Eleventh International Conference on Computational Fluid Dynamics (ICCFD11), Maui, HI, USA, July 11-15, 2022

ICCFD11-2022-0502

Deneb: An Open-source High-performance Flow Solver based on DRM-DG Method

Hojun You[*,**], Juhyun Kim[*], and Chongam Kim[*,**]
Corresponding author: chongam@snu.ac.kr

[*] Department of Aerospace Engineering, Seoul National University, Republic of Korea.
[**] Institute of Advanced Aerospace Technology, Seoul National University, Republic of Korea.

Abstract: High-order methods are being recognized as powerful tools for handling scale-resolving simulations over complex geometry. However, several obstacles still block their complete applications to practical engineering problems: a compromise between accuracy and efficiency on mixed-curved meshes, inherent vulnerability to numerical oscillations, and lack of open-source high-performance solvers for researchers. To address these issues, we present Deneb, an open-source high-order accurate numerical solver that enables high-performance scale-resolving simulations on PDE-based flow systems. Deneb uses the physical domain-based modal discontinuous Galerkin (DG) method; thus, it can provide an arbitrary high-order accurate solution on mixed-curved meshes and has the potential for handling polyhedral meshes as well. However, the DG method becomes significantly expensive on high-order curved elements when using quadrature rules, blocking its applicability to practical engineering problems. To avoid this, the direct reconstruction method (DRM) is applied to the DG volume and surface integrals to perform efficient numerical integration on non-affine elements without accuracy loss. The resulting DRM-DG method eliminates the severe cost of a quadrature-based approach on mixed-curved meshes. Deneb offers explicit and implicit Runge–Kutta methods as well to achieve high-order accuracy in time. In addition, Krylov subspace methods and preconditioners are available for high-performance linear system solving in parallel. Deneb possesses a strong capability to resolve multi-physical shocks without numerical instabilities with the aid of multi-dimensional limiting and artificial viscosity methods. In particular, the hierarchical multi-dimensional limiting process enables efficient computations of supersonic flows without time-step restriction. The current release of Deneb covers the simulations of hypersonic equilibrium and magneto-hydrodynamic flows as well as compressible Navier–Stokes equations, but it has the potential to solve any PDE-based multi-physical flow systems. Several benchmark problems are presented to highlight Deneb's capability to perform scale-resolving and multi-physical flow simulations. A scalability test is also presented to verify the scaling characteristics of Deneb for high-performance computing.

Keywords: Direct Reconstruction Method, Discontinuous Galerkin Method, Open-source Code, Multi-physical Flows, High-performance Computing.

1 Introduction

Finite element-based high-order methods have been widely embraced and explored by the computational fluid dynamics (CFD) community due to their unique features. Unlike high-order extensions in finite volume methods (FVM) employing wide stencils, finite element-based high-order methods can achieve an arbitrary high-order accuracy in smooth regions with compact stencils. This compactness makes the message passing process more straightforward and preserves high scalability on large-scale parallel computations. Using compact stencils is also beneficial for handling complex mesh systems flexibly. Combined with high-degree curved elements, high-order methods can accurately resolve flow physics along a curved surface, even with

coarse meshes. Furthermore, high-order methods are far more cost-effective than traditional second-order FVM, making scale-resolving simulations practically feasible. A simple analysis shows that numerical error decreases much faster than computational cost increases in higher-order methods. Finally, high-rank tensor operations that are ubiquitous in high-order methods have a high level of arithmetic intensity, and high performance on modern computing machines can be readily achieved.

During the last decade, there have been more efforts and progress than ever in developing high-order methods and their industrial applications using scale-resolving simulations [1, 2, 3]. At the same time, a series of International Workshop on High-Order CFD Methods have been held to establish well-defined benchmark test suites and identify pacing items for further research [4]. Based on these foundations, high-order methods provided several promising results in large-scale, high-fidelity simulations of viscous turbulent flows (e.g., [5, 6]). Although most studies were conducted by in-house solvers, developing open-source high-order CFD solvers is also actively underway [7, 8, 9, 10]. PyFR [7] is an open-source Python-based framework for solving advection–diffusion type problems based on the high-order energy-stable flux reconstruction method [11, 12, 13]. Nektar++ [8, 10] is an open-source software framework for solving partial differential equations using the high-order spectral/*hp* element method [14].

Despite such promising aspects, several obstacles still block the complete application of high-order methods to practical engineering problems [2, 4, 15]. First, developing a highly flexible, high-order numerical algorithm for complex grid systems is necessary. The numerical algorithm must be efficient on mixed-curved meshes without compromising high-order accuracy. In addition, due to the less diffusive nature of high-order methods, they are very susceptible to producing non-physical oscillations near discontinuities. Thus, robust and accurate shock-capturing methods are needed. Finally, open-source codes based on high-order methods are relatively less accessible compared to finite volume methods. This lack of accessibility raises the entry barrier and delays the widespread uptake of high-order methods in multi-physical engineering applications. Besides, fast and reliable mesh generation techniques, robust mesh adaptation, and scalable iterative methods to handle stiffness are also important issues.

The current paper presents Deneb, an open-source high-performance CFD solver based on the physical domain-based modal discontinuous Galerkin (DG) method, to address the aforementioned issues. This method can achieve the optimal order of accuracy with the optimal number of polynomial bases even on non-affine elements [16, 17]. Thus, Deneb provides the optimal accuracy on mixed-curved meshes than the solvers based on reference domain-based methods. The use of modal bases also enables further extensions to polyhedral meshes. However, the physical domain-based modal DG method suffers a substantially high computational cost when using quadrature rules on mixed-curved elements where mapping functions are high-degree polynomials [18]. To overcome this issue, the direct reconstruction method (DRM) [19, 20, 21] is developed and applied to the DG method for efficient numerical integration of the DG residual on mixed-curved elements without compromising accuracy. Deneb employs both artificial viscosity and limiting methods to suppress unwanted numerical oscillations near flow discontinuities. In particular, the hierarchical multi-dimensional limiting process (hMLP and hMLP_BD) [22, 23, 24, 25] enables efficient and robust computations of high-speed compressible flow simulations without time-step restriction. Like most open-source CFD solvers, the numerical methods adopted in Deneb are also applicable to any PDE-based flow system. The current release of Deneb supports hypersonic equilibrium and magneto-hydrodynamic flow simulations, which are not available in other open-source higher-order CFD solvers. Finally, we summarize Deneb's key features as follows:

- Arbitrary high-order accuracy on hybrid meshes with all types of mixed-curved elements in two and three dimensions;
- Various high-order explicit and implicit time-stepping methods with a range of preconditioned Krylov subspace methods;
- Shock-capturing with both artificial viscosity and multi-dimensional limiting methods that enable flow simulation containing multi-physical shock waves;
- High-performance scale-resolving simulation of viscous turbulent flows;
- Multi-physical flow simulation of hypersonic equilibrium and magneto-hydrodynamic systems;

- Efficient parallel computing showing strong scaling up to 2,000 cores and weak scaling up to 15,000 cores on the KISTI NURION supercomputer;
- Support for Linux and Windows operating systems.

The present paper is organized as follows: The high-order DRM-DG and various time-stepping methods are presented in Sec. 2. Shock-capturing methods are also described in the same section. Governing systems supported by the current release of Deneb are provided in Sec. 3. Selected benchmark tests demonstrating Deneb's simulation capability are illustrated in Sec. 4. Finally, conclusions are presented in Sec. 5.

2 Numerical Methods

2.1 Discontinuous Galerkin Spatial Discretization

We apply the high-order DG method to a hyperbolic-type system of equations given by

$$\begin{aligned} \frac{\partial \mathbf{q}}{\partial t} + \sum_{i=1}^{d} \frac{\partial}{\partial x^i} \mathbf{f}^i(\mathbf{q}, \boldsymbol{\nabla}\mathbf{q}) = \mathbf{v}(\mathbf{q}, \boldsymbol{\nabla}\mathbf{q}), \\ t \in [0, \infty), \quad \mathbf{x} \in \Omega \subset \mathbb{R}^d, \end{aligned} \tag{1}$$

where $\mathbf{q} = \left[q^1, \cdots, q^s\right]^T$ is a vector of state variables, $\mathbf{f}^i = \left[f^{i,1}, \cdots, f^{i,s}\right]^T$ is a x^i-directional flux vector, and $\mathbf{v} = \left[v^1, \cdots, v^s\right]^T$ is a source vector. The spatial domain Ω is partitioned into N_{elem} non-overlapping polyhedral elements as $\Omega = \cup_{i=1}^{N_{\text{elem}}} D_i$. Let $\partial D_i^{\text{B}} = \partial D_i \cap \partial\Omega$ and $\partial D_i^{\text{I}} = \partial D_i \setminus \partial D_i^{\text{B}}$. Thus, physical boundary conditions are applied to ∂D_i^{B} while numerical fluxes are to be evaluated along ∂D_i^{I}.

Since the DG discretization is applied elementwise, the element index is omitted for brevity. For the k^{th}-order approximation (DG-Pk), we assume the following form of numerical solution

$$\mathbf{q}(t, \mathbf{x}) \approx \mathbf{q}^h = \left(\mathbf{I}_s \otimes \boldsymbol{\phi}^T\right) \tilde{\mathbf{q}}, \tag{2}$$

where $\boldsymbol{\phi} = \left[\phi^1, \cdots, \phi^b\right]^T$ is a set of polynomial bases for the k^{th}-degree polynomial space $\mathbb{P}^k(D)$, $\mathbf{I}_s$ is a s-by-s identity matrix, $\tilde{\mathbf{q}}$ is a vector of modal coefficients of the solution defined by

$$\tilde{\mathbf{q}} = \left[q^{1,1}, \cdots, q^{1,b}, \cdots, q^{s,1}, \cdots, q^{s,b}\right]^T, \tag{3}$$

i.e, $q^i = \sum_{j=1}^{b} q^{i,j}\phi^j$. Here, $\otimes$ is the Kronecker product, and $b = \dim \mathbb{P}^k(D) =_{k+d} C_d$.

We use polynomials in physical coordinates for $\boldsymbol{\phi}$ due to their distinctive properties: they can handle polyhedral meshes and even meshes with arbitrarily shaped elements (e.g., [26, 18]); in addition, they can achieve the formal order of accuracy with the optimal number of polynomial bases on non-affine elements, such as mixed-curved elements. On the contrary, methods based on reference domain bases fail to achieve the designed order of accuracy on non-affine elements unless a solution space is enlarged, resulting in considerable computational costs [16, 17]. It is noted that keeping the same number of bases for all elements is beneficial to well-balanced memory partitioning and fast memory access. We employ orthonormal polynomials due to the computational benefit of the diagonal mass matrix. The orthonormal polynomial bases can be obtained from a set of prescribed monomials using the modified Gram–Schmidt process. The detailed algorithm can be found in [19].

The auxiliary variables, $\boldsymbol{\theta}$, are introduced to evaluate the gradient of the state vector as

$$\boldsymbol{\theta}_j = \frac{\partial \mathbf{q}^h}{\partial x^j} + \sum_{e \in \partial D} \mathbf{r}_{e,j}^{-}, \quad \boldsymbol{\theta}_{e,j}^{\pm} = \frac{\partial \mathbf{q}^{h\pm}}{\partial x^j} + \mathbf{r}_{e,j}^{\pm} \quad \forall e \in \partial D, \tag{4}$$

where $\mathbf{r}_{e,j}^{\pm}$'s are from the following lifting operator:

$$\int_D \mathbf{r}_{e,j}^{\pm} \otimes \boldsymbol{\phi}^{\pm} dV = \mp \int_e n^j (\widehat{\mathbf{q}} - \mathbf{q}^{h\pm}) \otimes \boldsymbol{\phi}^{\pm} dA. \tag{5}$$

Here, values with plus or minus symbol mean that they are calculated from a neighbor or owner element, respectively, and $\mathbf{n} = [n^1, \cdots, n^d]^T$ is a normal vector at e directing from the owner to neighbor element. We omit the minus symbol for brevity if it does not cause any confusion in the context. Following the BR2 approach [27], $\widehat{\mathbf{q}}$ becomes a central value of $\mathbf{q}^{h\pm}$ when $e \in \partial D^{\mathrm{I}}$. For $e \in \partial D^{\mathrm{B}}$, proper boundary conditions ($\widehat{\mathbf{q}} = \widehat{\mathbf{q}}^{\mathrm{B}}$) are applied. Being analogous to Eq. (2), let $\tilde{\mathbf{r}}_{e,j}^{\pm}$ be a vector of modal coefficients of $\mathbf{r}_{e,j}^{\pm}$ satisfying $\mathbf{r}_{e,j}^{\pm} = (\mathbf{I} \otimes \boldsymbol{\phi}^{\pm T}) \tilde{\mathbf{r}}_{e,j}^{\pm}$.

Multiplying the orthonormal polynomial bases to Eq. (1) and integrating it over D, the following weak formulation is obtained:

$$\begin{aligned} \int_D \frac{\partial \mathbf{q}^h}{\partial t} \otimes \boldsymbol{\phi} dV = \quad & - \sum_{e \in \partial D^{\mathrm{I}}} \underbrace{\int_e \widehat{\mathbf{F}} \cdot \mathbf{n} \otimes \boldsymbol{\phi} dA}_{= \mathbf{R}_e^{\mathrm{I}}} - \sum_{e \in \partial D^{\mathrm{B}}} \underbrace{\int_e \widehat{\mathbf{F}^{\mathrm{B}}} \cdot \mathbf{n} \otimes \boldsymbol{\phi} dA}_{= \mathbf{R}_e^{\mathrm{B}}} \\ & + \sum_{i=1}^{d} \underbrace{\int_D \mathbf{F}^i \otimes \frac{\partial \boldsymbol{\phi}}{\partial x^i} dV}_{= \mathbf{R}_i^f} + \underbrace{\int_D \mathbf{V} \otimes \boldsymbol{\phi} dV}_{= \mathbf{R}^v} = -\mathbf{R}, \end{aligned} \tag{6}$$

where $\widehat{\mathbf{F}} = \widehat{\mathbf{F}}(\mathbf{q}^{h\pm}, \boldsymbol{\theta}_{e,j}^{\pm}; \mathbf{n})$ is a numerical flux, $\widehat{\mathbf{F}^{\mathrm{B}}} = \widehat{\mathbf{F}^{\mathrm{B}}}(\mathbf{q}^h, \boldsymbol{\theta}_{e,j}; \mathbf{n})$ is a boundary flux, $\mathbf{F}^i = \mathbf{f}^i(\mathbf{q}^h, \boldsymbol{\theta}_j)$, and $\mathbf{V} = \mathbf{v}(\mathbf{q}^h, \boldsymbol{\theta}_j)$. The left hand side of Eq. (6) is simplified as

$$\int_D \frac{\partial \mathbf{q}^h}{\partial t} \otimes \boldsymbol{\phi} dV = \mathbf{I}_s \otimes \left(\int_D \boldsymbol{\phi}^T \otimes \boldsymbol{\phi} dV \right) \frac{d\tilde{\mathbf{q}}}{dt} = \frac{d\tilde{\mathbf{q}}}{dt} \tag{7}$$

owing to the orthonormality of $\boldsymbol{\phi}$.

2.2 Time Marching Schemes

Gathering the weak formulation of Eq. (6) for all elements results in the following system of time-dependent nonlinear ordinary differential equations:

$$\frac{d\bar{\mathbf{q}}}{dt} = -\bar{\mathbf{R}}(\bar{\mathbf{q}}), \tag{8}$$

where $\bar{\mathbf{q}}$ is an appropriate gathering of $\tilde{\mathbf{q}}$. The numerical solution is then advanced in time for unsteady simulation by a high-order explicit or implicit multi-stage Runge-Kutta (R-K) method. For an explicit case, we use a family of strong-stability-preserving R-K (SSPRK) methods given by

$$\begin{aligned} \bar{\mathbf{u}}_i &= -\bar{\mathbf{R}} \left(t^n + \alpha_i \Delta t, \bar{\mathbf{q}}^n + \sum_{j=1}^{i-1} a_{ij} \bar{\mathbf{u}}_j \right), \quad i = 1, \cdots, s, \\ \bar{\mathbf{q}}^{n+1} &= \bar{\mathbf{q}}^n + \sum_{j=1}^{s} m_j \bar{\mathbf{u}}_j, \end{aligned} \tag{9}$$

where the coefficients (α_i, a_{ij}, m_j) are given in [28, 29]. The time-step Δt is the minimum value of local time-steps Δt_i computed by

$$\Delta t_i = \sigma_{\mathrm{CFL}} \times \left[\sum_{j=1}^{d} \frac{\lambda_{j,c}}{h_j} + \frac{\lambda_{j,v}}{h_j^2} \right]^{-1}, \quad h_j = (2k+1) \frac{|D_i|}{D_{i,j}}, \tag{10}$$

where $|D_i|$ is the volume of D_i and $|D_{i,j}|$ is the projected area of D_i on the plane orthogonal to x^j-axis. $\lambda_{j,c}$ and $\lambda_{j,v}$ represent the maximum eigenvalue of convective and viscous fluxes along x^j-axis, respectively. The CFL number, σ_{CFL}, is usually set to 0.9 for explicit simulations.

For implicit simulation, a family of Rosenbrock-type R-K methods given by

$$\begin{aligned}\left(\frac{\mathbf{I}}{\Delta t\gamma_{ii}}+\mathbf{J}\right)\bar{\mathbf{u}}_i &= -\bar{\mathbf{R}}\left(t^n+\alpha_i\Delta t,\bar{\mathbf{q}}^n+\sum_{j=1}^{i-1}a_{ij}\bar{\mathbf{u}}_j\right)\\ &\quad+\sum_{j=1}^{i-1}\left(\frac{c_{ij}}{\Delta t}\bar{\mathbf{u}}_j\right)+\gamma_i\Delta t\frac{\partial\bar{\mathbf{R}}}{\partial t}\left(t^n,\bar{\mathbf{q}}^n\right),\quad i=1,\cdots,s,\\ \bar{\mathbf{q}}^{n+1} &= \bar{\mathbf{q}}^n+\sum_{j=1}^{s}m_j\bar{\mathbf{u}}_j,\end{aligned}\tag{11}$$

is employed due to computational efficiency [30, 31]. Here, $\mathbf{J}=\partial\bar{\mathbf{R}}/\partial\bar{\mathbf{q}}$ is the system Jacobian matrix. The coefficients of A-stable and L-stable Rosenbrock-type methods are well tabulated in [32, 33].

For steady simulation, Eq. (8) is solved with the implicit backward Euler method until $\bar{\mathbf{R}}(\bar{\mathbf{q}})$ becomes sufficiently small. Applying the pseudo-transient continuation (PTC) and Jacobian linearization, we have

$$\begin{aligned}\left(\frac{\mathbf{I}}{\Delta\tau}+\mathbf{J}\right)\Delta\bar{\mathbf{q}} &= -\bar{\mathbf{R}}(\bar{\mathbf{q}}^n),\\ \bar{\mathbf{q}}^{n+1} &= \bar{\mathbf{q}}^n+\Delta\bar{\mathbf{q}},\end{aligned}\tag{12}$$

where $\Delta\tau$ is a pseudo local time-step of each cell.

2.3 Direct Reconstruction Method

Computational efficiency and numerical accuracy of the DG method directly depend on how the residual $\mathbf{R}$ and the system Jacobian matrix $\mathbf{J}$ are treated. It is impractical or even impossible to analytically calculate $\mathbf{R}$ and $\partial\mathbf{R}/\partial\tilde{\mathbf{q}}$ due to the non-linearity of the flux functions. Thus, they are approximately computed using numerical integration.

Applying quadrature rules is the simplest way to perform the numerical integration. In the quadrature-based approach, the domain of integral needs to be transformed into the reference domain on which quadrature rules are defined. However, this leads to enormous computational costs and memory overhead when the integral function is defined on a curved element. This is because a mapping function to the reference domain makes the degree of the transformed integrand very high. In this case, computational costs scale as $(kN_{\mathfrak{T}})^d$ for volume integral functions [19], where $N_{\mathfrak{T}}$ is the degree of the mapping function. Thus, computational costs become even more severe in three dimensions.

DRM was proposed in a quadrature-free manner to overcome the severe computational costs of the quadrature-based approach [19, 20, 21]. The basic idea of DRM is to approximate flux functions by nodal reconstruction. This conceptually results in a hybrid approach combining modal and nodal methods; the numerical solution and flux functions are approximated by modal and nodal bases, respectively. Nodal expansion of flux functions is computed using generalized reconstruction, which corresponds to a general extension of interpolation. In DRM, the nonlinear mapping is avoided by directly applying the nodal reconstruction in the physical domain. Thus, it requires much fewer points than the quadrature-based approach, resulting in a considerable saving in computational costs.

Here, the basic formulation of DRM is briefly described. Assume that $\mathbf{F}^i$ in Eq. (6) is contained in a finite-dimensional polynomial space $\mathbb{S}(D)$, and let $\boldsymbol{\varphi}=\{\varphi^{(1)},\cdots,\varphi^{(d_s)}\}$ denote a set of bases of $\mathbb{S}(D)$ with $d_s=\dim\mathbb{S}(D)$. Note that d_s does not mean the spatial dimension of D. Consider a set of p ($\geq d_s$) points $\mathbf{X}=\{\mathbf{x}_1,\cdots,\mathbf{x}_p\}\subset D$ such that the generalized Vandermonde matrix

$$\mathbf{V}=\left(\varphi^{(i)}(\mathbf{x}_j)\right)_{ij}\in\mathbb{R}^{d_s\times p}\tag{13}$$

becomes full rank, i.e., $\text{Rank}(\mathbf{V}) = d_s$. Then it can be shown that

$$\mathbf{F}^i = \sum_{m=1}^{p} \mathbf{F}^i_m \ell^m, \tag{14}$$

where $\mathbf{F}^i_m = \mathbf{F}^i|_{\mathbf{x}=\mathbf{x}_m}$, $\mathbf{L} = [\ell^1, \cdots, \ell^p]^T = \mathbf{V}^{\text{g}}\boldsymbol{\varphi}$, and $\mathbf{V}^{\text{g}}$ is the generalized inverse matrix of $\mathbf{V}$ (i.e., $\mathbf{V} = \mathbf{V}\mathbf{V}^{\text{g}}\mathbf{V}$). Eq. (14) is called the generalized reconstruction of $\mathbf{F}^i$ since it becomes interpolation when $d_s = p$. Note that $\ell^i(\mathbf{x}_j) \neq \delta_{ij}$ in general when $p > d_s$. In DRM, $\mathbf{R}^f_i$ in Eq. (6) is computed using Eq. (14) as

$$\mathbf{R}^f_i = \sum_{m=1}^{p} \mathbf{F}^i_m \otimes \int_D \ell^m \frac{\partial \boldsymbol{\phi}}{\partial x^i} dV. \tag{15}$$

Similarly, we have

$$\frac{\partial \mathbf{R}^f_i}{\partial \tilde{\mathbf{q}}^{\pm}} = \sum_{m=1}^{p} \frac{\partial \mathbf{F}^i_m}{\partial \tilde{\mathbf{q}}^{\pm}} \otimes \int_D \ell^m \frac{\partial \boldsymbol{\phi}}{\partial x^i} dV. \tag{16}$$

In DRM, the generalized reconstruction is irrelevant to the mapping function degree, $N_{\mathfrak{T}}$. Thus, computational costs scale as k^d, which is significantly less than the quadrature-based approach. Other residual terms such as $\mathbf{R}^I_e$ can be computed similarly by defining $\mathbf{X}$ on e for surface integral functions [20]. Each flux function may have a different approximation space $\mathbb{S}(D)$ and reconstruction nodes $\mathbf{X}$. Thus, $\mathbb{S}(D)$ and $\mathbf{X}$ can be chosen flexibly according a user's preference. The approximation space and reconstruction nodes for each element type (i.e., triangle, quadrilateral, tetrahedron, hexahedron, prism, pyramid) used in Deneb are provided in [33] in detail.

2.4 Shock-capturing Methods

High-order methods are prone to generate spurious oscillations near flow discontinuities unless numerical dissipations are carefully managed. Spurious oscillations trigger numerical instabilities that lead to accuracy degradation, incorrect flow physics, and even failure of computations by non-physical flow data (e.g., negative pressure). Deneb adopts both artificial viscosity and limiting (hMLP and hMLP_BD) methods to suppress unwanted numerical oscillations.

In the artificial viscosity method, a dissipation term is added to the governing equations to stabilize the numerical solution. There have been many studies (e.g., [34, 35, 36, 37]) depending on how to control the amount of the dissipation term. In Deneb, a Laplacian form of dissipation ($\varepsilon\boldsymbol{\nabla}^2\mathbf{q}$) is adopted and the amount of artificial viscosity is controlled by [34]

$$\varepsilon = \begin{cases} 0 & \text{if } s_e < s_0 - \kappa, \\ \frac{\varepsilon_0}{2}\left(1 + \sin\frac{\pi(s_e - s_0)}{2\kappa}\right) & \text{if } s_0 - \kappa \leq s_e < s_0 + \kappa, \\ \varepsilon_0 & \text{otherwise,} \end{cases}, \quad s_e = \log_{10}\frac{\int_D \left(\mathbf{q}^h - \Pi^{k-1}\mathbf{q}^h\right)^2 dV}{\int_D {\mathbf{q}^h}^2 dV}, \tag{17}$$

where Π^m is a projection operator onto m^{th}-degree polynomial space. We follow the details in [35] for computing ε_0.

While artificial viscosity methods suffer time-step restriction due to the increase of viscous spectral radii, limiting methods have no such effect. Thus, they enable efficient supersonic flow computations. Deneb employs hMLP methods [22, 23, 24, 25]. In hMLP methods, troubled-cells are firstly detected by applying the *P1*-projected MLP condition at each vertex $v \in D$:

$$\overline{\mathbf{q}}^{\min}_v \leq \Pi^1\mathbf{q}^h(\mathbf{x}_v) \leq \overline{\mathbf{q}}^{\max}_v, \tag{18}$$

where $\mathbf{q}^h(\mathbf{x}_v)$ is a solution on D evaluated at the vertex v, and $\overline{\mathbf{q}}^{\min}_v$ and $\overline{\mathbf{q}}^{\max}_v$ denote the minimum and maximum of cell-average values among the cells sharing the vertex v, respectively. Solutions that violate Eq. (18) are considered troubled-cells containing discontinuities. Secondly, after imposing the *P1*-projected MLP condition, the smooth extrema detector is applied to preserve solution accuracy at smooth extrema.

In the smooth extrema detector, the numerical solution is decomposed into three parts:

$$\mathbf{q}^h(\mathbf{x}_v) = \overline{\mathbf{q}} + \underbrace{\left(\Pi^1\mathbf{q}^h(\mathbf{x}_v) - \overline{\mathbf{q}}\right)}_{Pk\text{-projected slope}} + \underbrace{\left(\mathbf{q}^h(\mathbf{x}_v) - \Pi^1\mathbf{q}^h(\mathbf{x}_v)\right)}_{P1\text{-filtered } Pk}. \tag{19}$$

Then, the following conditions are applied to detect local smooth extrema:

$\mathcal{C}1$. If there is a local maximum near the vertex v,

$$Pk\text{-projected slope} > 0, \quad P1\text{-filtered } Pk < 0, \quad \mathbf{q}^h(\mathbf{x}_v) > \overline{\mathbf{q}}_v^{\min}, \tag{20a}$$

$\mathcal{C}2$. If there is a local minimum near the vertex v,

$$Pk\text{-projected slope} < 0, \quad P1\text{-filtered } Pk > 0, \quad \mathbf{q}^h(\mathbf{x}_v) < \overline{\mathbf{q}}_v^{\max}. \tag{20b}$$

The MLP slope limiter is then applied to the cells violating Eqs. (18) and (20). We refer [22] for the detailed procedure.

The recent study [25] shows that the hMLP limiter [22] has two shortcomings: accuracy loss on mixed meshes and occurrence of subcell oscillations. The hMLP_BD limiter was developed to resolve these issues by introducing simplex decomposition and the troubled-boundary detector. hMLP_BD captures flow discontinuities without subcell oscillations on mixed meshes while preserving accuracy at smooth region. The detailed procedure is provided in [25].

3 Governing Systems

3.1 Compressible Navier–Stokes Equations

The system of compressible Navier–Stokes (N–S) equations is used to simulate laminar or turbulent flow simulation with the implicit LES (ILES) approach. It is given by

$$\mathbf{q} = \begin{bmatrix} \rho \\ \rho u^j \\ \rho E \end{bmatrix}, \mathbf{f}^i = \begin{bmatrix} \rho u^i \\ \rho u^i u^j + p\delta_{ij} - \tau_{ij} \\ u^i(\rho E + p) - \sum_{m=1}^{d} \tau_{im} u^m + Q^i \end{bmatrix}, \mathbf{v} = \mathbf{0}, \quad i, j \in \{1, \cdots, d\} \tag{21}$$

in a vector form, where ρ, u^j, E denote the density, velocity, and specific total energy, respectively. The viscous stress tensor and heat flux rate are

$$\tau_{ij} = \mu\left(\frac{\partial u^i}{\partial x^j} + \frac{\partial u^j}{\partial x^i} - \frac{2}{3}\sum_{m=1}^{d}\frac{\partial u^m}{\partial x^m}\right), \quad Q^i = -\kappa\frac{\partial T}{\partial x^i}, \tag{22}$$

where μ, κ, T denote the viscosity, thermal conductivity, and temperature, respectively. For a calorically perfect ideal gas, the pressure is computed by

$$p = (\gamma - 1)\,\rho e, \quad e = E - \frac{1}{2}\sum_{m=1}^{d} u^m u^m, \tag{23}$$

where $\gamma = 1.4$ is the heat capacity ratio. The maximum eigenvalues of the convective and viscous terms in the flux along x^i-axis are given by

$$\lambda_{i,c} = u^i + a, \quad \lambda_{i,v} = \frac{\mu}{\rho}\max(\frac{4}{3}, \frac{\gamma}{\mathrm{Pr}}), \tag{24}$$

where a and Pr denote the speed of sound and Prandtl number, respectively. The inviscid Euler equations are achieved by neglecting the effects of viscosity and heat conductivity from Eq. (21).

3.2 Hypersonic Equilibrium Flow

When the flight speed becomes high enough, the air behind a shock becomes high temperature and physical properties change due to chemical reactions, such as dissociation and ionization. Accurate modelling of physical properties is crucial for predicting aerothermodynamic loads on hypersonic vehicles. In thermochemical equilibrium, thermodynamic and transport properties of air become a function of two thermodynamic state variables. Curve-fit models have been widely used for computational efficiency [38, 39, 40].

The previous curve-fit models are based on piecewise polynomials and have several drawbacks: lack of accuracy, smoothness, diversity in property models, and extendability to a new dataset. IDEA [41, 42] was recently proposed based on artificial neural networks to overcome these limitations. IDEA provides more accurate, globally continuous, diverse, and trainable property models. Deneb uses IDEA to realize hypersonic flow simulations. The governing equations are identical to Eqs. (21) and (22), and unknown variables are determined by IDEA's property models:

$$p,\ \mu,\ \kappa,\ T = f_{\mathrm{IDEA}}(\rho,\ e), \quad e = E - \frac{1}{2}\sum_{m=1}^{d} u^m u^m. \tag{25}$$

3.3 Magneto-hydrodynamic Flow

We consider the non-relativistic compressible ideal magneto-hydrodynamic (MHD) flows. Starting with the inviscid Euler equations, the Lorentz force and magnetic field evolution equations are added to the system. The eight-wave MHD system [43, 44] is adopted as a baseline. Although it has a non-zero source term that makes the system non-conservative, it also makes the system Galilean invariant [45] and symmetrizable [46] and helps to convect the non-zero magnetic divergence out of the domain. Note that the system has zero eigenvalues without the source term.

Since the eight-wave system has no mechanism to remove the non-zero magnetic divergence in a stagnation region, we additionally adopt the generalized Lagrange multiplier (GLM) approach to realize a divergence-free magnetic field [47, 48, 44, 49]. In the current study, we apply the following GLM-MHD system:

$$\begin{aligned}
\mathbf{q} &= \begin{bmatrix} \rho \\ \rho u^j \\ \rho E + \|\mathbf{B}\|^2/2 + \psi^2/2 \\ B^j \\ \psi \end{bmatrix}, \mathbf{f}^i = \begin{bmatrix} \rho u^i \\ \rho u^i u^j - B^i B^j + \left(p + \|\mathbf{B}\|^2/2\right)\delta_{ij} \\ u^i(\rho E + p + \|\mathbf{B}\|^2) - B^i \sum_{m=1}^{d} u^m B^m + c_h \psi B^i \\ u^i B^j - u^j B^i + c_h \psi \delta_{ij} \\ c_h B^i \end{bmatrix}, \\
\mathbf{v} &= -\boldsymbol{\nabla}\cdot\mathbf{B}\begin{bmatrix} 0 \\ B^j \\ \sum_{m=1}^{d} u^m B^m \\ u^j \\ 0 \end{bmatrix} - (\boldsymbol{\nabla}\psi\cdot\mathbf{u} + \alpha\psi)\begin{bmatrix} 0 \\ \mathbf{0} \\ \psi \\ \mathbf{0} \\ 1 \end{bmatrix}, \quad i,j \in \{1,\cdots,d\},
\end{aligned} \tag{26}$$

where $\mathbf{B} = [B^1, \cdots, B^d]^T$ is a magnetic field and ψ is a mixed hyperbolic-parabolic ansatz with $\alpha = 2$. The hyperbolic propagation speed c_h is chosen to be

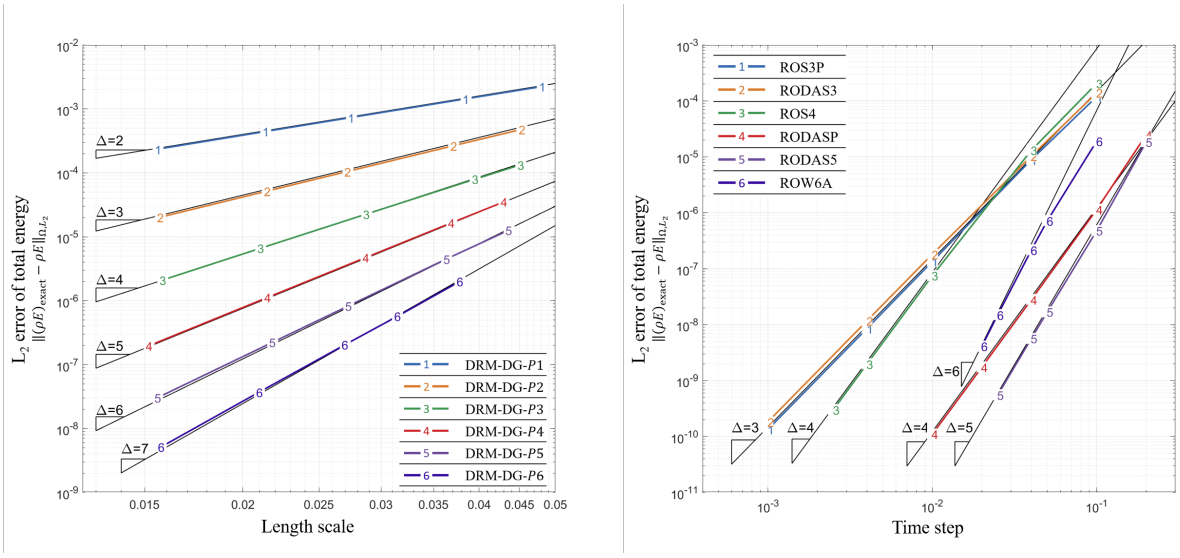
$$c_h = \lambda_{\max} - u_{\max,\Omega}, \tag{27}$$

where $\lambda_{\max}$ is the maximum eigenvalue of the system and $u_{\max,\Omega}$ is the maximum flow speed in the domain [44].

4 Highlights on Numerical Simulations

4.1 High-order Accuracy

The high-order accuracy of Deneb in space and time is verified by the method of manufactured solution (MMS). The compressible Navier–Stokes equations in two dimensions are simulated on a box domain with



(a) Spatial accuracy

(b) Temporal accuracy

Figure 1: Refinement study of the MMS problem on regular Cartesian meshes; Error were measured at $t = 1.0$.

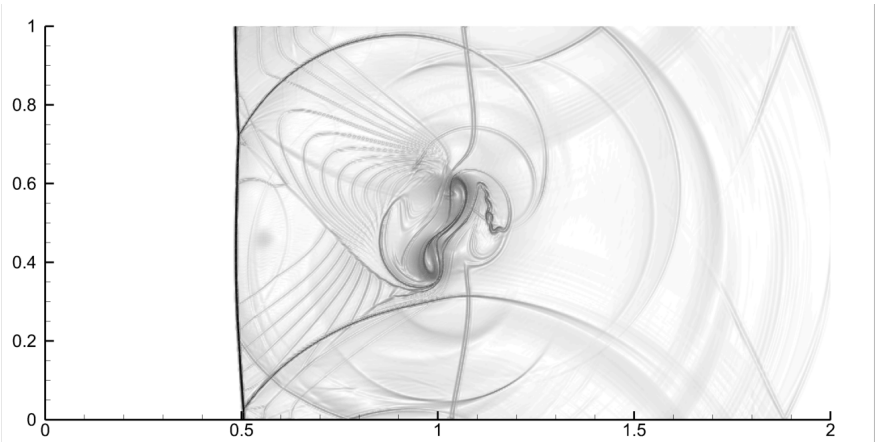
periodic boundary conditions. The governing system is modified to have an analytic spatiotemporal solution by adding a manufactured source term [50]. The flow parameters are (Re, Ma) = (600, 0.15). We tested various approximation orders of the DRM-DG method and implicit Rosenbrock-type R–K methods. Fig. 1 shows the results of the refinement study with the expected high-order accuracy of Deneb.

4.2 Complex Shock Flow Simulations

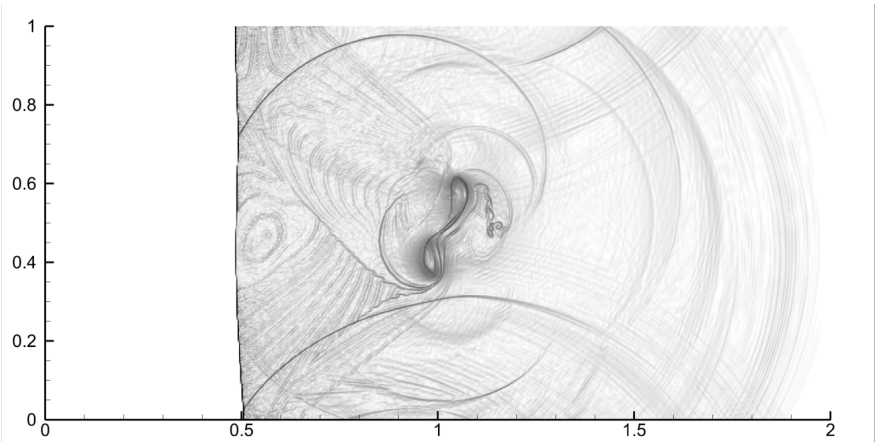
Two benchmark problems are presented to evaluate Deneb's capability to perform complex shock flow simulations. Both problems are governed by the Euler equations. The first benchmark is a strong vortex–shock wave interaction problem[1] [51]. This test aims to assess the performance of Deneb on complex flow arising from the interaction between a strong vortex and a shock wave. A strong vortex characterized by $\mathrm{Ma_v} = 0.9$ penetrates a standing shock with $\mathrm{Ma_s} = 1.5$, leading to complex shock pattern and vortex splitting. We use DRM-DG-*P2* and TVDRK methods. The mesh consists of 3.6×10^5 triangles in the computational domain of $[0, 2] \times [0, 1]$. Fig. 2 compares the results computed by each shock-capturing method implemented in Deneb. All three results successfully resolve key flow features, such as vortex splitting. The artificial viscosity method offers very sharp shock-capturing but produces some non-physical shock waves originating from the standing shock wave. Both artificial viscosity and *h*MLP methods produce negative flow variables, suffering a stability issue. For example, we observed that numerical solutions blow up on the Cartesian meshes with a similar resolution even though we adopted a scaling technique [52]. In artificial viscosity, numerical solutions were very sensitive to the tunable parameters, such as Pe, κ in [35]. On the contrary, *h*MLP_BD provides slightly more diffusive results, but they have no negative values and are robust to the mesh type. The *h*MLP_BD provides the fastest computation. For example, *h*MLP_BD is 3.6× faster than artificial viscosity.

The second benchmark is a shock–wedge interaction problem, also known as Schardin's problem [53]. A moving shock with Ma = 1.34 passes through a 30° wedge and generates complex flow structures, such as a slip layer with the Kelvin–Helmholtz instability, diverging acoustics, and a series of shock interactions over a long time scale. We use DRM-DG-*P3* and SSPRK methods. The mesh is irregular mixed and consists of about one million elements. Fig. 3 shows both computed and experimental [54] Schlieren results, confirming that *h*MLP and *h*MLP_BD accurately capture complex flow patterns. In particular, the series of vortices

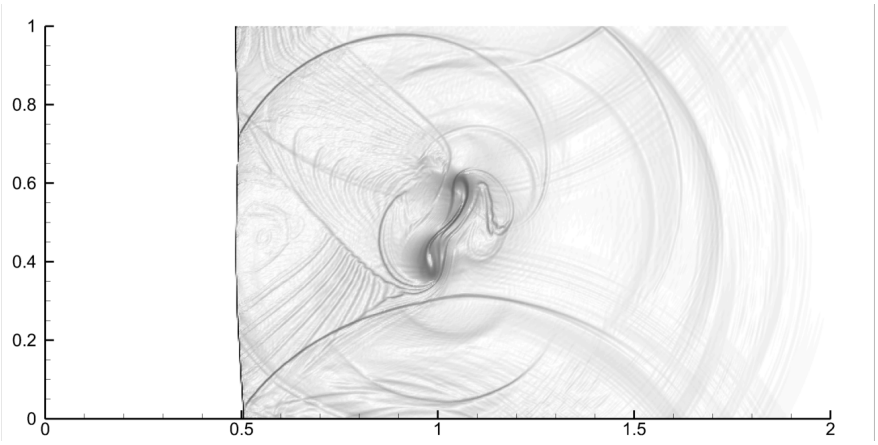
[1]This was one of the benchmark problems in the Fifth International Workshop on High-Order CFD Method (2018) and the High Fidelity CFD Workshop (2022).



(a) Artificial viscosity (Pe = 2, $\kappa = 4$)

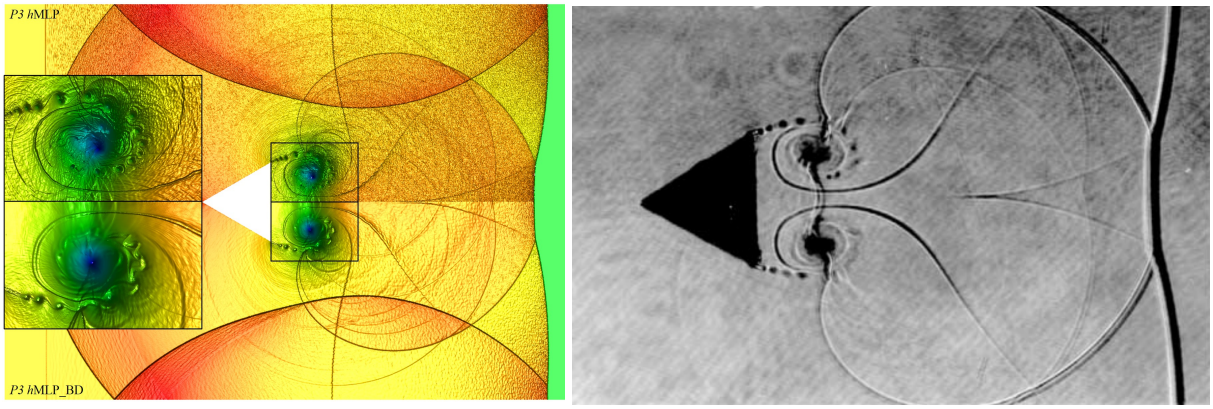


(b) hMLP



(c) hMLP_BD

Figure 2: Numerical Schlieren view of the strong vortex–shock wave interaction problem at $t = 0.7$



(a) Computed result at $t = 3.25$ (b) Experimental result

Figure 3: Computed and experimental results of the shock–wedge interaction problem

along the slip layer behind the wedge show a good agreement with the experiment. The result computed by hMLP contains small-scale wiggles over the entire computational domain from subcell oscillations created by the incoming shock. Subcell oscillations eventually produce negative flow variables, causing a stability problem. In the case of hMLP_BD, subcell oscillations across the shock wave are effectively removed and the downstream flow field is not compromised by numerical wiggles.

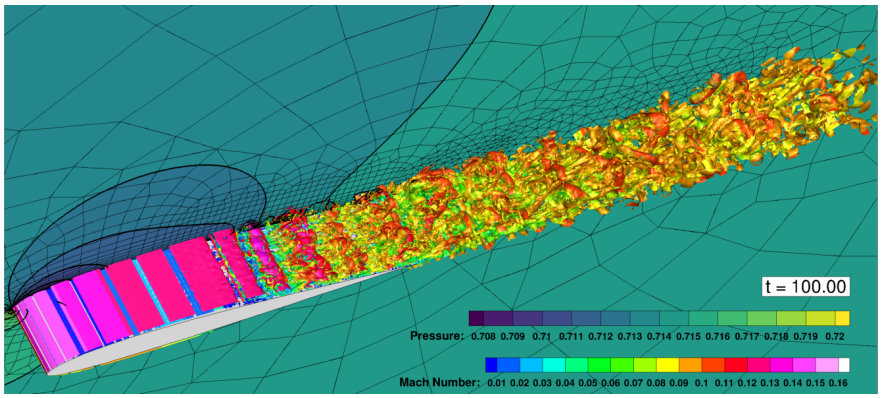
4.3 Scale-resolving Flow Simulations

Two challenging turbulent flow problems are presented to highlight Deneb's capability to perform scale-resolving simulations. Scale-modeled simulations, such as RANS, hardly capture the flow physics of this class of problems. The benefits of the DRM-DG method are also verified by comparing the computational costs with the quadrature-based method.

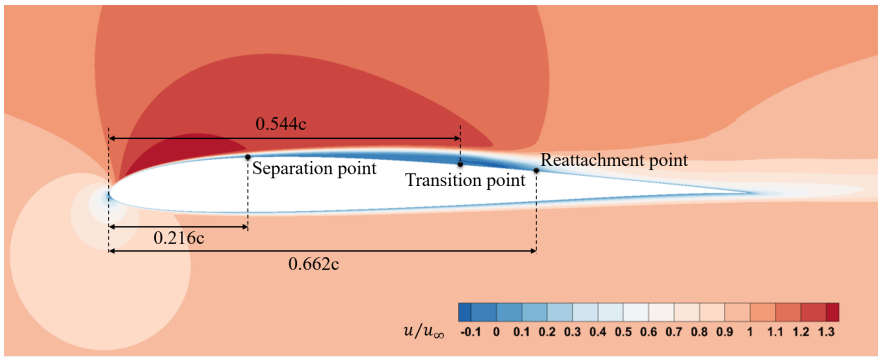
First, ILES is carried out for the transitional flow over the SD7003 airfoil at $(\mathrm{Re_c},\ \mathrm{Ma}) = (6 \times 10^4,\ 0.1)$ and an incidence angle of 4°. The flow is initially laminar and attached on the leading edge suction side but separated soon to develop roll-up vortices and their break-down. The transition to turbulence occurs in the detached shear layer, and it is reattached to the airfoil, producing a laminar separation bubble. The computational domain consists of *P3*-curved hexahedral and prism meshes with 35,730 elements. RODAS5 is used for time marching with a constant time-step of $\Delta t = 0.001$. As shown in Fig. 4, Deneb resolves multi-scale turbulent eddies and the laminar separation bubble. Although not presented here, spectral analyses in the time domain confirm that the resolved eddies follow the Kolmogorov $-5/3$ spectrum of turbulent energy cascade. The surface force coefficients are compared with previous studies [55, 56, 57, 58] in Fig. 5, showing that the current results are close to the DNS data [55]. It is noted that the RANS result failed to predict the skin friction after the flow is reattached.

The amounts of floating-point operations and computer memory are compared with the quadrature-based method in Tab. 1. DRM-DG requires approximately 30× less floating-point operations and 5.5× smaller memory usage. The elapsed time is also measured and compared in Tab. 2. These results confirm again the computational efficiency of DRM-DG compared to the quadrature-based method.

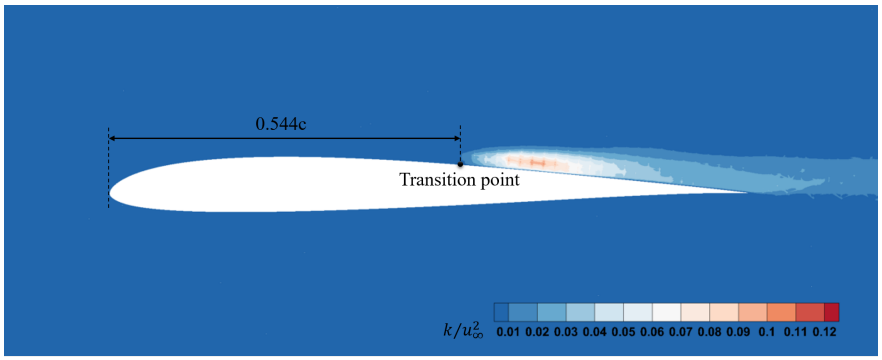
Second, ILES is carried out to the separated flow over a $3.5D$ tangent-ogive forebody at $(\mathrm{Re_D},\ \mathrm{Ma}) = (8 \times 10^5,\ 0.25)$ and the incident angle of 40°. In this flow condition, steady asymmetric vortices are generated around the nose, leading to a significant side force, even though the target geometry is axisymmetric. RANS computations have usually failed to capture the flow asymmetry without geometry perturbation and produced non-physical flow features by over-predicting the turbulent viscosity in massively separated vortical flow [59, 60, 61, 62, 63]. The forebody base is extended by $3.5D$ to make a cylindrical afterbody in the simulation. The mesh consists of 219,436 *P2*-curved elements containing all types of mixed elements; tetrahedrons, hexahedrons, prisms, and pyramids. The artificial viscosity [34] is applied adaptively to stabilize an aliasing-driven instability. We use DRM-DG-*P2* and RODAS3 methods with $\Delta t = 0.005$. Fig. 6 illustrates both instantaneous and averaged pressure distributions at the ogive surface, confirming the asymmetry of pressure



(a) Instantaneous field (iso-surface by Q-criterion)

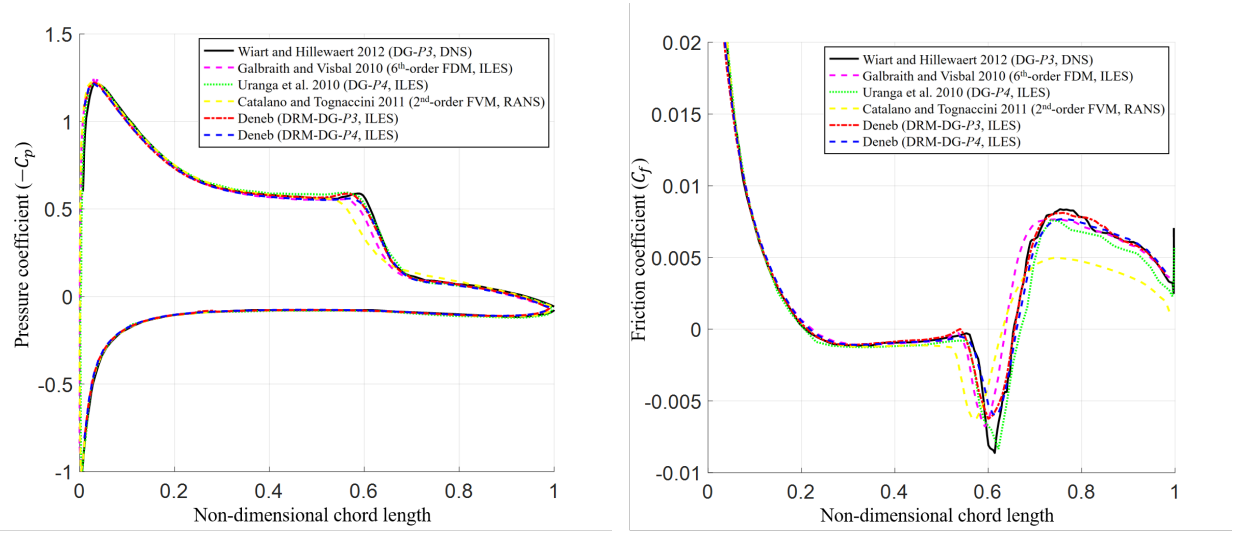


(b) Mean stream-wise velocity



(c) Mean turbulent kinetic energy

Figure 4: Both instantaneous and averaged flow fields around the SD7003 airfoil computed by DRM-DG-*P4*



(a) Pressure coefficient

(b) Friction coefficient

Figure 5: Pressure and friction coefficients at the surface of the SD7003 airfoil

Table 1: The number of floating-point operations and memory required to solve the SD7003 airfoil problem

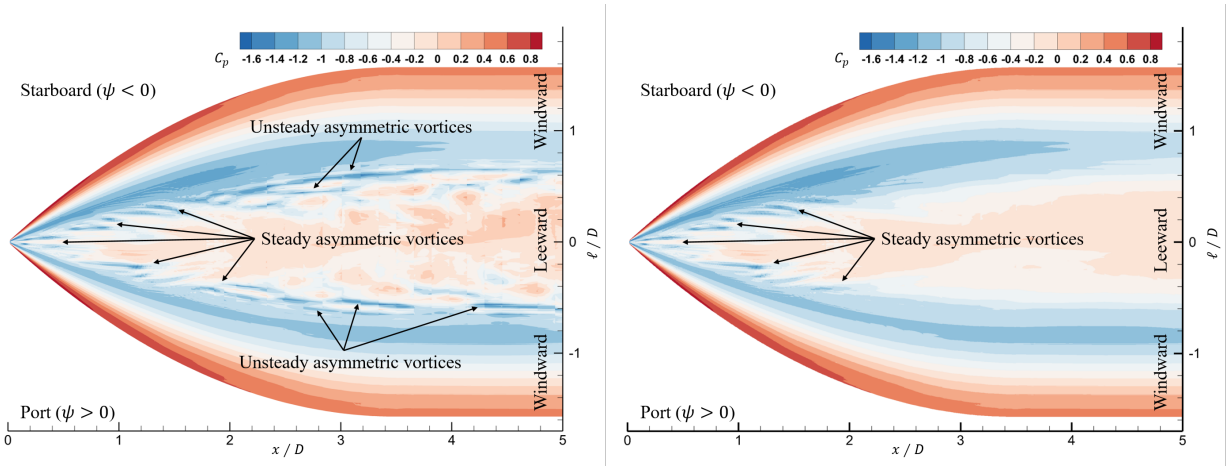
		Operation counts (Gflop)		Memory (GiB)	
		System matrix	Residual	Tensor coefficients	System matrix
DG-*P3*	DRM	2.5×10^3	1.8×10^1	8.2	1.8×10^1
	Quadrature	7.7×10^4	2.8×10^2	1.3×10^2	1.8×10^1
DG-*P4*	DRM	1.2×10^4	4.9×10^1	2.3×10^1	5.4×10^1
	Quadrature	3.8×10^5	8.4×10^2	4.0×10^2	5.4×10^1

Table 2: The elapsed time of DRM-DG and quadrature-based methods; The computing machine consists of 15 nodes; 30 Intel Xeon *E5-2650 v4* CPUs, 360 cores in total.

		Elapsed time relative to DRM-DG-*P3*†			
		Total	System matrix	GMRES	Others
DG-*P3*	DRM	1.00†	0.29	0.69	0.02
	Quadrature‡	7.80	6.88	0.74	0.18
DG-*P4*	DRM	3.98	1.32	2.62	0.04
	Quadrature‡	24.12	20.49	2.89	0.74

‡ It should be noted that DRM was still used for the surface integration in the case of the quadrature-based method, i.e., a quadrature rule is used only for the volume integration only.

distribution. In Fig. 7, steady asymmetric vortices are clearly observed near the nose, leading to the non-zero side force. The forebody has normal and side force coefficients of $C_{\mathrm{n}} = 1.3035$ and $C_{\mathrm{s}} = -0.3163$, respectively, which are consistent with the experimental data in [64]. Fig. 8 compares surface pressure coefficients at several locations on the forebody with the previous studies [65, 60]. The RANS result failed to resolve the flow asymmetry without a small asymmetric bump near the forebody apex, while the current ILES successfully captures the flow asymmetry. The current results are much more consistent with the experimental data, confirming Deneb's capability in scale-resolving simulations.



(a) Instantaneous contour

(b) Mean contour

Figure 6: Surface pressure distribution of the tangent-ogive forebody

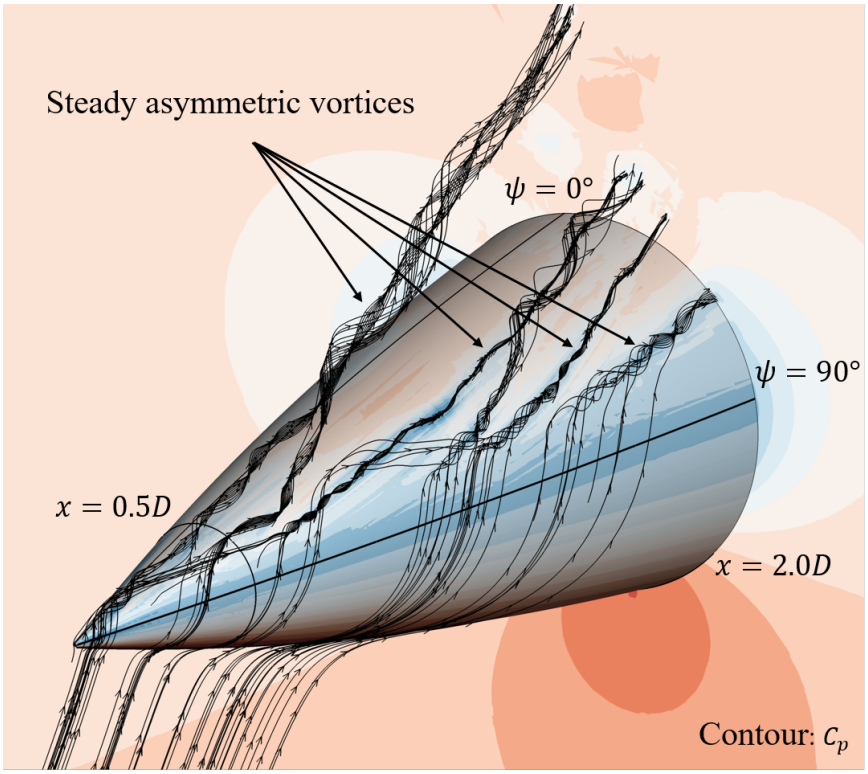
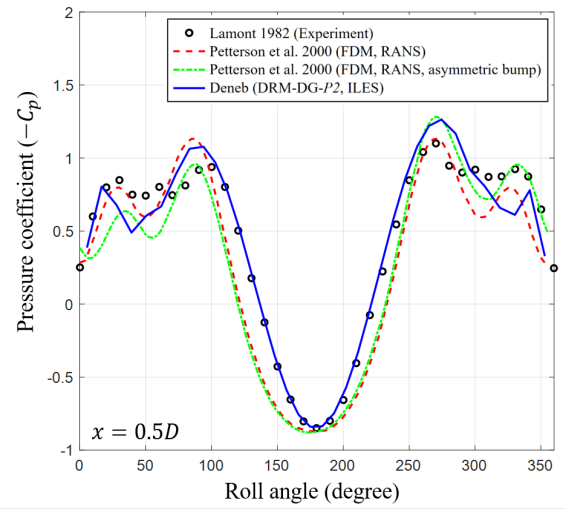
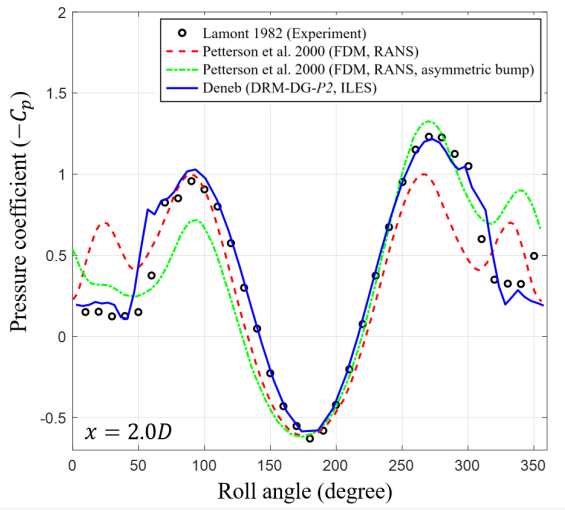


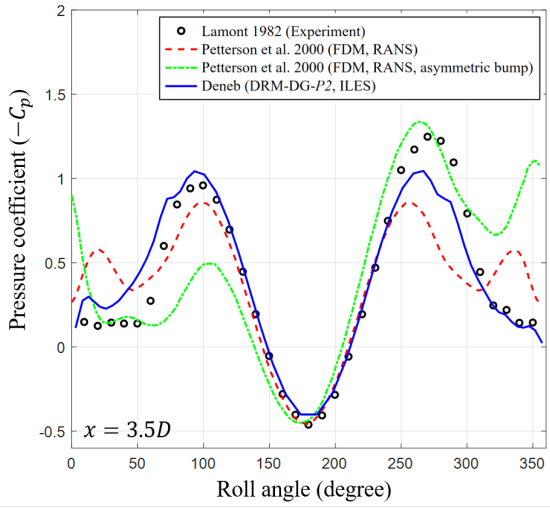
Figure 7: Steady asymmetric vorties around the tangent-ogive forebody (averaged flow field)



(a) $x = 0.5D$

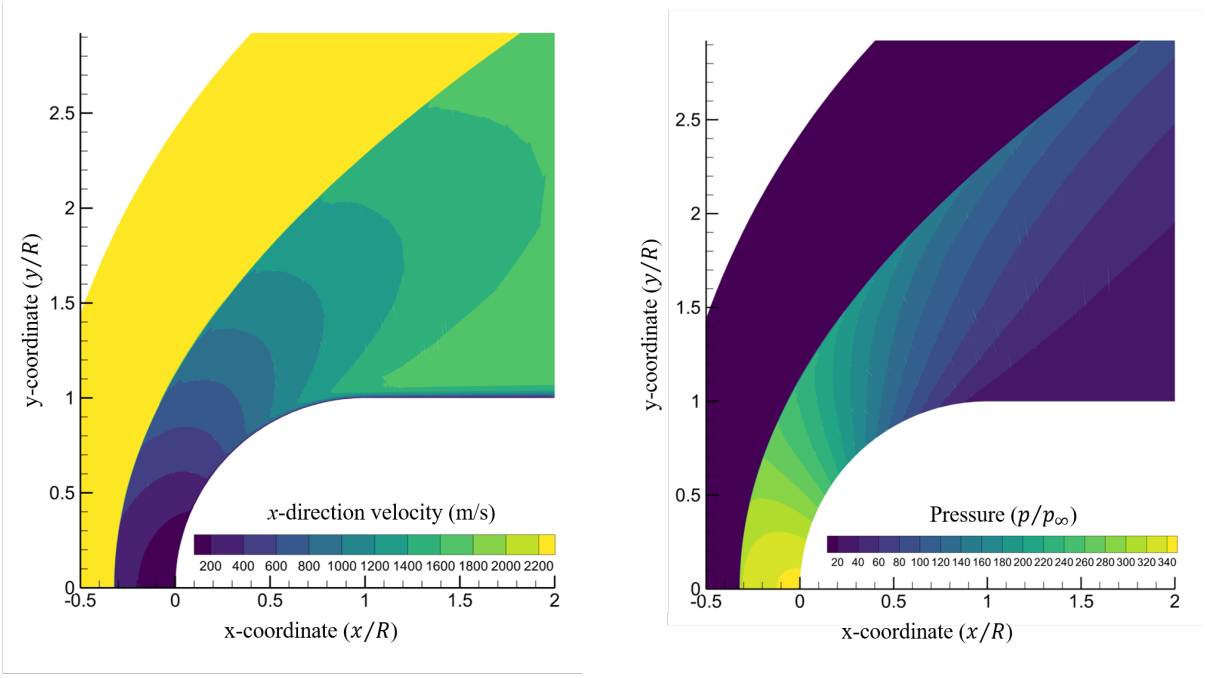


(b) $x = 2.0D$



(c) $x = 3.5D$

Figure 8: Pressure coefficients of the tangent-ogive cylinder



(a) x-direction velocity

(b) Pressure

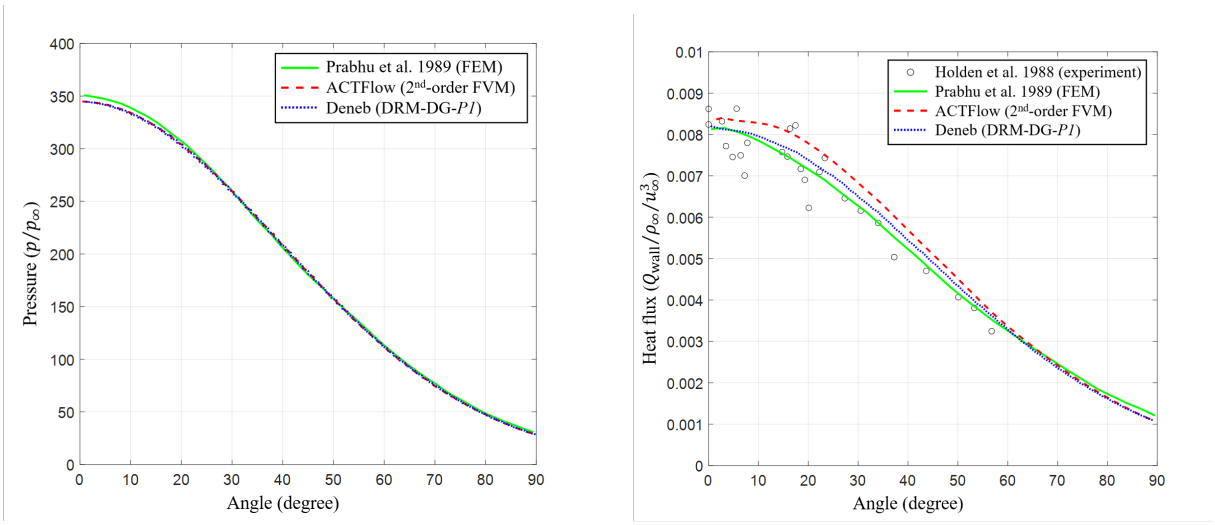
Figure 9: Solution of the hypersonic leading edge problem with a 280×44 *P2*-curved quadrilateral mesh on the symmetrical plane

4.4 Multi-physical Flow Simulations

Several benchmark problems are presented to highlight Deneb's capability to solve compressible multi-physical flow systems—hypersonic equilibrium and MHD flows. hMLP_BD is used to capture flow discontinuities, and TVDRK is adopted for time integration. The first benchmark is a hypersonic laminar flow past a 1.5 inch-radius cylindrical leading edge with Ma = 16.34 [66, 67]. The flow conditions are $(p,\ T)_\infty = (82.944\ \text{Pa},\ 52.183\ \text{K})$, which corresponds to $\text{Re}_\text{D} \approx 2.5 \times 10^5$, and the wall is isothermal with $T_\text{wall} = 294.4$ K. Fig. 9 presents the velocity and pressure contours, showing that both strong shock and boundary layer are well resolved. Fig. 10 compares the surface pressure and heat transfer rate with previous studies [68, 66, 69]. One more benchmark for hypersonic equilibrium flow is presented. A half-meter diameter sphere flies at the speed of 3.048 km/s at an altitude of 76.2km [70]. The free stream conditions are $(\rho,\ p,\ T)_\infty = \left(3.993 \times 10^{-5}\ \text{kg/m}^3,\ 2.089\ \text{Pa},\ 182.333\ \text{K}\right)$, corresponding to Ma = 11.26 and $\text{Re}_\text{D} = 4976$. The wall is isothermal with $T_\text{wall} = 1000$ K. Fig. 11 shows the velocity and pressure contours with a strong bow shock. The computed results are also consistent with previous studies [70, 71][2], as shown in Fig. 12.

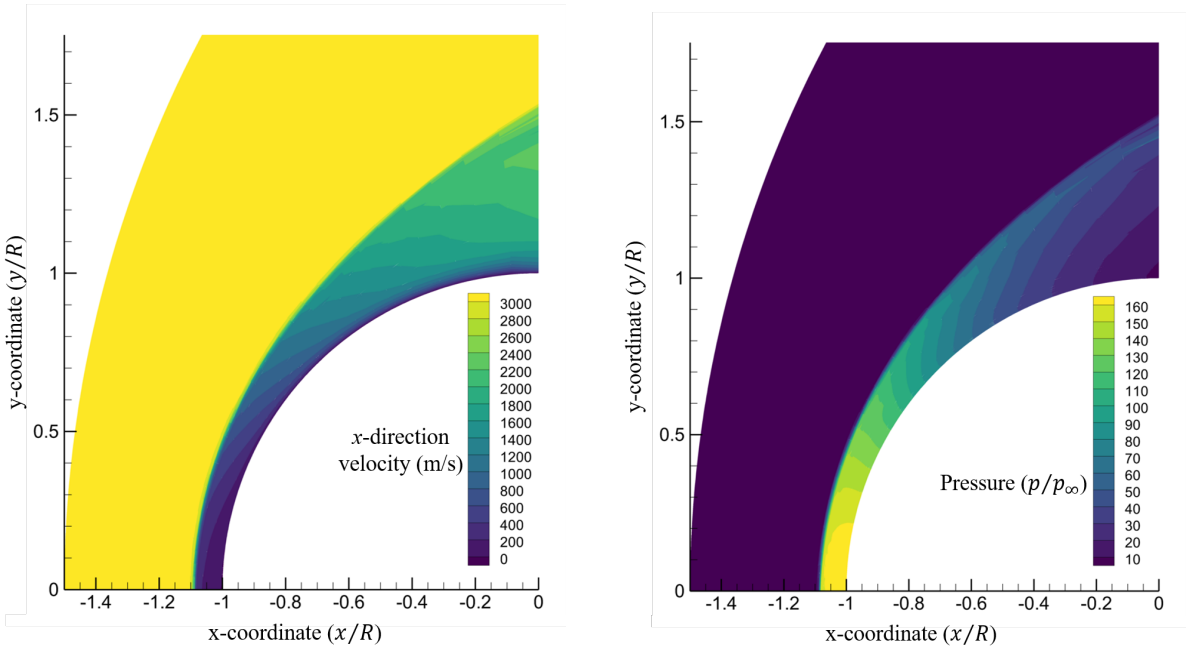
Orszag–Tang MHD vortex [72] is simulated to evaluate the performance of Deneb in resolving MHD turbulence. A 350×350 Cartesian mesh is used on the computational domain of $[0, 1]^2$ with periodic boundary conditions. While the initial flow is smooth, complex flow structures are gradually developed, such as shock–shock interaction and magnetic reconnection, which eventually lead to compressible turbulence, as shown in Fig. 13. The present results are compared with previous studies [49, 73] in Fig. 14, confirming that Deneb successfully resolves complex MHD flow physics. One more benchmark problem for MHD flow is presented. The MHD rotor problem [74] is closely related to the magnetic braking mechanism of a rotating interstellar cloud during star formation [75]. Initially, a dense spinning fluid body, corresponding to a protocluster cloud having high angular momentum, is surrounded by sparse stationary gas, representing a rarefied interstellar medium, under a constant perpendicular magnetic field. As shown in Fig. 15, the interaction between the rotating fluid and the magnetic field generates strong torsional Alfvén waves propagating outward and the

[2]The data of [71] are extracted from [70].



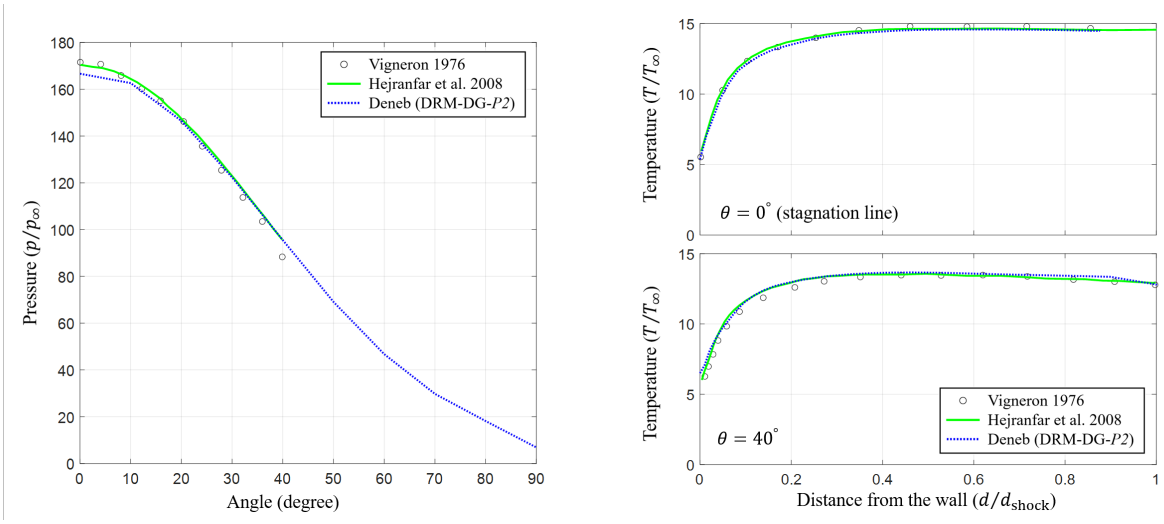
(a) Pressure (b) Heat transfer rate

Figure 10: Surface pressure and heat transfer rate of the hypersonic leading edge problem



(a) x-direction velocity (b) Pressure

Figure 11: Solution of the hypersonic sphere problem with a $60,000$ $P2$-curved hexahedral mesh on the full domain



(a) Pressure

(b) Temperature

Figure 12: Pressure profile along the surface and temperature profiles along the wall normal lines of the hypersonic sphere problem

loss of angular momentum. The current result is compared with previous studies [76, 49] in Fig. 16. It verifies again that Deneb captures shock locations without non-monotonic oscillations and resolves detonation physics governed by the MHD equations.

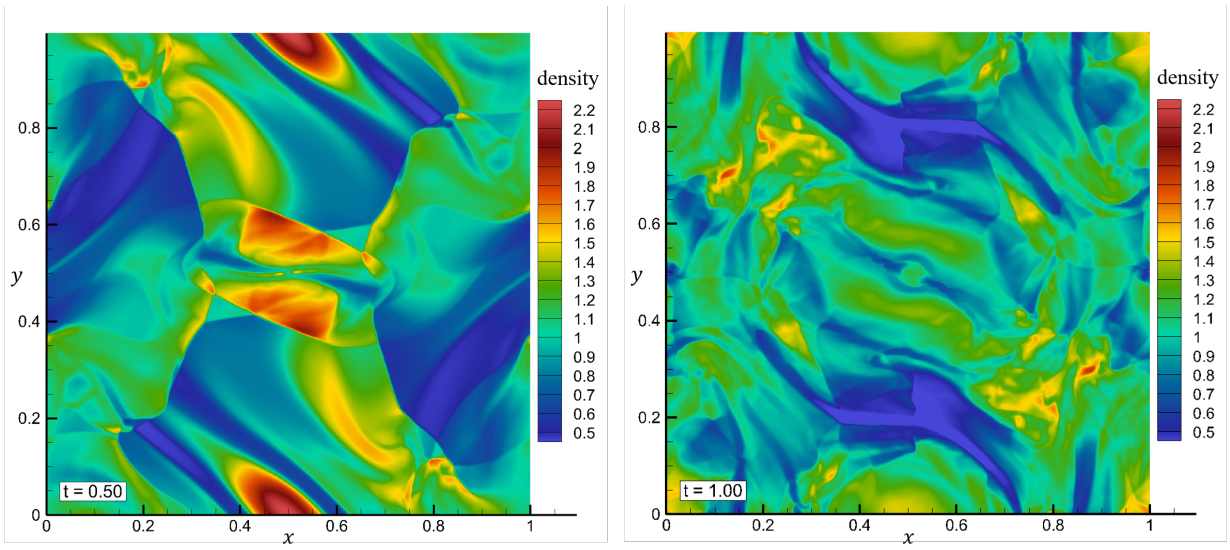
4.5 Scalability

The scalability characteristic of Deneb is evaluated on the Nurion supercomputer[3] operated by the Korea Institute of Science and Technology Information (KISTI). Nurion consists of computing nodes connected by Omni-Path interconnect networks, and each node has one *Intel Xeon Phi 7250* CPU. The compressible Navier–Stokes system in three dimensions is solved by the DRM-DG method. All runs are performed on the fixed box domain $[0,\ 1]^3$ with periodic boundary conditions. For simplicity, solutions are computed for two hundred time-steps by the SSPRK method, and then computational cost is measured.

Two metrics are used to evaluate the scalability performance. The strong scaling represents how computational cost varies with the number of cores N_{proc} for fixed total workloads, and ideally, computational cost scales as $1/N_{\text{proc}}$. In the strong scalability test, we fix the number of elements as $N_{\text{elem}} = 2.88 \times 10^5$ for DRM-DG-*P3*, $N_{\text{elem}} = 1.01 \times 10^5$ for DRM-DG-*P4*, and $N_{\text{elem}} = 4.32 \times 10^4$ for DRM-DG-*P5*. Fig. 17 illustrates the strong scaling of Deneb up to 4,000 cores. Deneb shows excellent scaling up to about 2,000 cores in this setting. The low performance at the largest cores is accounted for by the small number of elements assigned to each core; for example, most cores merely have eleven elements for DRM-DG-*P5*.

The weak scaling means how computational cost depends on the number of cores for fixed workloads per core. The computational cost remains unchanged for the ideal case. In the weak scalability test, the mesh is refined so that the number of elements is equal to the number of cores multiplied by the factor of one hundred. Fig. 18 presents the weak scaling of Deneb up to 15,000 cores. Deneb shows excellent scaling up to about 10,000 cores in this setting. The parallel efficiency increases with the approximation order due to the high level of arithmetic intensity. Hybrid parallelization, such as OpenMP–MPI, is necessary to achieve scaling beyond $O(10^4)$ cores.

[3]As of Nov. 2021, it was ranked the 38th in the world's top500 list supercomputers (top500.org).



(a) $t = 0.5$ (b) $t = 1$

Figure 13: Density contour of the Orszag–Tang vortex problem computed by DRM-DG-*P2*

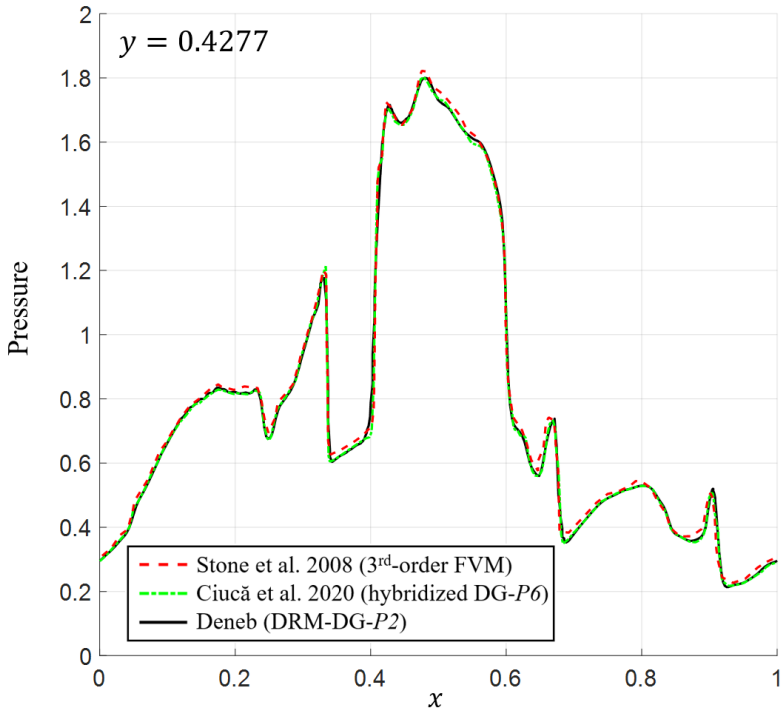
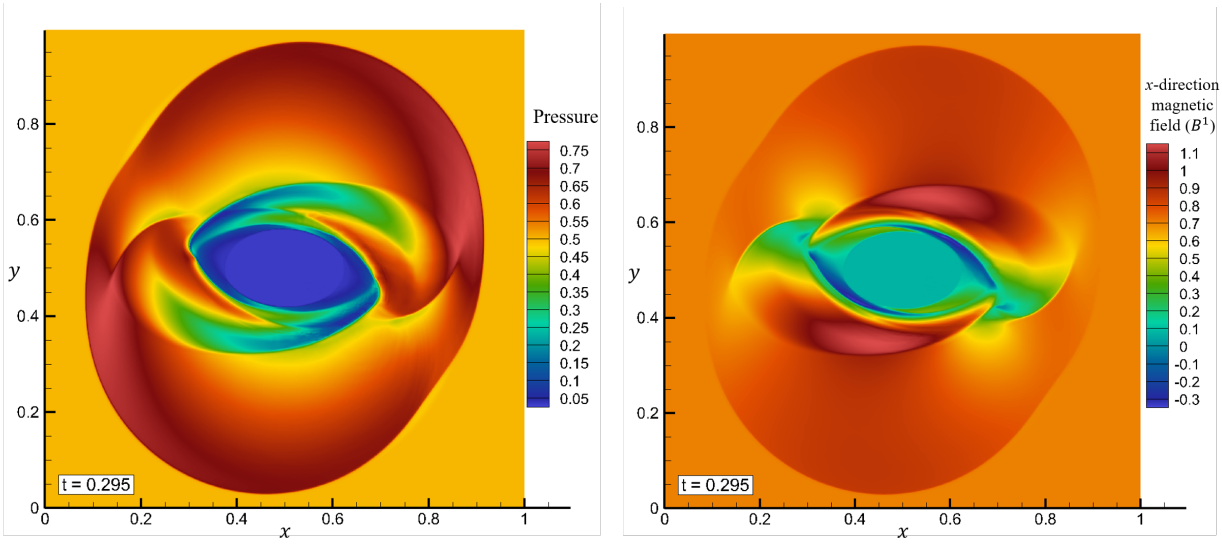


Figure 14: Pressure profile of the Orszag–Tang vortex problem at along $y = 0.4277$ at $t = 0.5$



(a) Pressure (b) x-direction magnetic field

Figure 15: Solution of the MHD rotor problem computed by DRM-DG-*P2*

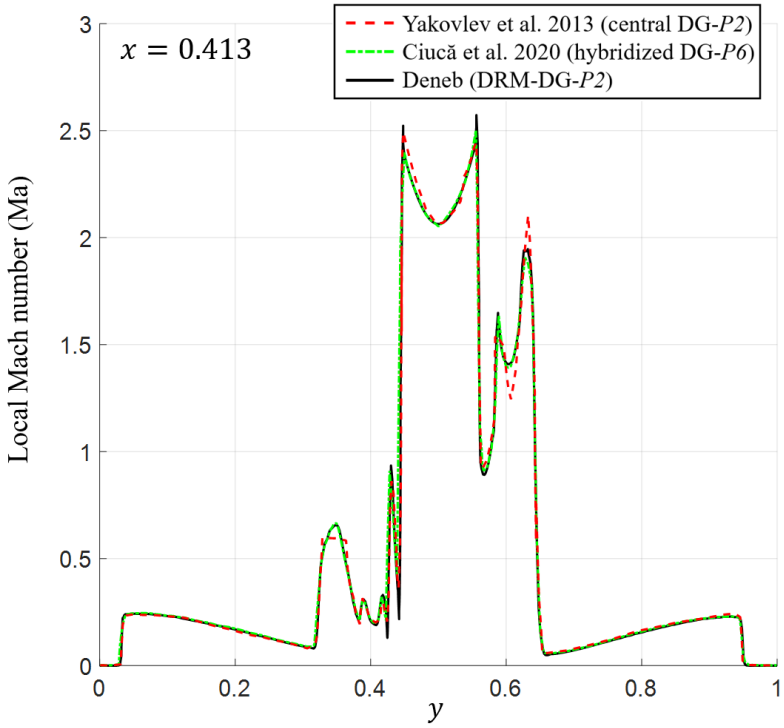
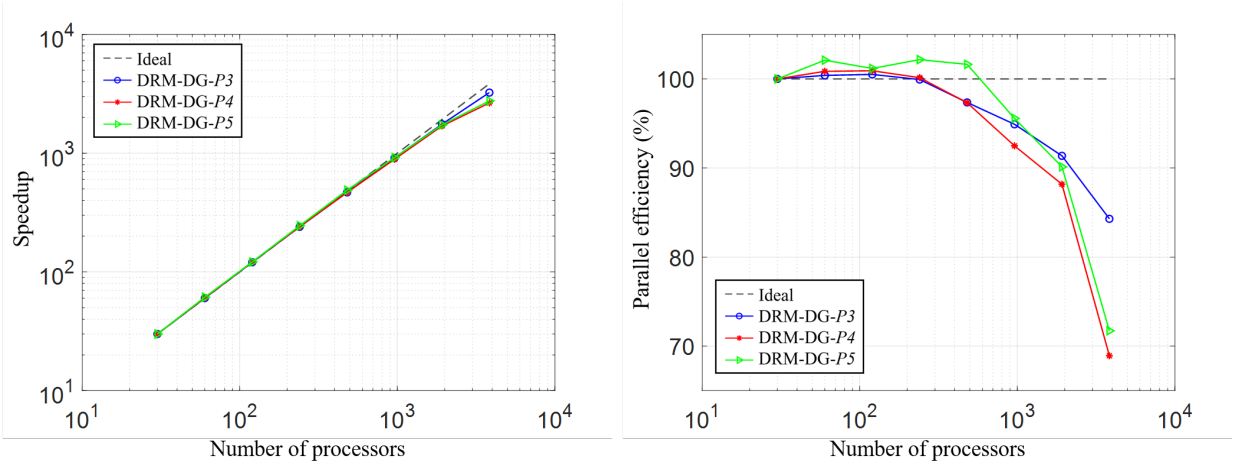


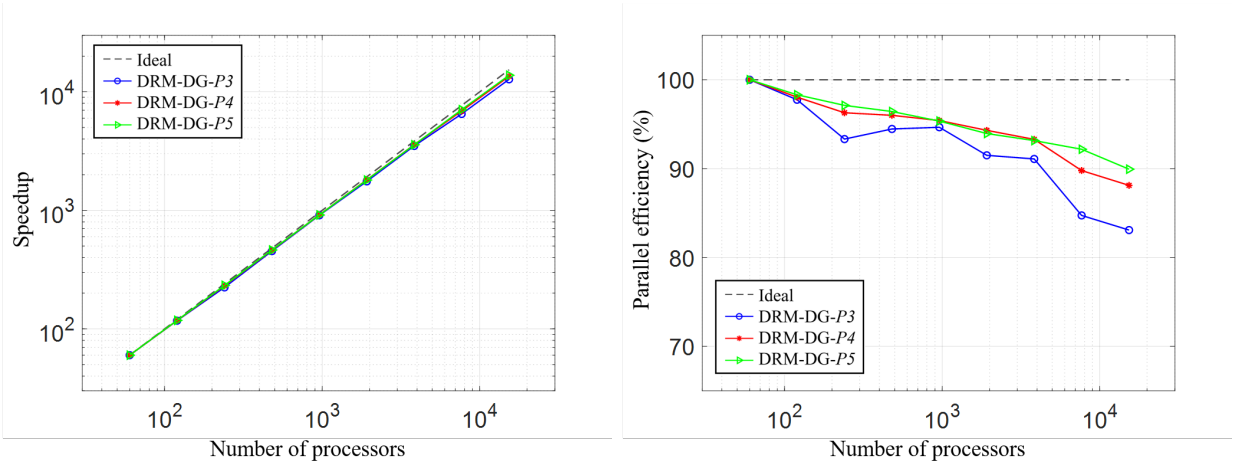
Figure 16: Local Mach number profile of the MHD rotor problem at along $x = 0.413$ at $t = 0.295$



(a) Speedup

(b) Parallel efficiency

Figure 17: Strong scaling of Deneb up to 4,000 cores; Each computing node was assigned 30 cores, and the scaling was calculated on the basis of one computing node.



(a) Speedup

(b) Parallel efficiency

Figure 18: Weak scaling of Deneb up to 15,000 cores; Each computing node was assigned 60 cores, and the scaling was calculated on the basis of one computing node.

5 Conclusion

The present paper introduces Deneb, an open-source high-order accurate numerical solver for high-performance scale-resolving simulations on PDE-based flow systems. Based on the physical domain-based modal DG method, Deneb provides an arbitrary high-order accurate solution with the optimal bases set on mixed-curved meshes. The novel DRM formulation is applied to the DG discretization, called DRM-DG, for the efficient integration of the DG volume and surface integrals without accuracy loss on high-order curved meshes, leading to the significant saving of the computational costs in the quadrature-based DG method. Various explicit and implicit Runge–Kutta methods are available in Deneb to achieve high-order accuracy in time. In addition, Deneb supports a range of high-performance parallel preconditioned Krylov subspace methods offered by the PETSc library. Both artificial viscosity and multi-dimensional limiting (hMLP and hMLP_BD) methods are available in Deneb to suppress numerical oscillations near flow discontinuities. In particular, the limiting methods enable the efficient computation of high-speed compressible flows with-

out time-step restriction. The current release of Deneb covers, aside from the compressible Navier–Stokes equations, hypersonic equilibrium and magneto-hydrodynamic flow systems, not previously available in open-source higher-order CFD solvers. Thanks to the easy extendability of the DRM formulation, Deneb has the potential for solving any PDE-based multi-physical flow system. A wide spectrum of benchmark problems demonstrates Deneb's capability to perform high-performance, scale-resolving, multi-physical compressible flow simulations. The scalability test confirms that Deneb has excellent scaling on high-performance supercomputer. Finally, we hope that Deneb contribute to the widespread uptake of higher-order methods in many multi-physical engineering and scientific applications, complementing other open-source codes, and lowering the entry barriers for researchers in various disciplines.

6 Acknowledgements

The authors appreciate the financial support provided by the Data-driven Flow Modeling Research Laboratory funded by the Defense Acquisition Program Administration under Grant UD200046CD. This research was also supported by the program of the National Research Foundation of Korea (NRF-2021R1A2C2008348).

References

[1] Norbert Kroll, Heribert Bieler, Herman Deconinck, Vincent Couaillier, Harmen van der Ven, and Kaare Sørensen. *ADIGMA—A European initiative on the development of adaptive higher-order variational methods for aerospace applications*. Springer Berlin Heidelberg, 2010.

[2] Norbert Kroll, Charles Hirsch, Francesco Bassi, Craig Johnston, and Koen Hillewaert. *IDIHOM: Industrialization of high-order methods—A top-down approach*. Springer International Publishing, 2015.

[3] Charles Hirsch, Koen Hillewaert, Ralf Hartmann, Vincent Couaillier, Jean-Francois Boussuge, Frederic Chalot, Sergey Bosniakov, and Werner Haase. *TILDA: Towards industrial LES/DNS in aeronautics*. Springer International Publishing, 2021.

[4] Z. J. Wang, Krzysztof Fidkowski, Rémi Abgrall, Francesco Bassi, Doru Caraeni, Andrew Cray, Herman Deconinck, Ralf Hartmann, Koen Hillewaert, H. T. Huynh, Norbert Kroll, Georg May, Per-Olof Persson, Bram van Leer, and Miguel Visbal. High-order CFD methods: current status and perspective. *International Journal for Numerical Methods in Fluids*, 72(8):811–845, 2013.

[5] Andrew C. Kirby, Michael J. Brazell, Zhi Yang, Rajib Roy, Behzad R. Ahrabi, Dimitri J. Mavriplis, Michael K. Stoellinger, and Jay Sitaraman. Wind farm simulations using an overset hp-adaptive approach with blade-resolved turbine models. *23rd AIAA Computational Fluid Dynamics Conference*, pages 1–24, 2017.

[6] Z. J. Wang and Salman Rahmani. Implicit large Eddy simulation of the NASA CRM high-lift configuration near stall. *Computers & Fluids*, 220:104887, 2021.

[7] F. D. Witherden, A. M. Farrington, and P. E. Vincent. PyFR: An open source framework for solving advection–diffusion type problems on streaming architectures using the flux reconstruction approach. *Computer Physics Communications*, 185(11):3028–3040, 2014.

[8] C. D. Cantwell, D. Moxey, A. Comerford, A. Bolis, G. Rocco, G. Mengaldo, D. De Grazia, S. Yakovlev, J. E. Lombard, D. Ekelschot, B. Jordi, H. Xu, Y. Mohamied, C. Eskilsson, B. Nelson, P. Vos, C. Biotto, R. M. Kirby, and S. J. Sherwin. Nektar++: An open-source spectral/*hp* element framework. *Computer Physics Communications*, 192:205–219, 2015.

[9] Thomas D. Economon, Francisco Palacios, Sean R. Copeland, Trent W. Lukaczyk, and Juan J. Alonso. SU2: An open-source suite for multiphysics simulation and design. *AIAA Journal*, 54(3):828–846, 2016.

[10] D. Moxey, C. D. Cantwell, Yan Bao, Andrea Cassinelli, Giacomo Castiglioni, Sehun Chun, Emilia Juda, Ehsan Kazemi, Kilian Lackhove, Julian Marcon, Gianmarco Mengaldo, Douglas Serson, Michael Turner, Hui Xu, Joaquim Peiró, Robert M. Kirby, and Spencer J. Sherwin. Nektar++: Enhancing the capability and application of high-fidelity spectral/*hp* element methods. *Computer Physics Communications*, 249:107110, 2020.

[11] H. T. Huynh. A flux reconstruction approach to high-order schemes including discontinuous Galerkin methods. *18th AIAA Computational Fluid Dynamics Conference*, 2007.

[12] H. T. Huynh. A reconstruction approach to high-order schemes including discontinuous Galerkin for diffusion. *47th AIAA Aerospace Sciences Meeting including The New Horizons Forum and Aerospace Exposition*, pages 1–34, 2009.

[13] Peter E. Vincent, P. Castonguay, and Antony Jameson. A new class of high-order energy stable flux reconstruction schemes. *Journal of Scientific Computing*, 47:50–72, 2011.

[14] George Karniadakis and Spencer Sherwin. *Spectral/hp element methods for CFD*. Oxford University Press, 2005.

[15] H. T. Huynh, Z. J. Wang, and P. E. Vincent. High-order methods for computational fluid dynamics: A brief review of compact differential formulations on unstructured grids. *Computers & Fluids*, 98:209–220, 2014.

[16] Lorenzo Botti. Influence of reference-to-physical frame mappings on approximation properties of discontinuous piecewise polynomial spaces. *Journal of Scientific Computing*, 52:675–703, 2012.

[17] David Moxey, Shankar P. Sastry, and Robert M. Kirby. Interpolation error bounds for curvilinear finite elements and their implications on adaptive mesh refinement. *Journal of Scientific Computing*, 78:1045–1062, 2019.

[18] Francesco Bassi, Lorenzo Botti, A. Colombo, D. A. Di Pietro, and P. Tesini. On the flexibility of agglomeration based physical space discontinuous Galerkin discretizations. *Journal of Computational Physics*, 231(1):45–65, 2012.

[19] Hojun You and Chongam Kim. Direct reconstruction method for discontinuous Galerkin methods on higher-order mixed-curved meshes I. Volume integration. *Journal of Computational Physics*, 395:223–246, 2019.

[20] Hojun You and Chongam Kim. Direct reconstruction method for discontinuous Galerkin methods on higher-order mixed-curved meshes II. Surface integration. *Journal of Computational Physics*, 416:109514, 2020.

[21] Hojun You and Chongam Kim. Direct reconstruction method for discontinuous Galerkin methods on higher-order mixed-curved meshes III. Code optimization via tensor contraction. *Computers & Fluids*, 215:104790, 2021.

[22] Jin Seok Park and Chongam Kim. Higher-order multi-dimensional limiting strategy for discontinuous Galerkin methods in compressible inviscid and viscous flows. *Computers & Fluids*, 96:377–396, 2014.

[23] Jin Seok Park and Chongam Kim. Hierarchical multi-dimensional limiting strategy for correction procedure via reconstruction. *Journal of Computational Physics*, 308:57–80, 2016.

[24] Jin Seok Park, Hojun You, and Chongam Kim. Higher-order multi-dimensional limiting process for DG and FR/CPR methods on tetrahedral meshes. *Computers & Fluids*, 154:322–334, 2017.

[25] Hojun You and Chongam Kim. High-order multi-dimensional limiting strategy with subcell resolution I. Two-dimensional mixed meshes. *Journal of Computational Physics*, 375:1005–1032, 2018.

[26] Gregor J. Gassner, Frieder Lörcher, Claus-Dieter Munz, and Jan S. Hesthaven. Polymorphic nodal elements and their application in discontinuous Galerkin methods. *Journal of Computational Physics*, 228(5):1573–1590, 2009.

[27] Francesco Bassi, Andrea Crivellini, Stefano Rebay, and Marco Savini. Discontinuous Galerkin solution of the Reynolds-averaged Navier–Stokes and k–ω turbulence model equations. *Computers & Fluids*, 34(4–5):507–540, 2005.

[28] Chi-Wang Shu and Stanley Osher. Efficient implementation of essentially non-oscillatory shock-capturing schemes. *Journal of Computational Physics*, 77(2):439–471, 1988.

[29] Sigal Gottlieb, Chi-Wang Shu, and Eitan Tadmor. Strong stability-preserving high-order time discretization methods. *SIAM Review*, 43(1):89–112, 2001.

[30] Francesco Bassi, A. Colombo, C. De Bartolo, N. Franchina, A. Ghidoni, and A. Nigro. Investigation of high-order temporal schemes for the discontinuous Galerkin solution of the Navier–Stokes equations. *Proceedings of the WCCM XI, ECCM V, ECFD VI*, 2014.

[31] Juhyun Kim, Hojun You, and Chongam Kim. On implicit temporal schemes for discontinuous Galerkin methods on curved-mixed mesh. *Proceedings of Korean Society for Aeronautical & Space Sciences 2018 Fall Conference*, 2018.

[32] Francesco Bassi, Lorenzo Botti, A. Colombo, A. Ghidoni, and F. Massa. Linearly implicit Rosenbrock-type Runge–Kutta schemes applied to the Discontinuous Galerkin solution of compressible and incompressible unsteady flows. *Computers & Fluids*, 118:305–320, 2015.

[33] Hojun You. *High-performance higher-order numerical algorithms for multi-physical flow simulations*. PhD thesis, Seoul National University, 2022.
[34] Per-Olof Persson and Jaime Peraire. Sub-cell shock capturing for discontinuous Galerkin methods. *44th AIAA Aerospace Sciences Meeting and Exhibit*, pages 1–13, 2006.
[35] M. L. Yu, F. X. Giraldo, M. Peng, and Z. J. Wang. Localized artificial viscosity stabilization of discontinuous Galerkin methods for nonhydrostatic mesoscale atmospheric modeling. *Monthly Weather Review*, 143(12):4823–4845, 2015.
[36] P. Fernandez, N. C. Nguyen, X. Roca, and J. Peraire. Implicit large-eddy simulation of compressible flows using the interior embedded discontinuous Galerkin method. *54th AIAA Aerospace Sciences Meeting*, pages 1–15, 2016.
[37] Juhyun Kim, Hojun You, and Chongam Kim. Shock-capturing model using PID controller for high-order discontinuous Galerkin method. *AIAA Scitech Forum*, pages 1–16, 2023.
[38] S. Srinivasan, J. C. Tannehill, and K. J. Weilmuenster. Simplified curve fits for the thermodynamic properties of equilibrium air. *NASA Reference Publication*, 1181, 1987.
[39] S. Srinivasan and J. C. Tannehill. Simplified curve fits for the transport properties of equilibrium air. *NASA Contractor Report*, 178411, 1987.
[40] Roop N. Gupta, Kam-Pui Lee, Richard A. Thompson, and Jerrold M. Yos. Calculations and curve fits of thermodynamic and transport properties for equilibrium air to 30000 K. *NASA Reference Publication*, 1260, 1991.
[41] Hojun You, Juhyun Kim, Kyeol Yune, and Chongam Kim. IDEA: Artificial neural network models for 11-species air properties at thermochemical equilibrium. *Manuscript in preparation*, 2022.
[42] Juhyun Kim, Hojun You, Kyeol Yune, and Chongam Kim. ANN-based air property models up to 25,000 K for hypersonic equilibrium flow simulations. *AIAA Aviation Forum*, pages 1–26, 2022.
[43] Kenneth G. Powell, Philip L. Roe, Timur J. Linde, Tamas I. Gombosi, and Darren L. De Zeeuw. A solution-adaptive upwind scheme for ideal magnetohydrodynamics. *Journal of Computational Physics*, 154(2):284–309, 1999.
[44] Dominik Derigs, Andrew R. Winters, Gregor J. Gassner, Stefanie Walch, and Marvin Bohm. Ideal GLM-MHD: About the entropy consistent nine-wave magnetic field divergence diminishing ideal magnetohydrodynamics equations. *Journal of Computational Physics*, 364:420–467, 2018.
[45] Germain Rousseaux. Forty years of Galilean electromagnetism (1973–2013). *The European Physical Journal Plus*, 128:81, 2013.
[46] S. K. Godunov. Symmetric form of the equations of magnetohydrohynamics. *Numerical Methods for Mechanics of Continuum Medium*, 1:26–34, 1972.
[47] C. D. Munz, P. Omnes, R. Schneider, E. Sonnendrücker, and U. Voß. Divergence correction techniques for Maxwell solvers based on a hyperbolic model. *Journal of Computational Physics*, 161(2):484–511, 2000.
[48] A Dedner, F. Kemm, D. Kröner, C. D. Munz, T. Schnitzer, and M. Wesenberg. Hyperbolic divergence cleaning for the MHD equations. *Journal of Computational Physics*, 175(2):645–673, 2002.
[49] C. Ciucă, P. Fernandez, A. Christophe, N. C. Nguyen, and Jamie Peraire. Implicit hybridized discontinuous Galerkin methods for compressible magnetohydrodynamics. *Journal of Computational Physics: X*, 5:100042, 2020.
[50] Hojun You, Juhyun Kim, and Chongam Kim. High-performance discontinuous Galerkin flow solver using direct reconstruction method. *AIAA Aviation 2021 Forum*, pages 1–21, 2021.
[51] Audrey Rault, Guillaume Chiavassa, and Rosa Donat. Shock-vortex interactions at high Mach numbers. *Journal of Scientific Computing*, 19:347–371, 2003.
[52] Zhiliang Xu, Yingjie Liu, and Chi-Wang Shu. Hierarchical reconstruction for discontinuous Galerkin methods on unstructured grids with a WENO-type linear reconstruction and partial neighboring cells. *Journal of Computational Physics*, 228(6):2194–2212, 2009.
[53] Professor H. Schardin. High frequency cinematography in the shock tube. *The Journal of Photographic Science*, 5(2):17–19, 1957.
[54] Se-Myong Chang and Keun-Shik Chang. On the shock–vortex interaction in Schardin's problem. *Shock Waves*, 10:333–343, 2000.
[55] C. Carton de Wiart and K. Hillewaert. DNS and ILES of transitional flows around a SD7003 using a high order discontinuous Galerkin method. *7th International Conference on Computational Fluid*

Dynamics (ICCFD7), pages 1–14, 2012.

[56] Marshall Galbraith and Miguel Visbal. Implicit large eddy simulation of low-Reynolds-number transitional flow past the SD7003 airfoil. *40th Fluid Dynamics Conference and Exhibit*, pages 1–32, 2010.

[57] A. Uranga, P.-O. Persson, M. Drela, and J. Peraire. Implicit large eddy simulation of transition to turbulence at low Reynolds numbers using a discontinuous Galerkin method. *International Journal for Numerical Methods in Engineering*, 87(1–5):232–261, 2010.

[58] P. Catalano and R. Tognaccini. RANS analysis of the low-Reynolds number flow around the SD7003 airfoil. *Aerospace Science and Technology*, 15(8):615–626, 2011.

[59] David Degani. Effect of geometrical disturbance on vortex asymmetry. *AIAA Journal*, 29(4):560–566, 1991.

[60] K. Petterson and D. I. A. Poll. Slender forebody aerodynamics at high alpha. *International Council of the Aeronautical Sciences (ICAS) 2000 Congress*, pages 294.1–294.9, 2000.

[61] Russell M. Cummings, James R. Forsythe, Scott A. Morton, and kyle D. Squires. Computational challenges in high angle of attack flow prediction. *Progress in Aerospace Sciences*, 39(5):369–384, 2003.

[62] Patrick Champigny, Sébastien Deck, Pascal Denis, and Sonia Magniant. Numerical simulation of forebody vortices at high angle of attack and their control using innovative systems. *Innovative Missile Systems*, RTO-MP-AVT-135:24–1–24–16, 2006.

[63] Sang Ho Kim and Wansub Kim. Asymmetric vortices around a body at high angle of attack supersonic flow. *Journal of the Korean Physical Society*, 55(5):2159–2165, 2009.

[64] Earl R. Keener, Gary T. Chapman, Lee Cohen, and Jamshid Taleghani. Side forces on a tangent ogive forebody with a fineness ratio of 3.5 at high angles of attack and Mach numbers from 0.1 to 0.7. *NASA Technical Memorandum*, NASA TM X-3437, 1977.

[65] P. J. Lamont. The complex asymmetric flow over a 3.5D ogive nose and cylindrical afterbody at high angles of attack. *20th Aerospace Sciences Meeting*, pages 1–10, 1982.

[66] Ramadas K. Prabhu, James R. Stewart, and Rajiv R. Thareja. A Navier–Stokes solver for high speed equilibrium flows and application to blunt bodies. *27th Aerospae Sciences Meeting*, pages 1–10, 1989.

[67] Datta Gaitonde and J. S. Shang. The performance of flux-split algorithms in high-speed viscous flows. *30th Aerospae Sciences Meeting*, pages 1–20, 1992.

[68] M. S. Holden, A. R. Wieting, J. R. Moselle, and C. Glass. Studies of aerothermal loads generated in regions of shock/shock interaction in hypersonic flow. *26th Aerospace Sciences Meeting*, pages 1–5, 1988.

[69] Changsoo Lee, Eunsa Kim, Yune Kyeol, and Chongam Kim. ACTFlow: A target-oriented finite volume solver for all-speed compressible turbulent flow simulations. *AIAA Scitech 2021 Forum*, pages 1–34, 2021.

[70] Kazem Hejranfar, Ramin Kamali Moghadam, and Vahid Esfahanian. Dual-code solution procedure for efficient computing equilibrium hypersonic axisymmetric laminar flows. *Aerospace Science and Technology*, 12(2):135–149, 2008.

[71] Y. C. Vigneron. Hypersonic viscous flow of equilibrium air around a blunt body. Master's thesis, Iowa State University, 1976.

[72] Steven A. Orszag and Cha-Mei Tang. Small-scale structure of two-dimensional magnetohydrodynamic turbulence. *Journal of Fluid Mechanics*, 90(1):129–143, 1979.

[73] James M. Stone, Thomas A. Gardiner, Peter Teuben, John F. Hawley, and Jacob B. Simon. Athena: A new code for astrophysical MHD. *The Astrophysical Journal*, 178(1):137–177, 2008.

[74] Dinshaw S. Balsara and Daniel S. Spicer. A staggered mesh algorithm using high order Godunov fluxes to ensure solenoidal magnetic fields in magnetohydrodynamic simulations. *Journal of Computational Physics*, 149(2):270–292, 1999.

[75] Telemachos Ch. Mouschovias and Efthimios V. Paleologou. Magnetic braking of an aligned rotator during star formation: An exact, time-dependent solution. *The Astrophysical Journal*, 237:877–899, 1980.

[76] Sergey Yakovlev, Liwei Xu, and Fengyan Li. Locally divergence-free central discontinuous Galerkin methods for ideal MHD equations. *Journal of Computational Science*, 4(1–2):80–91, 2013.