-1782.

[3] B. Ren and C. Parés, "High-order WENO finite-difference methods for hyperbolic nonconservative systems of partial differential equations," *Journal of Computational Physics,* vol. 535, pp. 114047 (2025).

[4] J. Li, C.-W. Shu, and J. Qiu, "Moment-based multi-resolution HWENO scheme for hyperbolic conservation laws," *Journal of Computational Physics,* vol. 32, pp. 364–400 (2022).

[5] Z. Zhao and J. Qiu, "An oscillation-free Hermite WENO scheme for hyperbolic conservation laws," *Science China Mathematics*, vol. 67, pp. 431–454 (2024).

[6] M. Zhang and Z. Zhao, "A fifth-order finite difference HWENO scheme combined with limiter for hyperbolic conservation laws," *Journal of Computational Physics*, vol. 472, pp. 111676 (2023).

[7] I. Wibisono, Yanuar, and E. A. Kosasih, "Fifth-order Hermite targeted essentially non-oscillatory schemes for hyperbolic conservation laws," *Journal of Scientific Computing*, vol. 87, pp. 69 (2021).

[8] D. Dhabliya, S. Kunche, S. Dingankar, S. S. Dari, R. Dhabliya, and V. Khetani, "Blockchain technology as a paradigm for enhancing cyber security in distributed systems," *Journal of Discrete Mathematical Sciences and Cryptography*, vol. 27, no. 2-B, pp. 729–740 (2024).

[9] P. Sahane, M. Gulhane, N. Rakesh, S. M. M. Naidu, M. Grover, and V. Mahajan, "Evaluating lightweight encryption schemes for resource-constrained devices in wireless sensor networks," *Journal of Discrete Mathematical Sciences and Cryptography,* vol. 28, no. 5-A, pp. 1803–1812 (2025).

[10] M. A. A. Sheela, K. Amulya, D. Lokesh, K. Yesubabu, and P. Ajay, "Federated multimodal language recognition: A deep learning approach for real-time applications," *International Journal of Recent Advances in Engineering & Technology* (IJRAET), vol. 14, no. 1, pp. 17–26 (Apr. 2025).

Received December, 2024