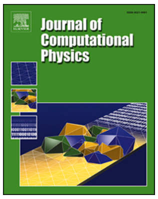
Journal of Computational Physics 541 (2025) 114332

Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp



Very high-order multi-layer compact schemes with shock-fitting method (MLC-SF) for compressible flow simulations

Yung-Tien Lin *, Xiaolin Zhong

Mechanical and Aerospace Engineering Department, University of California Los Angeles, Los Angeles, CA 90095, USA

ARTICLE INFO

Keywords:
High-order methods
Multi-layer compact schemes
Finite difference method
Upwind schemes
Shock-fitting method
Compressible flow

ABSTRACT

High-order numerical methods are commonly employed in direct numerical simulation (DNS) to achieve the required accuracy with fewer degrees of freedom, thereby improving computational efficiency. To further improve global spatial accuracy, Bai and Zhong proposed the multi-layer compact (MLC) schemes (JCP, 2019) to introduce spatial derivatives as new degrees of freedom and create a more compact stencil for the same spatial accuracy. Stability analysis showed MLC can achieve seventh-order global accuracy with closed boundaries, which surpasses most of the sixth-order conventional upwind finite difference schemes. Despite this high-order convergence rate, MLC faces challenges in supersonic flow simulations, primarily due to the Gibbs phenomenon across shock waves. The numerical oscillation can cause divergence in high-order numerical schemes if no additional treatment, such as shock-capturing or shock-fitting methods, is applied. Therefore, further studies are needed to enhance MLC's applicability to realistic high-speed flow applications, particularly in the context of shock treatments and boundary condition implementation. This paper develops a novel MLC method to improve its applicability for supersonic flow simulations. The proposed method integrates MLC with the shock-fitting method (MLC-SF), treating the shock wave as a computational boundary that separates upstream and downstream solutions. The shock-fitting method mitigates spurious numerical oscillations across the discontinuous interface, preserving the high-order accuracy of MLC-SF. Additionally, this paper introduces a physically consistent boundary condition for the MLC-SF spatial derivative layers behind the shock. This boundary condition uses the inversion of the flux Jacobian matrix to estimate the correct spatial derivatives, ensuring consistency between MLC-SF value and derivative layers at the inflow boundary. In order to systematically benchmark the proposed method, MLC-SF is applied to five simulation cases involving linear advection, Euler, and Navier-Stokes equations on one- and two-dimensional domains. The studied cases aim to compare the results of shock-fitting and shock-capturing methods, evaluate the performance of MLC-SF within the arbitrary Lagrangian-Eulerian (ALE) framework for moving grid applications, and test the MLC-SF derivative layers on fluid mechanics problems involving non-Cartesian grids. In both one-dimensional and two-dimensional shock wave interaction cases, MLC-SF with the proposed physically consistent inflow condition achieves seventh-order spatial accuracy, which outperforms the other four tested methods. Notably, in the one-dimensional shock-interaction results, the fifth-order WENO methods exhibit only first-order accuracy behind the shock wave, highlighting the necessity of adopting the shock-fitting approach to maintain the high spatial accuracy property in MLC-SF. In terms of computational efficiency. MLC-SF can save at least 30 % of the computational time compared to conventional high-order finite difference methods with shock-fitting for Shu-Osher-like problem. The overall objective of this study is to establish a high-order MLC framework suitable for compressible and high-speed fluid mechanics simulations.

---

* Corresponding author.
E-mail addresses: yungtlin@ucla.edu (Y.-T. Lin), xiaolin@seas.ucla.edu (X. Zhong).

https://doi.org/10.1016/j.jcp.2025.114332
Received 4 February 2025; Received in revised form 21 July 2025; Accepted 27 August 2025

1. Introduction

The growing demand for hypersonic research, driven by national security, space exploration, and advanced aircraft development, has increasingly received the attention of researchers. A breakthrough in hypersonic vehicle development could significantly reshape the aerospace industry's trajectory [1]. In the study of hypersonic boundary layer transitions, local maxima in skin friction and wall heat transfer typically occur near the end of the transitional zone [2]. Accurate prediction of the transition location and delaying turbulence generation can greatly reduce the vehicle's cooling system requirements [3]. However, studying fluid mechanics at such extremely high-speed flow poses experimental and numerical challenges. Hypersonic fluid mechanics includes not only the effects of gas compressibility and shock-turbulence interactions, but also the added complexities of high-temperature gas dynamics, rarefied flow regimes, and complex chemical-physical interaction between molecular reaction and chaotic turbulence [4]. Conducting hypersonic experiments is challenging due to difficulties in generating high enthalpy and noise-free inflow streams. At the same time, numerical simulations must resolve multiscale flow features both temporally and spatially, ranging from molecular energy mode excitations to turbulence energy cascades. To date, hypersonic fluid mechanics remains a significant challenge in aerospace engineering.

1.1. *DNS in hypersonic boundary layer transition flows*

Direct numerical simulation (DNS) has been adopted in hypersonic fluid mechanics research for years, particularly in the study of boundary layer transition flows [3]. Within boundary layers, hypersonic flows receive perturbations from environmental disturbances through receptivity mechanisms, leading to the formation of various eigenmodes. These eigenmodes grow, interact, and ultimately result in turbulence [5]. Turbulence models generally face limitations in accurately modeling laminar-to-turbulent transitions, shock wave-boundary layer interactions, and separated turbulence, due to the complex and multi-scale features of hypersonic flows [6]. Capturing the small yet critical nonlinear interactions of flow perturbations requires highly accurate DNS.

Zhong [4] showed a family of high-order upwind explicit (non-compact) and compact [7] finite difference methods can be applied to hypersonic transitional DNS. The study used the shock-fitting method to treat the sharp primary shock as a computational boundary. By separating the flow solution into upstream and downstream regions of the shock wave, high-order methods can simulate the supersonic flow with the discontinuous interface without incurring the Gibbs phenomenon and causing unstable simulations. Additionally, the study conducted von Neumann stability analysis for the explicit and compact upwind finite difference methods. The results indicated that seventh-order interior stencils could only be used with up to fifth-order boundary stencils. This implies conventional finite difference methods achieve at most sixth-order global accuracy for closed boundary problems. Duan et al. [8] applied the bandwidth-optimized WENO method [9] to simulate the hypersonic transitional boundary. Shock-capturing methods like WENO enable numerical schemes to resolve downstream locations where shocklets and turbulence interactions occur. Laible and Fasel [10] utilized upwind finite difference schemes in the downstream and wall-normal directions, complemented by a pseudo-spectral method in the azimuthal direction, to simulate hypersonic cone flow. They also applied low-pass filtering to suppress undesirable high-frequency numerical oscillations. Candler et al. [11] compared various numerical methods for reentry vehicles using the finite volume software, US3D, with kinetic energy consistent (KEC) flux [12]. Nguyen et al. [13] utilized a matrix-free discontinuous Galerkin method [14] in simulating hypersonic boundary layer transitions on a flared cone using calorically perfect gas assumption with implicit large eddy simulation (ILES), also referred to as under-resolved DNS (UDNS) [15].

1.2. *High-order numerical methods*

High-order numerical methods are computationally efficient to achieve the same accuracy for having at least a third-order spatial convergence rate [16]. This character makes high-order methods ideal for DNS applications that have relatively simple geometry and demanding accuracy requirements in space. The traditional high-order numerical representations are spectral [17], finite difference [7,18,19], and finite volume [20] methods. The spectral method is widely considered the most accurate numerical approach due to its exponential convergence rate, known as spectral accuracy [21], and has been used since the early years of DNS research [22]. However, the conventional spectral method is vulnerable to discontinuous interfaces in solutions or bounded computational domain because of its global basis function representation, which limits its application [23]. In contrast, finite difference and finite volume methods offer a slower algebraic convergence rate [24] but provide greater flexibility in imposing boundary conditions and integrating with parallel computing frameworks. Conventional explicit upwind finite difference can achieve sixth-order accuracy when applied to smooth problems [4]. The compact finite difference methods, such as Lele [7], reduce numerical dissipation for high frequency wave and achieve a spectral-like resolution. However, compact schemes require additional implicit solver to compute first-order derivative from neighboring values, which compromise the computing time of the method. On the other hand, high-order finite volume methods depend on accurate flux reconstruction on the cell faces. Proper flux reconstruction is crucial for achieving numerical conservation properties, such as kinetic energy [12] and entropy [25] conservation, which are essential for accurately capturing the correct flow physics.

There has been a recent rise in interest in the finite element-based (FEM) methods by increasing the degrees of freedom inside each element. Among the methods, discontinuous Galerkin (DG) method received the most attention in the community. One of the biggest strengths of DG or other finite element-based methods is the ease of implementation of high-order formulation on unstructured grids through the weak formulation. Conventional FEM approaches that utilize continuous basis functions often require additional stabilization techniques when applied to fluid mechanics problems. Cockburn and Shu [26] showed that using discontinuous basis functions between elements, combined with Runge-Kutta time integration, can be stably applied to the Navier-Stokes equations. This approach is known as the Runge-Kutta Discontinuous Galerkin (RKDG) method. However, RKDG requires high-memory storage due to the duplicated degrees of freedom on element faces. Memory-efficient DG methods such as hybridized DG [27] and embedded DG [28] are proposed to reduce the redundant degrees of freedom through minimizing the flux jump across element faces. The flexibility of DG has been widely used in various fluid mechanic applications. For hypersonic applications, Hoskin et al. [29] provided an overview of DG for high-speed flow, including discussions on both shock-capturing and shock-fitting techniques within the DG framework. Other basis function-based methods, such as spectral volume (SV) [30] and spectral difference (SD) [31] methods, are also gaining attention. Spectral volume methods subdivide SV cell into several control volumes and reconstruct fluxes from the neighboring states inside the cell. Spectral difference methods utilize the solutions on the Gaussian quadrature points to avoid direct surface or volume integrations. These methods can be considered as complement approaches to DG for FEM-based methods.

1.3. Multi-layer compact schemes

In addition to the traditional high-order methods and FEM-based approaches, Zhong and Bai [32–34] proposed the multi-layer compact (MLC) schemes which introduce first-order spatial derivatives as new degrees of freedom to the finite difference method. In MLC [32,33], the original solution space for the flow states is referred to the value layer and the newly introduced solution space for spatial derivatives of flow states is referred to the derivative layers. This method solves both value and derivative layer through time integration of the same governing equations but with different formulation. For the value layer, the first-order flux derivatives in the original governing equations can be analytically computed through the chain rule with the aid of the first-order spatial derivatives from the derivative layer. For the derivative layer, the time change of the spatial derivatives is estimated from the spatial differentiation of the original governing equations, which plays as the auxiliary equations to the problem. In the auxiliary equations, the second-order flux derivatives are numerically estimated from both neighboring state values and first-order spatial derivatives. By utilizing this multi-layer structure, MLC forms a more compact stencil than conventional finite difference methods, achieving spectral-like spatial resolution similar to that of the compact schemes from Lele [7]. Furthermore, MLC shares conceptual similarities with the discontinuous Galerkin (DG) method, as both introduce additional degrees of freedom at grid points or within cells. The derivative layer in MLC can be seen as analogous to a P_1 approximation in DG, similar to how the finite volume method aligns with a P_0 DG approximation. However, the fundamental solving procedures differ: MLC solves the governing equations in different differential forms for each layer, while DG relies on a weak formulation to compute the time change of the solution states.

From the previous study from Bai and Zhong [33], MLC can achieve up to seventh-order spatial accuracy on a closed boundary linear advection problem. In contrast, conventional finite difference methods can only achieve sixth-order global accuracy for closed boundary problems [4]. This limitation arises because conventional schemes typically rely on up to fifth-order one-sided differences at the boundaries, and higher-order implementations often lead to simulation divergence. As a consequence, even using a seventh or higher-order stencil for interior points, the conventional methods have a lower spatial global accuracy due to the usage of less accurate boundary stencils. Meanwhile, MLC can incorporate a seventh-order interior stencil with sixth-order boundary stencils without causing simulation divergence. Furthermore, the Fourier analysis from the study shows that MLC has spectral-like properties such as lower numerical dissipation for high wave number and less dispersive error. This spectral-like property can be also found in the compact schemes from Lele [7], which also treats first-order derivatives as unknowns and solve them globally through an implicit solver. Unlike conventional compact scheme, MLC does not require a global linear solver to compute the first-order derivatives. MLC compute the first-order derivatives through the time integration of the auxiliary equations explicitly. Also, since the value layer solutions can be computed analytically without numerical differentiation, MLC is computationally more efficient than the conventional methods in term of computer wall time for one-dimensional linear advection problems. To address weak numerical instability originating from the numerical approximation of the second-order cross derivatives for MLC on multi-dimensional problems, Bai and Zhong proposed the directional MLC (DMLC) to address this issue [34]. This method introduces all of the cross derivatives as another new degree of freedom to the system. For two-dimensional problems, the time change of the second-order cross derivatives is estimated from second-order auxiliary equations. The third-order flux derivatives in the DMLC auxiliary equations are numerically approximated by the first-order and second-order cross derivatives. The derivative estimation in DMLC avoids using the weakly unstable two-dimensional cross derivative stencils in MLC, which improves the method stability for multi-dimensional problems.

Even though MLC and its variants have demonstrated strong capabilities in terms of numerical accuracy and computational efficiency, previous studies [32–34] only applied the method to linear advection problems, which are far from complex, realistic scenarios such as hypersonic transitional boundary layer flows. Bridging the gap between simple one-dimensional wave problems and the complex multi-scale physical interactions in hypersonic flows requires further development. Key challenges remain, such as flux vector splitting for nonlinear hyperbolic equations, boundary condition implementations for Navier-Stokes equations, and treatments for discontinuous shock waves. While most of these issues have been investigated individually for other conventional methods, MLC requires adaptations of these techniques, particularly for the newly introduced derivative layers. Among these challenges, the treatment of discontinuous shock waves is the priority to be resolved, since improper shock handling can cause immediate simulation divergence. Therefore, developing a robust and effective shock treatment for MLC is critical to extending its applicability to supersonic flow simulations.

1.4. Shock-capturing methods

To apply high-order numerical schemes to high-speed flows, shock-capturing and shock-fitting methods are commonly employed to suppress numerical oscillations caused by the Gibbs phenomenon near shocks. The shock-fitting method represents a shock wave as either a computational interface or a shock-fitted mesh face, thereby explicitly treating the shock as a boundary [4,35,36]. This approach prevents numerical schemes from solving across the discontinuity, effectively eliminating the Gibbs phenomenon. However, shock-fitting typically requires prior knowledge of the shock topology, which limits its application primarily to relatively simple supersonic flow fields. In contrast, shock-capturing methods do not require explicit identification of shock locations. Instead, they introduce artificial dissipation, either explicitly or implicitly, to stabilize the simulation by smoothing the solution near shock waves [37]. Due to their robustness and flexibility, shock-capturing methods are well-suited for complex scenarios, including supersonic flows involving shocklets, triple points, and turbulence-shock interactions. Both shock-fitting and shock-capturing methods remain popular tools in the supersonic research community, each offering unique advantages depending on the complexity and requirements of the flow problem.

The robustness of shock-capturing methods makes them ideal for initial investigations of supersonic flow fields. One of the earliest shock-capturing approaches was the artificial viscosity method, introduced by von Neumann and Richtmyer [38], which added an artificial viscosity term proportional to the local velocity gradient to stabilize simulations. Jameson et al. [39] developed the Jameson-Schmidt-Turkel (JST) scheme, which incorporates both second- and fourth-order artificial dissipative fluxes. The second-order dissipation is activated near shock waves to enhance stability, while the fourth-order dissipation suppresses high-frequency oscillations caused by aliasing. For discontinuous Galerkin (DG) methods, Persson and Peraire [40] introduced a discontinuity sensor to selectively apply numerical diffusion. Other classical shock-capturing techniques, such as flux limiters [41,42] and slope limiters [43], are also widely adopted for stabilizing supersonic flow simulations.

In recent decades, the weighted essentially non-oscillatory (WENO) method has become a widely popular shock-capturing approach for simulating flows with shocks. The WENO family uses multiple sub-stencils to evaluate the flux, reconstructing it through a weighted averaging process. Jiang and Shu [44] enhanced the original WENO method developed by Liu et al. [45] by introducing the concept of minimizing the total variation in the smoothness indicator, thereby achieving optimal fifth-order accuracy for three-point stencils more effectively. This improved method, known as WENO-JS, became the foundation for subsequent WENO methods. Various WENO variants have since been developed to address specific limitations of WENO-JS, such as improving the weighting function to ensure faster convergence toward the optimal weights in smooth regions. Notable advancements include WENO-M [46], WENO-Z [47,48], and targeted essentially non-oscillatory (TENO) methods [49].

Among the WENO family, the Hermite WENO (HWENO) method [50,51] has a particularly close relationship with the multi-layer compact (MLC) schemes. Like MLC, HWENO uses both function values and first-order derivatives to reconstruct fluxes from neighboring cell averages. HWENO has been successfully applied to Runge-Kutta discontinuous Galerkin (RKDG) methods, maintaining compactness while delivering computational efficiency. Given its reliance on first-order derivative layers, HWENO could be seamlessly incorporated into the MLC framework, which inherently supports such derivative layers for flux reconstruction. This suggests a promising possibility for enabling shock-capturing simulations with MLC by leveraging HWENO.

However, shock-capturing schemes often resort to lower, sometimes even first-order accuracy in the vicinity of shocks. This approach, while stabilizing the solution, tends to smear the discontinuities and diminish the global accuracy of the solution. Additionally, many shock-capturing methods require fine-tuning of empirical parameters, which can vary depending on the flow field, further complicating their application. In contrast, the shock-fitting method, which treats shocks as computational boundaries, offers a more accurate result of shock wave physics. This makes it a suitable candidate for applications such as hypersonic boundary layer transition flows, where shock time-dependent behaviors are critical.

1.5. Shock-fitting methods

While shock-capturing methods provide robust approaches for simulating flow fields containing shock waves, they typically reduce the order of accuracy when encountering discontinuous interfaces. This reduction compromises the overall spatial accuracy of otherwise high-order schemes. An alternative approach, which is not mutually exclusive, is the shock-fitting method [4,35,36]. This technique, introduced for supersonic flow simulations in the 1960s [52], treats the shock wave as a computational boundary. By separating the upstream and downstream solutions at the shock, the shock-fitting method preserves the high spatial accuracy of high-order schemes. In Zhong [4], the shock boundary movement is estimated through the local shock acceleration from a time derivative of the Rankine-Hugoniot relations with a characteristic relation. This shock-fitting method with high-order upwind finite difference method has been used in various hypersonic boundary layer transition flow studies [3,53,54]. Johnsen et al. [55] conducted a comprehensive comparison of shock-fitting and shock-capturing methods. Their results showed that shock-fitting is generally more accurate than shock-capturing when the downstream solution is smooth and relatively simple. This characteristic makes shock-fitting particularly attractive for direct numerical simulations (DNS) of hypersonic boundary layer transitions, where precise modeling of time-dependent shock wave interactions is crucial [3,10]. Rawat and Zhong [56] compared different approaches for estimating shockfront motion and found that using the time derivative of the Rankine-Hugoniot relations with a characteristic relation, yielded the most accurate results. However, shock-fitting has a significant limitation: it requires a priori knowledge of the shock wave's location in the flow field. In realistic problems, shock locations are often not well-defined and may form and evolve over time, as seen in phenomena such as shock triple points, shocklets, and turbulence-shock interactions.

Shock detection has become a critical research focus in the supersonic simulation community, particularly for identifying shock waves in smeared and turbulent CFD simulation results. The primary purpose of shock detection is to identify shock locations and apply appropriate shock treatment techniques, such as refining grids near shock waves or deploying shock-capturing methods in marked cells. In this context, the smoothness indicator in WENO methods [44] can also be considered as a shock detection tool. The Ducros sensor [57] is widely used for shock detection due to its ability to distinguish shock waves from compressible turbulent structures effectively. Other approaches, such as edge detection techniques from image processing [58,59] and neural network-based classifiers [60–62], have received significant interest in recent years. To integrate shock detection into the shock-fitting method for simulating complex shock wave interactions, Paciorri and Bonfiglioli [63] proposed a framework for the finite volume method with unstructured grids. This method adaptively modifies grid point connectivity and interpolates shock-captured solutions from the background mesh based on detected shock locations. In a more recent study [64], the method was further enhanced to incorporate shock detection and pattern identification, enabling the creation of continuous shock boundaries from discrete marked cells. This combined approach has demonstrated the ability to capture intricate shock structures, such as those observed in three-dimensional shock reflections [65].

In a broader context, r-adaptation techniques that track shockfronts can also be classified as shock-fitting methods, even when shocks are not explicitly marked or tracked. Nguyen et al. [66] proposed an r-adaptation approach for hybridized DG methods in compressible flow simulations by solving the Helmholtz equation. A velocity divergence-based shock indicator was used as the source term for this equation, enabling the method to accurately track shock locations in various two-dimensional supersonic steady flows, including channel flow over a bump, double ramps, and flows past a cylinder. Beyond conventional CFD solvers, recent advancements in optimization-based shock-tracking methods have demonstrated the ability to eliminate the need for limiters or artificial viscosity to stabilize shocks. These methods reformulate supersonic flow simulations as optimization problems. For instance, the Moving Discontinuous Galerkin Finite Element Method with Interface Condition Enforcement (MDG-ICE), developed by Kercher and Corrigan [67], incorporates the DG residual, flux differences between elements, and geometric boundary conditions into an objective function. The shock-tracking process minimizes flux differences or enforces interface conditions by dynamically adjusting mesh points. This method naturally extends to unsteady problems by recasting the governing equations into a steady space-time formulation. Similarly, the High-Order Implicit Shock Tracking (HOIST) method, introduced by Zahr and Persson [68], treats the DG residual as an optimization constraint. It introduces a discontinuity tracker as an objective function to minimize, ensuring accurate shock representation while maintaining stability and high-order accuracy.

1.6. *Objectives*

This paper is aiming for developing a novel numerical method designed for high-order multi-layer compact (MLC) schemes in high-speed flow simulations to achieve seventh-order global accuracy. Unlike previous MLC studies by Bai and Zhong [33,34], which primarily focused on numerical analysis and implemented MLC on simple, periodic, linear advection-dominant problems, this work addresses the complex demands of realistic supersonic flow simulation scenarios. These simulations require sophisticated approaches to overcome challenges such as flux-vector splitting, shock treatment, and boundary condition formulation for compressible fluid dynamics equations. To bridge these critical gaps, this paper develops and implements MLC integrated with a high-order shock-fitting method, termed MLC-SF, for supersonic flow over a cylinder. This integration is not merely a technical enhancement; it identifies and incorporates the essential components necessary for practical supersonic flow applications. By advancing the development of MLC-SF, this study extends its applicability to more challenging and realistic scenarios. Furthermore, we benchmark MLC-SF against established shock interaction problems, including the cases from the comprehensive assessment of high-resolution methods by Johnsen et al. [55]. This comparative analysis demonstrates the robustness and effectiveness of MLC-SF in addressing specific problems while providing valuable insights for researchers evaluating MLC-SF's suitability for their applications.

This paper is organized as follows. Section 2 provides an overview of the numerical methods and techniques developed from existing studies for supersonic flow simulations. Topics include the governing equations, the conventional upwind finite difference method, the shock-fitting method, and the multi-layer compact (MLC) schemes. This section highlights the essential components of conventional approaches for supersonic flow simulations and introduces the multi-layer compact (MLC) scheme. Section 3 outlines the necessary adaptations for the MLC with shock-fitting (MLC-SF) framework. These include a modified flux-vector splitting scheme to mitigate the carbuncle phenomenon-a numerical instability that occurs near the centerline in blunt body flows [69]-the arbitrary Lagrangian-Eulerian (ALE) formulation for handling moving meshes, and the development of boundary conditions for the MLC derivative layers. This section highlights the essential components of conventional approaches to supersonic simulations and introduces MLC. Section 3 outlines the necessary adaptations for MLC-SF, including a modified flux-vector splitting scheme to mitigate the carbuncle phenomenon, a numerical error happened in the vicinity of the centerline [69], the arbitrary Lagrangian-Eulerian (ALE) formulation for moving meshes, and boundary conditions for MLC derivative layers. Section 4 verifies and demonstrates the capabilities of MLC and the proposed MLC-SF against other well-estabilished methods like FD-SF and WENO methods. In addition to the grid convergence rate studies, this Section also provides the numerical error versus computational time to justify computational cost of MLC-SF. In Section 5, MLC-SF is applied to more realistic shock-involving two-dimensional problems to showcase the applicability of the method. The test cases include two-dimensional vorticity-entropy wave interaction from Mahesh [70] and supersonic flow over a parabolic cylinder. Finally, Section 6 concludes the paper with remarks and suggestions for future work.

2. Numerical methods

2.1. *Conventional finite difference approach*

Fluid mechanics problems are often modeled using the Navier-Stokes or Euler equations, assuming the continuum hypothesis holds. Regardless of the governing equations used, when the Reynolds number is sufficiently high, the advection term plays a critical role in the flow physics. The advection term is generally expressed in conservation form as:

$$\frac{\partial U}{\partial t} + \frac{\partial F_j}{\partial x_j} = 0 \tag{1}$$

where F_j represents the fluxes for different direction, which can be nonlinear functions of the state variables U. One key challenge in fluid simulations is accurately estimating these nonlinear fluxes from neighboring numerical solutions. To address this, various numerical methods have been employed in fluid mechanics, including finite difference, finite volume, and discontinuous Galerkin methods.

Finite-difference methods, including both explicit and compact schemes, have been applied to DNS of hypersonic boundary layer transitions for many years. While these methods exhibit slower spatial convergence rates compared to spectral methods, they offer greater flexibility in implementing boundary conditions, making them suitable for simulating slightly complex geometries such as blunt objects with shock waves. Most high-order finite-difference methods are upwind-biased schemes, which introduce a small amount of dissipative error to stabilize convection-dominated hypersonic flows. These methods are well-established in the research community [4,8,10] and form a solid foundation for the development of MLC-SF on DNS applications.

2.1.1. Navier–Stokes equations

For low-altitude applications where the fluid continuum hypothesis can be applied, the hypersonic flow is mainly driven by the Navier-Stokes equations. In contrast to other fluid mechanics problems, hypersonic flow needs to consider molecular dissociation and ionization due to the high temperature in the flow field, where the ideal gas law may not be applied. One approach involves introducing the density of each species ρ_m and using an extra body force term W to model chemical reactions and other thermochemical nonequilibrium processes. Hence, the governing equations of a three-dimensional unsteady hypersonic flow can be expressed in a nonlinear conservation form, which is:

$$\frac{\partial U}{\partial t} + \frac{\partial F_j^{inv}}{\partial x_j} - \frac{\partial F_j^{vis}}{\partial x_j} = W, \quad j = \{1,2,3\} \tag{2}$$

where the conservative flow states U, inviscid flux components F_j^{inv}, and viscous flux components F_j^{vis} are:

$$U = \begin{bmatrix} \rho_m \\ \rho u_1 \\ \rho u_2 \\ \rho u_3 \\ e \end{bmatrix}, \quad F_j^{inv} = \begin{bmatrix} \rho_m u_j \\ \rho u_1 u_j + p\delta_{1j} \\ \rho u_2 u_j + p\delta_{2j} \\ \rho u_3 u_j + p\delta_{3j} \\ (e+p)u_j \end{bmatrix}, \quad F_j^{vis} = \begin{bmatrix} 0 \\ \tau_{1j} \\ \tau_{2j} \\ \tau_{3j} \\ \tau_{jk}u_k - q_j \end{bmatrix} \tag{3}$$

The viscous stress tensor τ_{ij} contains the shear and the bulk viscosity. For simplicity, one may apply Stokes' hypothesis which assumes the bulk viscosity is equal to zero and leads to the following:

$$\tau_{ij} = \mu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) - \frac{2}{3}\mu\frac{\partial u_k}{\partial x_k}\delta_{ij} \tag{4}$$

and the viscosity is modeled by Sutherland's law:

$$\mu(T) = \mu_0\left(\frac{T}{T_0}\right)^{3/2}\frac{T_0+S}{T+S} \tag{5}$$

For the thermal conduction term, the heat flux is modeled by Fourier's law with a thermal conductivity κ:

$$q_j = -\kappa\frac{\partial T}{\partial x_j} \tag{6}$$

One of the common approaches in aerospace applications for thermal conductivity κ is assuming the ratio of momentum diffusivity to thermal diffusivity remains constant. This ratio is known as the Prandtl number. The typical range of the Prandtl number for air is 0.71 to 0.78 based on different flight conditions [4,12,71]. With the above equations and relations, one may start to apply numerical methods to the model for simulating hypersonic flows.

2.1.2. Upwind finite difference method

Finite difference methods have been widely used in hypersonic research, as shown in the works of Zhong [4], Duan et al. [8], and Laible and Fasel [10]. The upwind approximation of the first-order derivative in finite difference D_x^+u at location i can be expressed in the following general form:

$$\sum_{k=-M_1}^{M_2} b_k (D_x^+u)_{i+k} = \frac{1}{h}\sum_{k=-N_1}^{N_2} a_k u_{i+k} + O(h^p) \tag{7}$$

where a and b are the finite difference coefficients that can be derived from the Taylor expansion, and the spatial convergence rate of the expression is pth-order. If $b_k \neq 0$ for $k \neq 0$ location, this formulation is referred to as compact schemes [7]. Compact schemes have a spectral-like convergence rate but require an implicit solver to compute derivatives globally. On the other hand, an explicit scheme $b_k = 0$ for $k \neq 0$, does not require an implicit solver. In Zhong [4], the coefficients of seventh-order explicit upwind schemes in interior points are:

$$\begin{aligned} &a_{\pm 4} = \mp\frac{1}{280} + \frac{1}{40320}\alpha, \quad a_{\pm 3} = \pm\frac{4}{105} - \frac{1}{5040}\alpha, \quad a_{\pm 2} = \mp\frac{1}{5} + \frac{1}{1440}\alpha, \\ &a_{\pm 1} = \pm\frac{4}{5} - \frac{1}{720}\alpha, \quad a_0 = \frac{1}{576}\alpha \quad b_0 = 1 \end{aligned} \tag{8}$$

where α is the upwind parameter for stabilizing the scheme which, however, introduces dissipation errors. In Bai and Zhong [33], the α is set to 36 for the seventh-order scheme for numerical scheme comparison.

The flux vector splitting technique is used to achieve an upwind-biased numerical solver during the flux evaluation step. This method splits the nonlinear advection terms into right-running (positive direction) and left-running (negative direction) waves using an approximate Riemann solver, which are then discretized using the corresponding upwind stencils:

$$\frac{\partial F_j}{\partial x_j} = D^+_{x_j} F^+_j + D^-_{x_j} F^-_j + O(h^p) \tag{9}$$

The local Lax-Friedrichs or Rusanov flux is used to approximate the Riemann solver. In addition, a predefined parameter ϵ is added to the maximum eigenvalue evaluation to smooth the solution near the stagnation points. A typical range of ϵ is 0.2 to 1.0 depending on the flow field requirements on the numerical dissipation inside the low-speed zone. This approach can mitigate the carbuncle phenomenon [69] at the centerline of the supersonic over a blunt object flow.

$$F^\pm_j = \frac{1}{2}\left(F_j \pm \lambda_j U\right), \quad \lambda_j = c + \sqrt{{u_j}^2 + (\epsilon c)^2} \tag{10}$$

With the above procedures for evaluating the spatial derivatives, the flow state can be advanced in time using an appropriate time integration method.

$$\frac{dU}{dt} = L(U,t) \tag{11}$$

The third-order low-storage strong stability preserving Runge-Kutta method (SSP-RK3), also known as the total variation diminishing (TVD) Runge-Kutta method [72], is used for time integration in most of the simulation cases presented in the following sections:

$$\begin{aligned} U^{(1)} &= U^n + \Delta t L(U^n, t^n) \\ U^{(2)} &= \frac{3}{4}U^n + \frac{1}{4}\left[U^{(1)} + \Delta t L(U^{(1)}, t^n + \Delta t)\right] \\ U^{n+1} &= \frac{1}{3}U^n + \frac{2}{3}\left[U^{(2)} + \Delta t L(U^{(2)}, t^n + \frac{\Delta t}{2})\right] \end{aligned} \tag{12}$$

For problems involving molecular reactions, the time scale of the thermo-chemical source terms can be small, leading to numerical stiffness in time integration. A hybrid approach can be used, such as the implicit-explicit (IMEX) time integration method, where the reaction terms are handled implicitly while the flow states are computed explicitly.

2.2. Shock-fitting methods

The shock-fitting method has a long-standing history in supersonic flow simulations, dating back to the 1960s [52]. This technique treats the discontinuous shock wave as a computational boundary, where the post-shock properties are determined using oblique shock relations. Zhong [4] successfully applied the shock-fitting method to hypersonic boundary layer transition DNS, enabling detailed investigations into hypersonic turbulence formation mechanism [3]. For shock movement computation, Rawat and Zhong [56] demonstrated that combining a conventional moving mesh with the time derivative of the Rankine-Hugoniot conditions, along with characteristic relations to estimate shock locations, effectively preserves the high-order spatial accuracy of upwind finite difference methods. The following section focuses on the implementation of this shock-fitting approach, adopting the notations from Zhong [4].

2.2.1. Moving mesh treatment

In the moving mesh approach for the shock-fitting method, the shockfront is treated as a fixed boundary in the computational domain, while the grid points move within the physical domain. Consequently, Eq. (2) must be modified to account for the moving mesh. One approach to achieve this is by applying the arbitrary Lagrangian-Eulerian (ALE) method to the three-dimensional governing equations, resulting in the following formulation:

$$\frac{\partial gU}{\partial t} + \frac{\partial F_1'}{\partial \xi} + \frac{\partial F_2'}{\partial \eta} + \frac{\partial F_3'}{\partial \zeta} = 0 \tag{13}$$

where

$$\begin{bmatrix} F_1' \\ F_2' \\ F_3' \end{bmatrix} = g \begin{bmatrix} \xi_x & \xi_y & \xi_z \\ \eta_x & \eta_y & \eta_z \\ \zeta_x & \zeta_y & \zeta_z \end{bmatrix} \begin{bmatrix} F_1 \\ F_2 \\ F_3 \end{bmatrix} + g \begin{bmatrix} \xi_t \\ \eta_t \\ \zeta_t \end{bmatrix} U \tag{14}$$

The detailed derivation in vector form can be found in the work by Persson et al. [73]. This approach maps the solutions from the physically moving domain (x, y, z) to the static computational domain (ξ, η, ζ) where g is associated with the Jacobian of the mapping function. ALE not only takes the coordinate transformation into account but also introduces the induced flux due to the mesh motion into the calculation. The MLC form of the moving mesh treatment can be simply obtained by applying spatial differentiation to the above equations with the chain rule. In the present study, the derivatives with respect to the computational coordinate, U_ξ and U_η, are used to represent the derivative layers in the two-dimensional MLC under the ALE formulation, instead of using physical coordinate derivatives, U_x and U_y. This choice is motivated by the fact that the spatial derivatives in Eq. (13) are defined in the computational coordinate. Using derivatives with respect to the physical coordinate would require additional steps to transform the computed fluxes back to the computational coordinate, slightly increasing the overall computational cost of the method.

2.2.2. Shockfront movement

Given supersonic free-stream conditions or solutions numerically calculated from the upstream supersonic flow field, the flow states immediately behind the shockfront can be determined using the Rankine-Hugoniot conditions. These conditions claim that the flux normal to and relative to the shock wave F' must be conserved:

$$F'_s - F'_0 = \left(\vec{F}_s - \vec{F}_0\right) \cdot \vec{a}_G + \left(U_s - U_0\right) b_G = 0 \tag{15}$$

where $\vec{a}_G$ acts as the normal vector of the shockfront, b_G is a transformed shockfront normal velocity, F_0 and U_0 are the fluxes and states in front of the shock, and F_s and U_s are the fluxes and states behind the shock. If the shockfront is located at the top boundary on the computational domain, i.e. $\eta = 1$, then $\vec{a}_G$ and b_G are equal to:

$$\begin{aligned} &\vec{a}_G = (g\eta_x)_G \hat{x} + (g\eta_x)_G \hat{y} + (g\eta_x)_G \hat{z} \\ &b_G = g\eta_t \end{aligned} \tag{16}$$

where the hat notations denote the unit vectors in the physical coordinate. Note that the normal vector of the shockfront, $\vec{a}_G$, depends on the shockfront's normal velocity, b_G, since the geometry of the shockfront is obtained through time integration of the shock velocity. Therefore, evaluating the shock velocity is another critical step in the shock-fitting method.

There are several ways to evaluate the shockfront velocity. In Rawat and Zhong's study [56], three different shock velocity evaluation methods are discussed. Among the methods, the time derivative of the Rankine-Hugoniot conditions with C^+ characteristics has the best spatial accuracy when applied to the shock-fitting method with high-order finite difference methods. The method computes the shockfront velocity through time integration of shock acceleration and the acceleration is given by Eq. (20).

The acceleration equation derivation starts with the time derivative of the Rankine-Hugoniot conditions:

$$\left(\frac{\partial \vec{F}_s}{\partial t} - \frac{\partial \vec{F}_0}{\partial t}\right) \cdot \vec{a}_G + \left(\vec{F}_s - \vec{F}_0\right) \cdot \frac{\partial \vec{a}_G}{\partial t} + \left(\frac{\partial U_s}{\partial t} - \frac{\partial U_0}{\partial t}\right) b_G + \left(U_s - U_0\right) \frac{\partial b_G}{\partial t} = 0 \tag{17}$$

Applying the chain rule to Eq. (14), the Jacobian of the flux in the direction normal to the shock wave, F'_2, is given by:

$$\mathbf{B}' = \frac{\partial F'_2}{\partial U} = \frac{\partial \vec{F}}{\partial U} \cdot \vec{a} + b_G \mathbf{I} \tag{18}$$

where $\mathbf{I}$ is the identity matrix. Substituting $\mathbf{B}'$ into Eq. (17) yields:

$$\mathbf{B}'_s \frac{\partial U_s}{\partial t} - \mathbf{B}'_0 \frac{\partial U_0}{\partial t} + \left(\vec{F}_s - \vec{F}_0\right) \cdot \frac{\partial \vec{a}_G}{\partial t} + \left(U_s - U_0\right) \frac{\partial b_G}{\partial t} = 0 \tag{19}$$

Then compute the eigenvalue and eigenvector for the flux Jacobian right behind the shockfront, $\mathbf{B}'_s$. There exists an eigenvalue, λ_s^+, and an eigenvector, I_s^+, which is associated with the C^+ characteristic line that propagated toward the shockfront. Finally, using the inner product of Eq. (19) with I_s^+ to obtain the shock acceleration equation:

$$\frac{\partial b_G}{\partial t} = \frac{-1}{[I_s^+ \cdot (U_s - U_0)]} \left[\lambda_s^+ I_s^+ \frac{\partial U_s}{\partial t} + I_s^+ (\vec{F}_s - \vec{F}_0) \cdot \frac{\partial \vec{a}_G}{\partial t} - (I_s^+ \mathbf{B}'_0) \frac{\partial U_0}{\partial t}\right] \tag{20}$$

With the acceleration of the shockfront, the shock velocity and location can be obtained through a time integration method.

2.3. The very high-order upwind multi-layer compact scheme

The multi-layer compact schemes (MLC) proposed by Bai and Zhong [33] introduces the spatial derivatives of the flow states U_x as additional degrees of freedom to the conventional finite difference method. The general form of the governing equations without body form term on MLC becomes:

$$\frac{\partial}{\partial t} \begin{bmatrix} U \\ U_{x_k} \end{bmatrix} + \frac{\partial}{\partial x_j} \begin{bmatrix} F_j \\ F_{j,x_k} \end{bmatrix} - \frac{\partial}{\partial x_j} \begin{bmatrix} F_{vj} \\ F_{vj,x_k} \end{bmatrix} = 0 \tag{21}$$

The auxiliary equations in the second row are derived from the spatial differentiation of the original governing. MLC formulation can be applied not only to the first-order derivative but also to other high-order derivatives. The following sections will demonstrate various MLC frameworks for one and multi-dimensional problems.

2.3.1. *One-dimensional formulation*

The fundamental idea of the MLC is to add additional degrees of freedom for each grid point. Spatial derivatives of the flow states are selected as extra simulated variables since they have a direct relation with flux derivatives. To obtain the auxiliary equations for the newly introduced variables, one can utilize spatial differentiation of the original governing equation. For example, in one dimensional MLC problem, a one-dimensional linear advection becomes:

$$\frac{\partial}{\partial t}\begin{bmatrix} U \\ U_x \end{bmatrix} + \frac{\partial}{\partial x}\begin{bmatrix} F_1 \\ F_{1,x} \end{bmatrix} = 0 \tag{22}$$

The flux derivative with respect to x, $\partial F_1/\partial x = F_{1,x}$, has a direct relation with U_x by using the flux Jacobian, i.e., $F_{1,x} = \mathbf{A}U_x$. Hence, the only remaining unknown in Eq. (22) is the flux's second derivative, $F_{1,xx}$. In MLC, the second derivative of a numerical state u can be approximated by using neighboring values of u and u_x:

$$(D^{+}_{xx}u)_i = \frac{1}{h^2}\sum_{l=-L_1}^{L_2} a_l u_{i+l} + \frac{1}{h}\sum_{m=-M_1}^{M_2} b_m (u_x)_{i+m} + O(h^p) \tag{23}$$

The above MLC coefficients, a and b, can be derived from the Taylor series with a uniform grid size, h. Fig. 1 shows the schematic of the stencil used in the above equation with $L_1 - L_2 - M_1 - M_2$. For instance, the seventh-order MLC coefficients with the upwind parameter, α, to stabilize the numerical schemes for hyperbolic equations:

$$\begin{aligned} &a_{\pm 2} = \frac{7}{85} \mp \frac{25}{3456}\alpha, \quad a_{\pm 1} = \frac{64}{27} \mp \frac{5}{108}\alpha, \quad a_0 = -5 \\ &b_{\pm 2} = \mp\frac{1}{36} + \frac{1}{576}\alpha, \quad b_{\pm 1} = \mp\frac{8}{9} + \frac{1}{36}\alpha, \quad b_0 = \frac{1}{16}\alpha \end{aligned} \tag{24}$$

For the actual value of the third and fifth-order MLC coefficients, Bai and Zhong's study [33] provides a more detailed derivation process and the definition of the upwind parameter. In the present study, the upwind parameters for different orders of accuracy are given in Table 1 which are the same values provided in Bai and Zhong's stability study. With the above formulations, the second derivative of the flux, $F_{1,xx}$, in Eq. (22) can be approximated using the MLC scheme in Eq. (23), with F_1 and $F_{1,x}$ provided. Then the equation can be applied with a time integration method to simulate the one-dimensional problem.

2.3.2. *Two-dimensional formulation*

For two dimensional MLC problem, the linear advection with MLC can be expressed as:

$$\frac{\partial}{\partial t}\begin{bmatrix} U \\ U_x \\ U_y \end{bmatrix} + \frac{\partial}{\partial x}\begin{bmatrix} F_1 \\ F_{1,x} \\ F_{1,y} \end{bmatrix} + \frac{\partial}{\partial y}\begin{bmatrix} F_2 \\ F_{2,x} \\ F_{2,y} \end{bmatrix} = 0 \tag{25}$$

Similar to the one-dimensional problem, the first derivatives of fluxes are derived from flux Jacobians, and the second derivatives, $F_{1,xx}$ and $F_{2,yy}$, can be approximated by Eq. (23). So the remaining unknowns in the two-dimensional equation are second-order mixed derivative terms, $F_{1,xy}$ and $F_{2,xy}$. The mixed derivative, u_{xy}, can be approximated by using u, u_x and u_y:

$$\begin{aligned} (u_{xy})_{ij} = {} & \frac{1}{h^2}\sum_{l_y=L_1}^{L_2}\sum_{l_x=L_1}^{L_2} a_{l_x,l_y} u_{i+l_x,j+l_y} \\ & + \frac{1}{h}\sum_{m_y=M_1}^{M_2}\sum_{m_x=M_1}^{M_2}\left[b_{m_x,m_y}(u_x)_{i+m_x,j+m_y} + c_{m_x,m_y}(u_y)_{i+m_x,j+m_y}\right] + O(h^r) \end{aligned} \tag{26}$$

Noting that most of the coefficients inside the stencil remain zero; only a few of the points have non-zero values. This is due to the two-dimensional stencil involving too many points than the required number to achieve targeting accuracy. The point selection

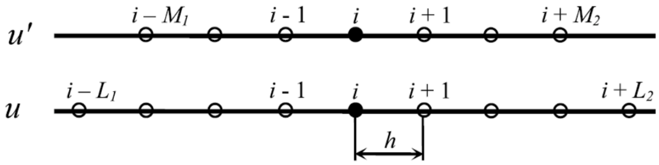


Fig. 1. Schematic of $L_1 - L_2 - M_1 - M_2$ scheme for MLC second-derivative approximation.

Table 1
MLC spatial accuracy and corresponding upwind parameter.

MLC Scheme	Order of Accuracy, p	Upwind Parameter, α
1-1-1-1	3	1.5
2-2-1-1	5	-1
2-2-2-2	7	12

process is based on using as few points as possible to reduce the computational cost and suppress round-off error accumulation. For simplicity, the detailed point selection procedure and the associated coefficient values are referred to the study by Bai and Zhong [33].

2.3.3. Directional multi-layer compact (DMLC) Scheme

In Bai and Zhong [32], the above mixed derivative approximation exhibits weak numerical instabilities over a small range of wavenumbers. To address this issue, Bai and Zhong proposed another approach [34] to solve the mixed derivative instability in multi-dimensional MLC. This approach is called the directional multi-layer compact (DMLC) schemes which introduces cross derivatives as new unknowns. For instance, Eq. (25) in DMLC becomes:

$$\frac{\partial}{\partial t}\begin{bmatrix} U \\ U_x \\ U_y \\ U_{xy} \end{bmatrix} + \frac{\partial}{\partial x}\begin{bmatrix} F_1 \\ F_{1,x} \\ F_{1,y} \\ F_{1,xy} \end{bmatrix} + \frac{\partial}{\partial y}\begin{bmatrix} F_2 \\ F_{2,x} \\ F_{2,y} \\ F_{2,xy} \end{bmatrix} = 0 \tag{27}$$

With the introduction of U_{xy} to the system, one can use the flux Jacobian and the chain rule to obtain $F_{1,xy}$ and $F_{2,xy}$, e.g., $F_{1,xy} = \mathbf{A}_y U_x + \mathbf{A} U_{xy}$. Hence, the remaining unknowns in the system are the third-order derivative $F_{1,xxy}$ and $F_{2,xyy}$. However, these terms can be solved by using the one-dimensional MLC approximation. For instance, $F_{1,xxy}$ is approximated by $F_{1,y}$ and $F_{1,xy}$ in x direction and, similarly, $F_{2,xyy}$ is approximated by $F_{2,x}$ and $F_{2,xy}$ in y direction. So the above process avoids DMLC using the unstable cross derivative approximation and yields more stable results. In the numerical results section, both MLC and DMLC will be discussed and compared for two-dimensional problems.

3. Proposed adaptations for MLC-SF

Bai and Zhong [33,34] conducted stability analysis for MLC and DMLC, demonstrating that these methods maintain stability with up to sixth-order boundary closure schemes while achieving seventh-order global accuracy for linear advection problems. They applied these methods to one-dimensional and two-dimensional linear advection scenarios and two-dimensional entropy wave transportation. However, these test cases are limited to linear advection wave-dominant problems with fixed grids and periodic boundary conditions. Extending MLC to more realistic applications, such as simulating supersonic flow over blunt bodies, requires significant adaptations. An appropriate Riemann solver must be employed to split zeroth- and first-order fluxes into upwind and downwind directions for accurate estimation of second-order derivatives. Furthermore, integrating the dynamic grid shock-fitting method requires an MLC computational framework incorporating the arbitrary Lagrangian-Eulerian (ALE) formulation. Proper handling of closed domain boundary conditions-including inflow, outflow, and wall boundaries-is essential for managing derivative layers. This section outlines the critical adaptations necessary for applying MLC to supersonic flow simulations over blunt objects. These key advancements form the foundation of the proposed multi-layer compact schemes with shock-fitting method (MLC-SF).

3.1. Flux vector splitting

The flux vector splitting technique is a crucial component of the upwind numerical methods. The technique introduces the idea similar to the approximated Riemann solver by determining the underlying wave direction and applying the upwind- and downwind-biased stencils accordingly. To introduce the flux vector splitting to MLC, the deriving procedure starts with the conservative form of a one-dimensional nonlinear advection equation:

$$\frac{\partial}{\partial t}[U] + \mathbf{A}\frac{\partial}{\partial x}[U] = 0 \tag{28}$$

where $\mathbf{A} = \partial F_1/\partial U$ is the flux Jacobian. The objective of flux vector splitting is to convert the nonlinear equation into two pure left and right direction problems based on the local and neighboring states while minimizing numerical dissipation and maintaining computational efficiency. For instance, the local Lax-Friedrichs method shown in Eq. (10) computes the absolute value of the maximum eigenvalue λ from $\mathbf{A}$. Then, the method splits the above nonlinear advection equation into right- and left-running wave-only equations and deploys upwind and downwind discretization, respectively. Even though Lax-Friedrichs introduces certain amounts of numerical dissipation to the results, the method is still applicable for high-order supersonic flow simulations.

For the MLC adaptation, the above conservative form of a one-dimensional nonlinear advection equation becomes the below expression by attaching the first-order auxiliary equation to the bottom row:

$$\frac{\partial}{\partial t}\begin{bmatrix} U \\ U_x \end{bmatrix} + \underbrace{\begin{bmatrix} \mathbf{A} & 0 \\ \mathbf{A}_x & \mathbf{A} \end{bmatrix}}_{\mathbf{B}} \frac{\partial}{\partial x}\begin{bmatrix} U \\ U_x \end{bmatrix} = 0 \tag{29}$$

One may notice that the MLC system Jacobian $\mathbf{B}$ has the same eigenvalues as the original Jacobian $\mathbf{A}$ through the Schur complement and assume $\mathbf{A}$ is invertible. The physical interpretation of this statement is that the underlying wave speeds for both the value layer and the derivative layer are the same, which is reasonable since the values and their derivatives should travel at the same speed along the characteristics. Henceforth, the MLC adaptation of the local Lax-Friedrichs method, as shown in Eq. (10), remains valid when differentiating via the chain rule:

$$F^{\pm}_{j,x} = \frac{1}{2}\left[F_{j,x} \pm \left(\lambda_j U_x + \lambda_{j,x} U\right)\right], \quad \lambda_j = c + \sqrt{{u_j}^2 + (\epsilon c)^2}, \quad \lambda_{j,x} = c_x + \frac{u_j u_{j,x} + \epsilon^2 c c_x}{\sqrt{{u_j}^2 + (\epsilon c)^2}} \tag{30}$$

With $F^{\pm}_{j,x}$ from above and $F^{\pm}_{j}$ from Eq. (10), both upwind and downwind second-order flux derivatives $D^{\pm}_{xx} F_j$ can be approximated by the split fluxes with corresponding direction MLC stencils.

In Bai and Zhong [33], the maximum eigenvalue λ_j used in the local Lax-Friedrichs method is based on the local maximum eigenvalues inside the stencil. This procedure requires a local maximum eigenvalue search from the neighboring values used by the stencil. In the proposed approach, all of the variables in Eq. (30) are local and do not require a neighboring search process. Also, since the MLC system Jacobian $\mathbf{B}$ has the same maximum eigenvalue of $\mathbf{A}$, the proposed method is capable of splitting Eq. (29) into two pure left and right direction problems. Compared to Bai and Zhong's approach, the additional $\lambda_{j,x} U$ term preserves the differentiation relation between $F^{\pm}_{j}$ and $F^{\pm}_{j,x}$, which slightly improves simulation accuracy. Furthermore, the usage of smoothing parameter ϵ ensures the flux is differentiable in the vicinity of the bow shock center line and suppresses the carbuncle phenomenon, which can lead to a diverged simulation.

3.2. Arbitrary Lagrangian–Eulerian formulation

For the moving grid shock-fitting method, the arbitrary Lagrangian-Eulerian (ALE) formulation is crucial for accounting for changes in the solution state due to the motion of Lagrangian grid points within an Eulerian flow field. This formulation allows the solutions behind the shock wave to respond appropriately to the shockfront movement. For a two-dimensional system, Eq. (13) can be expressed as follows for time integration:

$$\frac{\partial U}{\partial t} = -\frac{1}{g}\left(\frac{\partial F_1'}{\partial \xi} + \frac{\partial F_2'}{\partial \eta} - U\frac{\partial g}{\partial t}\right) \tag{31}$$

This is the standard formulation for applying the finite difference method within the ALE framework. In the finite difference method, the spatial derivatives of fluxes require numerical approximation, as discussed in the previous chapter. For MLC, the time derivative of the value layer U can be computed analytically through the chain rule if the derivative layers U_ξ and U_η are given. However, the derivative layers require time integration from the auxiliary equations shown below:

$$\begin{aligned}
\frac{\partial U_\xi}{\partial t} &= \frac{g_\xi}{g^2}\left(\frac{\partial F_1'}{\partial \xi} + \frac{\partial F_2'}{\partial \eta} - U\frac{\partial g}{\partial t}\right) - \frac{1}{g}\left(\frac{\partial^2 F_1'}{\partial \xi^2} + \frac{\partial^2 F_2'}{\partial \xi \partial \eta} - U_\xi \frac{\partial g}{\partial t} - U\frac{\partial g_\xi}{\partial t}\right)\\
\frac{\partial U_\eta}{\partial t} &= \frac{g_\eta}{g^2}\left(\frac{\partial F_1'}{\partial \xi} + \frac{\partial F_2'}{\partial \eta} - U\frac{\partial g}{\partial t}\right) - \frac{1}{g}\left(\frac{\partial^2 F_1'}{\partial \xi \partial \eta} + \frac{\partial^2 F_2'}{\partial \eta^2} - U_\eta \frac{\partial g}{\partial t} - U\frac{\partial g_\eta}{\partial t}\right)
\end{aligned} \tag{32}$$

Similar to the finite difference discretization for first-order derivatives, the second-order derivatives in the auxiliary equations are approximated using upwind MLC discretization. Most terms in the auxiliary equations can be derived through the chain rule and applied consistently across different equations. Appendix A.1 provides detailed relations between the required derivative terms for MLC calculations. For directional MLC (DMLC) or three-dimensional implementations, a similar approach can be followed to derive the necessary expressions.

3.3. Inflow conditions for MLC-SF

For the shock-fitting method on MLC-SF, the flow states right behind the shock, U_s, are derived from Eq. (15) but the state derivatives remain undetermined. To ensure the MLC derivative layers' boundary condition is consistent with the above shock acceleration computation, this study adopts the same idea of using Rankine-Hugoniot conditions and flow characteristics. To simplify the deriving procedure, two-dimensional Euler equations are used and the shockfront tangential and normal directions are ξ and η, respectively.

For the derivative layer tangential to the shockfront surface, $(U_\xi)_s$, the derivative can be directly derived from the spatial differentiation of Eq. (15), the Rankine-Hugoniot conditions:

$$\left(\vec{F}_s \cdot \vec{a}_G + U_s b_G\right)_\xi = \left(\vec{F}_0 \cdot \vec{a}_G - U_0 b_G\right)_\xi \tag{33}$$

This is due to the flow states behind the shock being derived from the upstream condition and shockfront geometry. Hence, the derivatives along the shockfront surface are independent of the downstream flow states.

For the derivative layers in the shockfront normal direction, $(U_\eta)_s$, the boundary condition deriving process starts from rearranging Eq. (19):

$$\frac{\partial U_s}{\partial t} = \mathbf{B}'^{-1}_s\left[\mathbf{B}'_0 \frac{\partial U_0}{\partial t} - \left(\vec{F}_s - \vec{F}_0\right)\cdot \frac{\partial \vec{a}_G}{\partial t} - \left(U_s - U_0\right)\frac{\partial b_G}{\partial t}\right] \tag{34}$$

Then the time derivative of the flow states is obtained, and the spatial derivative of the flow state, $(U_\eta)_s$, can be acquired from the inverse of the Euler equations with the flow Jacobian in the η direction, $\mathbf{B}_s$:

$$\frac{\partial U_s}{\partial \eta} = -\mathbf{B}_s{}^{-1}\left(\frac{\partial U_s}{\partial t} + \mathbf{A}_s \frac{\partial U_s}{\partial \xi}\right) \tag{35}$$

The physical meaning of the above process is utilizing the wave propagation direction in flow characteristics. In Eq. (20), the shock acceleration depends on the time derivatives associated with the C^+ characteristics line. This is due to C^+ being the only characteristic conveying information toward the shockfront. The rest of the characteristics, which originated from the shockfront, should depend on upstream states. Hence, Eq. (34) implicitly computes the rest of the state derivatives with the state derivatives associated with C^+ fixed. This new boundary condition at shockfront provides more accurate and numerically stable results than simply using extrapolation or one-sided finite difference to estimate the derivatives normal to the shock $U_{s,\eta}$. The testing results of this boundary condition are shown in the one-dimensional shock-density wave problem and two-dimensional entropy-vorticity wave interaction with shock problems. For the Navier-Stokes equations, Eq. (35) needs to consider the viscous effect in the state-inversion process.

3.4. Outflow conditions for MLC

In the supersonic flow over a blunt object simulation, both supersonic and subsonic outflow conditions are required to apply the Navier-Stokes equations. This is because the flow field consists of the subsonic viscous boundary layer in the vicinity of the no-slip wall and the supersonic flow outside the layer. In the Navier-Stokes characteristic boundary conditions (NSCBC) proposed by Poinsot and Lele [74], the supersonic and subsonic outflow conditions should be treated differently due to the characteristic propagation directions. The method derives the boundary condition by converting a multi-dimensional into the local one-dimensional inviscid (LODI) relation and applying the method of characteristics to determine which Riemann invariant-like variables $\mathcal{L}$ should be computed. This procedure makes the NSCBC a robust boundary condition which can be applied to high-order numerical schemes without the usage of interpolation or extrapolation. This study borrows the idea of NSCBC to derive boundary conditions for the MLC derivative layers when applicable. Appendix A.2 provides the two-dimensional NSCBC with MLC.

For supersonic outflow conditions, the flow states on the boundary can be simply computed through time integration from the interior points due to the characteristics all leaving the computational domain, and no additional information is needed to flow in.

$$\begin{aligned} \left.\frac{dU}{dt}\right|_{out} &= L(U, U_\xi, U_\eta, t) \\ \left.\frac{dU_\xi}{dt}\right|_{out} &= L_\xi(U, U_\xi, U_\eta, t) \\ \left.\frac{dU_\eta}{dt}\right|_{out} &= L_\eta(U, U_\xi, U_\eta, t) \end{aligned} \tag{36}$$

For the inviscid subsonic non-reflecting outflow, building a perfectly non-reflecting boundary condition might not lead to a well-posed problem. If one specifies the amplitude of the incoming characteristics $\mathcal{L}_1$ to zero, the simulation could lead to a diverged solution due to a potential ill-posedness of the problem. The NSCBC suggests converting the boundary condition into a partially non-reflecting condition by imposing a pressure at infinity p_∞. This reference variable represents the static pressure outside the computational domain. This outflow condition will be based on the pressure difference between the pressure on the boundary and far-field pressure to generate a weak $\mathcal{L}_1$ into the computational domain where $\mathcal{L}_1$ takes the below form:

$$\mathcal{L}_1 = K(p - p_\infty) \tag{37}$$

where K is a constant: $K = \sigma(1 - M^2)c/L$ which is based on the maximum Mach number in the flow M, the characteristic size of the domain L, and a scaling parameter σ. For $\sigma = 0$ condition, this outflow condition will become the perfectly non-reflecting outflow condition, which may cause an unstable simulation. One should select an appropriate σ for different flow fields.

By introducing a similar concept to the MLC inviscid subsonic non-reflecting outflow condition, the $\mathcal{L}_1$ and the corresponding spatial derivatives can be estimated as follows through direct differentiation:

$$\begin{aligned} \mathcal{L}_1 &= K(p - p_\infty) \\ \mathcal{L}_{1,\xi} &= K_\xi(p - p_\infty) + K p_\xi \\ \mathcal{L}_{1,\eta} &= K_\eta(p - p_\infty) + K p_\eta \end{aligned} \tag{38}$$

With the $\mathcal{L}_1$ computed from the above relation, the rest of the boundary flow states can be obtained through the procedure from Appendix A.2.

For the viscous subsonic outflow condition, Poinsot and Lele [74] found that fixing the normal derivatives of the tangential viscous stresses and the normal heat flux appears to be the best choice in terms of simulation stability. However, a similar approach for MLC leads to unstable simulation results. For the current MLC implementation for viscous subsonic outflow conditions, a compact extrapolation based on the interior points is used for the value and wall-normal derivative layers. The wall-tangential derivative layer is obtained through a central finite difference method from the extrapolated value layer. Even though this extrapolation approach has limited accuracy, MLC is able to converge for the supersonic flow over blunt object simulations by using this approach.

3.5. Wall conditions for MLC

Both inviscid slip wall and viscous isothermal no-slip wall boundary conditions are considered in the present study. Even though these boundary conditions are also discussed in the NSCBC by Poinsot and Lele [74], the current MLC implementations utilizing the NSCBC deriving process lead to unstable simulation results. The reason for the difference between the original NSCBC approach and the MLC version is more than the discrepancy in the usage of the numerical scheme. The introduction of arbitrary Lagrangian-Eulerian formulation to MLC and the curvilinear mapping for the curve walls can also affect the requirement of the boundary condition. In the present study, the wall boundary conditions heavily rely on extrapolation and interpolation from the interior points. This approach has relatively stronger numerical stability by controlling the order of the accuracy of the extrapolation process. However, the extra/interpolation-based approach can only be used with at most fifth-order approximations with MLC, which limits the global accuracy of the final simulation results.

The physical condition of the inviscid slip wall is zero wall-normal velocity for preventing the flow from penetrating the wall. This condition allows the tangential component of velocity and the rest of the flow properties to change freely. For the current MLC implementation of this boundary condition, all of the value and derivative layers are obtained through extrapolation from the interior points. Then, the zero wall-normal velocity condition is imposed on the velocity components, and the velocity tangential derivative layer is recalculated through finite difference discretization.

For the viscous isothermal no-slip wall, the direct extrapolation technique used in the inviscid slip wall boundary will diverge when the grid size is reduced. The viscous boundary condition requires an additional treatment when applying MLC. The current MLC implementation for this boundary type still uses direct extrapolation on the wall-normal derivative layer for the velocity components and temperature. Extrapolations on the value and wall-tangential layers are not required since these values are given in the isothermal no-slip wall boundary. However, the pressure gradients are estimated based on the idea of all of the velocity components staying zero all the time on the wall, which means the time derivative in the x and y-momentum equations are:

$$\frac{\partial \rho u}{\partial t} = \frac{\partial \rho v}{\partial t} = 0 \tag{39}$$

With the above conditions, one may apply the no-slip boundary condition and rewrite the momentum equation in ALE formulation to obtain the below relation:

$$\begin{aligned} y_\eta \frac{\partial p}{\partial \xi} - y_\xi \frac{\partial p}{\partial \eta} &= y_\eta \frac{\partial \tau_{xx}}{\partial \xi} - x_\eta \frac{\partial \tau_{xy}}{\partial \xi} - y_\xi \frac{\partial \tau_{xx}}{\partial \eta} + x_\xi \frac{\partial \tau_{xy}}{\partial \eta} \\ -x_\eta \frac{\partial p}{\partial \xi} + x_\xi \frac{\partial p}{\partial \eta} &= y_\eta \frac{\partial \tau_{xy}}{\partial \xi} - x_\eta \frac{\partial \tau_{yy}}{\partial \xi} - y_\xi \frac{\partial \tau_{xy}}{\partial \eta} + x_\xi \frac{\partial \tau_{yy}}{\partial \eta} \end{aligned} \tag{40}$$

This relation describes how the pressure gradients react to the wall shear stresses in a transformed coordinate. Since the velocity value and gradients are known, through either the boundary conditions or extrapolation, the shear stress can be computed, and pressure gradients can be obtained by solving the relation with matrix inversion:

$$\begin{bmatrix} p_\xi \\ p_\eta \end{bmatrix} = J \begin{bmatrix} x_\xi & y_\xi \\ x_\eta & y_\eta \end{bmatrix} \begin{bmatrix} C_1 \\ C_2 \end{bmatrix} \tag{41}$$

where

$$\begin{aligned} C_1 &= y_\eta \frac{\partial \tau_{xx}}{\partial \xi} - x_\eta \frac{\partial \tau_{xy}}{\partial \xi} - y_\xi \frac{\partial \tau_{xx}}{\partial \eta} + x_\xi \frac{\partial \tau_{xy}}{\partial \eta} \\ C_2 &= y_\eta \frac{\partial \tau_{xy}}{\partial \xi} - x_\eta \frac{\partial \tau_{yy}}{\partial \xi} - y_\xi \frac{\partial \tau_{xy}}{\partial \eta} + x_\xi \frac{\partial \tau_{yy}}{\partial \eta} \end{aligned} \tag{42}$$

Then the pressure value layer p on the wall can be computed from the wall-normal derivatives with a reversed finite difference discretization. This additional pressure treatment improves the MLC stability with the viscous isothermal no-slip wall boundary when a finer grid is applied.

4. Numerical results – verifications

In this section, three test cases are selected to verify and benchmark the performance of the proposed multi-layer compact schemes with the shock-fitting method (MLC-SF) in fifth- and seventh-order accuracy formulations. For comparison, this paper also evaluates the conventional upwind finite difference method with the shock-fitting method (FD-SF) [4], along with three weighted essentially non-oscillatory (WENO) schemes. The tested WENO schemes include WENO5-JS [44], WENO5-M [46], and WENO5-Z [47], which represent the simulation results obtained using shock-capturing techniques. Detailed parameter settings for the WENO methods will be provided in the one-dimensional linear advection problem section. For the two-dimensional moving mesh problem, the dynamic multi-layer compact (DMLC) method will also be discussed. Time integration for all three verification cases is conducted using the fourth-order Runge-Kutta method, with the time step set sufficiently small to minimize temporal errors. The corresponding Courant-Friedrichs-Lewy (CFL) number for the numerical tests ranges from 0.001 to 0.01. For the Euler equations, the modified local Lax-Friedrichs method [4] is adopted for flux-vector splitting.

4.1. One-dimensional linear advection wave

The one-dimensional linear advection problem with periodic boundaries from Henrick et al [46] is given as follows:

$$\frac{\partial u}{\partial t}+\frac{\partial u}{\partial x}=0 \quad \text{on } x\in[-1,1], \quad t\in[0,2] \tag{43}$$

and the initial condition:

$$u(x,t=0)=sin\left(\pi x-\frac{sin(\pi x)}{\pi}\right) \tag{44}$$

The problem is designed to examine WENO methods when the first derivative vanishing points existed in the system. From Henrick et al. [46] works, WENO5-JS has convergence issues near the vanishing point even though the solution is considered smooth. Hence, both WENO-M and WENO-Z are designed to address this issue and converge with optimal spatial accuracy. The parameter, ϵ, of WENO local smoothness indicator in WENO5-JS is set to 10^{-6} and for WENO5-M and WENO5-Z are 10^{-40}.

Fig. 2 illustrates the L^1-norm error convergence rate versus the mesh number, N, and the solutions at the simulation's end time for the fifth-order methods. Among these methods, WENO5-JS demonstrates the slowest convergence rate, consistent with its known convergence properties. In contrast, WENO5-M and WENO-Z achieve the optimal fifth-order convergence rate. A key distinction between these two methods lies in their computational cost: WENO-Z requires 30 % less computing time compared to WENO-M [48], as it evaluates smoothness indicators using absolute differences rather than recalculating weights for each indicator from the mapping function.

For the shock-fitting methods, the upwind finite difference method maintains a steady fifth-order convergence rate, with slightly lower errors than the modified WENO methods at equivalent mesh densities. Meanwhile, MLC demonstrates the best overall spatial accuracy among all methods. This superior accuracy can be attributed to the additional degrees of freedom in MLC, which is twice as many as the other methods for the same number of grid points. Even when accounting for this disparity by normalizing the degrees of freedom, MLC still achieves lower errors than the others.

This trend of superior spatial accuracy for MLC extends to the seventh-order results, as shown in Fig. 3. The results from the one-dimensional linear advection problem highlight MLC's capability to handle smooth solutions with exceptional precision. Fig. 4 shows

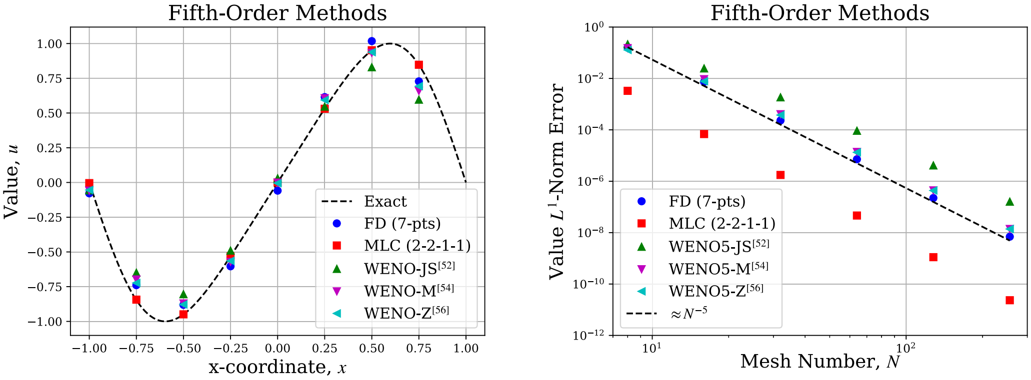


Fig. 2. One-dimensional linear advection: fifth-order methods comparison for solutions at $t=2$ with $N=8$ (left) and L^1-norm error convergence rate (right) .

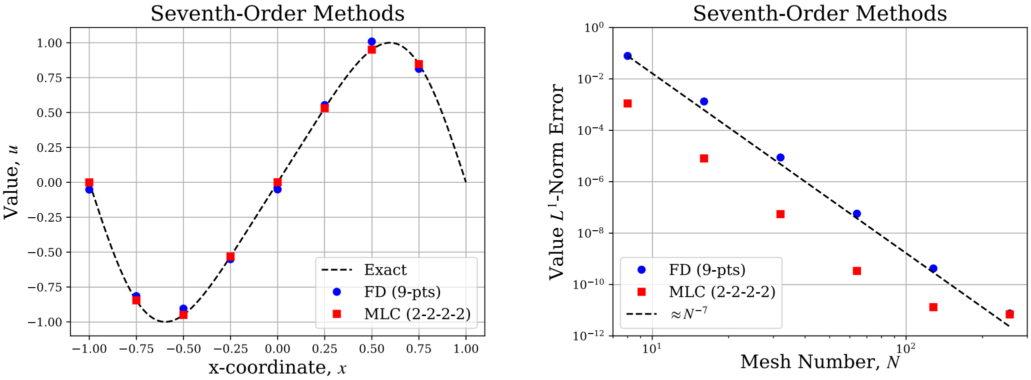


Fig. 3. One-dimensional linear advection: seventh-order methods comparison for solutions at $t=2$ with $N=8$ (left) and L^1-norm error convergence rate (right).

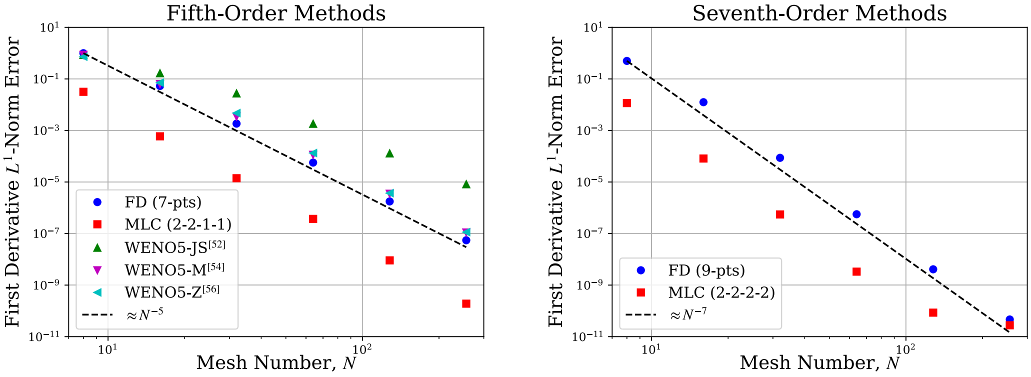


Fig. 4. One-dimensional linear advection: first derivative L^1-norm error convergence rate for fifth-order (left) and seventh-order (right) methods.

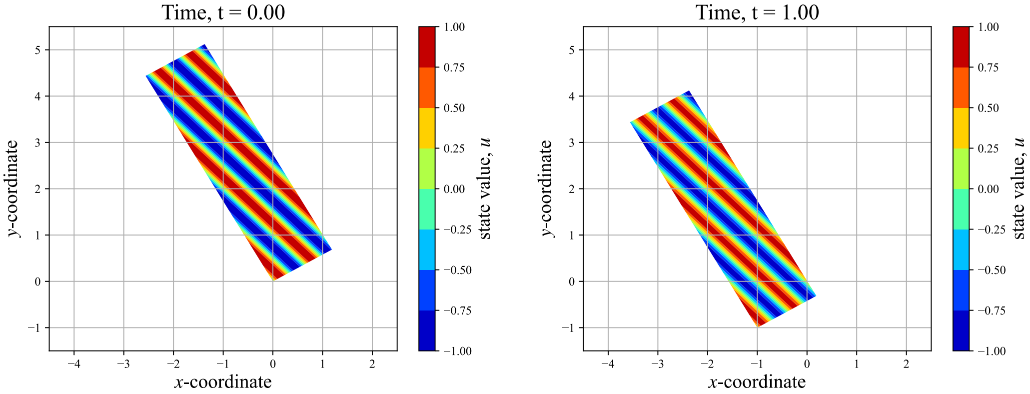


Fig. 5. Two-dimensional linear advection with a moving mesh: solution contours of the initial state (left) and the end state (right).

the error convergence rates of the first derivative for different methods. The convergence behavior of the first derivative follows a similar trend to that of the solution value, except that WENO-JS exhibits an even slower convergence rate. As before, the MLC scheme achieves the lowest error for a given mesh number and degrees of freedom among all tested methods.

4.2. Two-dimensional linear advection wave on a moving grid

The two-dimensional linear advection with moving mesh problem aims to verify MLC and DMLC with a two-dimensional advection equation with moving mesh treatment. The problem is given as a rectangular computational domain with a tilt angle $\theta = 30°$ counter-clock wisely and the entire domain is moving uniformly in $\vec{v} = (-1, -1)$. The governing equation for this problem can be derived from Eq. (13):

$$\frac{\partial gU}{\partial t} + \frac{\partial F_1'}{\partial \xi} + \frac{\partial F_2'}{\partial \eta} = 0 \quad \text{on } \xi, \eta \in [0,1], \quad t \in [0,1] \tag{45}$$

where the state and fluxes in the physical domain are $U = F_1 = F_2 = u$. For the initial condition:

$$u(x, y, t = 0) = \sin(2\pi(x + y)) \tag{46}$$

and the coordinate transformation function:

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} D_x \cdot \cos(\theta) & -D_y \cdot \sin(\theta) \\ D_x \cdot \sin(\theta) & D_y \cdot \cos(\theta) \end{bmatrix} \begin{bmatrix} \xi \\ \eta \end{bmatrix} \tag{47}$$

Note that D_x and D_y are properly selected to ensure the state value is consistent with the periodic boundaries. Fig. 5 shows the problem's initial and end solution contours in the physical domain.

Fig. 6 illustrates the solution profiles along the $\eta = 0$ line for FD, MLC, and DMLC methods. For both fifth- and seventh-order approaches, the FD method exhibits the least accuracy compared to the two MLC-based methods. In the fifth-order results, DMLC demonstrates slightly higher accuracy than MLC. However, for the seventh-order cases, the differences between MLC and DMLC are negligible and visually indistinguishable.

Fig. 7 presents the error convergence plots for the upwind finite difference method, MLC, and DMLC. In fifth-order methods, MLC and DMLC show very similar error trends with respect to the mesh number, both outperforming the upwind finite difference

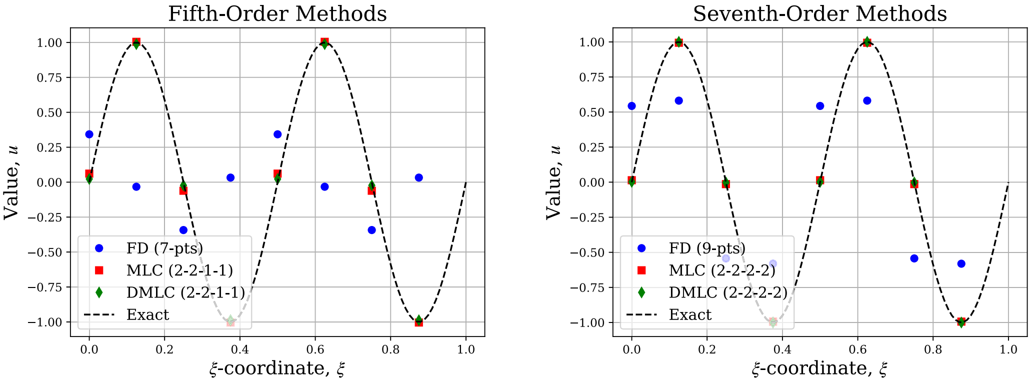


Fig. 6. Two-dimensional linear advection with a moving mesh: Solution comparison between different methods with $N = 8$ and $\eta = 0$ for fifth-order methods (left) and seventh-order methods (right).

method in terms of accuracy per degree of freedom and computational efficiency. For the seventh-order methods, all three approaches exhibit comparable error distributions relative to degrees of freedom, although MLC slightly deviates from the trend for certain grid resolutions.

In terms of computational efficiency, DMLC significantly outperforms the upwind finite difference method. To achieve an L^1-norm error of approximately 10^{-9}, the seventh-order DMLC required 54.7 s, whereas the seventh-order upwind finite difference method took 163.4 s. This represents a 2.99 times speedup for DMLC, underscoring its efficiency for high-accuracy simulations.

4.3. One-dimensional entropy-shock interaction wave

The one-dimensional shock-density wave interaction presented in this study is based on Shu-Osher's problem [75] and revised by Suresh [76] for improving problem smoothness. The problem simulates a normal shock moving at Mach number, M, relative to the upstream and interacting with a density wave. The governing equations are the Euler equations:

$$\frac{\partial}{\partial t}\begin{bmatrix}\rho\\ \rho u\\ \rho e\end{bmatrix}+\frac{\partial}{\partial x}\begin{bmatrix}\rho u\\ \rho u^2+p\\ (\rho e+p)u\end{bmatrix}=0 \quad \text{on } x\in[-1,1], \quad t\in[0,0.36] \tag{48}$$

with the initial condition for downstream, $x \le x_G$:

$$\begin{aligned}\rho(x,t=0)&=\left[(\gamma+1)M^2\right]/\left[(\gamma-1)M^2+2\right]\\ u(x,t=0)&=\left[2\sqrt{\gamma}(M^2-1)\right]/[(\gamma+1)M]\\ p(x,t=0)&=1+\left[2\gamma(M^2-1)\right]/(\gamma+1)\end{aligned} \tag{49}$$

and for upstream, $x > x_G$:

$$\begin{aligned}\rho(x,t=0)&=1.0+\delta\cdot\sin^4(2.5\pi x)\\ u(x,t=0)&=0\\ p(x,t=0)&=1\end{aligned} \tag{50}$$

where $\gamma = 1.4$, $M = 3$, $\delta = 0.2$ and $x_G = -0.8$. The problem selected is due to fluxes relative to the shockfront being C^3 continuous when $t = 0$. This smoothness property provides a more gentle start for the density interaction process and yields a better convergence rate for high-order methods with the shock-fitting method.

Fig. 9 shows the global solution profile and error for fifth-order FD-SF, MLC-SF, and WENO5-JS with mesh number $N = 1,600$. The reference solution is from fifth-order FD with $N = 102,400$. From the density error distribution plot, a smooth region, Ω_1, is defined with $x \in [0.18, 0.43]$. This region is the slow entropy wave region and will be used for error evaluation in the later part of this test. Also, the global error is defined by $x \in [-1, 0.43]$ to avoid errors near the shock.

Fig. 8 presents the simulation results of mesh number $N = 100$ for both fifth and seventh-order methods. Among the fifth-order methods, the WENO family exhibits noticeable smearing in the slow entropy wave region. This behavior likely stems from the WENO methods employing less accurate stencils near the shock, which compromises resolution in adjacent regions. In contrast, FD-SF captures the initial oscillations at the first few peaks of the slow entropy wave region, though some smearing is evident toward the end of the region. Overall, FD outperforms pure shock-capturing methods in this test case.

Notably, MLC-SF delivers the most accurate results for both fifth- and seventh-order simulations. The fifth-order MLC-SF successfully tracks all peaks in the slow entropy wave region in a level of precision that even the seventh-order FD-SF method does not achieve. These results highlight MLC-SF's superior ability to resolve fine details in complex flow regions behind the shock when using the shock-fitting method.

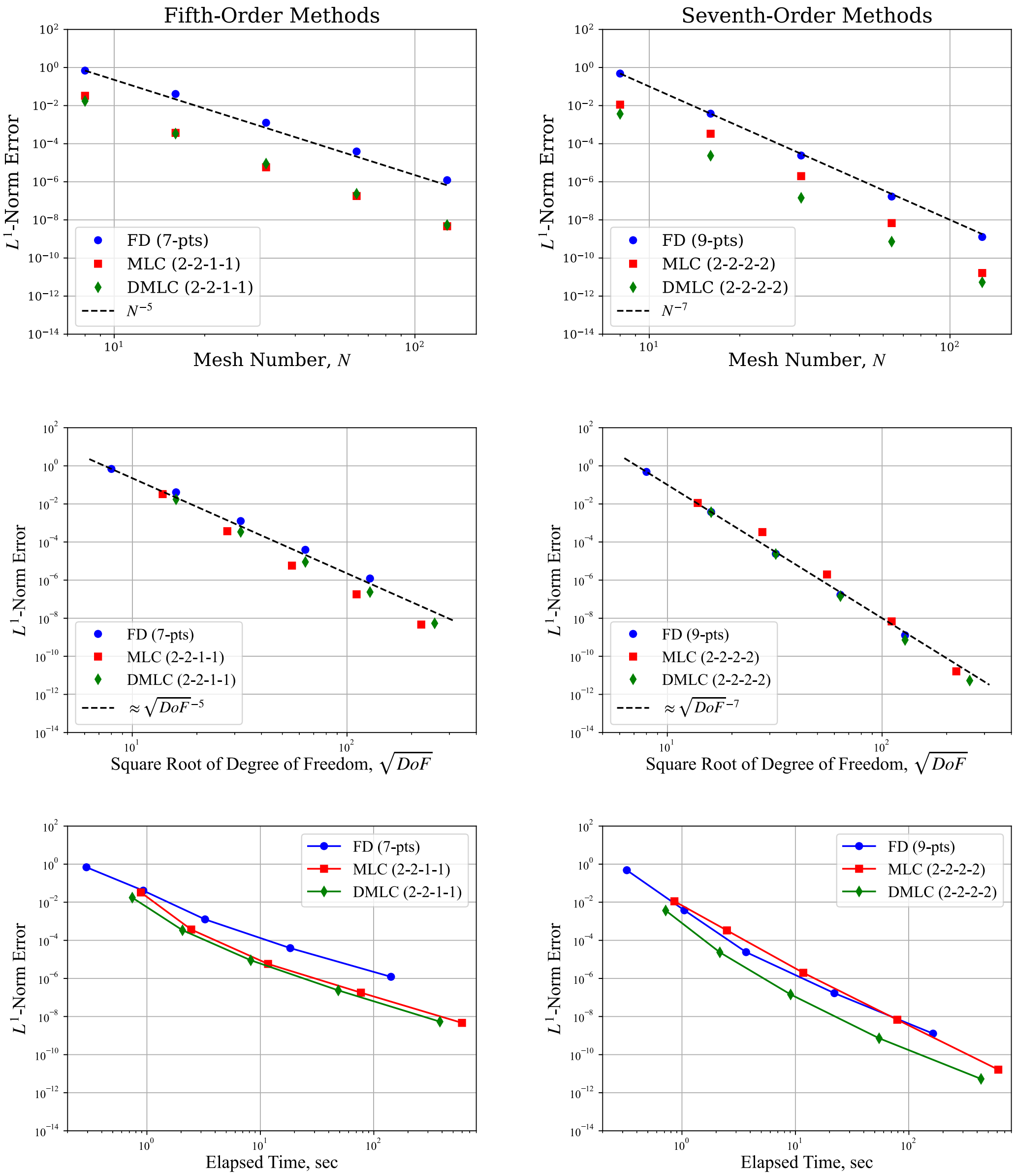


Fig. 7. Two-dimensional linear advection with a moving mesh: L^1-norm error versus mesh number N (top), degrees of freedom (middle), and computing elapsed time (bottom) for fifth (left) and seventh-order (right) methods.

Table 2 shows the global density L^1-norm error for the fifth-order methods. For the shock-fitting methods, the upwind FD-SF converges around fifth-order and MLC-SF can achieve a convergence rate with $r = 5.5$ in certain points. In contrast, the fifth-order shock-capturing schemes, without shock-fitting applied, are converged around first-order even for the two advanced WENO methods. Table 3 shows the error convergence rate of fifth-order methods in the smooth region, Ω_1. For the shock-fitting methods, the upwind finite difference method achieves fifth-order accuracy faster than the global error and MLC-SF's convergence rate is capped at sixth-order. On the other hand, WENO methods' performance is improved but still below second-order.

For seventh-order methods, Table 4 shows the L^1-norm error in both global and smooth regions. Both the upwind finite difference method and MLC are below seventh-order accuracy in the global region. One of the possible reasons for the reduction in spatial accuracy is the upstream flux being C^3 continuous across the shock. The discontinuity in higher-order derivatives causes the high-order approximation to fail and leads to incorrect derivative estimation. The error due to the discontinuity can be observed in Fig. 9 at the interfaces of different system waves (x around -0.5 and 0.2). One way to reduce the discontinuity error is using C^∞ function in the upstream such as the bump function. Nevertheless, the measured spatial accuracy is recovered to the expected order when the solution is evaluated within the smooth region, Ω_1. The FD-SF is capped at sixth-order accuracy due to the usage of the fifth-order

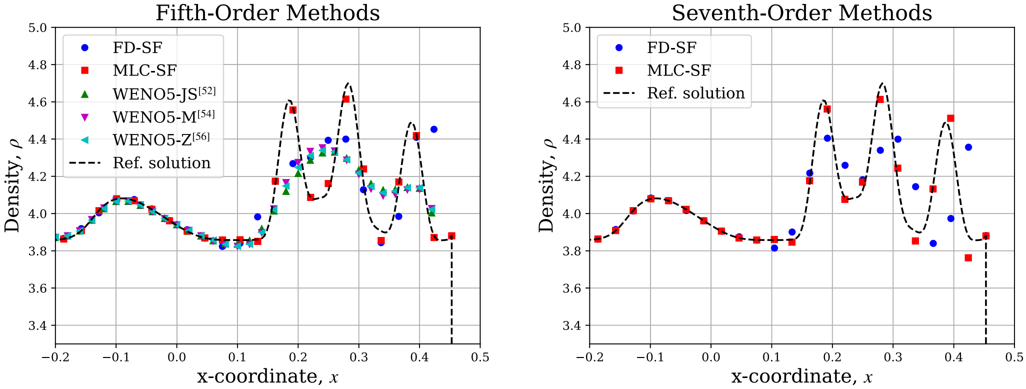


Fig. 8. One-dimensional shock-density wave interaction: Density comparison between different methods with $N = 100$ for fifth-order methods (left) and seventh-order methods (right).

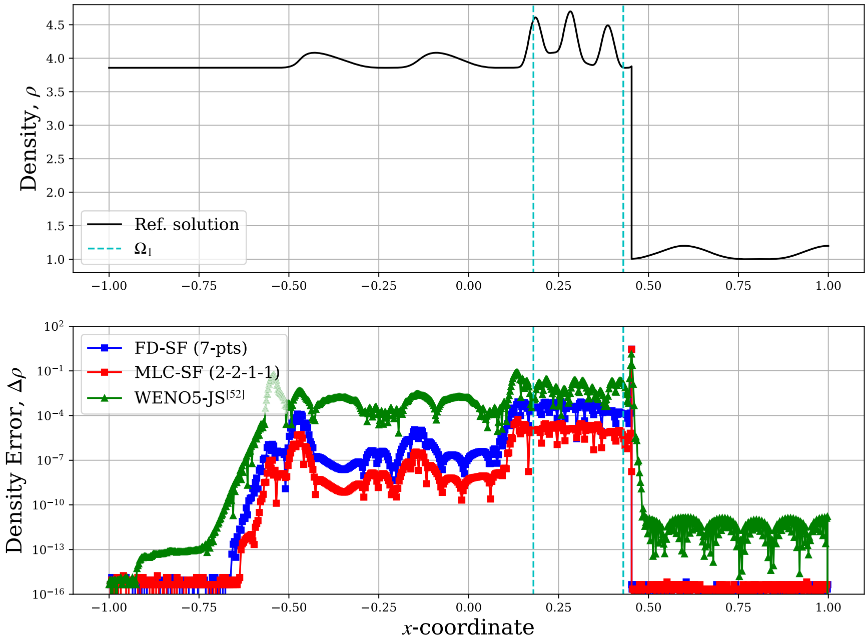


Fig. 9. One-dimensional shock-density wave interaction: density distribution at t = 0.36 (top) and the corresponding density error distribution for fifth-order methods.

Table 2
One-dimensional shock-density wave interaction: global density L^1-norm error and convergence rate for fifth-order methods.

	FD-SF (7-pts)		MLC-SF (2-2-1-1)		WENO5-JS		WENO5-M		WENO5-Z	
N	L^1 error	r	L^1 error	r	L^1 error	r	L^1 error	r	L^1 error	r
50	$3.43 \cdot 10^{-2}$		$6.21 \cdot 10^{-3}$		$4.91 \cdot 10^{-2}$		$4.79 \cdot 10^{-2}$		$4.85 \cdot 10^{-2}$	
100	$1.24 \cdot 10^{-2}$	1.5	$4.65 \cdot 10^{-4}$	3.7	$2.48 \cdot 10^{-2}$	1.0	$2.88 \cdot 10^{-2}$	0.7	$2.82 \cdot 10^{-2}$	0.8
200	$1.86 \cdot 10^{-3}$	2.7	$1.00 \cdot 10^{-5}$	5.5	$1.02 \cdot 10^{-2}$	1.3	$1.25 \cdot 10^{-2}$	1.2	$1.14 \cdot 10^{-2}$	1.3
400	$8.58 \cdot 10^{-5}$	4.4	$2.23 \cdot 10^{-7}$	5.5	$4.13 \cdot 10^{-3}$	1.3	$5.24 \cdot 10^{-3}$	1.3	$4.31 \cdot 10^{-3}$	1.4
800	$3.04 \cdot 10^{-6}$	4.8	$5.79 \cdot 10^{-9}$	5.3	$1.58 \cdot 10^{-3}$	1.4	$2.06 \cdot 10^{-3}$	1.3	$1.63 \cdot 10^{-3}$	1.4
1600	$1.05 \cdot 10^{-7}$	4.9	$2.06 \cdot 10^{-10}$	4.8	$6.85 \cdot 10^{-4}$	1.2	$8.55 \cdot 10^{-4}$	1.3	$6.90 \cdot 10^{-4}$	1.2
3200	$3.78 \cdot 10^{-9}$	4.8	$1.01 \cdot 10^{-11}$	4.4	$3.13 \cdot 10^{-4}$	1.1	$3.66 \cdot 10^{-4}$	1.2	$3.19 \cdot 10^{-4}$	1.1

Table 3

One-dimensional shock-density wave interaction: smooth region (Ω_1) density L^1-norm error and convergence rate for fifth-order methods.

	FD-SF (7-pts)		MLC-SF (2-2-1-1)		WENO5-JS		WENO5-M		WENO5-Z	
N	L^1 error	r	L^1 error	r	L^1 error	r	L^1 error	r	L^1 error	r
50	$1.70 \cdot 10^{-1}$		$3.36 \cdot 10^{-2}$		$2.54 \cdot 10^{-1}$		$2.61 \cdot 10^{-1}$		$2.57 \cdot 10^{-1}$	
100	$5.50 \cdot 10^{-2}$	1.6	$2.30 \cdot 10^{-3}$	3.9	$2.04 \cdot 10^{-1}$	0.3	$2.03 \cdot 10^{-1}$	0.4	$2.03 \cdot 10^{-1}$	0.3
200	$8.84 \cdot 10^{-3}$	2.6	$4.55 \cdot 10^{-5}$	5.7	$9.06 \cdot 10^{-2}$	1.2	$1.14 \cdot 10^{-1}$	0.8	$1.08 \cdot 10^{-1}$	0.9
400	$3.82 \cdot 10^{-4}$	4.5	$7.80 \cdot 10^{-7}$	5.9	$3.29 \cdot 10^{-2}$	1.5	$4.56 \cdot 10^{-2}$	1.3	$3.89 \cdot 10^{-2}$	1.5
800	$1.29 \cdot 10^{-5}$	4.9	$1.26 \cdot 10^{-8}$	5.9	$1.02 \cdot 10^{-2}$	1.7	$1.72 \cdot 10^{-2}$	1.4	$1.19 \cdot 10^{-2}$	1.7
1600	$4.12 \cdot 10^{-7}$	5.0	$2.00 \cdot 10^{-10}$	6.0	$2.83 \cdot 10^{-3}$	1.9	$5.46 \cdot 10^{-3}$	1.7	$3.15 \cdot 10^{-3}$	1.9
3200	$1.30 \cdot 10^{-8}$	5.0	$3.04 \cdot 10^{-12}$	6.0	$8.18 \cdot 10^{-4}$	1.8	$1.57 \cdot 10^{-3}$	1.8	$8.96 \cdot 10^{-4}$	1.8

Table 4

One-dimensional shock-density wave interaction: global and the smooth region density L^1-norm error and convergence rate for seventh-order methods.

	Global				Smooth Region			
	FD-SF (9-pts)		MLC-SF (2-2-2-2)		FD-SF (9-pts)		MLC-SF (2-2-2-2)	
N	L^1 error	r	L^1 error	r	L^1 error	r	L^1 error	r
50	$4.45 \cdot 10^{-2}$		$8.50 \cdot 10^{-3}$		$2.47 \cdot 10^{-1}$		$4.62 \cdot 10^{-2}$	
100	$8.10 \cdot 10^{-3}$	2.5	$1.82 \cdot 10^{-4}$	5.5	$4.18 \cdot 10^{-2}$	2.6	$7.15 \cdot 10^{-4}$	6.0
200	$2.45 \cdot 10^{-4}$	5.1	$2.10 \cdot 10^{-6}$	6.4	$1.13 \cdot 10^{-3}$	5.2	$3.47 \cdot 10^{-6}$	7.7
400	$4.63 \cdot 10^{-6}$	5.7	$5.92 \cdot 10^{-8}$	5.1	$1.92 \cdot 10^{-5}$	5.9	$1.22 \cdot 10^{-8}$	8.2
800	$9.48 \cdot 10^{-8}$	5.6	$1.75 \cdot 10^{-9}$	5.1	$2.75 \cdot 10^{-7}$	6.1	$4.69 \cdot 10^{-11}$	8.0
1600	$2.80 \cdot 10^{-9}$	5.1	$9.31 \cdot 10^{-11}$	4.2	$3.77 \cdot 10^{-9}$	6.2	$1.82 \cdot 10^{-12}$	4.7
3200	$1.19 \cdot 10^{-10}$	4.6	$5.21 \cdot 10^{-12}$	4.2	$5.45 \cdot 10^{-11}$	6.1	$8.69 \cdot 10^{-13}$	1.1

boundary closure schemes for stability. In contrast, MLC-SF can even reach an eighth-order convergence rate with the proposed boundary condition and eventually reduce to first-order due to the emergence of round-off errors.

Regarding computational efficiency, Fig. 10 and Fig. 11 show the L^1-norm error in the global and smooth regions concerning the mesh number, degrees of freedom, and elapsed time. Either in the plots of degrees of freedom or elapsed time, MLC-SF is the fastest method in converging errors. This implies MLC-SF is the most memory and computing time-efficient method among the compared methods. In seventh-order methods, the upwind finite difference method requires 162.7 s to achieve an error of 10^{-10} in the global and smooth regions. On the other hand, MLC-SF only took 122.4 and 48.7 s to reach the error for global and smooth regions respectively. In other words, MLC-SF provides a 1.32 times speedup in the global region and 3.34 times speedup in the smooth region over the upwind finite difference for obtaining density L^1-norm error equal to 10^{-10}.

5. Numerical results – applications

This section extends FD-SF and MLC-SF from the previous section to more realistic shock-involving problems like two-dimensional vorticity-entropy wave interaction with a shock wave [70] and supersonic flow over a parabolic cylinder. These case studies provide the grid convergence rate plots to demonstrate the current capability of MLC-SF compared with well-established FD-SF. In addition to the grid convergence rate studies, preliminary results for MLC with the 1-1-1-1 (third-order) scheme applied to two-dimensional supersonic flow over a cylinder are also presented. Although further study is needed on improving wall boundary conditions for the derivative layers in order to achieve the seventh-order global accuracy, the results demonstrate that the current implementation can converge to a proper solution, highlighting the method's potential for real-world applications.

5.1. Two-dimensional supersonic vorticity–entropy wave interaction with a shock wave

This testing case aims to test the numerical accuracy of the MLC-SF proposed in section 3.3 on the two-dimensional Euler equations. The problem is based on Mahesh's linear perturbation problem [70], which is a supersonic inviscid flow with a normal shock sitting in the middle of the computational domain ($x_G \approx 0$). Fig. 12 shows the initial density fluctuation distribution ($\rho^* = \rho' / A_e \bar{\rho}_1$) in the simulation domain. The left boundary is a supersonic inlet with a given Mach number M_∞ and the right boundary is a subsonic outlet. For the top and bottom boundaries, a periodic interface is applied to these boundaries to connect the simulation solutions on the horizontal borders virtually. The initial fluctuation terms for the upstream zone are applied as follows:

$$\frac{p_1'}{\bar{p}_1} = 0$$

$$\frac{u_1'}{\bar{u}_1} = l A_v \cos(mx + ly - \bar{u}_1 m t)$$

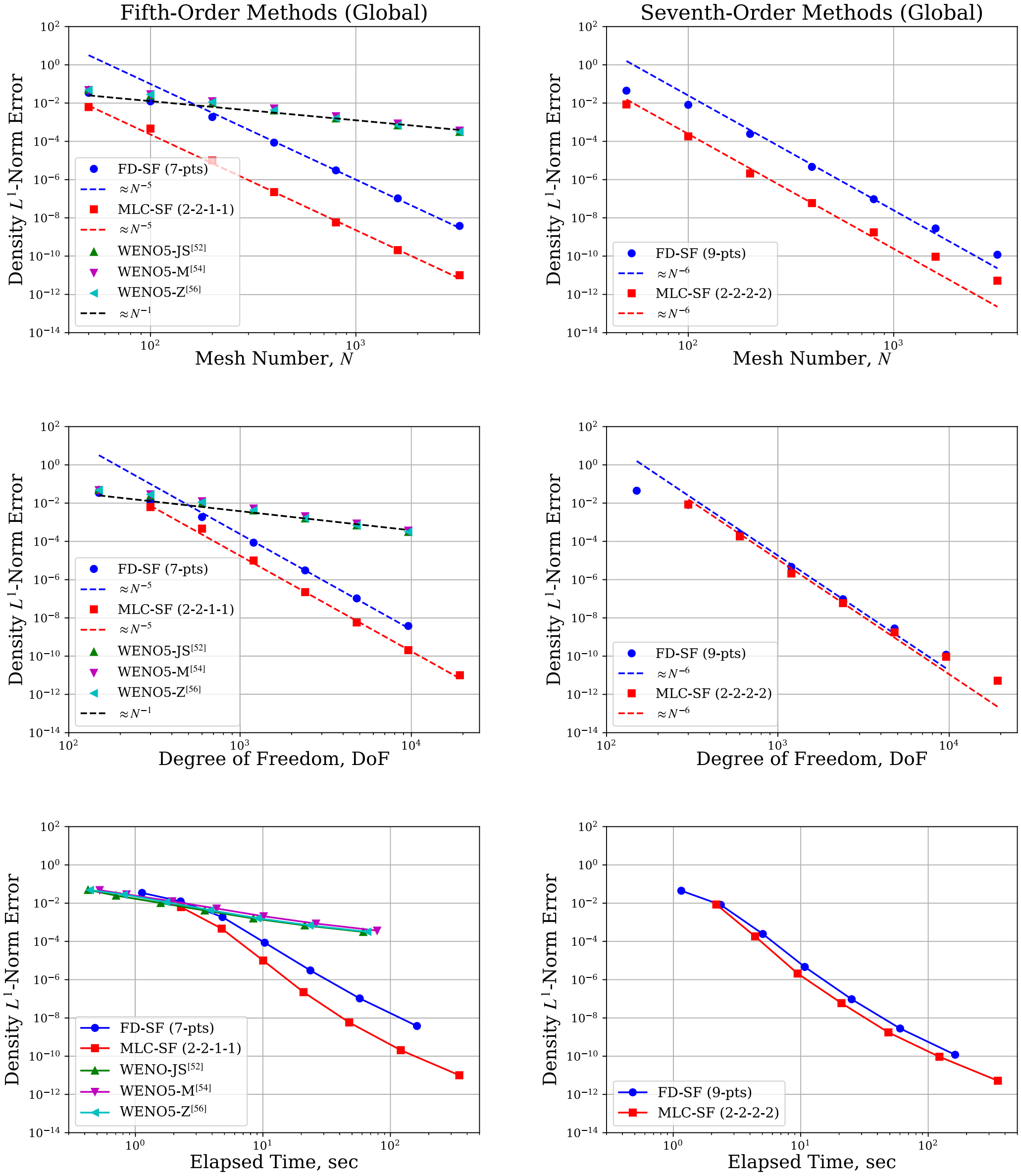


Fig. 10. One-dimensional shock-density wave interaction: global L^1-norm error versus mesh number N (top), degrees of freedom (middle), and computing elapsed time (bottom) for fifth (left) and seventh-order (right) methods.

$$\frac{v_1'}{\bar{u}_1} = -mA_v \cos(mx + ly - \bar{u}_1 mt)$$

$$\frac{\rho_1'}{\bar{\rho}_1} = A_e \cos(mx + ly - \bar{u}_1 mt) \qquad (51)$$

where $l = \sin(\psi_1)$ and $m = \cos(\psi_1)$. In this testing problem, the given freestream and fluctuation parameters are $M_\infty = 1.5$, $A_v = A_e = 10^{-4}$, and $\psi_1 = 15°$. For the initial solutions behind the shock wave, the downstream fluctuations and the shape of the shock wave are derived based on the linear perturbation theory. The linearized solution can be found in Mahesh's work [70]. For the numerical simulation setting, the subsonic outlet boundary condition is implemented using the non-reflecting characteristic boundary from Poinsot and Lele [74]. The entire computational domain, $\Omega = [-1.5\pi, 2\pi] \times [0, 2\pi)$, is split into two zones: upstream and downstream. The number of grid points for each zone is $(N+1) \times N$. The third-order strong stability preserving Runge-Kutta method [72] is used for the time integration and the CFL number ranges from 0.05 to 0.3 for different grid sizes. The total simulation time is $T^* = T/T_{ref} \approx 6.95$ where the reference time is the perturbation traveling time in the spanwise direction, $T_{ref} = 2\pi l/\bar{u}_1 m$. The simulation time is set to prevent the outlet numerical error, a C^- acoustic wave, from reaching the solutions behind the shock wave.

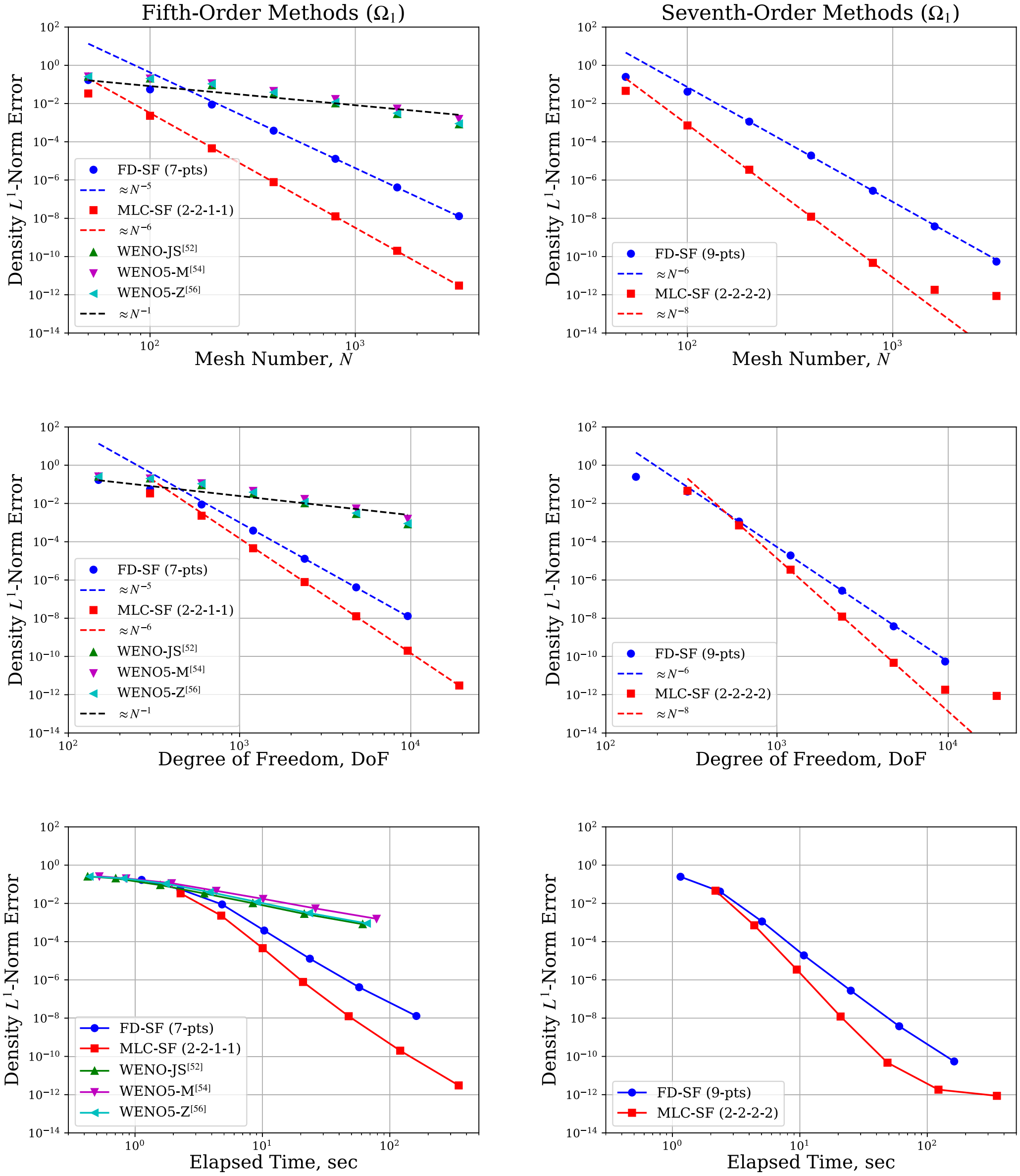


Fig. 11. One-dimensional shock-density wave interaction: L^1-norm error in the smooth region, Ω_1, versus mesh number N (top), degrees of freedom (middle), and computing elapsed time (bottom) for fifth (left) and seventh-order (right) methods.

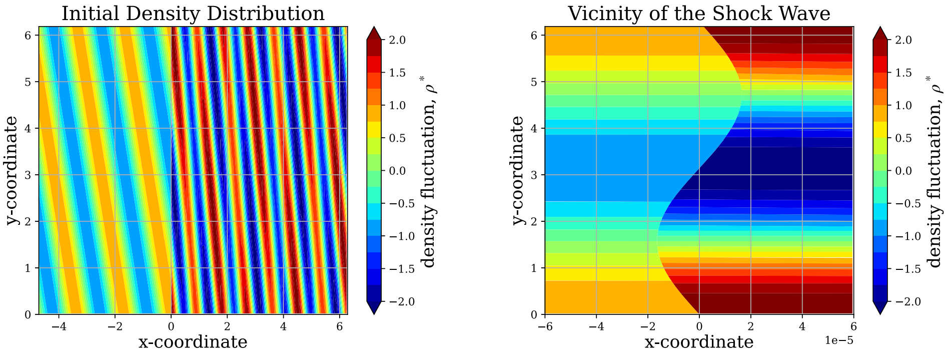


Fig. 12. Two-dimensional shock interaction: the initial density fluctuation distribution ($\rho^* = \rho'/A_e\rho_1$) in the entire computational domain (left) and in the vicinity of the shock wave (right).

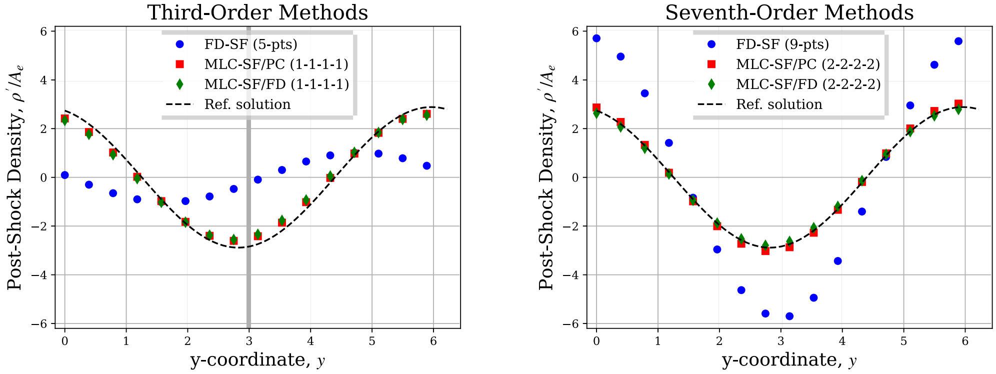


Fig. 13. Two-dimensional shock interaction: post-shock density fluctuation distribution comparison for third-order methods (left) and the seventh-order methods (right).

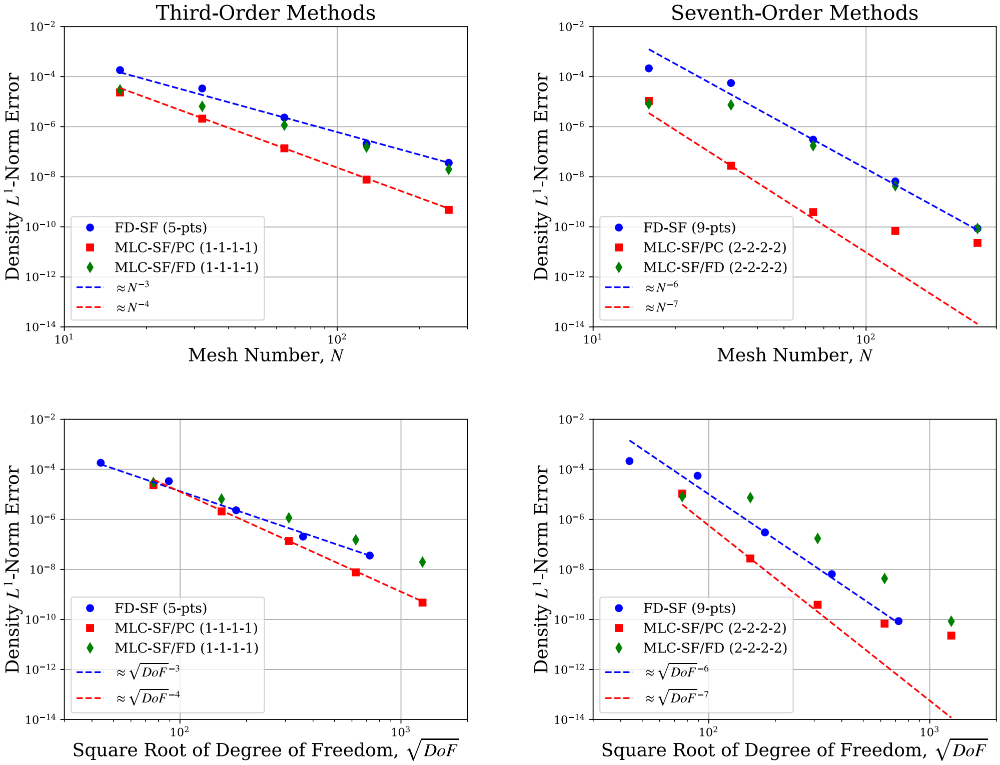


Fig. 14. Two-dimensional shock interaction: L^1-norm error versus mesh number N (top) and degree of freedom (bottom) for third (left) and seventh-order (right) methods.

Fig. 13 shows the simulation results of the post-shock density distribution using third- and seventh-order methods. For the same mesh number, the MLC methods demonstrate higher accuracy than the FD methods. Fig. 14 shows the L^1-norm error with different mesh sizes and orders of the method used. The error is computed based on the density error right behind the shock wave, and the reference solution is the finest grid point solution ($N = 512$) from the seventh-order MLC-SF. In addition to comparing the results between FD-SF and MLC-SF, this testing case also compares different boundary condition implementations for MLC-SF for the stream-wise state derivative behind the shock wave $U_{s,\eta}$. The two boundary condition implementations are the physically consistent method mentioned in section 3.3 (MLC-SF/PC) and the finite difference method (MLC-SF/FD). The MLC-SF/FD uses $(r-1)$th-order one-sided difference schemes to estimate the derivative behind the shock from the downstream state values U. For seventh-order MLC-SF, the MLC-SF/FD approach uses $(r-2)$th-order (fifth-order) one-sided difference due to the numerical stability issue when sixth-order is used. Consequently, the L^1-norm error of the MLC-SF/FD is about the same as the FD-SF for the same mesh number. On the other hand, the MLC-SF/PC approach can estimate the U_η derivative without using one-sided difference. This makes the method outperform the other two methods and achieve a seventh-order global convergence rate for coarser grid points. The testing case

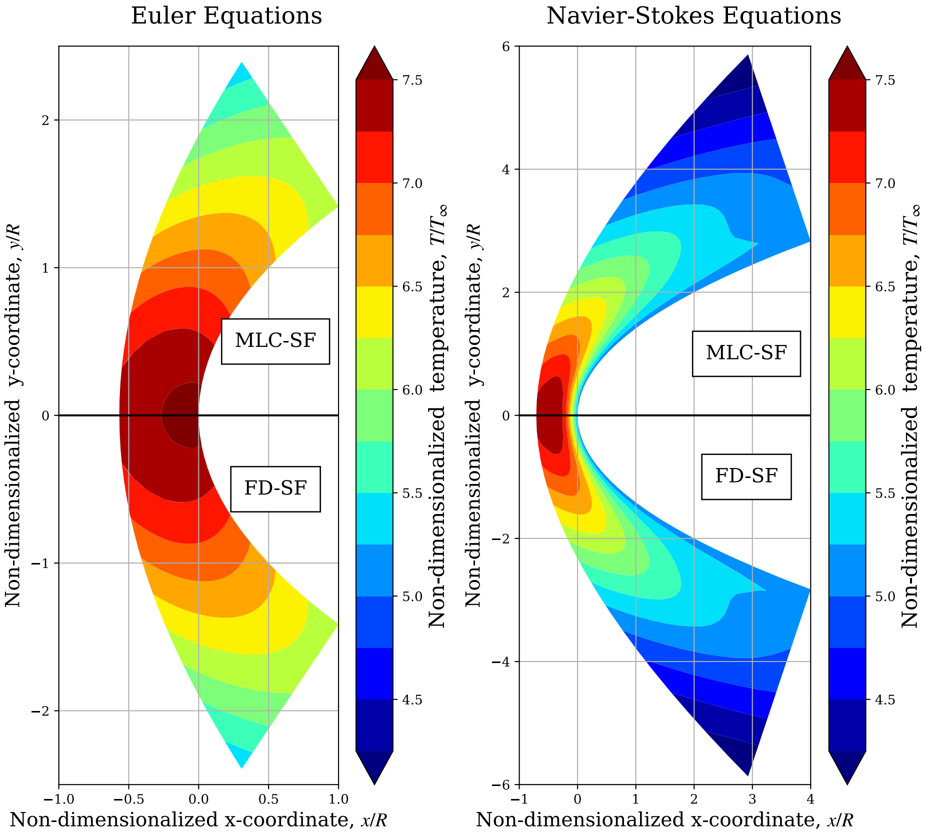


Fig. 15. Two-dimensional supersonic flow over parabolic cylinder: temperature contour for the Euler equations (left) and the Navier-Stokes equation (right) with third-order upwind finite difference method, FD-SF, (lower section) and multi-layer compact scheme, MLC-SF, (upper section).

shows the importance of the boundary condition used for the MLC-SF to achieve the desired global convergence rate for simulation. The key to constructing an accurate MLC boundary condition is to ensure the condition's physical meaning between the value and derivative layers is consistent.

5.2. Two-dimensional supersonic flow over a blunt object

The last simulation case is the two-dimensional supersonic flow over a parabolic cylinder. In this problem, both the Euler and Naver-Stokes equations are considered. For the geometry of the cylinder, the parabola is defined based on the radius of curvature at the parabola center R. The explicit wall's coordinate function is:

$$x_w = \frac{1}{2R} {y_w}^2 \tag{52}$$

The freestream flow conditions are $M_\infty = 5.73$, $Re_R = 100$, and $T_\infty = 250$ K. For simplicity, this problem only considers the ideal gas model with $\gamma = 1.4$ and $Pr = 0.77$. The rest of the flow conditions can be found in Zhong's supersonic flow over a cylinder study [4].

The tested schemes are the third-order FD-SF (five-point stencil) and MLC-SF (1-1-1-1) for this problem set. In the computational domain, ξ is defined as the clockwise azimuthal direction to the parabola, and η is defined as the wall-normal direction toward the shockfront. The parabola endpoints for the Euler equation are located at $x_{end} = R$ and for the Navier-Stokes equations are $x_{end} = 4R$. Also, the mesh number in η direction for Navier-Stokes problems is doubled compared to the inviscid flow. The discrepancy in the domain size and mesh number used for the two equations is due to the Navier-Stokes equations having convergence issues at the outlet for shorter computational domains and strong velocity and temperature gradient in the vicinity of the stagnation point. For boundary condition implementations, a mirror condition is used at the centerline to reduce the computational cost. The outlet boundary states in the Navier-Stokes simulations are computed through rth-order polynomial extrapolation for both value and derivative layers from the interior points. The inlet states are obtained from the shock-fitting method described in section 3.3 (MLC-SF/PC). For the Euler wall boundary, the inviscid non-penetrable wall is applied. The pressure term is computed from the pressure gradient based on the momentum equation in the wall-normal direction. Non-slip and isothermal conditions are applied to the Navier-Stokes wall boundary. The wall temperature $T_w = 1250$ K, and the pressure term is extrapolated from the interior points. In addition, MLC-SF uses polynomial extrapolation to obtain velocity derivatives on the wall. All of the simulations use third-order strong stability preserving Runge-Kutta

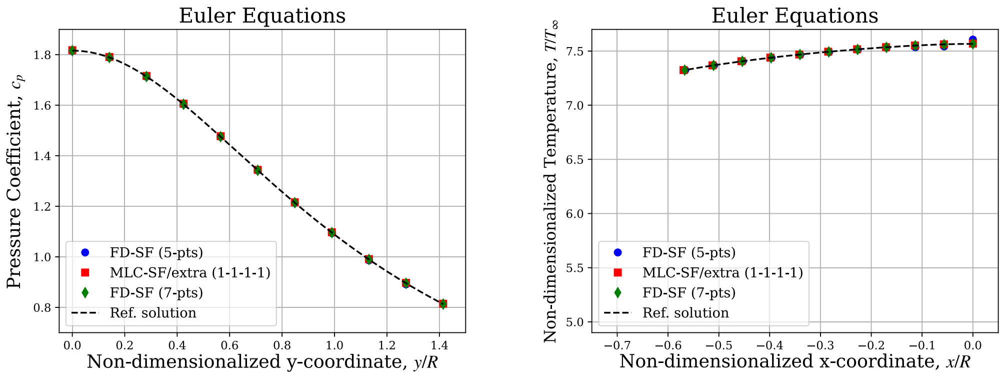


Fig. 16. Two-dimensional supersonic flow over parabolic cylinder (Euler equations): Pressure coefficient on the wall (left) and non-dimensionalized temperature at the centerline (right) with $N = 10$ for different methods.

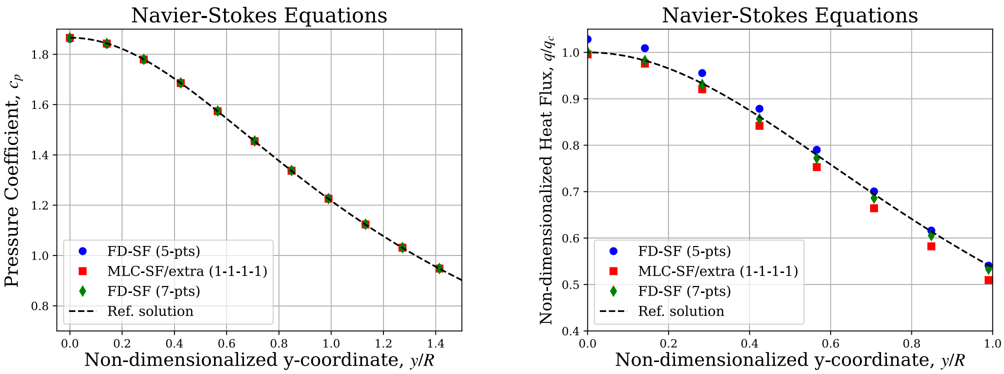


Fig. 17. Two-dimensional supersonic flow over parabolic cylinder (Navier-Stokes equations): Pressure coefficient on the wall (left) and non-dimensionalized heat flux on the wall (right) with $N = 10$ for different methods.

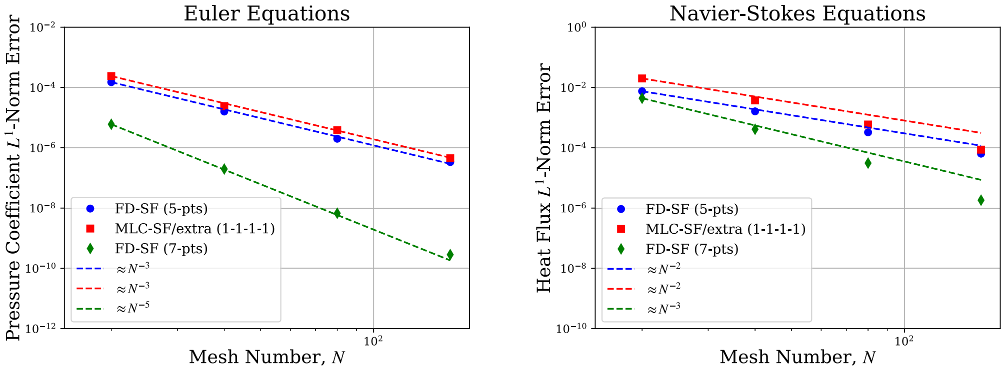


Fig. 18. Two-dimensional supersonic flow over parabolic cylinder: L^1-norm error for the pressure coefficient on the wall versus mesh number N for the Euler equations (left) and L^1-norm error for the non-dimensionalized heat flux on the wall versus mesh number N for the Navier-Stokes equations (right).

method [72] for time integration with CFL number ranging from 0.05 to 0.6 for different mesh numbers. Simulations are considered converged when the non-dimensional velocity change over steps is at least less than 10^{-4}.

Fig. 15 shows the side-by-side comparison of the third-order FD-SF and MLC-SF in a non-dimensional temperature contour for the two equations with mesh number $(N_\xi, N_\eta) = (21, 21)$. The FD-SF method is based on Zhong's work [4], which has been applied to various studies [54,77,78] and can be treated as the reference method to the problem. For the current implementation, the MLC-SF has similar results as the FD-SF, which shows the consistency of the solution between the two methods.

The reference solutions used for the following error evaluation, for both Euler and Navier-Stokes equations, are computed using the fifth-order FD-SF method on a fine mesh with $(N_\xi, N_\eta) = (321, 321)$. Additionally, the reference heat flux at the center point in the Navier-Stokes simulation is $q_c = -358.780\ kW/m^2$, as obtained from the reference solution. In Fig. 16 and Fig. 17, both Euler and

Navier-Stokes simulations have similar accuracy for both MLC and FD when a coarse mesh ($N = 10$) is used. However, as shown in Fig. 18, the L^1-norm error, whether in terms of the pressure coefficient for the Euler equations or the heat flux for the Navier-Stokes equations, is larger for MLC-SF compared to FD-SF on the same mesh. This is due to the current MLC-SF wall boundaries and subsonic outlet implementations not being physically consistent with both value and derivative layers. Most of the implementations rely on extrapolation or one-sided finite difference, which introduces numerical instability or uncertainty to the MLC-SF simulations. Also, as shown in the two-dimensional shock interaction case, a non-physically consistent boundary, like MLC-SF/FD, will deteriorate MLC-SF error convergence. As a result, deriving physically consistent boundary conditions for both wall boundary and subsonic outlet, such as MLC-proposed in the previous case, is necessary for MLC-SF to achieve the desired order of accuracy globally.

6. Conclusions

This paper establishes the multi-layer compact schemes with shock-fitting method (MLC-SF) applied to one-dimensional and two-dimensional flow for supersonic flow over a blunt object simulation. To fully leverage the high spatial accuracy of MLC, the shock-fitting method is introduced to suppress the Gibbs phenomenon caused by discontinuous shock wave. To integrate MLC with the dynamic grid shock-fitting method, the arbitrary Lagrangian-Eulerian (ALE) formulation is derived for the MLC derivative layers and uses the derivatives based on the computational domain U_ξ and U_η as the MLC derivative layers. This enables the application of MLC-SF on non-Cartesian grids and for problems with moving boundaries. In addition, a physically consistent inflow boundary condition is proposed for the shock-fitting method (MLC-SF/PC) to maintain high-order spatial accuracy in MLC-SF. This proposed boundary condition for the shock-normal derivative layer utilizes the C^+ characteristics propagated toward the shockfront, which is consistent with the inflow condition for the value layer. This physically consistent approach eliminates the need for polynomial extrapolation or one-side finite difference, allowing MLC to achieve global seventh-order spatial accuracy. Finally, the proposed MLC framework was tested and compared with other high-order methods in both one-dimensional and two-dimensional problems to evaluate the method's spatial accuracy.

For the one-dimensional linear advection problems, the upwind finite difference (FD) method and MLC have better spatial convergence rates than the tested WENO methods. This is because the upwind methods do not require a smoothness indicator to recover the optimal spatial accuracy. For the one-dimensional Shu-Osher-like problem, the methods incorporating the shock-fitting approach achieve significantly higher spatial convergence rates than pure shock-capturing methods. The shock-capturing methods exhibit only about first-order convergence behind the shock wave, even when employing fifth-order WENO schemes. The discontinuous solution across the shock wave forces shock-capturing methods to rely on lower-order formulations, preventing them from achieving their designed optimal spatial accuracy. In contrast, the upwind methods with shock-fitting can achieve sixth-order accuracy for FD-SF and up to eighth-order accuracy for MLC-SF in the smooth region. This is due to the shock-fitting method only simulating the flow behind the shock wave which prevents the numerical schemes from solving the flow field across the discontinuous interface. Among the numerical schemes implemented with the shock-fitting method, MLC is more computationally efficient than the upwind finite difference for obtaining the same order of errors. The seventh-order MLC can have a 1.32 times speedup over the seventh-order upwind finite difference method for the same global L^1-norm error and 3.34 times speedup in the smooth region.

For the two-dimensional system, both MLC and DMLC were implemented on the linear advection with a moving mesh problem. The cross derivative approximation in MLC makes the method less accurate in seventh-order form and potentially introduces numerical instability to the simulation. On the other hand, DMLC resolves the stability issue by introducing the cross derivative term as new degrees of freedom and has a more consistent and stable error convergence rate. The resulting speedup for DMLC over the finite difference method is 2.99. From the studied cases and compared methods, MLC is the suitable method for the one-dimensional problem and DMLC is the most efficient and stable method for the two-dimensional linear advection problem. In the two-dimensional shock-interaction problem, the proposed inflow condition for MLC-SF is physically consistent between both value and derivative layers. This property stabilizes the MLC boundary conditions and enables the schemes to achieve global seventh-order accuracy. However, the current boundary condition implementations for the supersonic flow past parabolic cylinder for both Euler and Navier-Stokes equations fail to achieve the desired seventh-order accuracy. This limitation arises from the wall boundary and subsonic outlet conditions, which rely on polynomial extrapolation or one-sided differences that are insufficient to stabilize simulations while maintaining high spatial accuracy. These testing cases show the importance of the boundary condition implementations in fully utilizing MLC-SF's potential.

Although MLC-SF demonstrates promising spatial accuracy and computational efficiency in the tested problems, the current framework faces limitations when applied to three-dimensional supersonic flow over a blunt object while maintaining seventh-order global accuracy. One of the key challenges of MLC-SF is designing boundary conditions that match the spatial accuracy of interior stencils. As shown by Zhong [4], the boundary conditions using conventional extrapolation or one-sided finite difference discretization can only achieve fifth-order spatial order accuracy without causing simulations to diverge. However, MLC-SF requires at least sixth-order boundary conditions to achieve the seventh-order global accuracy. Boundary condition implementation like Navier-Stokes characteristic boundary conditions (NSCBC) [74] can be an ideal approach, as they do not rely on extrapolation or one-sided finite difference schemes. However, implementing NSCBC in MLC is significantly more complex than in conventional finite difference methods. The process requires addressing additional derivative layers in MLC and defining appropriate physical interaction processes between MLC-SF solutions and closed boundaries. Consequently, developing high-order boundary conditions for MLC-SF will be a primary focus to improve the applicability of the method.

Another challenge for the proposed MLC-SF framework is the method's capability on complex shock interactions such as flow fields involving triple points of shock waves or shocklets. While the shock-fitting method offers high-order accuracy, it relies on relatively simple shock structures where the shock location can be predefined before simulations. For fluid flows with complex or multiple shock interactions, shock-capturing methods remain the more suitable approach. Ideally, a hybrid approach combining shock-fitting and shock-capturing methods could be developed to suppress the Gibbs phenomenon in the vicinity of the primary shock while simultaneously simulating downstream shocklet interactions.

Finally, MLC-SF requires more implementation time due to its multi-layer structure. The auxiliary equations derived from the spatial differentiation of the governing equations demand additional development time and debugging efforts, particularly due to the extra terms introduced by the chain rule. Although MLC-SF has demonstrated computational efficiency in simple one- and two-dimensional problems, the additional auxiliary equations in the three-dimensional formulation may hinder computational speed. However, the higher-order spatial convergence rate of MLC can offset the extra computational cost when more mesh points are used. MLC could become more efficient if a more effective method for computing the derivative layers is developed. Overall, the work presented in this paper aims to demonstrate MLC-SF's potential in more realistic simulation problems. Studying these MLC-SF adaptations is not just about applying MLC on more complex problems but also provide a different perspective to the current existing simulation frameworks.

CRediT authorship contribution statement

Yung-Tien Lin: Writing – original draft, Visualization, Validation, Software, Resources, Methodology, Investigation, Funding acquisition, Formal analysis, Data curation, Conceptualization; **Xiaolin Zhong:** Writing – review & editing, Supervision, Resources, Methodology, Conceptualization.

Data availability

Data will be made available on request.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Yung-Tien Lin reports financial support was provided by Executive Yuan Republic of China Ministry of Education. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

Y.-T. Lin thanks the 2023 Government Scholarship for Study Abroad (GSSA) from the Taiwan Ministry of Education for financially supporting the completion of this study.

Appendix A.

A.1. Two-dimensional ALE formulation for MLC-SF

To implement a dynamic grid for the shock-fitting method, the arbitrary Lagrangian-Eulerian (ALE) formulation is introduced and integrated with the MLC framework. The auxiliary equations for MLC with ALE are derived directly using the chain rule. The general form of the governing equations in the ALE formulation is expressed as:

$$\begin{aligned}
\frac{\partial U}{\partial t} &= -(1/g)A \\
\frac{\partial U_\xi}{\partial t} &= -\left[(1/g)_\xi A + (1/g)A_\xi\right] \\
\frac{\partial U_\eta}{\partial t} &= -\left[(1/g)_\eta A + (1/g)A_\eta\right]
\end{aligned} \tag{A.1}$$

where g is the determinant of the coordinate transformation Jacobian.
The total ALE fluxes, A, which include the fluxes accounting for grid motion, $g_t U$, and coordinate transformation F_i':

$$\begin{aligned}
A &= g_t U + F'_{1,\xi} + F'_{2,\eta} \\
A_\xi &= g_{t\xi} U + g_t U_\xi + F'_{1,\xi\xi} + F'_{2,\xi\eta} \\
A_\eta &= g_{t\eta} U + g_t U_\eta + F'_{1,\xi\eta} + F'_{2,\eta\eta}
\end{aligned} \tag{A.2}$$

The zeroth- and first-order flux derivatives are computed analytically. The transformed fluxes in the ξ direction, F_1':

$$\begin{aligned}
F_1' &= y_\eta F_1 - x_\eta F_2 + (g\xi_t)U \\
F'_{1,\xi} &= \left(y_{\xi\eta} F_1 + y_\eta F_{1,\xi}\right) - \left(x_{\xi\eta} F_2 + x_\eta F_{2,\xi}\right) + (g\xi_t)_\xi U + (g\xi_t)U_\xi
\end{aligned}$$

$$F'_{1,\eta} = \left(y_{\eta\eta}F_1 + y_\eta F_{1,\eta}\right) - \left(x_{\eta\eta}F_2 + x_\eta F_{2,\eta}\right) + (g\xi_t)\eta U + (g\xi_t)U_\eta \tag{A.3}$$

The transformed fluxes in the η direction, F'_2:

$$\begin{aligned}
F'_2 &= -y_\xi F_1 + x_\xi F_2 + (g\eta_t)U \\
F'_{2,\xi} &= -\left(y_{\xi\xi}F_1 + y_\xi F_{1,\xi}\right) + \left(x_{\xi\xi}F_2 + x_\xi F_{2,\xi}\right) + (g\eta_t)_\xi U + (g\eta_t)U_\xi \\
F_{2,\eta} &= -\left(y_{\xi\eta}F_1 + y_\xi F_{1,\eta}\right) + \left(x_{\xi\eta}F_2 + x_\xi F_{2,\eta}\right) + (g\eta_t)_\eta U + (g\eta_t)U_\eta
\end{aligned} \tag{A.4}$$

Note that the fluxes in the physical coordinate, F_i, are generally consisted of the inviscid and viscous terms:

$$\begin{aligned}
F_1 &= F_1^{inv} - F_2^{vis} \\
F_2 &= F_2^{inv} - F_2^{vis}
\end{aligned} \tag{A.5}$$

For the inviscid flux in x direction, F_1^{inv}:

$$F_1^{inv} = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho u v \\ (e+p)u \end{bmatrix} \tag{A.6}$$

and the inviscid flux in y direction, F_2^{inv}:

$$F_2^{inv} = \begin{bmatrix} \rho v \\ \rho u v \\ \rho v^2 + p \\ (e+p)v \end{bmatrix} \tag{A.7}$$

The first-order derivatives of the fluxes can be simply computed through direct differentiation with respect to any given direction, δ:

$$\frac{\partial F_1^{inv}}{\partial \delta} = \frac{\partial}{\partial \delta}\begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho u v \\ (e+p)u \end{bmatrix} = \begin{bmatrix} (\rho u)_\delta \\ (\rho u)_\delta u + (\rho u)u_\delta + p_\delta \\ (\rho u)_\delta v + (\rho u)v_\delta \\ (e_\delta + p_\delta)u + (e+p)u_\delta \end{bmatrix} \tag{A.8}$$

and

$$\frac{\partial F_2^{inv}}{\partial \delta} = \frac{\partial}{\partial \delta}\begin{bmatrix} \rho v \\ \rho u v \\ \rho v^2 + p \\ (e+p)v \end{bmatrix} = \begin{bmatrix} (\rho v)_\delta \\ (\rho v)_\delta u + (\rho v)u_\delta \\ (\rho v)_\delta v + (\rho v)v_\delta + p_\delta \\ (e_\delta + p_\delta)v + (e+p)v_\delta \end{bmatrix} \tag{A.9}$$

The ideal gas relation is used to determine the relation between the conservative states and the primitive variables:

$$\begin{aligned}
e &= \frac{p}{\gamma - 1} + \frac{1}{2}\rho(u^2 + v^2) \\
p &= (\gamma - 1)\cdot[e - \frac{1}{2}\rho(u^2 + v^2)] \\
T &= \frac{\gamma - 1}{\rho R_{gas}}\left[e - \frac{1}{2}\frac{(\rho u)^2 + (\rho v)^2}{\rho}\right]
\end{aligned} \tag{A.10}$$

Similarly, the spatial derivatives of primitive variables can be obtained by the chain rule with respect to any given direction, δ:

$$\begin{aligned}
u_\delta &= \frac{1}{U_1}U_{2,\delta} - \frac{U_2}{{U_1}^2}U_{1,\delta} \\
v_\delta &= \frac{1}{U_1}U_{3,\delta} - \frac{U_3}{{U_1}^2}U_{1,\delta} \\
T_\delta &= -\frac{(\gamma - 1)U_{1,\delta}}{{U_1}^2 R_{gas}}\left[U_4 - \frac{1}{2}\frac{(U_2)^2 + (U_3)^2}{U_1}\right] + \frac{\gamma - 1}{U_1 R_{gas}}\left[U_{4,\delta} - \frac{U_2 U_{2,\delta} + U_3 U_{3,\delta}}{U_1} + \frac{(U_2)^2 + (U_3)^2}{2(U_1)^2}U_{1,\delta}\right] \\
p_\delta &= U_{1,\delta}R_{gas}T + U_1 R_{gas}T_\delta
\end{aligned} \tag{A.11}$$

Recall the conservative states are $\vec{U} = [\rho, \rho u, \rho v, e]^T$.
For the viscous flux in x direction, F_1^{vis}:

$$F_1^{vis} = \begin{bmatrix} 0 \\ \tau_{xx} \\ \tau_{xy} \\ kT_x + u\tau_{xx} + v\tau_{xy} \end{bmatrix} \tag{A.12}$$

and viscous flux in y direction, F_2^{vis}:

$$F_2^{vis} = \begin{bmatrix} 0 \\ \tau_{xy} \\ \tau_{yy} \\ kT_y + u\tau_{xy} + v\tau_{yy} \end{bmatrix} \tag{A.13}$$

The first-order spatial derivatives of the viscous flux in x direction with respect to any given direction, δ:

$$F_{1,\delta}^{vis} = \begin{bmatrix} 0 \\ (\tau_{xx})_\delta \\ (\tau_{xy})_\delta \\ (kT_x)_\delta + u_\delta \tau_{xx} + u(\tau_{xx})_\delta + v_\delta \tau_{xy} + v(\tau_{xy})_\delta \end{bmatrix} \tag{A.14}$$

and spatial derivatives of the viscous flux in y direction:

$$F_{2,\delta}^{vis} = \begin{bmatrix} 0 \\ (\tau_{xy})_\delta \\ (\tau_{yy})_\delta \\ (kT_y)_\delta + u_\delta \tau_{xy} + u(\tau_{xy})_\delta + v_\delta \tau_{yy} + v(\tau_{yy})_\delta \end{bmatrix} \tag{A.15}$$

For the viscous stress terms, the fluid is assumed to be Newtonian, with viscosity modeled using Sutherland's law. Additionally, the Stokes hypothesis is applied ($\lambda = -2/3\mu$):

$$\begin{aligned} \tau_{xx} &= 2\mu u_x + \tau_D \\ \tau_{xy} &= \mu\left(u_y + v_x\right) \\ \tau_{yy} &= 2\mu v_y + \tau_D \\ \tau_D &= \lambda \nabla \cdot \vec{u} \end{aligned} \tag{A.16}$$

The first-order spatial derivatives of the viscous stress terms with respect to any given direction, δ:

$$\begin{aligned} (\tau_{xx})_\delta &= 2\left[\mu_\delta u_x + \mu u_{x,\delta}\right] + \tau_{D,\delta} \\ (\tau_{xy})_\delta &= \mu_\delta\left[u_y + v_x\right] + \mu\left[u_{y,\delta} + v_{x,\delta}\right] \\ (\tau_{yy})_\delta &= 2\left[\mu_\delta v_y + \mu v_{y,\xi}\right] + \tau_{D,\xi} \end{aligned} \tag{A.17}$$

For the thermal conduction terms:

$$\begin{aligned} q_x &= kT_x \\ q_y &= kT_y \end{aligned} \tag{A.18}$$

and their first-order spatial differentiation with respect to δ direction:

$$\begin{aligned} q_{x,\delta} &= k_\delta T_x + kT_{x,\delta} \\ q_{y,\delta} &= k_\delta T_y + kT_{y,\delta} \end{aligned} \tag{A.19}$$

A.2. Two-dimensional NSCBC for MLC-SF

The Navier-Stokes characteristic boundary conditions (NSCBC), proposed by Poinsot and Lele [74], can be applied to the MLC framework to prevent the use of lower-order extrapolation or one-side finite difference for determining the derivative on the computational boundaries. NSCBC utilizes the concept of characteristic propagation to ensure the solution states inside the computational domain can properly interact with the numerical boundaries. The method begins by using transformed governing equations. For example, the continuity equation and its corresponding derivative layers:

$$\begin{aligned} &\frac{\partial \rho}{\partial t} + d_1 + m_{2,y} = 0 \\ &\frac{\partial \rho_\xi}{\partial t} + d_{1,\xi} + m_{2,y,\xi} = 0 \\ &\frac{\partial \rho_\eta}{\partial t} + d_{1,\eta} + m_{2,y,\eta} = 0 \end{aligned} \tag{A.20}$$

where the surface normal of the numerical boundary is assumed to be in x-direction.
The x-momentum equation, $m_1 = \rho u$:

$$\begin{aligned} &\frac{\partial m_1}{\partial t} + ud_1 + \rho d_3 + (m_1 v)_y = 0 \\ &\frac{\partial m_{1,\xi}}{\partial t} + u_\xi d_1 + ud_{1,\xi} + \rho_\xi d_3 + \rho d_{3,\xi} + (m_1 v)_{y,\xi} = 0 \end{aligned}$$

$$\frac{\partial m_{1,\eta}}{\partial t} + u_\eta d_1 + u d_{1,\eta} + \rho_\eta d_3 + \rho d_{3,\eta} + (m_1 v)_{y,\eta} = 0 \tag{A.21}$$

The y-momentum, $m_2 = \rho v$:

$$\begin{aligned}
&\frac{\partial m_2}{\partial t} + v d_1 + \rho d_4 + (m_2 v + p)_y = 0\\
&\frac{\partial m_{2,\xi}}{\partial t} + v_\xi d_1 + v d_{1,\xi} + \rho_\xi d_4 + \rho d_{4,\xi} + (m_2 v + p)_{y,\xi} = 0\\
&\frac{\partial m_{2,\eta}}{\partial t} + v_\eta d_1 + v d_{1,\eta} + \rho_\eta d_4 + \rho d_{4,\eta} + (m_2 v + p)_{y,\eta} = 0
\end{aligned} \tag{A.22}$$

The energy equation:

$$\begin{aligned}
&\frac{\partial e}{\partial t} + \frac{1}{2}(u_k u_k) d_1 + \frac{d_2}{\gamma - 1} + m_1 d_3 + m_2 d_4 + [(e+p)v]_y = 0\\
&\frac{\partial e_\xi}{\partial t} + (u u_\xi + v v_\xi) d_1 + \frac{1}{2}(u_k u_k) d_{1,\xi} + \frac{d_{2,\xi}}{\gamma - 1} + m_{1,\xi} d_3 + m_1 d_{3,\xi}\\
&\quad + m_{2,\xi} d_4 + m_2 d_{4,\xi} + [(e+p)v]_{y,\xi} = 0\\
&\frac{\partial e_\eta}{\partial t} + (u u_\eta + v v_\eta) d_1 + \frac{1}{2}(u_k u_k) d_{1,\eta} + \frac{d_{2,\eta}}{\gamma - 1} + m_{1,\eta} d_3 + m_1 d_{3,\eta}\\
&\quad + m_{2,\eta} d_4 + m_2 d_{4,\eta} + [(e+p)v]_{y,\eta} = 0
\end{aligned} \tag{A.23}$$

The transformed variable d_1:

$$\begin{aligned}
d_1 &= \frac{1}{c^2}\left[\mathcal{L}_2 + d_2\right]\\
d_{1,\xi} &= -2\frac{c_\xi}{c^3}\left[\mathcal{L}_2 + d_2\right] + \frac{1}{c^2}\left[\mathcal{L}_{2,\xi} + d_{2,\xi}\right]\\
d_{1,\eta} &= -2\frac{c_\eta}{c^3}\left[\mathcal{L}_2 + d_2\right] + \frac{1}{c^2}\left[\mathcal{L}_{2,\eta} + d_{2,\eta}\right]
\end{aligned} \tag{A.24}$$

Transformed variable d_2:

$$\begin{aligned}
d_2 &= 0.5(\mathcal{L}_4 + \mathcal{L}_1)\\
d_{2,\xi} &= 0.5(\mathcal{L}_{4,\xi} + \mathcal{L}_{1,\xi})\\
d_{2,\eta} &= 0.5(\mathcal{L}_{4,\eta} + \mathcal{L}_{1,\eta})
\end{aligned} \tag{A.25}$$

Transformed variable d_3:

$$\begin{aligned}
d_3 &= \frac{\mathcal{L}_4 - \mathcal{L}_1}{2\rho c}\\
d_{3,\xi} &= -\frac{\mathcal{L}_4 - \mathcal{L}_1}{2(\rho c)^2}(\rho_\xi c + \rho c_\xi) + \frac{\mathcal{L}_{4,\xi} - \mathcal{L}_{1,\xi}}{2\rho c}\\
d_{3,\eta} &= -\frac{\mathcal{L}_{4,\eta} - \mathcal{L}_{1,\eta}}{2(\rho c)^2}(\rho_\eta c + \rho c_\eta) + \frac{d_{2,\eta}}{2\rho c}
\end{aligned} \tag{A.26}$$

Transformed variable d_4:

$$\begin{aligned}
d_4 &= \mathcal{L}_3\\
d_{4,\xi} &= \mathcal{L}_{3,\xi}\\
d_{4,\xi} &= \mathcal{L}_{3,\eta}
\end{aligned} \tag{A.27}$$

Characteristics $\mathcal{L}_1$:

$$\begin{aligned}
\mathcal{L}_1 &= \lambda_1(p_x - \rho c u_x)\\
\mathcal{L}_{1,\xi} &= \lambda_{1,\xi}(p_x - \rho c u_x) + \lambda_1[p_{x,\xi} - (\rho_\xi c + \rho c_\xi)u_x + \rho c u_{x,\xi}]\\
\mathcal{L}_{1,\eta} &= \lambda_{1,\eta}(p_x - \rho c u_x) + \lambda_1[p_{x,\eta} - (\rho_\eta c + \rho c_\eta)u_x + \rho c u_{x,\eta}]
\end{aligned} \tag{A.28}$$

Characteristics $\mathcal{L}_2$:

$$\begin{aligned}
\mathcal{L}_2 &= \lambda_2(c^2 \rho_x - p_x)\\
\mathcal{L}_{2,\xi} &= \lambda_{2,\xi}(c^2 \rho_x - p_x) + \lambda_2(2 c c_\xi \rho_x + c^2 \rho_{x,\xi} - p_{x,\xi})\\
\mathcal{L}_{2,\eta} &= \lambda_{2,\eta}(c^2 \rho_x - p_x) + \lambda_2(2 c c_\eta \rho_x + c^2 \rho_{x,\eta} - p_{x,\eta})
\end{aligned} \tag{A.29}$$

Characteristics $\mathcal{L}_3$:

$$\mathcal{L}_3 = \lambda_3 v_x$$

$$\begin{aligned}\mathcal{L}_{3,\xi} &= \lambda_{3,\xi} v_x + \lambda_3 v_{x,\xi}\\ \mathcal{L}_{3,\eta} &= \lambda_{3,\eta} v_x + \lambda_3 v_{x,\eta}\end{aligned} \tag{A.30}$$

Characteristics $\mathcal{L}_4$:

$$\begin{aligned}\mathcal{L}_4 &= \lambda_4(p_x + \rho c u_x)\\ \mathcal{L}_{4,\xi} &= \lambda_{4,\xi}(p_x + \rho c u_x) + \lambda_4[p_{x,\xi} + (\rho_\xi c + \rho c_\xi)u_x + \rho c u_{x,\xi}]\\ \mathcal{L}_{4,\eta} &= \lambda_{4,\eta}(p_x + \rho c u_x) + \lambda_4[p_{x,\eta} + (\rho_\eta c + \rho c_\eta)u_x + \rho c u_{x,\eta}]\end{aligned} \tag{A.31}$$

The characteristic speed λ_1:

$$\begin{aligned}\lambda_1 &= u - c\\ \lambda_{1,\xi} &= u_\xi - c_\xi\\ \lambda_{1,\eta} &= u_\eta - c_\eta\end{aligned} \tag{A.32}$$

Characteristic speed $\lambda_2 = \lambda_3$:

$$\begin{aligned}\lambda_2 &= \lambda_3 = u\\ \lambda_{2,\xi} &= u_\xi\\ \lambda_{2,\eta} &= u_\eta\end{aligned} \tag{A.33}$$

Characteristic speed λ_4:

$$\begin{aligned}\lambda_4 &= u + c\\ \lambda_{4,\xi} &= u_\xi + c_\xi\\ \lambda_{4,\eta} &= u_\eta + c_\eta\end{aligned} \tag{A.34}$$

Depends on the boundary type, different condition will be applied and the rest of unspecified variables are calculated through the transformed governing equations.

References

[1] H.M. Sayler, Hypersonic weapons: Background and issues for Congress, Congressional Research Service USA, 2019.
[2] M. Malik, T. Zang, D. Bushnell, Boundary layer transition in hypersonic flows, in: 2nd International Aerospace Planes Conference, 1990, p. 5232.
[3] X. Zhong, X. Wang, Direct numerical simulation on the receptivity, instability, and transition of hypersonic boundary layers, Annu. Rev. Fluid Mech. 44 (1) (2012) 527–561.
[4] X. Zhong, High-order finite-difference schemes for numerical simulation of hypersonic boundary-layer transition, J. Computat. Phys. 144 (2) (1998) 662–709.
[5] A. Fedorov, Transition and stability of high-speed boundary layers, Annu. Rev. Fluid Mech. 43 (1) (2011) 79–95.
[6] C.J. Roy, F.G. Blottner, Review and assessment of turbulence models for hypersonic flows, Progr. Aerospace Sci. 42 (7-8) (2006) 469–530.
[7] S.K. Lele, Compact finite difference schemes with spectral-like resolution, J. Computat. Phys. 103 (1) (1992) 16–42.
[8] L. Duan, I. Beekman, M.P. Martin, Direct numerical simulation of hypersonic turbulent boundary layers. Part 2. Effect of wall temperature, J. Fluid Mech. 655 (2010) 419–445.
[9] M.P. Martín, E.M. Taylor, M. Wu, V.G. Weirs, A bandwidth-optimized WENO scheme for the effective direct numerical simulation of compressible turbulence, J. Computat. Phys. 220 (1) (2006) 270–289.
[10] A. Laible, H. Fasel, Numerical investigation of hypersonic transition for a flared and a straight cone at Mach 6, in: 41st AIAA Fluid Dynamics Conference and Exhibit, 2011, p. 3565.
[11] G.V. Candler, H.B. Johnson, I. Nompelis, V.M. Gidzak, P.K. Subbareddy, M. Barnhardt, Development of the US3D code for advanced compressible and reacting flow simulations, in: 53rd AIAA Aerospace Sciences Meeting, 2015, p. 1893.
[12] P.K. Subbareddy, G.V. Candler, A fully discrete, kinetic energy consistent finite-volume scheme for compressible flows, J. Computat. Phys. 228 (5) (2009) 1347–1364.
[13] C. Nguyen, S. Terrana, J. Peraire, Implicit Large eddy simulation of hypersonic boundary-layer transition for a flared cone, in: AIAA SCITECH 2023 Forum, 2023, p. 0659.
[14] N. Cuong Nguyen, S. Terrana, J. Peraire, Large-eddy simulation of transonic buffet using matrix-free discontinuous Galerkin method, AIAA J. 60 (5) (2022) 3060–3077.
[15] P. Fernandez, N.C. Nguyen, J. Peraire, The hybridized discontinuous Galerkin method for implicit large-eddy simulation of transitional turbulent flows, J. Computat. Phys. 336 (2017) 308–329.
[16] Z.J. Wang, K. Fidkowski, R. Abgrall, F. Bassi, D. Caraeni, A. Cary, H. Deconinck, R. Hartmann, K. Hillewaert, H.T. Huynh, et al., High-order CFD methods: current status and perspective, Int. J. Numer. Methods Fluids 72 (8) (2013) 811–845.
[17] D. Gottlieb, S.A. Orszag, Numerical analysis of spectral methods: theory and applications, SIAM, 1977.
[18] C.K.W. Tam, J.C. Webb, Dispersion-relation-preserving finite difference schemes for computational acoustics, J. Computat. Phys. 107 (2) (1993) 262–281.
[19] M.R. Visbal, D.V. Gaitonde, On the use of higher-order finite-difference schemes on curvilinear and deforming meshes, J. Computat. Phys. 181 (1) (2002) 155–185.
[20] C. Ollivier-Gooch, M. Van Altena, A high-order-accurate unstructured mesh finite-volume scheme for the advection–diffusion equation, J. Computat. Phys. 181 (2) (2002) 729–752.
[21] L.N. Trefethen, Spectral methods in MATLAB, SIAM, 2000.
[22] R.D. Moser, J. Kim, N.N. Mansour, Direct numerical simulation of turbulent channel flow up to Reτ= 590, Phys. Fluids 11 (4) (1999) 943–945.
[23] W.S. Don, Numerical study of pseudospectral methods in shock wave applications, J. Computat. Phys. 110 (1) (1994) 103–111.
[24] Z.J. Wang, High-order methods for the Euler and Navier–Stokes equations on unstructured grids, Progr. Aerospace Sci. 43 (1-3) (2007) 1–41.
[25] A.E. Honein, P. Moin, Higher entropy conservation and numerical stability of compressible turbulence simulations, J. Computat. Phys. 201 (2) (2004) 531–545.
[26] B. Cockburn, C.-W. Shu, Runge–Kutta discontinuous Galerkin methods for convection-dominated problems, J. Sci. Comput. 16 (2001) 173–261.
[27] B. Cockburn, J. Gopalakrishnan, R. Lazarov, Unified hybridization of discontinuous Galerkin, mixed, and continuous Galerkin methods for second order elliptic problems, SIAM J. Numer. Anal. 47 (2) (2009) 1319–1365.

[28] N.C. Nguyen, J. Peraire, B. Cockburn, A class of embedded discontinuous Galerkin methods for computational fluid dynamics, J. Computat. Phys. 302 (2015) 674–692.
[29] D.S. Hoskin, R.L. Van Heyningen, N.C. Nguyen, J. Vila-Pérez, W.L. Harris, J. Peraire, Discontinuous Galerkin methods for hypersonic flows, Progr. Aerospace Sci. 146 (2024) 100999.
[30] Z.J. Wang, Spectral (finite) volume method for conservation laws on unstructured grids. basic formulation: Basic formulation, J. Computat. Phys. 178 (1) (2002) 210–251.
[31] Y. Liu, M. Vinokur, Z.J. Wang, Spectral difference method for unstructured grids I: Basic formulation, J. Computat. Phys. 216 (2) (2006) 780–801.
[32] Z. Bai, X. Zhong, New very high-order upwind multilayer compact schemes with spectral-like resolution for flow simulations, AIAA, 2017, pp. 1–42.
[33] Z. Bai, X. Zhong, New very high-order upwind multi-layer compact (MLC) schemes with spectral-like resolution for flow simulations, J. Computat. Phys. 378 (2019) 63–109.
[34] Z. Bai, X. Zhong, A new very high-order upwind directional multi-layer compact (DMLC) scheme for multi-dimensional flows, Comput. Fluids 197 (2020) 104356.
[35] M.D. Salas, A shock-fitting primer, CRC Press, 2009.
[36] M. Onofri, R. Paciorri, Shock fitting: classical techniques, recent developments, and memoirs of Gino Moretti, Springer, 2017.
[37] S. Pirozzoli, Numerical methods for high-speed flows, Annu. Rev. Fluid Mech. 43 (1) (2011) 163–194.
[38] J. VonNeumann, R.D. Richtmyer, A method for the numerical calculation of hydrodynamic shocks, J. Appl. Phys. 21 (3) (1950) 232–237.
[39] A. Jameson, W. Schmidt, E. Turkel, Numerical solution of the Euler equations by finite volume methods using Runge Kutta time stepping schemes, in: 14th fluid and plasma dynamics conference, 1981, p. 1259.
[40] P.-O. Persson, J. Peraire, Sub-cell shock capturing for discontinuous Galerkin methods, in: 44th AIAA aerospace sciences meeting and exhibit, 2006, p. 112.
[41] B. Van Leer, Towards the ultimate conservative difference scheme. V. A second-order sequel to Godunov's method, J. Computat. Phys. 32 (1) (1979) 101–136.
[42] P.K. Sweby, High resolution schemes using flux limiters for hyperbolic conservation laws, SIAM J. Numer. Anal. 21 (5) (1984) 995–1011.
[43] H.C. Yee, A class of high-resolution explicit and implicit shock-capturing methods, High Resolution Upwind and TVD Methods for the Compressible Flow Equations (1994).
[44] G.-S. Jiang, C.-W. Shu, Efficient implementation of weighted ENO schemes, J. Computat. Phys. 126 (1) (1996) 202–228.
[45] X.-D. Liu, S. Osher, T. Chan, Weighted essentially non-oscillatory schemes, J. Computat. Phys. 115 (1) (1994) 200–212.
[46] A.K. Henrick, T.D. Aslam, J.M. Powers, Mapped weighted essentially non-oscillatory schemes: achieving optimal order near critical points, J. Computat. Phys. 207 (2) (2005) 542–567.
[47] R. Borges, M. Carmona, B. Costa, W.S. Don, An improved weighted essentially non-oscillatory scheme for hyperbolic conservation laws, J. Computat. Phys. 227 (6) (2008) 3191–3211.
[48] M. Castro, B. Costa, W.S. Don, High order weighted essentially non-oscillatory WENO-Z schemes for hyperbolic conservation laws, J. Computat. Phys. 230 (5) (2011) 1766–1792.
[49] L. Fu, Review of the high-order TENO schemes for compressible gas dynamics and turbulence, Arch. Computat. Methods Eng. 30 (4) (2023) 2493–2526.
[50] J. Qiu, C.-W. Shu, Hermite WENO schemes and their application as limiters for Runge–Kutta discontinuous Galerkin method: one-dimensional case, J. Computat. Phys. 193 (1) (2004) 115–135.
[51] J. Li, C.-W. Shu, J. Qiu, Multi-resolution HWENO schemes for hyperbolic conservation laws, J. Computat. Phys. 446 (2021) 110653.
[52] G. Moretti, M. Abbett, A time-dependent computational method for blunt body flows, Aiaa J. 4 (12) (1966) 2136–2141.
[53] Z. Zou, X. Zhong, A new very high-order finite-difference method for linear stability analysis and bi-orthogonal decomposition of hypersonic boundary layer flow, J. Computat. Phys. 512 (2024) 113135.
[54] S. He, X. Zhong, The effects of nose bluntness on broadband disturbance receptivity in hypersonic flow, Phys. Fluids 34 (5) (2022).
[55] E. Johnsen, J. Larsson, A.V. Bhagatwala, W.H. Cabot, P. Moin, B.J. Olson, P.S. Rawat, S.K. Shankar, B. Sjögreen, H.C. Yee, et al., Assessment of high-resolution methods for numerical simulations of compressible turbulence with shock waves, J. Computat. Phys. 229 (4) (2010) 1213–1237.
[56] P.S. Rawat, X. Zhong, On high-order shock-fitting and front-tracking schemes for numerical simulation of shock–disturbance interactions, J. Computat. Phys. 229 (19) (2010) 6744–6780.
[57] F. Ducros, V. Ferrand, F. Nicoud, C. Weber, D. Darracq, C. Gacherieu, T. Poinsot, Large-eddy simulation of the shock/turbulence interaction, J. Computat. Phys. 152 (2) (1999) 517–549.
[58] J. Canny, A computational approach to edge detection, IEEE Trans. Pattern Anal. Mach. Intell. (6) (1986) 679–698.
[59] T.R. Fujimoto, T. Kawasaki, K. Kitamura, Canny-Edge-Detection/Rankine-Hugoniot-conditions unified shock sensor for inviscid and viscous flows, J. Computat. Phys. 396 (2019) 264–279.
[60] Y. Liu, Y. Lu, Y. Wang, D. Sun, L. Deng, F. Wang, Y. Lei, A CNN-based shock detection method in flow visualization, Comput. Fluids 184 (2019) 1–9.
[61] A.D. Beck, J. Zeifang, A. Schwarz, D.G. Flad, A neural network based shock detection and localization approach for discontinuous Galerkin methods, J. Computat. Phys. 423 (2020) 109824.
[62] H. Blumhoefer, A. Sharma, Accelerating shock detection using a multigrid approach, AIAA J. (2024) 1–8.
[63] R. Paciorri, A. Bonfiglioli, A shock-fitting technique for 2D unstructured grids, Comput. Fluids 38 (3) (2009) 715–726.
[64] R. Paciorri, A. Bonfiglioli, Accurate detection of shock waves and shock interactions in two-dimensional shock-capturing solutions, J. Computat. Phys. 406 (2020) 109196.
[65] A. Assonitis, R. Paciorri, M. Ciallella, M. Ricchiuto, A. Bonfiglioli, L. Cirrottola, Numerical simulations of shock interactions on 3D structured grids using a shock-fitting approach, in: AIAA SCITECH 2023 Forum, 2023, p. 2135.
[66] N.C. Nguyen, R.L. Van Heyningen, J. Vila-Pérez, J. Peraire, Optimal transport for mesh adaptivity and shock capturing of compressible flows, J. Computat. Phys. 508 (2024) 113005.
[67] A. Corrigan, A.D. Kercher, D.A. Kessler, A moving discontinuous Galerkin finite element method for flows with interfaces, Int. J. Numer. Methods Fluids 89 (9) (2019) 362–406.
[68] M.J. Zahr, P.-O. Persson, An optimization-based approach for high-order accurate discretization of conservation laws with discontinuous solutions, J. Computat. Phys. 365 (2018) 105–134.
[69] K. Peery, S. Imlay, Blunt-body flow simulations, in: 24th joint propulsion conference, 1988, p. 2904.
[70] K. Mahesh, The interaction of a shock wave with a turbulent shear flow, Stanford University, 1996.
[71] L. Duan, Y. Liu, M.M. Choudhari, K.M. Casper, R.M. Wagnild, Direct numerical simulation of hypersonic boundary layer transition in a digital wind tunnel at Mach 8, in: AIAA SciTech Forum, 2021.
[72] S. Gottlieb, C.-W. Shu, Total variation diminishing Runge-Kutta schemes, Math. Computat. 67 (221) (1998) 73–85.
[73] P.-O. Persson, J. Bonet, J. Peraire, Discontinuous Galerkin solution of the Navier–Stokes equations on deformable domains, Comput. Methods Appl. Mech. Eng. 198 (17-20) (2009) 1585–1595.
[74] T.J. Poinsot, S.K. Lele, Boundary conditions for direct simulations of compressible viscous flows, J. Computat. Phys. 101 (1) (1992) 104–129.
[75] C.-W. Shu, S. Osher, Efficient implementation of essentially non-oscillatory shock-capturing schemes, II, in: Upwind and High-Resolution Schemes, Springer, 1989, pp. 328–374.
[76] A. Suresh, Interaction of a shock with a density disturbance via shock fitting, J. Computat. Phys. 206 (1) (2005) 6–15.
[77] C.P. Knisely, X. Zhong, Sound radiation by supersonic unstable modes in hypersonic blunt cone boundary layers. II. Direct numerical simulation, Phys. Fluids 31 (2) (2019).
[78] C. Haley, X. Zhong, Supersonic mode in a low-enthalpy hypersonic flow over a cone and wave packet interference, Phys. Fluids 33 (5) (2021).