

Article

High-Order CFD Solvers on Three-Dimensional Unstructured Meshes: Parallel Implementation of RKDG Method with WENO Limiter and Momentum Sources

Weicheng Pei ¹, Yuyan Jiang ¹ and Shu Li ^{1,2,*}

¹ School of Aeronautics Science and Engineering, Beihang University, 37 Xueyuan Road, Haidian District, Beijing 100191, China; weicheng.pei@icloud.com (W.P.); jiang.yu.yan@icloud.com (Y.J.)

² Ningbo Institute of Technology, Beihang University, Ningbo 315800, China

* Correspondence: lishu@buaa.edu.cn

Abstract: In aerospace engineering, high-order computational fluid dynamics (CFD) solvers suitable for three-dimensional unstructured meshes are less developed than expected. The Runge–Kutta discontinuous Galerkin (RKDG) finite element method with compact weighted essentially non-oscillatory (WENO) limiters is one of the candidates, which might give high-order solutions on unstructured meshes. In this article, we provide an efficient parallel implementation of this method for simulating inviscid compressible flows. The implemented solvers are tested on lower-dimensional model problems and real three-dimensional engineering problems. Results of lower-dimensional problems validate the correctness and accuracy of these solvers. The capability of capturing complex flow structures even on coarse meshes is shown in the results of three-dimensional applications. For solving problems containing rotary wings, an unsteady momentum source model is incorporated into the solvers. Such a finite element/momentum source hybrid method eliminates the reliance on advanced mesh techniques, which might provide an efficient tool for studying rotorcraft aerodynamics.



Citation: Pei, W.; Jiang, Y.; Li, S. High-Order CFD Solvers on Three-Dimensional Unstructured Meshes: Parallel Implementation of RKDG Method with WENO Limiter and Momentum Sources. *Aerospace* **2022**, *9*, 372. <https://doi.org/10.3390/aerospace9070372>

Academic Editor: Gokhan Inalhan

Received: 9 June 2022

Accepted: 7 July 2022

Published: 11 July 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: high-order CFD solvers; three-dimensional unstructured meshes; RKDG method; compact WENO limiter; parallel implementation; momentum source models

1. Introduction

Computational fluid dynamics (CFD) solvers, widely used in aerospace engineering, are based either on finite difference (FD) schemes or on finite volume (FV) schemes. However, FD schemes are restricted on structured meshes, while most FV schemes have only first- or second-order accuracy. One way to develop higher-order schemes suitable for unstructured meshes is to use the discontinuous Galerkin (DG) finite element (FE) methods. As with other FE methods, DG methods assume continuous approximation in each element (i.e., cell). However, they allow discontinuities to exist on intercell boundaries. Such discontinuities are then resolved by numerical fluxes, e.g., exact or approximate Riemann solvers, as with Godunov-type schemes [1]. The original DG scheme was not proposed for CFD but for solving the linear neutron transport equation [2]. For nonlinear conservation laws, Chavent and Salzano [3] designed an explicit Euler DG scheme, which is second-order accurate in space but only first-order accurate in time. To improve the accuracy of time discretization, Cockburn and Shu [4] replaced the first-order forward Euler method with high-order Runge–Kutta (RK) methods, which led to the family of Runge–Kutta discontinuous Galerkin (RKDG) schemes for scalar conservation laws [5,6], for one-dimensional conservation law systems [7], for multi-dimensional scalar conservation laws [8] and for multi-dimensional conservation law systems [9]. The last case includes the Euler system that depicts inviscid compressible flows, which is the first application of RKDG in CFD. We will give a concise derivation of this scheme in Sections 2.1 and 2.2.

The Euler system is hyperbolic, and might have discontinuities in its solutions. Approximating such solutions with finite terms of continuous functions would generate numerical oscillations near discontinuities (known as the Gibbs phenomenon [10]). Therefore, any useful high-order scheme must incorporate some mechanism for suppressing such spurious oscillations. For second-order FD and FV schemes, there are slope and flux limiters [11]. These limiters work well on shock capturing but might reduce the order of accuracy when used in smooth regions. Furthermore, they are not applicable to higher-order schemes. To overcome these drawbacks, the essentially non-oscillatory (ENO) [12] and weighted ENO (WENO) [13] limiters for high-order FD schemes, which can essentially suppress spurious oscillations and maintain high-order accuracy in smooth regions, were developed. Similar ideas were later incorporated into FV [14] and DG [15,16] schemes, which are suitable for unstructured meshes. However, such early versions of unstructured WENO limiters have to query not only a cell's immediate neighbors, but also the neighbors' neighbors, and so on. To preserve the compactness of RKDG schemes, Zhong [17] and Zhu [18] proposed a compact WENO limiter for two-dimensional structured and unstructured meshes. In our early work [19], we extended it to three-dimensional and presented an efficient parallel implementation. We will summarize the key steps of this compact WENO limiter in Section 2.3.

The parallel implementation of the RKDG scheme with a compact WENO limiter can be used for solving many practical problems in aerospace engineering. However, for problems with moving boundaries (such as rotary wings on a helicopter or blades on a wind turbine), advanced meshing techniques (such as dynamic meshing, overset meshes or adaptive mesh refinement) have to be applied. On the other hand, typical lifting bodies (such as rotary wings) have been successfully modeled as various momentum sources in FD/FV schemes. Rajagopalan and Mathur [20,21] proposed the first FD/FV-based momentum source model, which models the time-averaged effect of a rotor as an actuator disk. Such a steady model was used by Kang and Sun [22,23] for solving the flow fields of rotors in ground effect. Shi [24] compared this model with an unsteady overset mesh method, and found that the steady momentum source model can predict complex aerodynamic interactions between a rotor and a flight deck at a much cheaper cost. Shen [25] and Kim [26] refined the model by replacing the actuator disk for the entire rotor with an actuator surface for each blade of the rotor, so that unsteady effects of rotary wings can be captured. Such early works are either oversimplified by ignoring unsteady effects (such as [20–23]), or restricted to structured meshes (such as [24,25]). To overcome these drawbacks, our parallel RKDG solvers are augmented with an unsteady actuator line model, which successfully captures unsteady effects of a rotor on unstructured meshes. The formulation of this finite element/momentum source hybrid method will be given in Section 2.4. Some demonstrative results are presented in Section 3.3.

2. The RKDG Method with WENO Limiters and Momentum Sources

Mathematically, the problems that we are going to solve in Section 3 are defined by the three-dimensional Euler system

$$\partial_t \begin{bmatrix} \rho \\ \rho u_x \\ \rho u_y \\ \rho u_z \\ \rho e_0 \end{bmatrix} + \partial_x \begin{bmatrix} \rho u_x \\ \rho u_x u_x + p \\ \rho u_y u_x \\ \rho u_z u_x \\ \rho h_0 u_x \end{bmatrix} + \partial_y \begin{bmatrix} \rho u_y \\ \rho u_x u_y \\ \rho u_y u_y + p \\ \rho u_z u_y \\ \rho h_0 u_y \end{bmatrix} + \partial_z \begin{bmatrix} \rho u_z \\ \rho u_x u_z \\ \rho u_y u_z \\ \rho u_z u_z + p \\ \rho h_0 u_z \end{bmatrix} = \begin{bmatrix} 0 \\ b_x \\ b_y \\ b_z \\ 0 \end{bmatrix} \quad (1)$$

with certain boundary and initial conditions, which describes inviscid compressible flows. In Equation (1),

- Variables in the first column are called conservative variables;
- The density ρ , the velocity components u_x, u_y, u_z and the pressure p are called primitive variables;

- The thermodynamic quantities e_0 and h_0 are the specific total energy and the specific total enthalpy, respectively; for ideal gases, they can be written as functions of the five primitive variables:

$$e_0 = \frac{p/\rho}{\gamma - 1} + \frac{u_x^2 + u_y^2 + u_z^2}{2}, \quad h_0 = e_0 + \frac{p}{\rho},$$

in which γ is the heat capacity ratio of the gas, and we use $\gamma = 1.4$ in this article;

- The body force components b_x, b_y, b_z are assumed to be 0s, except in Section 3.3.

The method that we use for solving the Euler system (1) is the RKDG method with a WENO limiter. A detailed formulation of this method for sourceless problems has been given in our previous work [19]. In this section, we give a more complete formulation, which takes source terms into consideration. If all source terms were canceled, it would reduce to the form in [19].

2.1. The DG Space Discretization

Equation (1) can be written as

$$\partial_t \underline{U} + \nabla \cdot \underline{\vec{F}} = \underline{Q}, \tag{2}$$

in which $\underline{\vec{F}} = \underline{F}^x \vec{e}_x + \underline{F}^y \vec{e}_y + \underline{F}^z \vec{e}_z$ is the flux vector whose components F^x, F^y, F^z , and $\underline{U}, \underline{Q}$ are all 5×1 matrices, each row of which is a scalar function depending on position \vec{x} and time t . By multiplying both sides with an arbitrary test function $\psi(\vec{x})$ and integrating the products on the i -th element (i.e., cell) E_i , and applying integral by parts and Gauss's divergence theorem, Equation (2) can be turned into a weak form:

$$\int_{E_i} (\psi \partial_t \underline{U} - \underline{\vec{F}} \cdot \nabla \psi) + \int_{\partial E_i} (\vec{v} \cdot \underline{\vec{F}}) \psi = \int_{E_i} \underline{Q} \psi, \tag{3}$$

where \vec{v} is the outer normal unit vector of ∂E_i (which is the boundary of E).

By choosing an orthonormal basis of the linear space spanned by polynomials less than the p -th degree over E , denoted as $\underline{\phi}(\vec{x}) := [\phi_1(\vec{x}) \ \cdots \ \phi_L(\vec{x})]$, the unknowns \underline{U} and the arbitrary test function ψ can be approximated as

$$\underline{U}(\vec{x}, t) \approx \underline{U}^h(\vec{x}, t) = \sum_{l=1}^L \hat{U}_l(t) \phi_l(\vec{x}), \quad \psi(\vec{x}) \approx \psi^h(\vec{x}) = \sum_{l=1}^L \hat{\psi}_l \phi_l(\vec{x}),$$

in which each $\hat{U}_l(t)$ is a 5×1 matrix of temporal functions, and each $\hat{\psi}_l$ is an arbitrary constant number. Substituting them into the weak form (Equation 2) gives

$$\sum_l \hat{\psi}_l \left[\sum_k \left(\int_{E_i} \phi_l \phi_k \right) \frac{d\hat{U}_k}{dt} + \int_{E_i} (\underline{\vec{F}} \cdot \nabla \phi_l - \underline{Q} \phi_l) + \oint_{\partial E_i} \phi_l \underline{F}^v \right] = \underline{O}, \tag{4}$$

where $\vec{v} \cdot \underline{\vec{F}} =: \underline{F}^v$ is the normal flux at some point on ∂E_i , whose value could be given by an exact or approximate Riemann solver of Equation (2); see [27] for details.

Recall the arbitrariness of $\{\hat{\psi}_l\}_{l=1}^L$ and adopt the inner product notation

$$\langle f | g \rangle := \int_{E_i} f(\vec{x}) g(\vec{x}),$$

and we could turn Equation (4) into a system of ordinary differential equations:

$$\frac{d\hat{U}}{dt} = \underline{R}(\hat{U}), \tag{5}$$

in which

$$\hat{U}(t) = \begin{bmatrix} \langle U_1 | \phi_1 \rangle & \cdots & \langle U_1 | \phi_L \rangle \\ \vdots & \ddots & \vdots \\ \langle U_5 | \phi_1 \rangle & \cdots & \langle U_5 | \phi_L \rangle \end{bmatrix}$$

is a $5 \times L$ matrix of temporal functions (which will be solved in Section 2.2), and

$$\underline{R} = \int_{E_i} \left(\begin{bmatrix} \underline{F}^x & \underline{F}^y & \underline{F}^z \end{bmatrix} \begin{bmatrix} \partial_x \underline{\phi} \\ \partial_y \underline{\phi} \\ \partial_z \underline{\phi} \end{bmatrix} - \underline{Q} \underline{\phi} \right) - \oint_{\partial E_i} \underline{F}^v \underline{\phi} \tag{6}$$

is a variable matrix depending on \hat{U} . The integrals in Equation (6) are evaluated by Gaussian quadrature rules, such as those given by [28].

2.2. The RK Time Discretization

Equation (5) is a typical system of nonlinear ordinary differential equations. In this article, we use the following explicit Runge–Kutta methods:

first-order:

$$\hat{U}^{n+1} = \hat{U}^n + \underline{R}^n \Delta t;$$

second-order:

$$\begin{aligned} \hat{U}^{n+1/2} &= \hat{U}^n + \underline{R}^n \Delta t, \\ \hat{U}^{n+1} &\equiv \hat{U}^{n+2/2} = \frac{1}{2} \hat{U}^n + \frac{1}{2} \left(\hat{U}^{n+1/2} + \underline{R}^{n+1/2} \Delta t \right); \end{aligned}$$

third-order:

$$\begin{aligned} \hat{U}^{n+1/3} &= \hat{U}^n + \underline{R}^n \Delta t, \\ \hat{U}^{n+2/3} &= \frac{3}{4} \hat{U}^n + \frac{1}{4} \left(\hat{U}^{n+1/3} + \underline{R}^{n+1/3} \Delta t \right), \\ \hat{U}^{n+1} &\equiv \hat{U}^{n+3/3} = \frac{1}{3} \hat{U}^n + \frac{2}{3} \left(\hat{U}^{n+2/3} + \underline{R}^{n+2/3} \Delta t \right); \end{aligned} \tag{7}$$

in which the $\underline{R}^\mu := \underline{R}(\hat{U}^\mu)$ s are the values mapped by the nonlinear operator \underline{R} from the corresponding \hat{U}^μ s.

It has been proven by Gottlieb and Shu that they are all total variation diminishing (TVD) [29] or strong stability preserving (SSP) [30] (which is a desired feature for hyperbolic problems), and fourth-order Runge–Kutta methods cannot be TVD or SSP without introducing the adjoint operator of \underline{R} , denoted as \underline{R}^\dagger , which satisfies

$$\hat{U}^{n+1} = \hat{U}^n - \underline{R}^\dagger(\hat{U}^n) \Delta t.$$

It is non-trivial to implement such an adjoint operator and thus we do not implement Runge–Kutta methods whose accuracy order is higher than three.

2.3. The Compact WENO Limiter

Following our previous work [19], we use the three-dimensional extension of the two-dimensional WENO limiters by Zhong [17] and Zhu [18]. For simplicity, we denote the index set of E_i 's neighbors as K_i . For each $k \in K_i$, we first transform Equation (2) to the ν -split form on the interface shared by E_i and its neighbor E_k :

$$\partial_t \underline{U} + \partial_\nu \underline{F}^\nu \equiv (\partial_t + \underline{A}^\nu \partial_\nu) \underline{U} = \underline{Q}, \tag{8}$$

in which the flux Jacobian can be approximated by the average value of \underline{U} on E_i :

$$\underline{A}^\nu \equiv \frac{\partial \underline{F}^\nu}{\partial \underline{U}} \approx \frac{\partial \underline{F}^\nu}{\partial \underline{U}} \Big|_{\langle \underline{U} \rangle_i}. \tag{9}$$

From the hyperbolicity of Equation (2), there always exists the eigenvalue decomposition

$$\underline{A}^v = \underline{R}^v \begin{bmatrix} u_v - a & & & & \\ & u_v & & & \\ & & u_v & & \\ & & & u_v & \\ & & & & u_v + a \end{bmatrix} (\underline{R}^v)^{-1}, \tag{10}$$

in which

$$\underline{R}^v = \begin{bmatrix} 1 & 1 & 0 & 0 & 1 \\ u_x - av_x & u_x & \sigma_x & \pi_x & u_x + av_x \\ u_y - av_y & u_y & \sigma_y & \pi_y & u_y + av_y \\ u_z - av_z & u_z & \sigma_z & \pi_z & u_z + av_z \\ h_0 - u_v a & \frac{u_x^2 + u_y^2 + u_z^2}{2} & u_\sigma & u_\pi & h_0 + u_v a \end{bmatrix}, \quad \begin{bmatrix} u_v \\ u_\sigma \\ u_\pi \end{bmatrix} = \begin{bmatrix} v_x & \sigma_x & \pi_x \\ v_y & \sigma_y & \pi_y \\ v_z & \sigma_z & \pi_z \end{bmatrix}^{-1} \begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix},$$

$$(\underline{R}^v)^{-1} = \begin{bmatrix} \frac{1}{2}(B_2 + \frac{u_v}{a}) & -\frac{1}{2}(B_1 u_x + \frac{v_x}{a}) & -\frac{1}{2}(B_1 u_y + \frac{v_y}{a}) & -\frac{1}{2}(B_1 u_z + \frac{v_z}{a}) & \frac{1}{2}B_1 \\ 1 - B_2 & B_1 u_x & B_1 u_y & B_1 u_z & -B_1 \\ -u_\sigma & \sigma_x & \sigma_y & \sigma_z & 0 \\ -u_\pi & \pi_x & \pi_y & \pi_z & 0 \\ \frac{1}{2}(B_2 - \frac{u_v}{a}) & -\frac{1}{2}(B_1 u_x - \frac{v_x}{a}) & -\frac{1}{2}(B_1 u_y - \frac{v_y}{a}) & -\frac{1}{2}(B_1 u_z - \frac{v_z}{a}) & \frac{1}{2}B_1 \end{bmatrix},$$

where $B_1 := (\gamma - 1)/a^2$ and $B_2 := B_1(u_v^2 + u_\sigma^2 + u_\pi^2)$. The original conservative variable \underline{U} is then projected into the space spanned by the columns of \underline{R} , which defines the characteristic variable:

$$\underline{V} := (\underline{R}^v)^{-1} \underline{U}.$$

Each scalar component of \underline{V} can now be treated as independent functions, to which any suitable scalar WENO limiter (such as the one in [17,18]) can be applied. Once obtaining the reconstructed characteristic variable \underline{V}^{new} , it is turned back to the original conservative variable:

$$\underline{U}_{i|k}^{new} := \underline{R}^v \underline{V}^{new}, \tag{11}$$

The subscript $i|k$ emphasizes the fact that it is a function defined on E_i with the help of E_k . There is one such reconstructed $\underline{U}_{i|k}^{new}$ for each $k \in K_i$, so the final step is to weight them by the volume of the corresponding adjacent element:

$$\underline{U}_i^{new} := \frac{\sum_{k \in K_i} \underline{U}_{i|k}^{new} |E_k|}{\sum_{k \in K_i} |E_k|}. \tag{12}$$

2.4. The Momentum Source Model

To model a fixed or rotary wing with a high aspect ratio, we use the actuator line model, which is a special case of the more general momentum source model. In this model, a wing is sliced into a series of thin pieces, each of which is perpendicular to the wing’s axis. When moving relative to the air, a piece feels a pair of aerodynamic forces:

$$\begin{bmatrix} L \, ds \\ D \, ds \end{bmatrix} = \frac{1}{2} \rho (u^2 + w^2) \begin{bmatrix} C_L(\alpha) \\ C_D(\alpha) \end{bmatrix} c \, ds,$$

in which

- c is the chord length of the piece, and s is the arc length parameter of the axis;
- α is the angle of attack, which is related to u and w (velocity components resolved in the airfoil’s local frame) by $\tan \alpha = w/u$;
- C_L and C_D are the lift and drag coefficients of the airfoil, respectively;
- L and D are the lift and drag per unit length, respectively.

Using Newton’s third Law, the force per unit length felt by the air surrounding the piece is

$$\vec{b}_L(\vec{r}(s)) = -[\vec{e}_L(s) \quad \vec{e}_D(s)] \begin{bmatrix} L(s) \\ D(s) \end{bmatrix},$$

in which \vec{e}_L and \vec{e}_D are unit vectors along and perpendicular to the direction of airflow relative to the airfoil. To avoid ambiguity, we denote the force per unit length as \vec{b}_L and use \vec{b}_V to denote the force per unit volume. If the intersection of a wing’s axis \overline{PQ} and a DG element E is the line segment \overline{RT} , then the body integral of the source term in Equation (6) is actually a line integral:

$$\int_{E_i} \underline{Q} \underline{\phi} = \int_{E_i} \begin{bmatrix} 0 \\ \vec{b}_V(\vec{r}) \\ 0 \end{bmatrix} \underline{\phi}(\vec{r}) = \begin{bmatrix} 0 \\ \int_{\overline{RT}} \vec{b}_L(\vec{r}) \underline{\phi}(\vec{r}) \\ 0 \end{bmatrix}, \tag{13}$$

By parameterizing the line segment \overline{RT} , the line integral in Equation (13) can be evaluated as a definite integral:

$$\int_{RT} \vec{b}_L(\vec{r}) \underline{\phi}(\vec{r}) = \int_{\xi_R}^{\xi_T} \vec{b}_L(\vec{r}(\xi)) \underline{\phi}(\vec{r}(\xi)) \left\| \frac{\partial \vec{r}}{\partial \xi} \right\| d\xi,$$

to which the standard Gaussian quadrature can be applied.

3. Results of Model Problems and Engineering Problems

In this section, we give the results of various problems to show the accuracy and performance of the methods described in Section 2.

3.1. Lower-Dimensional Model Problems

3.1.1. Shock Tube Problems

These problems are usually defined as one-dimensional problems, but we treat them as three-dimensional ones. All these problems are considered in a $[0.0, 5.0] \times [0.0, 1.0] \times [0.0, 0.5]$ box with all boundaries closed but the left and right ends open. Using the method described in [27], their exact solutions can be obtained, which can be used for assessing the accuracy of our solvers. In our earlier work [19], we have given the results obtained by running our solvers on an unstructured hexahedral mesh. This time, we use an unstructured tetrahedral mesh instead.

Problem 1 (Sod). Solve the Euler system Equation (1) for $t \in [0.0, 1.0]$ with the initial condition

$$[\rho \quad u \quad v \quad w \quad p]_{t=0} = \begin{cases} [1 \quad 0 \quad 0 \quad 0 \quad 1], & x < 2.5; \\ [\frac{1}{8} \quad 0 \quad 0 \quad 0 \quad \frac{1}{10}], & x > 2.5. \end{cases}$$

This is a classical problem of inviscid compressible flows. It contains all three types of discontinuities: a shock wave and a contact discontinuity running towards the right and an expansion wave running towards the left. In Figure 1, we plot the density contour given by our third-order solver with an EigenWeno limiter. In Figure 2, we compare the density distributions given by various solvers along the longitudinal axis (on which $y = 0.5$ and $z = 0.25$) of the box. The accuracy of our solvers and the effect of p -refinement are clear in these figures.

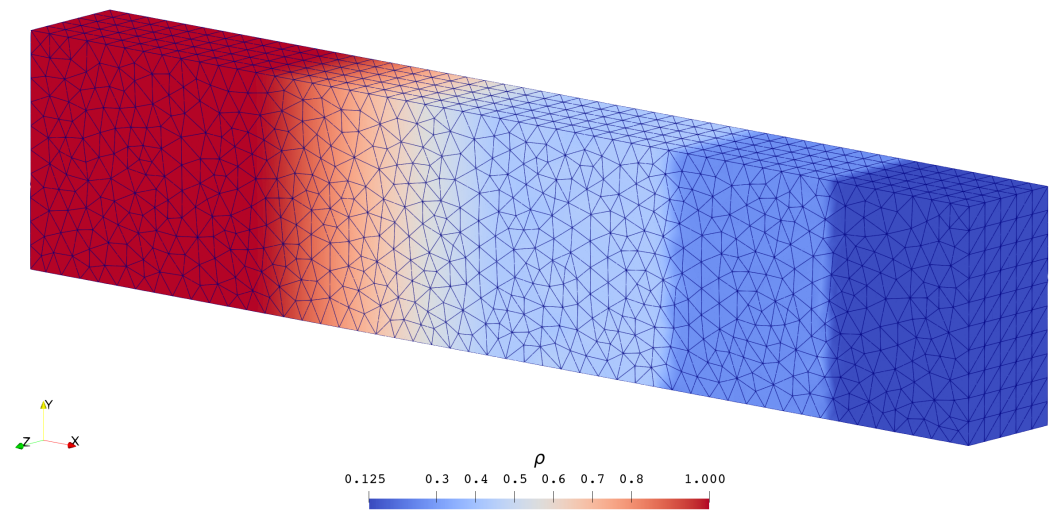


Figure 1. Third-order solution of $\rho(t = 1.0)$ in Problem 1.

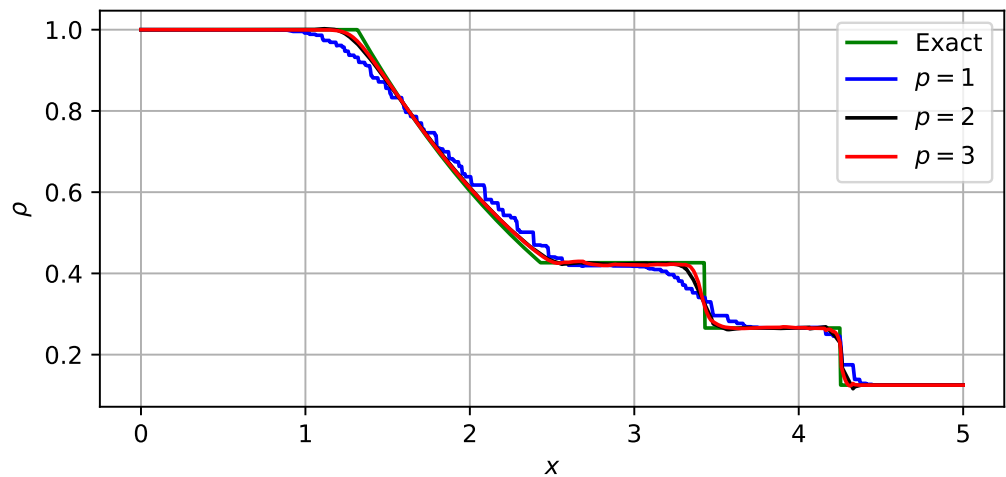


Figure 2. Comparison between solutions of $\rho(t = 1.0)$ in Problem 1.

It is predictable that both mesh refinement (decreasing h) and order increment (increasing p) can help to improve accuracy. To compare the performance of solvers with different orders more fairly, it is better to use finer meshes for running lower-order solvers. After a few trials, we find that the solutions given by the h - p pairs listed in Table 1 roughly have the same level of accuracy, as shown in Figure 3. It is clear, at least for Problem 1, that high-order solvers are better than low-order ones in the sense of obtaining the same level of accuracy with less time and space costs. Similar conclusions were drawn in our previous work [19], in which the $p = 3$ solution of a linear advection problem on an $h \approx 1/4$ mesh defeated the $p = 1$ solution of the same problem on an $h \approx 1/32$ mesh in accuracy but saved quite a lot of time. These encouraging results justify our efforts to implement higher-order solvers.

Table 1. Comparison between time costs and file sizes of various h - p pairs.

h	p	#Cells	#Steps	Time Cost	File Size
1/50	1	2469675	1500	1492.16 s	1.23 GB
1/18	2	117045	500	279.119 s	213 MB
1/10	3	19815	250	201.034 s	89.1 MB

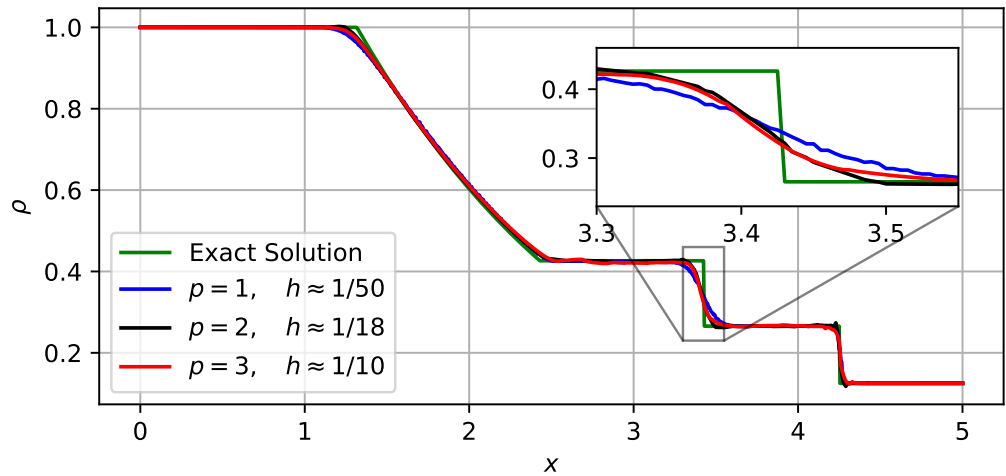


Figure 3. Comparison between solutions given by various h - p pairs.

Problem 2 (Lax). Solve the Euler system Equation (1) for $t \in [0.0, 0.6]$ with the initial condition

$$[\rho \ u \ v \ w \ p]_{t=0} = \begin{cases} [0.445 \ 0.698 \ 0.000 \ 0.000 \ 3.528], & x < 2.5; \\ [0.500 \ 0.000 \ 0.000 \ 0.000 \ 0.571], & x > 2.5. \end{cases}$$

This is another problem that involves all three types of discontinuities. It is more difficult than the previous one in the sense that its solution contains values beyond the range of initial values and the discontinuities are much steeper. As before, we plot the density contour of our third-order solution in Figure 4 and compare the density distributions given by various solvers in Figure 5. Both figures demonstrate the effect of p -refinement and the ability of our high-order solvers to suppress numerical oscillations.

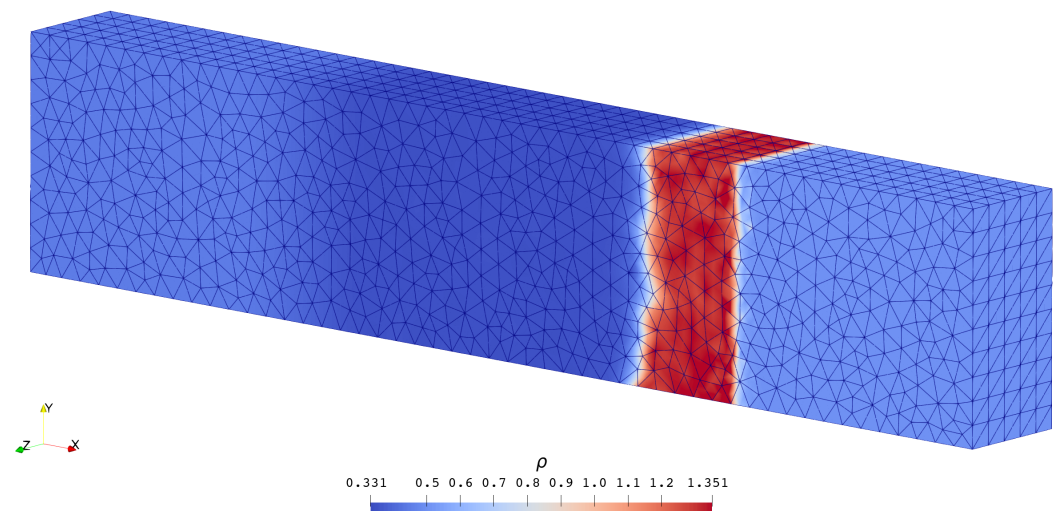


Figure 4. Third-order solution of $\rho(t = 0.5)$ in Problem 2.

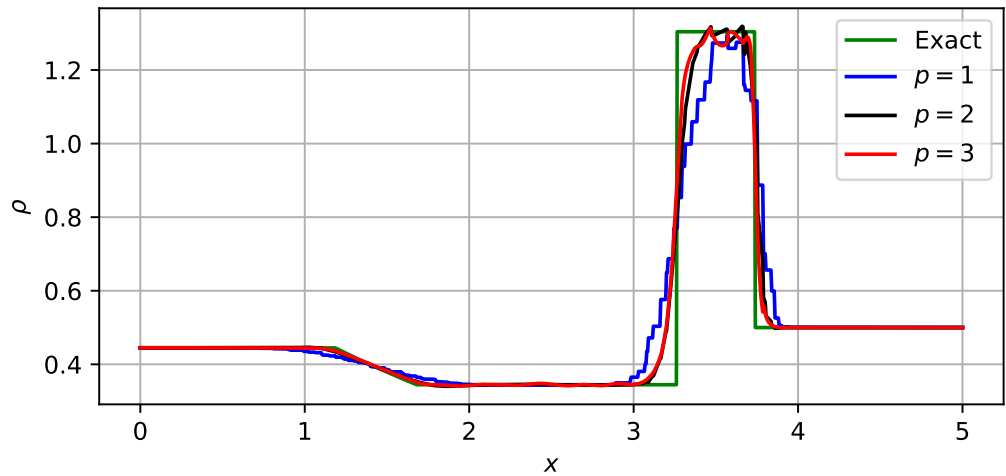


Figure 5. Comparison between solutions of $\rho(t = 0.5)$ in Problem 2.

Problem 3 (Vacuum). Solve the Euler system Equation (1) for $t \in [0.0, 0.4]$ with the initial condition

$$[\rho \ u \ v \ w \ p]_{t=0} = \begin{cases} [1 \ -4 \ 0 \ 0 \ 0.4], & x < 2.5; \\ [1 \ +4 \ 0 \ 0 \ 0.4], & x > 2.5. \end{cases}$$

This problem is easier than the previous two in the sense that no strong discontinuities (shock or contact discontinuity) occur in the solution. However, there is a region of vacuum, which does not occur in the previous problems, generated between the left- and right-running expansion waves. The velocity u inside the vacuumed region (denoted by a $*$ in subscript) is undefined, but its boundary values should be determined from the left (denoted by an L in subscript) and right (denoted by an R in subscript) initial values by the two Riemann invariants:

$$u_{*L} = u_L + \frac{2a_L}{\gamma - 1}, \quad u_{*R} = u_R - \frac{2a_R}{\gamma - 1}.$$

See [27] for more detailed discussions. Unfortunately, approximate Riemann solvers usually (have to) neglect this condition. For this reason, we use an exact Riemann solver to obtain fluxes on cell boundaries. As before, we plot the results in Figures 6 and 7, which again validate the correctness and robustness of our solvers.

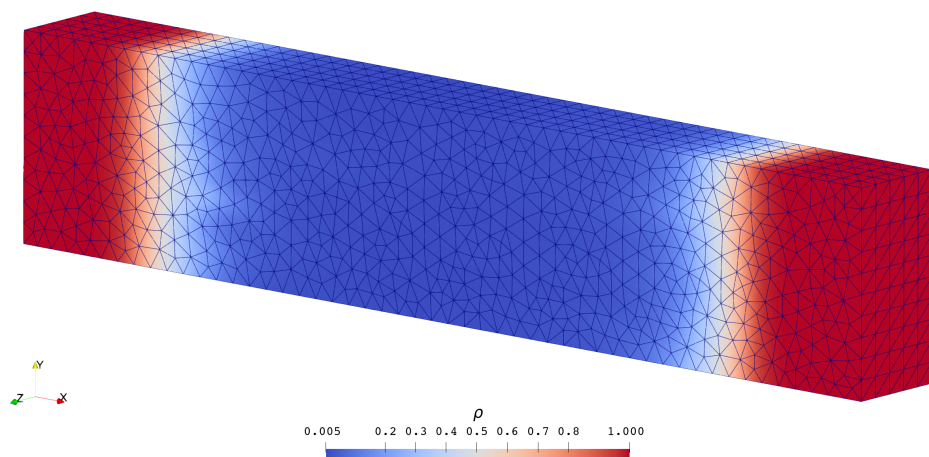


Figure 6. Third-order solution of $\rho(t = 0.4)$ in Problem 3.

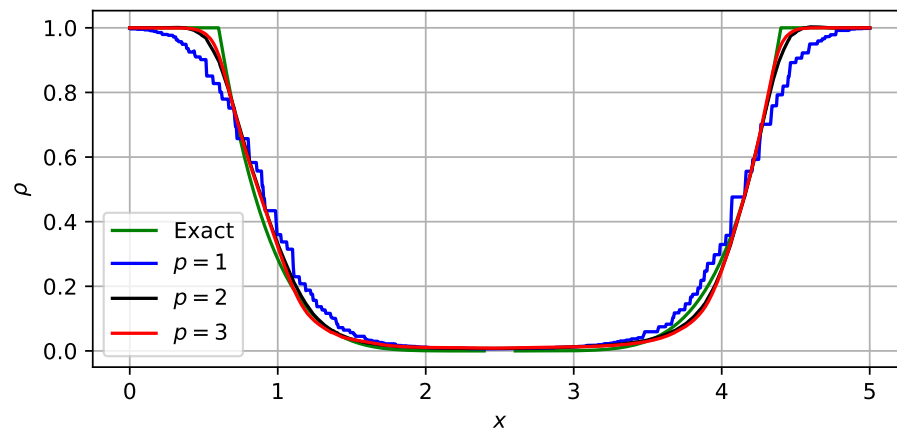


Figure 7. Comparison between solutions of $\rho(t = 0.4)$ in Problem 3.

3.1.2. The Forward Step Problem

This is a classical two-dimensional problem [31], but we treat it as a three-dimensional one:

Problem 4. Solve the Euler system (Equation 1) in a $[0, 3] \times [0, 1] \times [0, Z]$ box (representing a wind tunnel), where the thickness Z could be any positive value, with a $[0.6, 3] \times [0, 0.2] \times [0, Z]$ box removed (representing a forward-facing step). The $x = 0$ surface is open as an inlet and the $x = 3$ surface is open as an outlet. All other boundary surfaces are closed as solid walls. The initial condition is given as a uniform state:

$$[\rho \quad u \quad v \quad w \quad p]_{t=0} = [1.4 \quad 3.0 \quad 0.0 \quad 0.0 \quad 1.0].$$

To show the applicability of our solvers on structured meshes, we solve this problem on meshes such as the one in Figure 8. As a common practice, we plot the density contours at various time steps in Figures 9–19.

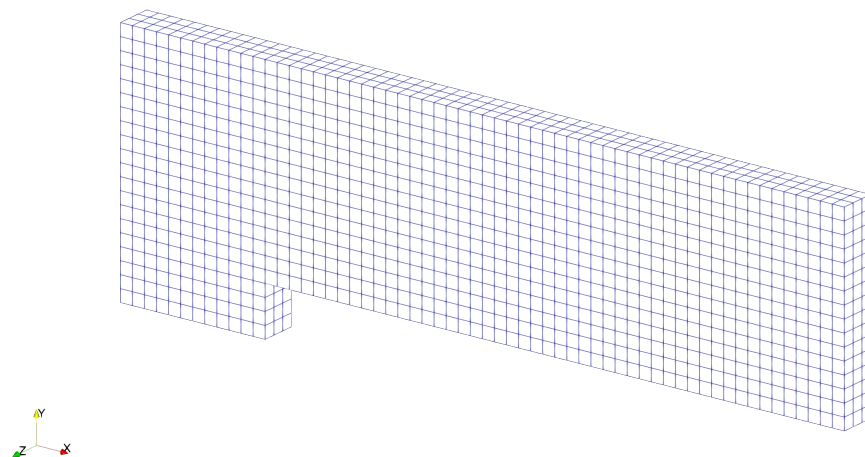


Figure 8. A coarse ($h = 1/20$) mesh for solving Problem 4. This mesh is too coarse to capture details of the flow field but clear enough to demonstrate the distribution of nodes and cells. Actually, we use a much finer ($h = 1/200$) mesh to produce Figures 9–19.

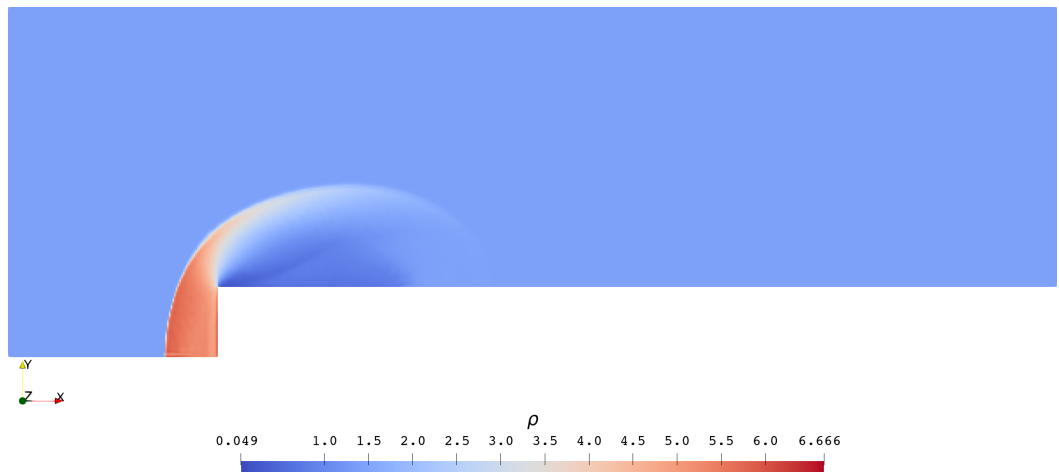


Figure 9. Third-order solution of Problem 4 at $t = 0.2$ on an $h = 1/200$ mesh.

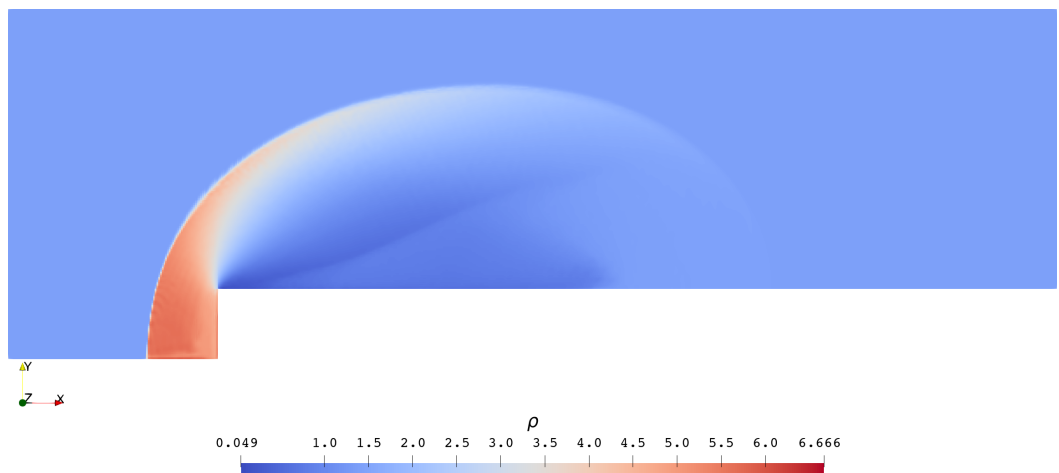


Figure 10. Third-order solution of Problem 4 at $t = 0.4$ on an $h = 1/200$ mesh.

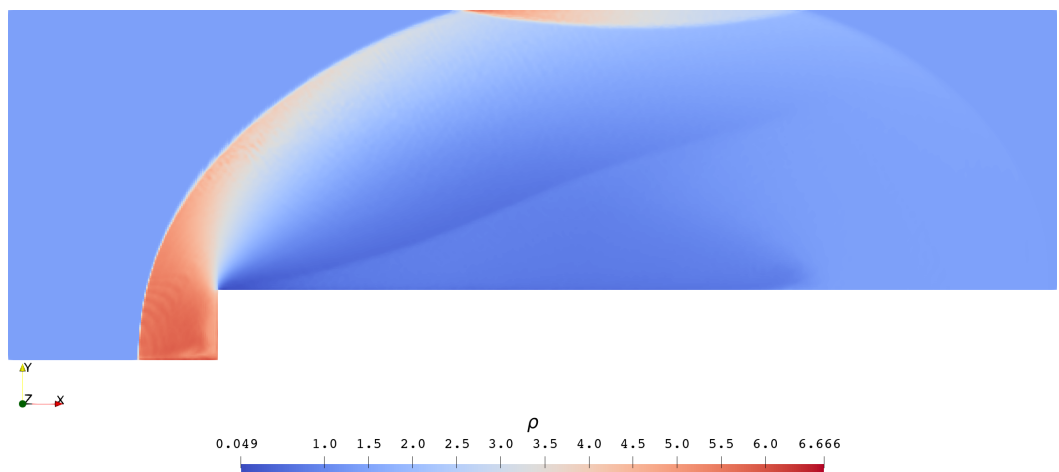


Figure 11. Third-order solution of Problem 4 at $t = 0.6$ on an $h = 1/200$ mesh.

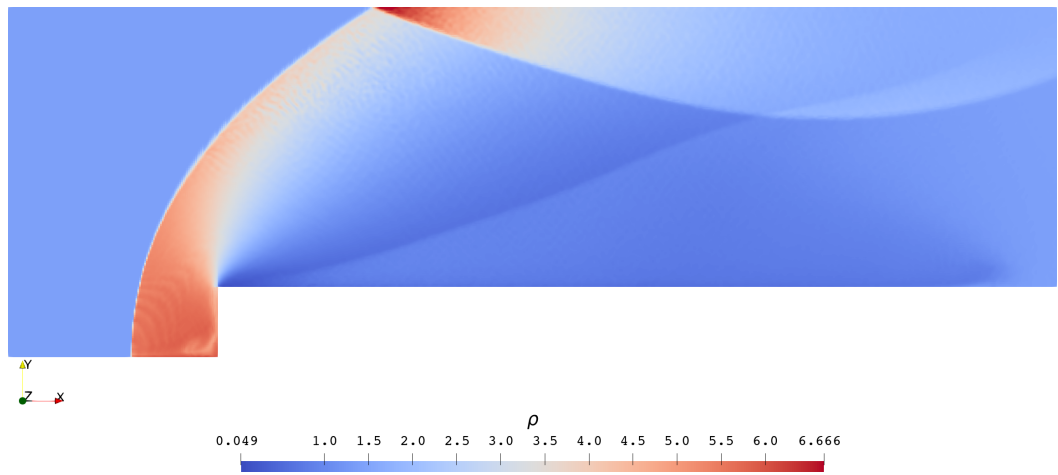


Figure 12. Third-order solution of Problem 4 at $t = 0.8$ on an $h = 1/200$ mesh.

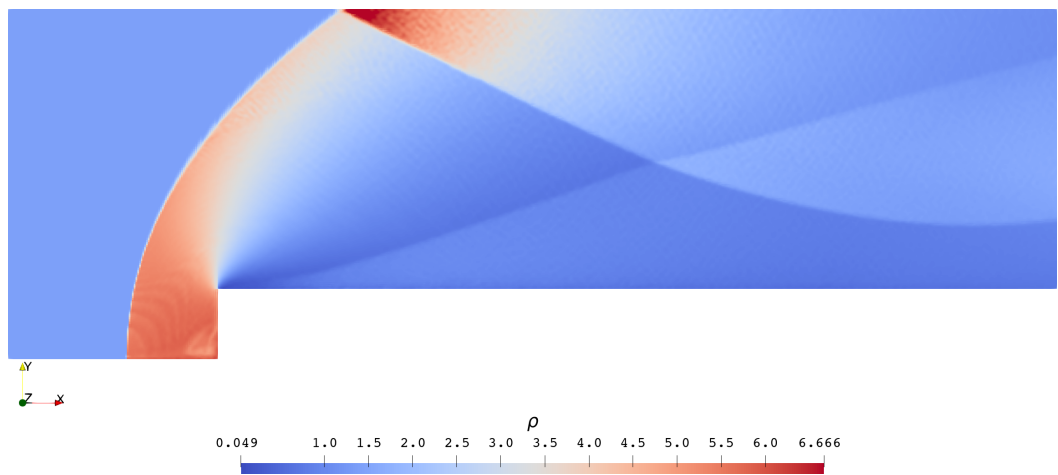


Figure 13. Third-order solution of Problem 4 at $t = 1.0$ on an $h = 1/200$ mesh.

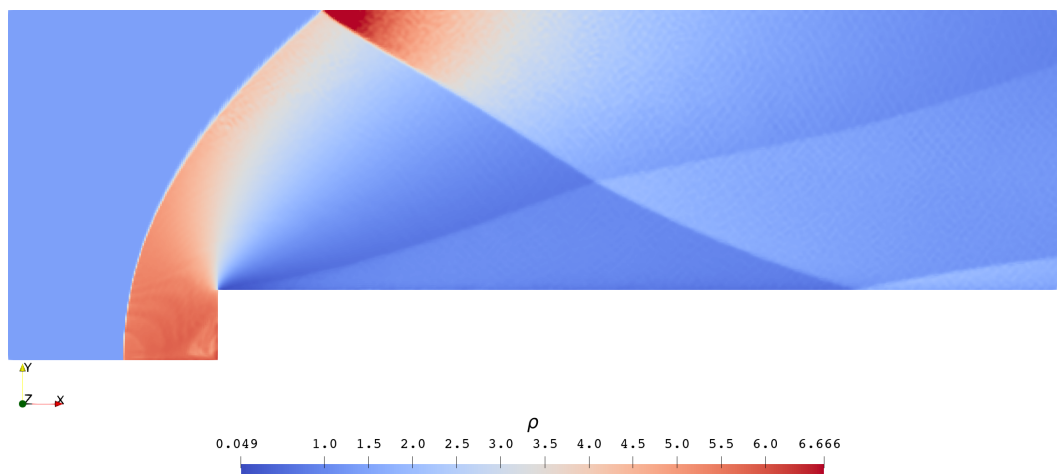


Figure 14. Third-order solution of Problem 4 at $t = 1.2$ on an $h = 1/200$ mesh.

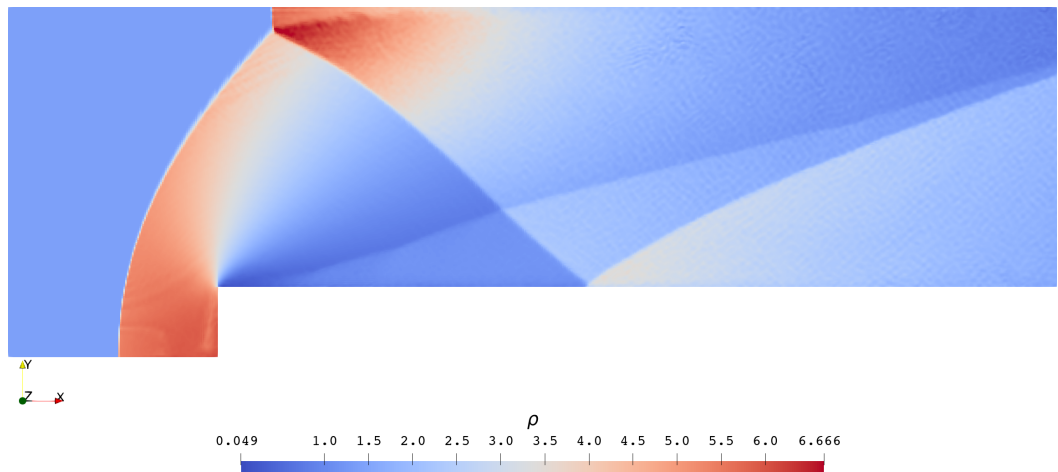


Figure 15. Third-order solution of Problem 4 at $t = 2.0$ on an $h = 1/200$ mesh.

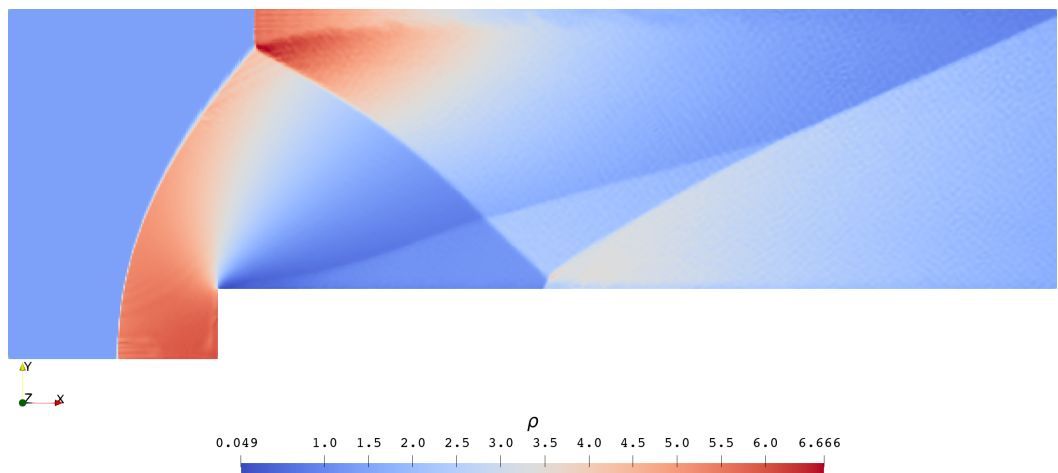


Figure 16. Third-order solution of Problem 4 at $t = 2.4$ on an $h = 1/200$ mesh.

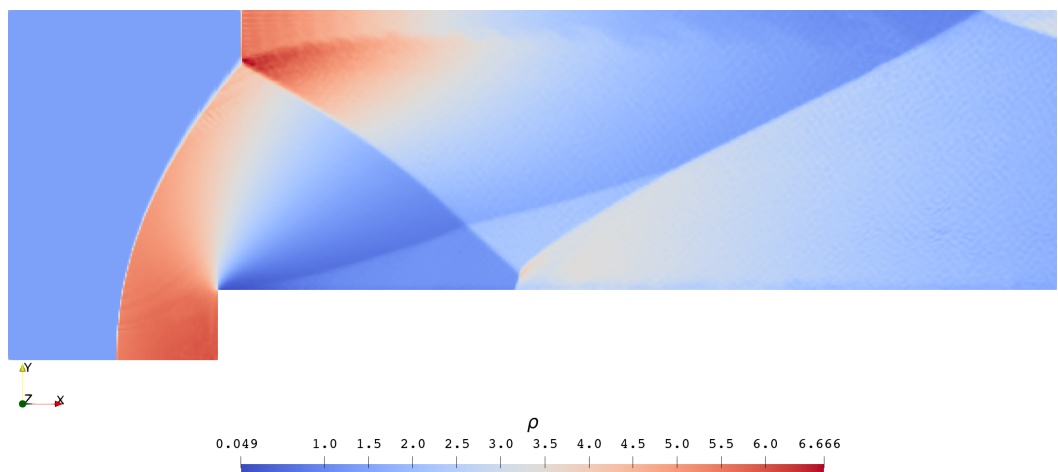


Figure 17. Third-order solution of Problem 4 at $t = 2.8$ on an $h = 1/200$ mesh.

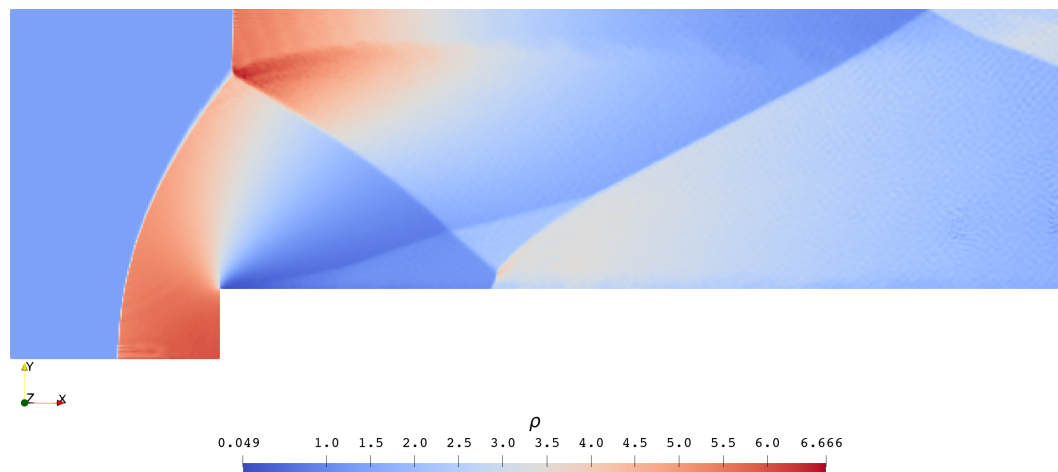


Figure 18. Third-order solution of Problem 4 at $t = 3.2$ on an $h = 1/200$ mesh.

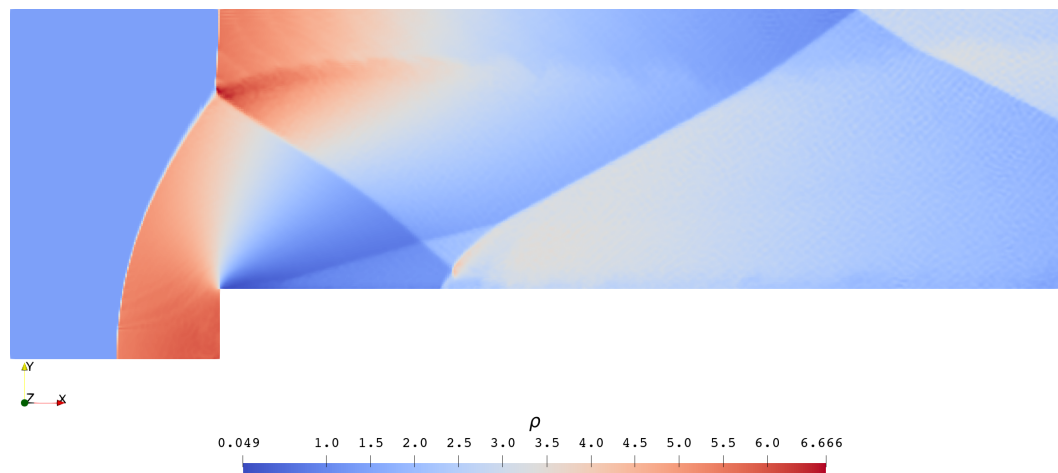


Figure 19. Third-order solution of Problem 4 at $t = 4.0$ on an $h = 1/200$ mesh.

The flow starts from a uniform supersonic ($M = 3.0$) state. In Figures 9 and 10, a curved bow shock wave generates in front of the forward-facing step and a fan-shaped expansion wave generates at the corner ($x = 0.6, y = 0.2$). The bow shock then hits the top of the tunnel (Figure 11) and reflects from it (Figures 12 and 13). At approximately $t = 1.2$ (Figure 14), the reflected shock hits the bottom of the tunnel (the top of the step) and reflects from it. Both of the two reflections are regular at these moments. Shortly before $t = 2.0$ (Figure 14), a Mach stem that characterizes a Mach reflection is generated from the first reflection point. A wavy contact discontinuity generating from the triple point is captured by Figures 16–19. This flow structure would be smeared off if we solved Problem 4 using lower-order solvers or coarser meshes.

3.2. Three-Dimensional Engineering Problems

In this section, we solve the Euler system Equation (1) on a three-dimensional unstructured mesh, which discretizes the space around a YF-17 aircraft by tetrahedral cells, as shown in Figure 20.

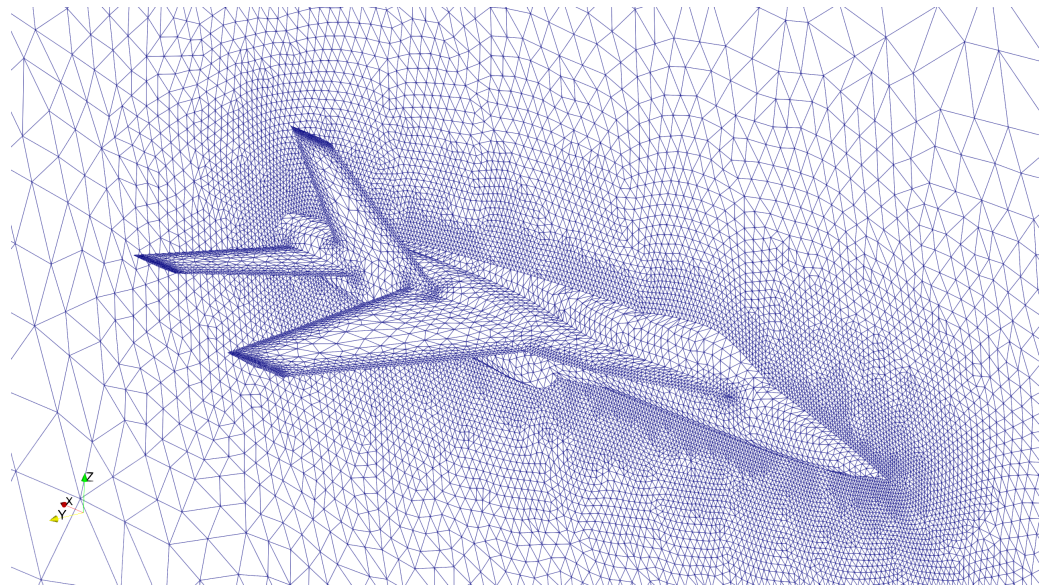


Figure 20. A zoomed-in view of the most important boundaries of the mesh for solving problems in Section 3.2.

3.2.1. YF-17 in Subsonic Flight

Problem 5. Solve the Euler system Equation (1) in the surroundings of a YF-17 aircraft. The initial condition is given as a subsonic flow ($M = 0.3$) with a 20-deg angle of attack:

$$[\rho \quad u \quad v \quad w \quad p]_{t=0} = [1.4 \quad 0.2819 \quad 0.0 \quad 0.1026 \quad 1.0].$$

All boundaries of the aircraft are defined as solid walls, except the intake of the engine, which is defined as a subsonic outlet of the condition equal to the initial condition, and the exhaust of the engine, which is defined as a supersonic ($M = 3.0$) inlet of the condition:

$$[\rho \quad u \quad v \quad w \quad p]_{ex.} = [1.4 \quad 2.4 \quad 0.0 \quad 0.0 \quad 1.44].$$

We solve this problem via two of our finite element solvers and plot the streamlines released from the strake and the wing at $t = 10$ in Figures 21 and 22. It can be seen that the streamlines obtained from the high-order ($p = 3$) solution are smoother than those from the low-order ($p = 1$) solution, which shows the benefit of p -refinement. The difference in accuracy is more obvious in Figure 23, which clearly shows a more detailed flow structure in the third-order solution (left half) and the piecewise constantness of the first-order solution (right half). Both solutions are able to capture the vortex trailing from the wing tip, which is generated from the the pressure difference between the upper and lower surfaces of the wing. When the wing generates positive lift, the pressure on the lower wing surface is higher than that on the upper wing surface. Under the action of this pressure difference, the air under the wing rolls up around the tip and flows backward, and the tip vortex is thus formed. The vortex trailing from the strake is generated from a similar mechanism.

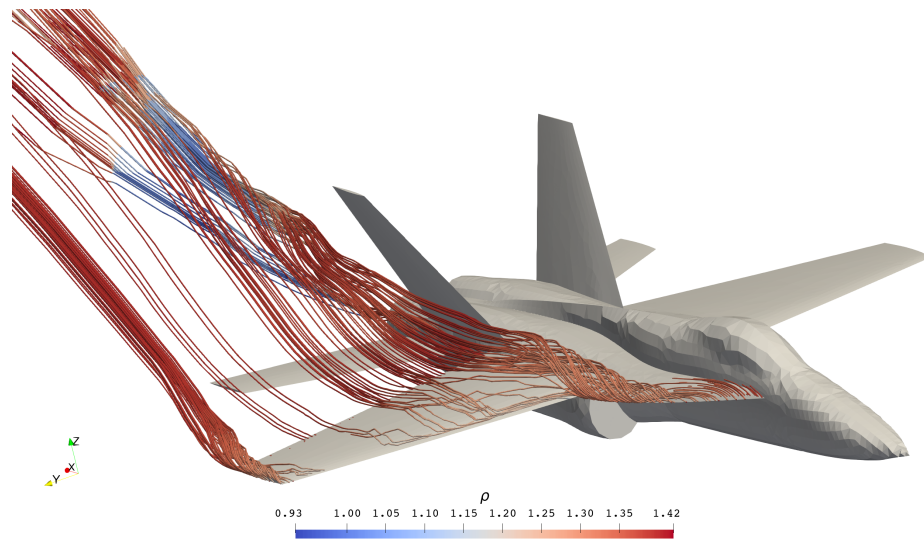


Figure 21. Streamlines given by the first-order solution of Problem 5.

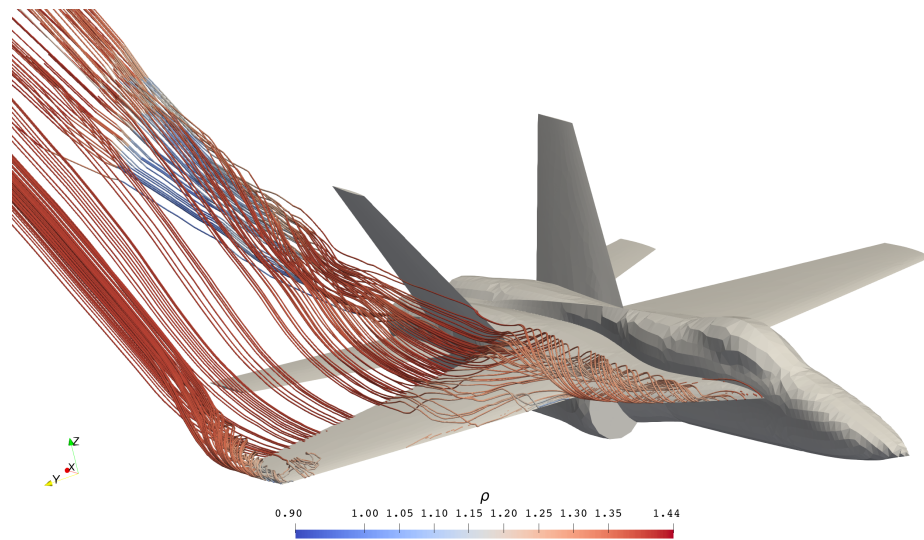


Figure 22. Streamlines given by the third-order solution of Problem 5.

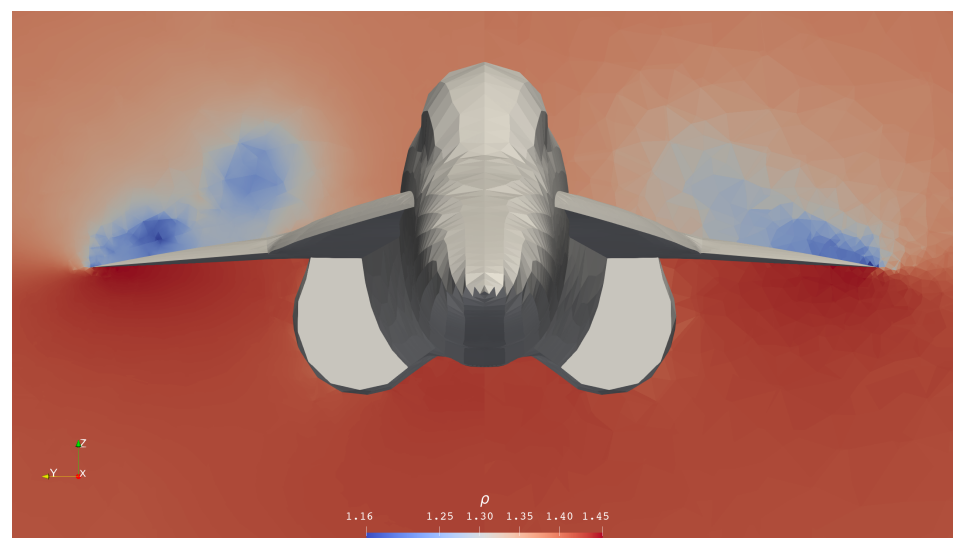


Figure 23. Comparison between the third-order solution (left half) and the first-order solution (right half) of Problem 5.

3.2.2. YF-17 in Supersonic Flight

Problem 6. Solve the Euler system Equation (1) in the surroundings of a YF-17 aircraft. The initial condition is given as a supersonic flow ($M = 2.0$) with a 0-deg angle of attack:

$$[\rho \quad u \quad v \quad w \quad p]_{t=0} = [1.4 \quad 2.0 \quad 0.0 \quad 0.0 \quad 1.0].$$

All boundaries of the aircraft are defined as solid walls, except the intake of the engine, which is defined as a supersonic outlet of the condition equal to the initial condition, and the exhaust of the engine, which is defined as a supersonic ($M = 3.0$) inlet of the condition:

$$[\rho \quad u \quad v \quad w \quad p]_{ex.} = [1.4 \quad 2.4 \quad 0.0 \quad 0.0 \quad 1.44].$$

We solve this problem using the same solvers as for Problem 5. Density contours on the $y = 0$ surface obtained from the first- and the third-order solutions are given in Figures 24 and 25, respectively. It is obvious that shock waves are captured well (without spurious oscillations) by both of them. To show the difference more clearly, we compare the two solutions in Figures 26 and 27. As in the subsonic case, the third-order solution (which is piecewise quadratic) outperforms the first-order one (which is piecewise constant). Shock waves (red) are generated on surfaces facing the wind, since the relative speed of air is larger than the speed of sound. Expansion waves (blue) are generated on leeward surfaces, since the solid body contracts there, which leaves more room for the supersonic flow.

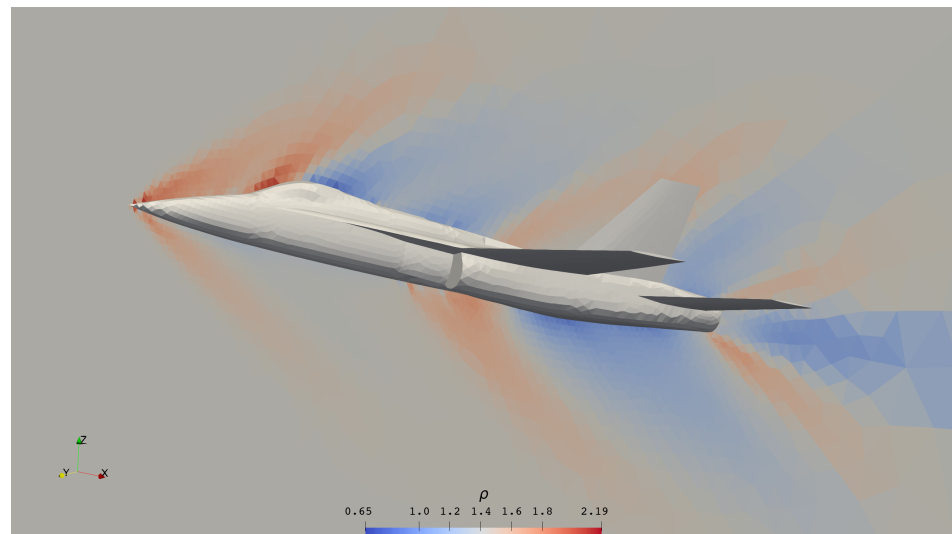


Figure 24. Density contour obtained from the first-order solution of Problem 6.

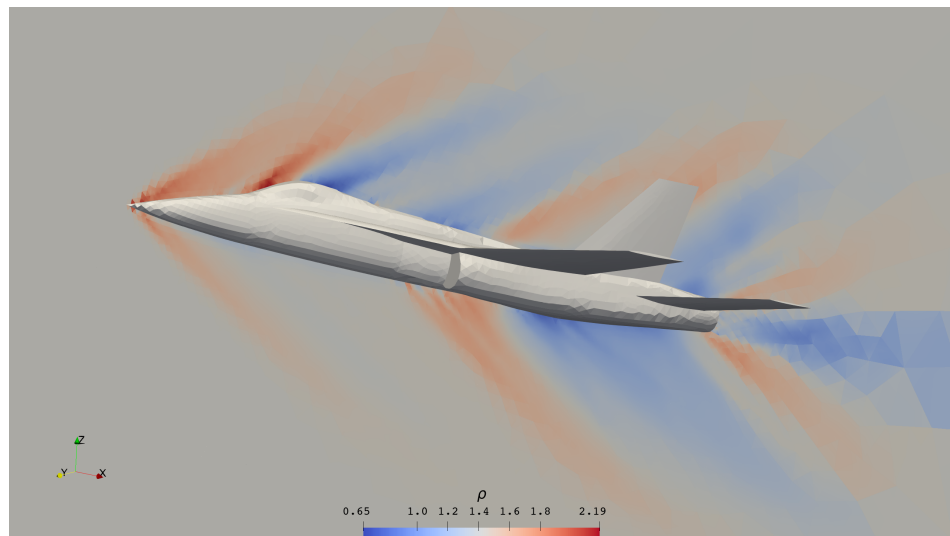


Figure 25. Density contour obtained from the third-order solution of Problem 6.



Figure 26. Comparison between the third-order solution (upper half) and the first-order solution (lower half) of Problem 6 on the $z = 0$ surface (top view).

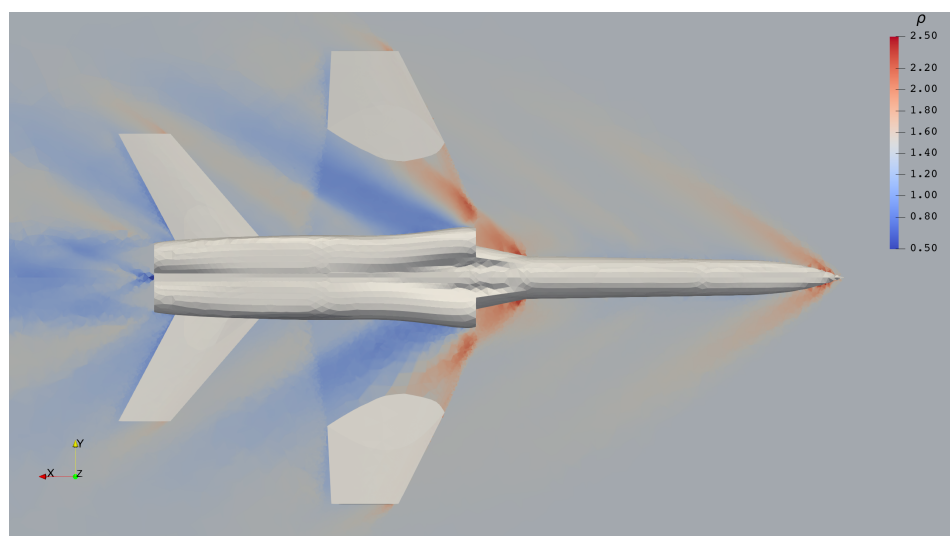


Figure 27. Comparison between the third-order solution (upper half) and the first-order solution (lower half) of Problem 6 on the $z = 0$ surface (bottom view).

3.2.3. Parallel Efficiency

Since the mesh in Figure 20 is highly unstructured and non-uniform, simple geometric partitioning cannot achieve a relatively balanced distribution of computational effort. In our code, we use the METIS library [32] to partition the dual graph of the mesh, which gives the results in Figures 28 and 29. The fluctuation of cell numbers is under 2%, which is quite good since the optimal partitioning of an unstructured mesh is an NP-hard problem. With such an approximately optimal partitioning, the parallel efficiency, which is defined as

$$E = \frac{T_{\text{serial}}}{PT_{\text{parallel}}} \times 100\%,$$

in which P is the number of processes, could surpass 99% in theory. In practice, however, the E values given by our numerical experiments are only around 80%, as shown in Figure 30, which is derived from the measured time costs given in Table 2. The main reason for the gap between theory and practice is that in the derivation of the ideal E value, inter-process communications are assumed to overlap in time with inner-cell computations, which is an over-optimistic assumption. Nevertheless, the acceleration is still significant, which reduces the wall clock time to an acceptable level.

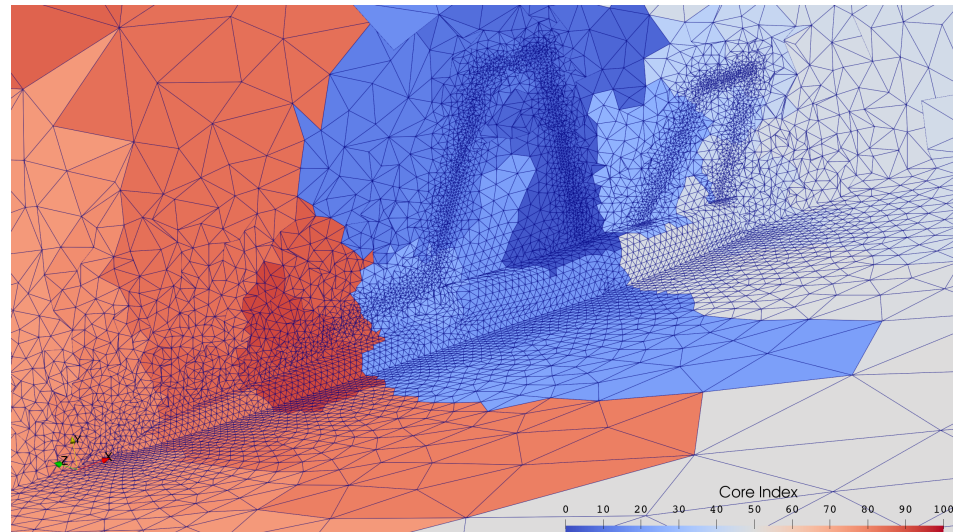


Figure 28. A 100-part partitioning of the mesh used for solving problems in Section 3.2.

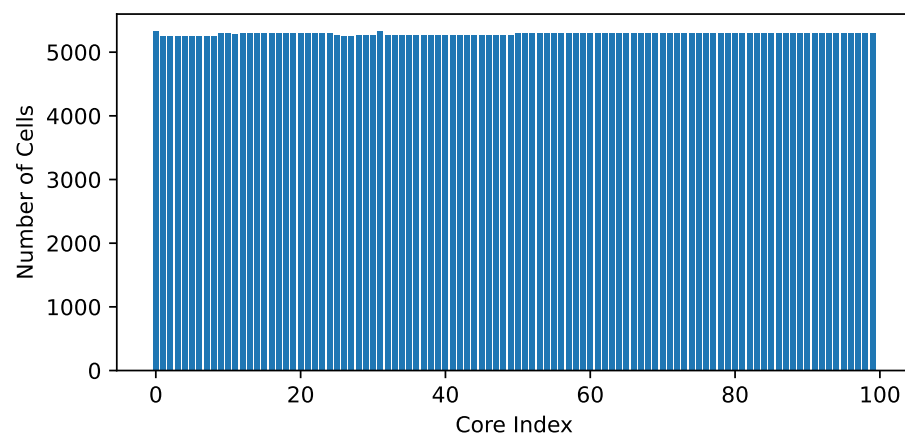


Figure 29. Distribution of cells in the mesh partitioning given in Figure 28.

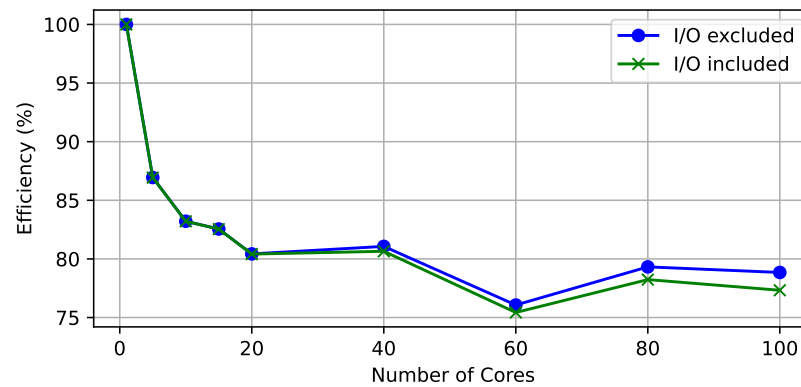


Figure 30. Parallel efficiency of the third-order solver for solving Problem 6.

Table 2. Time costs of the third-order solver running on different numbers of cores.

P	T_{100}	T_{399}	T_{400}	$P \frac{T_{399} - T_{100}}{399 - 100}$	$P \frac{T_{400} - T_{100}}{400 - 100}$
1	10,511.3	42,466.5	42,593.0	106.873	106.939
5	2389.25	9740.13	9769.96	122.925	123.012
10	1246.37	5086.62	5101.95	128.436	128.519
15	839.554	3420.17	3430.75	129.462	129.560
20	645.737	2632.32	2640.57	132.882	132.989
40	325.269	1310.72	1319.73	131.833	132.595
60	222.890	923.043	931.754	140.499	141.773
80	168.862	672.439	681.445	134.736	136.689
100	137.787	543.081	552.682	135.550	138.298

3.3. Problems with Momentum Sources

In this section, we solve two problems of rotorcraft aerodynamics using the momentum source model described in Section 2.4. The problems are set to simulate wind tunnel tests of a rotor (a pair of rotary wings), using the mesh shown in Figure 31. All of the boundaries of the outside box are solid walls, except the left and right ends, which are set as the inlet and outlet, respectively. The spherical region is the circumscribed sphere of the rotor, whose rotating axis could point in any direction. The smaller box enclosing the sphere is used for refining the mesh in the surrounding and downstream regions of the rotor.

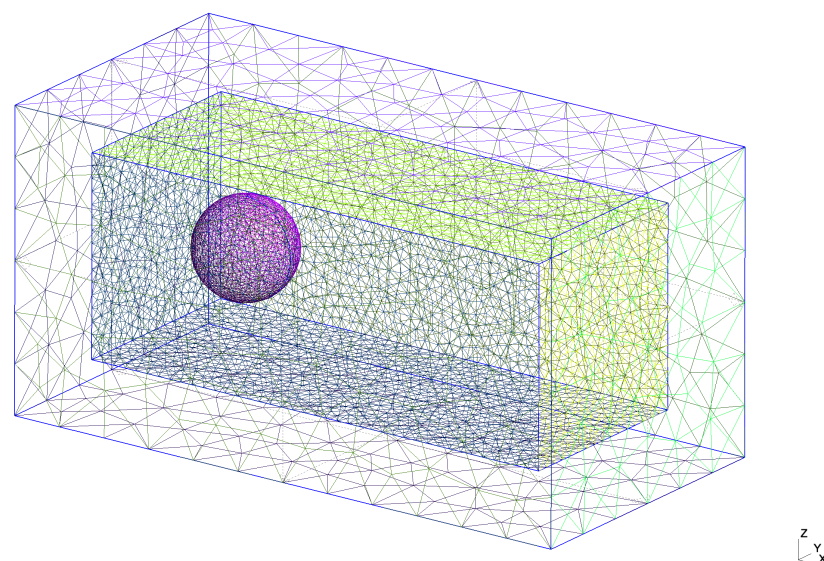


Figure 31. A schematic mesh for solving problems in Section 3.3.

3.3.1. A Climbing Rotor

Problem 7. Solve Equation (1) in the surroundings of a rotor, whose rotating axis points in the direction of $(-1, 0, 0)$. The initial condition is given as a uniform flow:

$$[\rho \quad u \quad v \quad w \quad p]_{t=0} = [1.29 \quad 1.0 \quad 0.0 \quad 0.0 \quad 101325.0],$$

which is also the background state at the two ends of the wind tunnel.

Third- and first-order solutions at the same moment ($t = 1$) are shown in Figures 32 and 33, respectively, in which density contours and velocity directions are plotted on selected slices. Large-scale flow structures, such as the contraction of airflow near the rotor disk and the rolled-up airwake, are captured in both figures. However, finer details, such as the ripples generated from the rotor disk and the strong vortex ring in the downstream, are only visible in the third-order solution. The nearly axisymmetric flow structure is caused by the periodic movement of the rotary wings. The density of air immediately below the rotor (red) is greater than that above the rotor (blue), which means that the rotor is compressing the air flowing across it. According to Newton's third law, the rotor must be experiencing a force in the opposite direction exerted by the air. This is where the rotor thrust comes from.

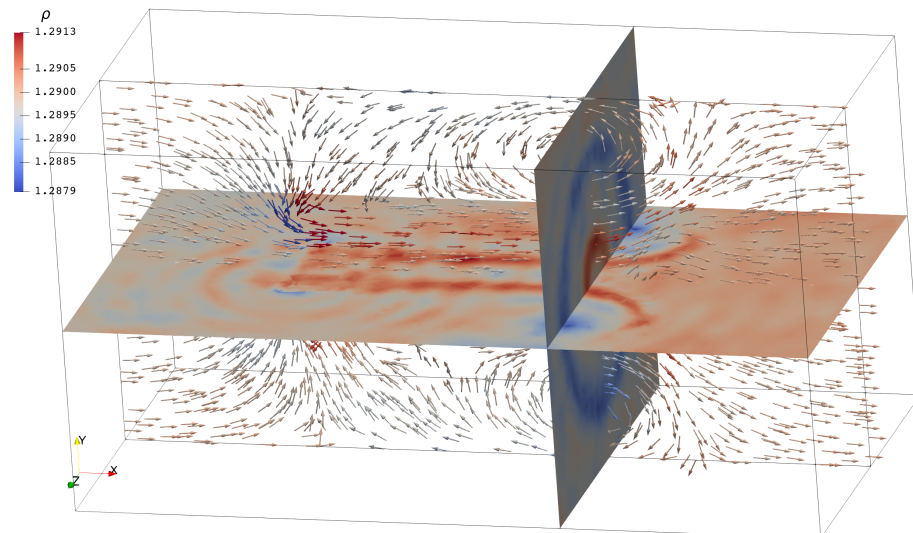


Figure 32. Third-order solution of Problem 7 at $t = 1$.

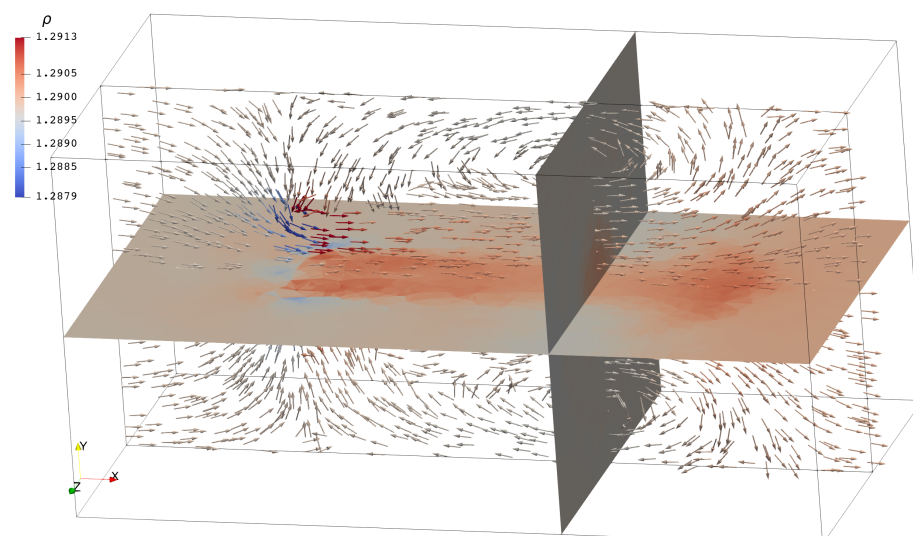


Figure 33. First-order solution of Problem 7 at $t = 1$.

3.3.2. A Rotor in Forward Flight

Problem 8. Solve Equation (1) in the surrounding of a rotor, whose rotating axis points in the direction of $(0,0,1)$. The initial condition is given as a uniform flow:

$$[\rho \quad u \quad v \quad w \quad p]_{t=0} = [1.29 \quad 10.0 \quad 0.0 \quad 0.0 \quad 101325.0],$$

which is also the background state at the two ends of the wind tunnel.

As with the climbing problem, we plot the third- and first-order solution at the same moment ($t = 1$) in Figures 34 and 35, respectively. As before, both of them can capture large-scaled structures, such as the skewed and rolled-up airwakes. The rolled-up structure is more clear in Figure 36, in which streamlines are plotted and colored by the magnitude of velocity. The piecewise constant first-order solution is much vaguer and loses many details, which shows again the value of developing high-order solvers.

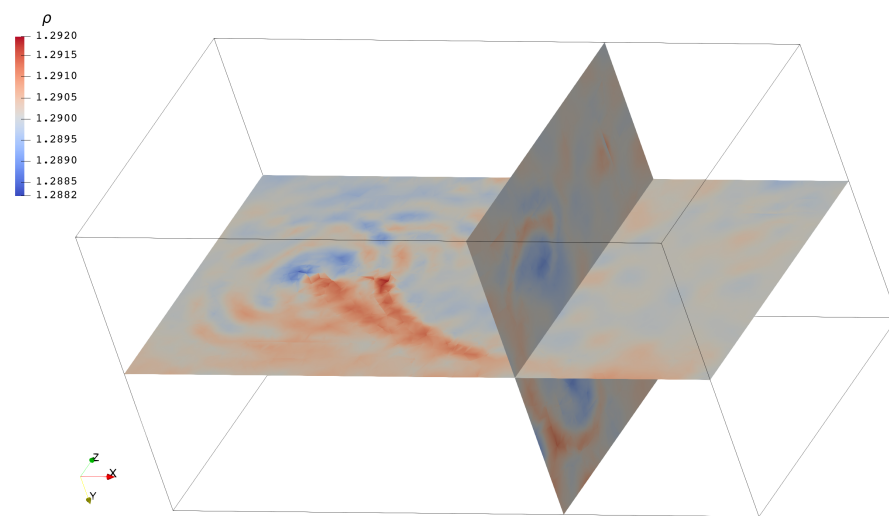


Figure 34. Third-order solution of Problem 8 at $t = 1$.

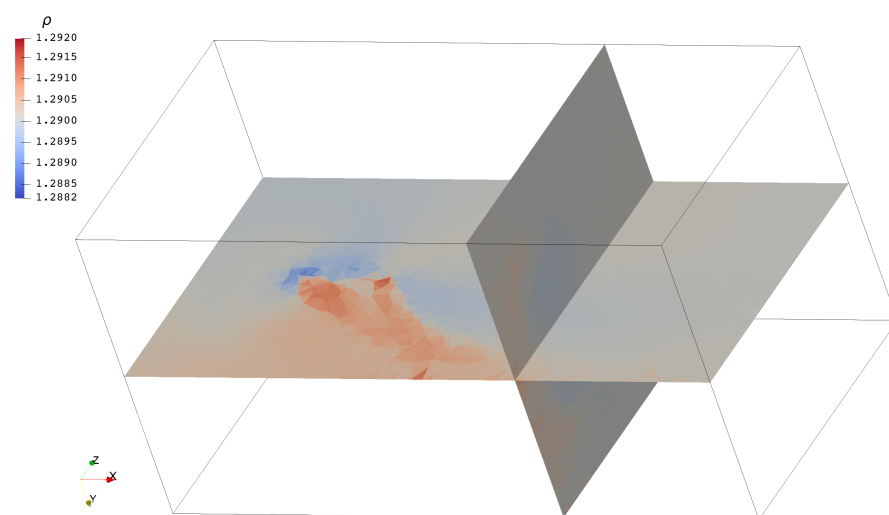


Figure 35. First-order solution of Problem 8 at $t = 1$.

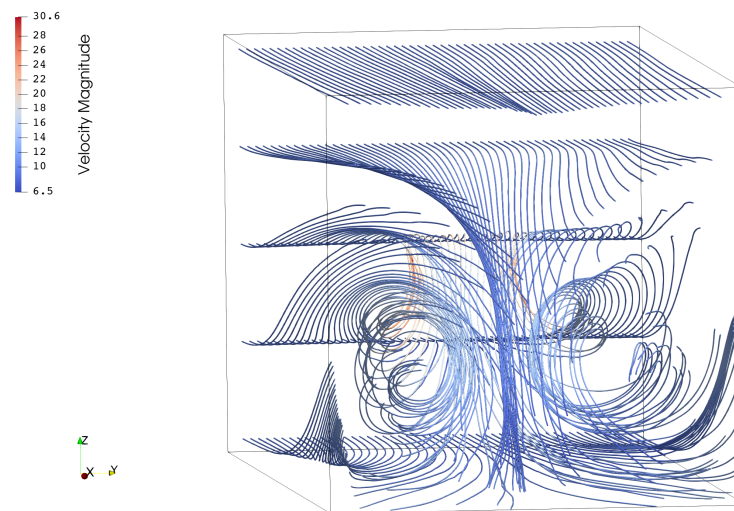


Figure 36. Streamlines of the solution in Figure 34.

4. Discussion

In this article, we give a concise but complete formulation of an RKDG scheme with a compact WENO limiter for solving inviscid compressible flows, possibly with momentum sources, on three-dimensional unstructured meshes. The algorithms are implemented on top of public available libraries, which support unstructured mesh partitioning, inter-process communication and distributed memory parallelization. The correctness and accuracy of the solvers are validated by lower-dimensional reference problems. Results of real three-dimensional applications are also presented, in which complex flow structures are captured by the third-order solver even on very coarse unstructured meshes. Future works may include implementing higher-order solvers under this framework and testing them on larger meshes and on larger high-performance computing platforms.

Author Contributions: Conceptualization, W.P. and S.L.; methodology, W.P.; software, W.P. and Y.J.; validation, W.P. and Y.J.; formal analysis, W.P. and Y.J.; investigation, W.P.; resources, W.P.; data curation, W.P.; writing—original draft preparation, W.P.; writing—review and editing, W.P., Y.J. and S.L.; visualization, W.P.; supervision, S.L.; project administration, S.L.; funding acquisition, S.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National High-Tech R&D Program of China (863 Program), grant number 2012AA112201.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this article are all generated from the source code publicly available in our GitHub repository at <https://github.com/pvc1989/miniCFD> accessed on 6 July 2022 (or the mirror site <https://gitee.com/pvc1989/miniCFD> accessed on 6 July 2022), except the mesh used in Section 3.2, which was downloaded from <https://cgns.github.io/CGNSFiles.html> accessed on 13 May 2022.

Acknowledgments: The authors would like to express their deepest appreciation to Shen Xin for his professional guidance on computer science, and to Zhou Yukai for his patient assistance with mathtyping.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

Abbreviations and Nomenclature

The following abbreviations are used in this manuscript:

CFD	Computational Fluid Dynamics
FD	Finite Difference
FV	Finite Volume
FE	Finite Element
DG	Discontinuous Galerkin
RK	Runge–Kutta
TVD	Total Variation Diminishing
SSP	Strong Stability Preserving
ENO	Essentially Non-Oscillatory
WENO	Weighted ENO
$\partial_t, \partial_x, \partial_y, \partial_z$	partial derivative with respect to t, x, y, z
α	angle of attack
γ	heat capacity ratio of the gas
$\vec{\nu}$	outer normal vector at some point on ∂E_i
$\vec{\pi}, \vec{\sigma}$	tangential vectors at some point on ∂E_i
ρ	density of the gas
ϕ_k	k -th function in an orthonormal basis
ψ	an arbitrary test function
ψ^h	approximation of ψ
$\hat{\psi}_k$	projection of ψ on ϕ_k
\underline{A}^v	Jacobian of \underline{F}^v with respect to \underline{U}
\vec{b}_L	body force per unit length
\vec{b}_V	body force per unit volume
b_x, b_y, b_z	x, y, z -component of body force per unit volume
c	chord length of a blade
E_i and ∂E_i	i -th element (cell) and its boundary
e_0	specific total energy of the gas
\vec{F}	flux vector whose components are matrices
\underline{F}^v	projection of \vec{F} on $\vec{\nu}$
$\underline{F}^x, \underline{F}^y, \underline{F}^z$	x, y, z -component of \vec{F}
h_0	specific total enthalpy of the gas
p	pressure of the gas
\underline{Q}	column matrix of source terms
\underline{R}	changing rate of \underline{U}
\underline{R}^v	matrix whose j -th column is the j -th eigenvector of \underline{A}^v
\underline{U}	column matrix of conservative variables
\underline{U}^h and $\underline{U}^{\text{new}}$	approximation of \underline{U} and its WENO reconstruction
$\underline{U}_{i k}^{\text{new}}$	WENO reconstruction of \underline{U}^h on E_i with the help of E_k
\hat{U}	coefficient matrix whose k -th column is \hat{U}_k
\hat{U}_k	projection of \underline{U} on ϕ_k
\hat{U}^n and \hat{R}^n	\hat{U} and \hat{R} at the n -th time step
u_x, u_y, u_z	x, y, z -component of velocity
\underline{V}	column matrix of characteristic variables
\underline{V}^h and $\underline{V}^{\text{new}}$	approximation of \underline{V} and its WENO reconstruction

References

1. Harten, A.; Lax, P.D.; van Leer, B. On Upstream Differencing and Godunov-Type Schemes for Hyperbolic Conservation Laws. *SIAM Rev.* **1983**, *25*, 35–61. [[CrossRef](#)]
2. Reed, W.H.; Hill, T.R. *Triangular Mesh Methods for the Neutron Transport Equation*; Technical Report LA-UR-73-479; Los Alamos Scientific Lab.: Los Alamos, MN, USA, 1973.
3. Chavent, G.; Salzano, G. A finite-element method for the 1-D water flooding problem with gravity. *J. Comput. Phys.* **1982**, *45*, 307–344. [[CrossRef](#)]

4. Cockburn, B.; Shu, C.W. The Runge-Kutta local projection P1-discontinuous-Galerkin finite element method for scalar conservation laws. In Proceedings of the 1st National Fluid Dynamics Conference, Cincinnati, OH, USA, 25–28 July 1988; American Institute of Aeronautics and Astronautics: Reston, VA, USA, 1988. [[CrossRef](#)]
5. Shu, C.W.; Osher, S. Efficient implementation of essentially non-oscillatory shock-capturing schemes. *J. Comput. Phys.* **1988**, *77*, 439–471. [[CrossRef](#)]
6. Shu, C.W.; Osher, S. Efficient implementation of essentially non-oscillatory shock-capturing schemes, II. *J. Comput. Phys.* **1989**, *83*, 32–78. [[CrossRef](#)]
7. Cockburn, B.; Lin, S.Y.; Shu, C.W. TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws III: One-dimensional systems. *J. Comput. Phys.* **1989**, *84*, 90–113. [[CrossRef](#)]
8. Cockburn, B.; Hou, S.; Shu, C.W. The Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws. IV. The multidimensional case. *Math. Comput.* **1990**, *54*, 545–581. [[CrossRef](#)]
9. Cockburn, B.; Shu, C.W. The Runge-Kutta Discontinuous Galerkin Method for Conservation Laws V: Multidimensional Systems. *J. Comput. Phys.* **1998**, *141*, 199–224. [[CrossRef](#)]
10. Gottlieb, D.; Shu, C.W. On the Gibbs Phenomenon and Its Resolution. *SIAM Rev.* **1997**, *39*, 644–668. [[CrossRef](#)]
11. Sweby, P.K. High Resolution Schemes Using Flux Limiters for Hyperbolic Conservation Laws. *SIAM J. Numer. Anal.* **1984**, *21*, 995–1011. [[CrossRef](#)]
12. Harten, A.; Engquist, B.; Osher, S.; Chakravarthy, S.R. Uniformly High Order Accurate Essentially Non-oscillatory Schemes, III. *J. Comput. Phys.* **1997**, *131*, 3–47. [[CrossRef](#)]
13. Jiang, G.S.; Shu, C.W. Efficient Implementation of Weighted ENO Schemes. *J. Comput. Phys.* **1996**, *126*, 202–228. [[CrossRef](#)]
14. Hu, C.; Shu, C.W. Weighted Essentially Non-oscillatory Schemes on Triangular Meshes. *J. Comput. Phys.* **1999**, *150*, 97–127. [[CrossRef](#)]
15. Qiu, J.; Shu, C.W. Runge-Kutta Discontinuous Galerkin Method Using WENO Limiters. *SIAM J. Sci. Comput.* **2005**, *26*, 907–929. [[CrossRef](#)]
16. Zhu, J.; Qiu, J.; Shu, C.W.; Dumbser, M. Runge-Kutta discontinuous Galerkin method using WENO limiters II: Unstructured meshes. *J. Comput. Phys.* **2008**, *227*, 4330–4353. [[CrossRef](#)]
17. Zhong, X.; Shu, C.W. A simple weighted essentially nonoscillatory limiter for Runge-Kutta discontinuous Galerkin methods. *J. Comput. Phys.* **2013**, *232*, 397–415. [[CrossRef](#)]
18. Zhu, J.; Zhong, X.; Shu, C.W.; Qiu, J. Runge-Kutta discontinuous Galerkin method using a new type of WENO limiters on unstructured meshes. *J. Comput. Phys.* **2013**, *248*, 200–220. [[CrossRef](#)]
19. Pei, W.; Jiang, Y.; Li, S. An Efficient Parallel Implementation of the Runge-Kutta Discontinuous Galerkin Method with Weighted Essentially Non-Oscillatory Limiters on Three-Dimensional Unstructured Meshes. *Appl. Sci.* **2022**, *12*, 4228. [[CrossRef](#)]
20. Rajagopalan, R.G.; Mathur, S.R. Three dimensional analysis of a rotor in forward flight. In Proceedings of the 20th Fluid Dynamics, Plasma Dynamics and Lasers Conference, Buffalo, NY, USA, 12–14 July 1989; American Institute of Aeronautics and Astronautics: Reston, VA, USA, 1989. [[CrossRef](#)]
21. Rajagopalan, R.G.; Mathur, S.R. Three Dimensional Analysis of a Rotor in Forward Flight. *J. Am. Helicopter Soc.* **1993**, *38*, 14–25. [[CrossRef](#)]
22. Kang, N.; Sun, M. Technical Note: Prediction of the Flow Field of a Rotor in Ground Effect. *J. Am. Helicopter Soc.* **1997**, *42*, 195–198. [[CrossRef](#)]
23. Kang, N.; Sun, M. Simulated Flowfields in Near-Ground Operation of Single- and Twin-Rotor Configurations. *J. Aircr.* **2000**, *37*, 214–220. [[CrossRef](#)]
24. Shi, Y.; Xu, Y.; Zong, K.; Xu, G. An investigation of coupling ship/rotor flowfield using steady and unsteady rotor methods. *Eng. Appl. Comput. Fluid Mech.* **2017**, *11*, 417–434. [[CrossRef](#)]
25. Shen, W.Z.; Zhang, J.H.; Sørensen, J.N. The Actuator Surface Model: A New Navier-Stokes Based Model for Rotor Computations. *J. Sol. Energy Eng.* **2009**, *131*. [[CrossRef](#)]
26. Kim, T.; Oh, S.; Yee, K. Improved actuator surface method for wind turbine application. *Renew. Energy* **2015**, *76*, 16–26. [[CrossRef](#)]
27. Toro, E.F. *Riemann Solvers and Numerical Methods for Fluid Dynamics*; Springer: Berlin/Heidelberg, Germany, 2009. [[CrossRef](#)]
28. Zhang, L.; Cui, T.; Liu, H. A Set of Symmetric Quadrature Rules on Triangles and Tetrahedra. *J. Comput. Math.* **2009**, *27*, 89–96.
29. Gottlieb, S.; Shu, C.W. Total variation diminishing Runge-Kutta schemes. *Math. Comput.* **1998**, *67*, 73–85. doi: 10.1090/s0025-5718-98-00913-2. [[CrossRef](#)]
30. Gottlieb, S.; Shu, C.W.; Tadmor, E. Strong Stability-Preserving High-Order Time Discretization Methods. *SIAM Rev.* **2001**, *43*, 89–112. [[CrossRef](#)]
31. Woodward, P.; Colella, P. The numerical simulation of two-dimensional fluid flow with strong shocks. *J. Comput. Phys.* **1984**, *54*, 115–173. [[CrossRef](#)]
32. Karypis, G.; Kumar, V. A Fast and High Quality Multilevel Scheme for Partitioning Irregular Graphs. *SIAM J. Sci. Comput.* **1998**, *20*, 359–392. [[CrossRef](#)]