A High-Order Discontinuous Galerkin Method for Solving Preconditioned Euler Equations

Huanqin Gao, Jiale Zhang, Hongquan Chen *, Shengguan Xu and Xuesong Jia

Key Laboratory of Non-Steady Aerodynamics and Flow Control of MIIT, College of Aerospace Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China; hjingao@nuaa.edu.cn (H.G.); zhjl_nuaa@nuaa.edu.cn (J.Z.); shengguanxu@nuaa.edu.cn (S.X.); xuesongjia@nuaa.edu.cn (X.J.)

* Correspondence: hqchenam@nuaa.edu.cn

Abstract: A high-order discontinuous Galerkin (DG) method is presented for solving the preconditioned Euler equations with an explicit or implicit time marching scheme. A detailed description is given of a practical implementation of a precondition matrix of the type of Weiss and Smith and of the DG spatial discretization scheme employed, with particular emphasis on the artificial viscosity-based shock capturing techniques. The curved boundary treatment is proposed through adopting a NURBS surface equipped with a radial basis function interpolation to propagate the boundary displacement to the interior of the mesh. The resulting methods are verified by simulating flows over two-dimensional airfoils, such as symmetric NACA0012 or asymmetric RAE2822, and over three-dimensional bodies, such as an academic hemispherical headform or aerodynamic ONERA M6 wing. Numerical results show that the present method functions for both transonic and nearly incompressible flow simulations, and the proposed treatment of curved boundaries, play an important role in improving the accuracy of the obtained solutions, which are in good agreement with available experimental data or other numerical solutions reported in literature.

Keywords: discontinuous Galerkin; Euler equations; preconditioning; shock capturing; curved boundary treatment

1. Introduction

Under the dominance of second order accurate methods in computational fluid dynamics (CFD) for real applications, the discontinuous Galerkin (DG) method started becoming increasingly popular in recent years due to its ability to achieve higher order spatial accuracy rigorously, even on an unstructured grid [1–4]. The compressible DG method is used in a wide spectrum of flow regimes [5–10]. In order to be able to capture shocks existing in the flow fields, several key ingredients, such as slope limiter [5], weighted essentially non-oscillatory (WENO) based limiter [6,7], and artificial viscosity [8,9] are usually imposed. The application of the slope limiter requires the identification of the elements lying in the shock region, and also requires for the reduction of the interpolating polynomial to be first order in those identified elements, which may destroy the high-order accuracy in the shocked region. In order to maintain the required accuracy, a WENO-based limiter can be selected at the cost of high computational time consuming. In contrast, the artificial viscosity-based approach is reported to be efficient in view of a relatively low cost [8] and to be flexible in terms of independence on the approximation order with different types of mesh elements [9]. Thus, in the present work, artificial viscosity is intended to be imposed on the preconditioned DG method, capable of compressible shocked flow simulations with explicit or implicit time integration schemes.

In general, the schemes with implicit time integration have the advantage of the time steps not being restricted by the severely strict stability limits in comparison to the explicit counterpart [2,10], and hence so-called implicit acceleration can often be achieved in terms of the convergence rate. In fact, successful efforts and applications were reported using
classical implicit schemes, including the generalized minimal residual method [11–13], implicit Runge–Kutta method [3,14] and the lower-upper symmetric Gauss–Seidel (LU-SGS) method [15] etc. In contrast, an LU-based implicit scheme is proven to be robust in calculations over a wide range of Mach numbers [16]. However, for the flows over aerodynamic bodies at low Mach numbers [10,17–19], the undesirable effects, such as low convergence speed (stiffness problem [10,19]) and loss of accuracy (reaching unphysical solution [18]), may appear due to the large disparity of wave speeds of the governing equations. Such effects are reported to be greatly ameliorated through adopting techniques such as preconditioning [19]. It can be learned that most previous research concerning the preconditioned DG method are focused on the flows at low Mach numbers without the presence of shock waves.

This paper presents an effort to develop a high-order accurate DG method with the key ingredients of preconditioning and shock capturing, capable of both nearly incompressible and compressible transonic shocked flow simulations. The preconditioning is realized by multiplying a matrix of the type of Weiss and Smith to the time derivative of the Euler equations, and the shock capturing ability is then enforced through adding an artificial viscosity term to the right side of the resulting preconditioned Euler equations. A detailed description is given of the DG spatial discretization scheme employed, with particular emphasis on the proposed treatment of curved boundaries through adopting a non-uniform rational basis spline (NURBS) surface equipped with a radial basis function (RBF) interpolation to propagate the boundary curve fit motion to the interior of the mesh. An explicit Runge–Kutta scheme associated with time integration is first applied to update the solutions, and the implicit counterpart is then formed through adopting the LU-SGS algorithm. The resulting methods are tested and analyzed by simulating flows over two-dimensional (2D) airfoils, such as symmetric NACA0012 or asymmetric RAE2822, and over three-dimensional (3D) bodies, such as an academic hemispherical headform or an aerodynamic ONERA M6 wing. The numerical results are presented in comparison with available experimental data or other numerical solutions reported in literature. It can be learned that the present method does function for compressible transonic shocked flow simulations, as well as for nearly incompressible flow simulations at low Mach numbers, and the proposed curved boundary treatment does introduce a notable improvement in the accuracy of the numerical approximation.

The rest of the paper is arranged as follows: Key aspects of a high-order accurate numerical method, including governing equations, preconditioning, numerical flux modification, DG spatial discretization, shock capturing, and the enforcement of boundary conditions are described in Section 2. In Section 3, a practical treatment of curved boundaries is proposed. In order to have high-order boundary approximation, RBF interpolation is introduced to propagate the boundary motion to the interior of the mesh. Some details of the time-stepping schemes are then described in Section 4, with particular emphasis on the implicit LU-SGS algorithm. In Section 5, a set of 2D- and 3D-test cases are arranged to demonstrate the ability of the resulting methods. Finally, conclusions are drawn together with possible further extensions in Section 6.

2. High-Order Accurate Numerical Method

2.1. Governing Equations

The three-dimensional compressible Euler equations governing inviscid flows can be expressed in a dimensionless conservative form [20] as follows:

$$\frac{\partial Q}{\partial t} + \nabla \cdot F = 0, \tag{1}$$
where the unknown vector $Q$ of conserved variables and the Cartesian components of the flux function $F = (f_1, f_2, f_3)$ are given by

$$
Q = \begin{pmatrix}
\rho \\
\rho u \\
\rho v \\
\rho w \\
\rho E
\end{pmatrix}, \quad 
.f_1 = \begin{pmatrix}
\rho u \\
\rho u^2 + p \\
\rho uv \\
\rho uw \\
u(\rho E + p)
\end{pmatrix}, \\
.f_2 = \begin{pmatrix}
\rho v \\
\rho v^2 + p \\
\rho vw \\
\rho vw \\
v(\rho E + p)
\end{pmatrix}, \\
.f_3 = \begin{pmatrix}
\rho w \\
\rho w^2 + p \\
\rho uw \\
\rho vw \\
\rho T + \rho C_p
\end{pmatrix},
$$

where $\rho$, $p$, and $E$ are the density, pressure, and total energy per unit mass, respectively, and $u$, $v$, and $w$ are the Cartesian components of the velocity vector $V$. This system can be closed up by adding the following equation of state:

$$
\rho E = \frac{p}{\Gamma - 1} + \frac{1}{2\rho} (u^2 + v^2 + w^2),
$$

which is valid for perfect gas, $\Gamma = \frac{C_p}{C_v}$ is the ratio of specific heats and taken as $\Gamma = 1.4$ for air.

The Equation (1) in conservative variables are reported to be ill-conditioned for the flows at low Mach numbers [21]. In order to single out the propagation of acoustic waves [22], the governing equations in primitive variables $q = (p, u, v, w, T)^T$ are considered here. By transforming the dependent variables in Equation (1) from conserved quantities to such primitive variables, Equation (1) can be written as

$$
P \frac{\partial q}{\partial t} + \nabla \cdot F = 0,
$$

where $P$ is the transformation matrix

$$
P = \frac{\partial Q}{\partial q} = \begin{bmatrix}
\rho_p & 0 & 0 & 0 & \rho_T \\
\rho_p u & \rho & 0 & 0 & \rho_T u \\
\rho_p v & 0 & \rho & 0 & \rho_T v \\
\rho_p w & 0 & 0 & \rho & \rho_T w \\
\rho_p H - 1 & \rho u & \rho v & \rho w & \rho_T H + \rho C_p
\end{bmatrix},
$$

here, $H = E + p/\rho$ is total enthalpy per unit mass of the fluid. For inviscid flows of interest, the derivatives that appear in Equation (5) can be directly determined by the equation of state, $p = \frac{1}{\Gamma} \rho T$, which reads:

$$
\rho_p = \frac{\partial \rho}{\partial p} \bigg|_T = \text{const} = \frac{\Gamma}{\Gamma - 1}, \quad \rho_T = \frac{\partial \rho}{\partial T} \bigg|_p = \text{const} = -\frac{\rho}{T},
$$

where $c$ is the acoustic velocity. The derivative $\rho_p$ that multiplies the pressure time derivative in the continuity equation controls the speed of propagation of acoustic waves [23] in the system (4). This is consistent with the notion of infinite pressure wave speeds in an incompressible flow due to $\rho_p = 0$ for constant density flows. The disparity of wave speeds of the system (4), mainly between acoustic waves and entropy waves, becomes very large for low Mach number flows, and can be ameliorated through adopting techniques such as preconditioning [10,23].

Following the well-known preconditioning technique of Weiss and Smith [23], the matrix $P$ in Equation (4) is modified through replacing the derivative $\rho_p$, with one proportional to the inverse of the local velocity squared to control the disparity of the wave
speeds of the system (4). The resultant matrix, termed as “preconditioning matrix $\Gamma$”, can be expressed as follows:

$$\Gamma = \begin{bmatrix} \Theta & 0 & 0 & 0 & \rho T \\ \Theta u & \rho & 0 & 0 & \rho T u \\ \Theta v & 0 & \rho & 0 & \rho T v \\ \Theta w & 0 & 0 & \rho & \rho T w \\ \Theta H - 1 & \rho u & \rho v & \rho w & \rho T H + \rho C_p \end{bmatrix},$$

(6)

where $\Theta$ is computed by

$$\Theta = \frac{1}{U_r^2} - \frac{\rho_T}{\rho C_p}$$

(7)

Here $U_r$ is a reference velocity defined as follows:

$$U_r = \begin{cases} \epsilon c, & \text{if } |v| \leq \epsilon c \\ |v|, & \text{if } \epsilon c < |v| \leq c \\ c, & \text{if } |v| > c \end{cases}$$

(8)

where $\epsilon$ is a small number defined by the user for preventing singularities at stagnation points. In the present paper, $\epsilon = O(M)$ is adopted as chosen in Reference [10]. In so doing, the governing equations of interest become the preconditioned Euler equations, which can be expressed as follows:

$$\Gamma \frac{\partial q}{\partial t} + \nabla \cdot F = 0.$$ 

(9)

In a strict sense, Equation (9) is not conservative for the time-dependent flows due to the fact that the time accuracy of the equations is destroyed by preconditioning. However, solutions in the steady state remain unchanged [10,23]. Thus, it is not a problem to employ Equation (9) for present steady calculations.

The eigenvalues of the preconditioned system (9) in the direction of the unit vector $n = (n_x, n_y, n_z)$, which can be given by

$$\lambda \left( \Gamma^{-1} \left( \frac{\partial f_1}{\partial q} n_x + \frac{\partial f_2}{\partial q} n_y + \frac{\partial f_3}{\partial q} n_z \right) \right) = u_n, u_n, u_n, u_n', c', u_n' - c',$$

(10)

where all the eigenvalues that indicate wave speeds of the resulting system are modified to be the same magnitude order (see Reference [23] for detailed expressions of eigenvalues).

2.2. Spatial Discretization

In this work, the discontinuous Galerkin finite element method is considered to be implemented on each element with polynomial functions of a maximum degree equal to $p$, which results in the method being $(p+1)$th order accurate. Let $d$ be the number of space dimensions, and the number of degrees of freedom per equation per element is $N_p = \prod_{i=1}^{d} (p + i) / i$.

Assume that the computational domain $\Omega \subset R^d$ is subdivided into a collection of non-overlapping finite elements $\{\Omega_i\}$ and define the following space for both solution and test function:

$$V^p_m = \left\{ v_h \in [L_2(\Omega)]^m : v_h|_{\Omega_i} \in \left[ V^m_p \right] \forall \Omega_i \in \Omega \right\},$$

(11)

where $V^m_p$ is the space of vector-valued polynomials of a degree, at most $p$, and $m$ is the dimension of the conservative state vector (note: $m = 4$ for present 2D case and $m = 5$ for 3D).
Multiplying Equation (9) by a test function $W_h \in V_h^p$ and integrating by parts on each element $\Omega_i$, we obtain the weak formulation of the DG method:

$$\int_{\Omega_i} W_h^T \frac{\partial q_h}{\partial t} d\Omega + \int_{\partial \Omega_i} W_h^T F(q_h) \cdot n d\sigma - \int_{\Omega_i} \nabla W_h^T \cdot F(q_h) d\Omega = 0 \quad \forall W_h \in V_h^p,$$

where $q_h$ is the approximate solution at element $\Omega_i$ and can be represented by basis functions $\phi = \{ \phi_j \} \in V_h^p$ as follows:

$$q_h = \sum_{j=1}^{N_p} \tilde{q}_{ij}(t) \phi_j(x),$$

where the unknowns $\tilde{q}_{ij}(t)$ are the solution coefficients of the element to be solved. According to Equation (12), we have the following system of $N_p$ equations for each $\Omega_i$

$$\int_{\Omega_i} \phi_j \frac{\partial q_h}{\partial t} d\Omega + \int_{\partial \Omega_i} \phi_j F(q_h) \cdot n d\sigma - \int_{\Omega_i} \nabla \phi_j \cdot F(q_h) d\Omega = 0 \quad 1 \leq j \leq N_p. \quad (14)$$

Thus, the unknowns $\tilde{q}_{ij}(t)$ can be determined through solving system (14).

Keep in mind that the flux terms are not uniquely defined at element interface due to discontinuous function approximation. In order to treat this issue properly, the flux term $F(q_h) \cdot n$, appearing in the second term of Equation (14), is usually required to be replaced by a numerical flux $H(q_h^+, q_h^-, n)$, where $q_h^+$ is the neighboring element interface state, $q_h^-$ the internal interface state, and $n$ the direction normal to the interface. It is reported that a Roe-type flux function [24] satisfies the flux consistency conditions [25] associated with ensuring the formal accuracy of the scheme, and hence is adopted in the present paper. As mentioned above, the wave speeds of the governing equations were changed by preconditioning, and the related dissipative part of the numerical fluxes $H(q_h^+, q_h^-, n)$ are supposed to be modified accordingly.

Following the work of Reference [21], the resultant numerical flux can be written as

$$H(q_h^+, q_h^-, n) = \frac{1}{2} \left( F(q_h^+) + F(q_h^-) \right) \cdot n - \frac{1}{2} \sigma \Gamma |A_f| \Delta q \cdot n. \quad (15)$$

It can be noted that the effect of preconditioning is considered in the dissipative part of the last term of the Equation (15). $\sigma \in (0, 1]$ is a user-tuned factor that controls the amount of the dissipation [26].

### 2.3. Shock Capturing

In order to simulate high-speed shocked flows, a kind of shock capturing technique [8] is adopted through adding a small amount of artificial viscosity to the preconditioned governing Equation (9) in order to deal with possible strong shocks, which reads:

$$\Gamma \frac{\partial q}{\partial t} + \nabla \cdot F = \nabla \cdot (\mu \nabla q), \quad (16)$$

where $\nabla q$ is the gradient of the primitive variable vector and is computed by the BR2 scheme [27] in the present work, and $\mu$ is the artificial viscosity coefficient. Keep in mind that the artificial viscosity introduced is intended to suppress non-physical oscillations in the vicinities of the strong shocks without affecting the solution accuracy in smooth regions.

The smoothness of solution is measured by $s_i = \log_{10} S_i$ with $S_i = \frac{\int_{\Omega_i} (c_h - \hat{c}_h)^2 d\Omega}{\int_{\Omega_i} c_h^2 d\Omega}$, in which $c_h = \sum_{j=1}^{N_p} c_i(t) \phi_j(x) = \hat{c}_h + \sum_{i=\hat{N}_p}^{N_p} c_i(t) \phi_i(x)$ is the selected numerical primitive
variable, which could be pressure or density. In the present paper, pressure is used. Following the work of Reference [8], the resultant $\mu_i$ associated with $\Omega_i$ is computed by

$$
\mu_i = \begin{cases} 
0 & s_i \leq s_0 - \kappa \\
\frac{\mu_0}{\pi} \left( 1 + \sin \frac{\pi(s_i-s_0)}{2\kappa} \right) & s_0 - \kappa < s_i \leq s_0 + \kappa \\
\mu_0 & s_i > s_0 + \kappa
\end{cases}
$$

(17)

where $\mu_0$, $s_0$, and $\kappa$ are the control parameters that are usually chosen empirically by the users. It can be noted that the coefficients for each component in expansion (13) decay very quickly in smooth regions, while the strength of the discontinuity in shock regions will slow down the rate of such decay [8]. In other words, the elemental artificial viscosity coefficient $\mu_i$ is close to zero in smooth regions and increases in shock regions. By adopting such an artificial viscosity coefficient, the shock capturing technique can restrain the spurious numerical oscillation in shock regions, causing almost no degrading of solution accuracy in the smooth regions. However, the above piecewise constant artificial viscosities are discontinuous at the element interfaces, which may still cause small numerical oscillations in the vicinities of the strong shocks [28]. The artificial viscosities, made to be continuous, are usually preferred. In the present work, such continuity is achieved through adopting a post-smoothing process, realized in two steps:

Step 1. Transferring the elemental viscosity to the vertex of the element. The viscosity $v_k$ at vertex $k$ is determined by

$$
v_k = \sum_{i=1}^{n} w_i \mu_i,
$$

where $n$ is the number of the adjacent elements with weights $w_i = \frac{\mid\Omega_i\mid}{\sum_{j=1}^{n} \mid\Omega_j\mid}$.

Step 2. Computing the viscosity at each integral point of the element through a linear interpolation.

It can be noted that a kind of smoothing is realized through a weighted average process in Step 1.

In so doing, the semi-discrete weak form of the Equation (16) can be obtained from Equation (14) through adding the corresponding artificial viscosity terms, which reads:

$$
\int_{\Omega_i} \phi_j \frac{\partial q^b}{\partial t} \, d\Omega + \int_{\partial \Omega_i} \phi_j H(q^b^+, q^b^-, n) \, d\sigma - \int_{\Omega_i} \nabla \phi_j \cdot F(q^b) \, d\Omega \\
- \int_{\partial \Omega_i} \phi_j (v \nabla q^b) \cdot n \, d\sigma + \int_{\Omega_i} \nabla \phi_j \cdot (v \nabla q^b) \, d\Omega = 0 \quad 1 \leq j \leq N_p.
$$

(18)

Following the treatment of BR2 scheme [27], the additional flux term associated with artificial viscosity in Equation (18) is determined by

$$
v \nabla q^b \cdot n \big|_{\partial \Omega_i} = \begin{cases} 
\frac{1}{2}v \left( \nabla q^b_+ + \nabla q^b_- \right) \cdot n & \text{on } \left[ \Omega_i \cap \Omega_j, \forall \Omega_j \in \Omega \right] \\
v \nabla q^b \cdot n & \text{on } \partial \Omega
\end{cases}
$$

The quantities at the boundary are determined through enforcing the related boundary conditions.

2.4. Boundary Conditions

It can be learned from Equation (18) that the enforcement of high-order accurate boundary conditions is straightforward and is mainly involved with the numerical fluxes $H(q^b, q^b^-, n)$, which are weakly prescribed at the integral points of the boundary. The state of the primitive variables at each integral point of the boundary, $q^b = (p^b, u^b, v^b, w^b, T^b)^T$, is determined by boundary conditions, such as far field and solid wall boundary conditions.
2.4.1. Subsonic Far Field Boundary Condition

The far field boundary conditions are usually implemented to be nonreflecting. It is well known that the boundary state \( \mathbf{q}^b \) determined by Riemann invariants is nonreflecting for traditional Euler equations. However, relatively complicated characteristic boundary conditions \[29\] should be used for the present preconditioned Euler equations due to the fact that the wave speeds of the governing equations were changed by preconditioning. In the present paper, an alternative simplified treatment is adopted following the work of Turkel \[29\].

In order to make it clear, take the far field subsonic flow as an example; the boundary state at each integral point can be assigned simply by

\[
p^b = p^- , \quad u^b = u^\infty , \quad v^b = v^\infty , \quad w^b = w^\infty , \quad T^b = T^\infty ,
\]

where superscript ‘\( \infty \)’ denotes the freestream variables, and superscript ‘\( - \)’ the internal variables. Similar treatment can be taken for the subsonic outflow situation of interest.

2.4.2. Solid Wall Boundary Condition

In the case of inviscid flows, the fluid slips over the wall surface. In other words, the normal component of the velocity vanishes at the solid wall boundary. Such a slip wall condition employed is based on following boundary state:

\[
p^b = p^- , \quad \mathbf{u}^b = \mathbf{u}^- - (\mathbf{V} \cdot \mathbf{n})^- \mathbf{n} , \quad \mathbf{v}^b = \mathbf{v}^- - (\mathbf{V} \cdot \mathbf{n})^- \mathbf{n} , \quad \mathbf{w}^b = \mathbf{w}^- - (\mathbf{V} \cdot \mathbf{n})^- \mathbf{n} , \quad T^b = T^- .
\]

With the boundary variables determined by Equation (20), the numerical flux of Equation (15) at the solid wall becomes \( \mathbf{F}(\mathbf{q}^b) \cdot \mathbf{n} \). This means that the computation of fluxes on the wall boundary are in the same manner as the one in non-preconditioned DG schemes.

3. Treatment of Curved Boundaries

As pointed out in the Reference \[10\], an appropriate representation of the possibly curved boundary geometry is essential for preserving numerical accuracy. Here, some details concerning the present treatment of the curved boundary are described.

3.1. Geometry Representation

The integrals that appear in Equation (18) are intended to be calculated in reference elements \[30\] using the Gaussian quadrature rules. The physical coordinate \( \mathbf{x} = (x, y, z) \) associated with the reference coordinate \( \mathbf{\xi} = (\xi, \eta, \zeta) \) can be expressed as

\[
x = \sum_{i=1}^{n_d} \psi_i(\mathbf{\xi}) x_i \quad \forall \mathbf{\xi} \in \kappa,
\]

where \( n_d \) is the total number of the control points \( x_i \), and \( \psi_i(\mathbf{\xi}) \) are the corresponding shape functions that are fixed in the reference element \( \kappa \).

As illustrated in Figure 1, with the mapping relationship (21), the linear reference element can be used for representing simple straight-edged elements exactly, while quadratic reference elements can be applied for approximating curve-edged elements only. For enhancing the accuracy of high-order approximation, special treatment, so-called high-order re-mesh, is further required \[9\] and addressed in the following.
3. Treatment of Curved Boundaries

As pointed out in the Reference [10], an appropriate representation of the possibly curved boundary geometry is essential for preserving numerical accuracy. Here, some details concerning the present treatment of the curved boundary are described.

3.1. Geometry Representation

The integrals that appear in Equation (18) are intended to be calculated in reference elements [30] using the Gaussian quadrature rules. The physical coordinate \( x = (x, y, z) \) associated with the reference coordinate \( \zeta = (\zeta, \eta, \zeta) \) can be expressed as

\[
R = \sum_{i} \zeta_{i} \phi_{i}(\zeta), \quad \forall \zeta \in \mathcal{V},
\]

where \( n_{V} \) is the total number of the control points \( x_{i} \), and \( \phi_{i}(\zeta) \) are the corresponding shape functions that are fixed in the reference element \( \mathcal{V} \).

As illustrated in Figure 1, with the mapping relationship (21), the linear reference element can be used for representing simple straight-edged elements exactly, while quadratic reference elements can be applied for approximating curve-edged elements only. For enhancing the accuracy of high-order approximation, special treatment, so-called high-order re-mesh, is further required [9] and addressed in the following.

![Figure 1. Tetrahedron element in reference space (left) and physical space (right). The bold integer numbers are the indexes of the corresponding mapping points.](image)

3.2. High-Order Re-Mesh

It is recognized that the order of the DG method is limited by the order of the boundary representation [25]. Thus, it is necessary to have a proper treatment of the curved boundary. Here, a practical approach is proposed through high-order re-meshing in view of easy implementation and curved-mesh quality.

The approach is NURBS surface-based, and a NURBS surface is enhanced with a RBF interpolation in the high-order re-mesh procedure to have an appropriate treatment of curved boundaries. Specifically, NURBS-based surface representations are firstly generated based on the initial surface elements of the curved boundaries (often available in AutoCAD shapes) (see Figure 2a), and then used to create high-order-required control points on the curved surfaces or edges (see Figure 2b–d).

A key to form an appropriate set of high-order-required control points lies in adding new control points, which can be easily operated from the initial surface elements. Take the second order approximation as an example; the midpoints of the vertexes of the initial surface elements are usually selected to be control points, as illustrated by the red points in Figure 2c. Such newly added control points are certainly required to be mapped to the curved NURBS surface generated, which can be done by solving the problem associated with the rule of shortest displacement [11], which reads:

\[
\min_{a,b} \left| x_{\text{new}} - S(a,b) \right|^2, \quad \text{s.t.} \quad 0 \leq a, b \leq 1,
\]

where \( S(a,b) \) denotes a point on NURBS surface and \( x_{\text{new}} \) is a newly added control point. Keep in mind that with the NURBS surface generated, the gradient at the point \( S(a,b) \) can be directly computed. Thus, the above problem can be efficiently solved by gradient-based
optimization, such as the Gauss–Newton algorithm, to obtain the required new control points (see red points in Figure 2d).

Figure 2. Illustration of the high-order surface re-mesh. (a) Surface of curved boundary; (b) initial boundary mesh; (c) midpoints introduced; and (d) high order curved surface. The black points are the vertexes of the mesh, and the red points are the newly added control points.

However, as pointed out in References [9,31], such kind mapping may cause low quality or even invalid elements in the vicinity of the curved boundary with high curvature (see intersected surfaces existed at the vertex B of the element in Figure 3b). Such a problem is reported to be ameliorated by improving the quality of the mesh near the boundary through adopting optimization processes [32]. An alternative approach used widely is to propagate the boundary movement to the interior through mesh-deforming techniques, including the RBF interpolation method [31], the linear elasticity method [9,33], etc. Among them, the RBF interpolation method has the ability to achieve high-quality meshes at a relatively low computational costs [31], and hence, whether the invalid elements exist or not, the RBF interpolation is applied as the last step of the present approach to retain the element quality (see Figure 3c). In order to assure the quality of the resulting mesh, an additional step of quality check is usually required. But it is learned from the test cases presented (see Section 5) that it is not necessary after having the RBF interpolation.
Applying the RBF interpolation method, the displacement $s(x)$ of the point $x$ in the computational domain can be expressed as

$$s(x) = \sum_{j=1}^{n_b} a_j \psi \left( \| x - x_{b_j} \| \right),$$  

(22)

where $x_{b_j}, j = 1, 2, \ldots, n_b$ are the positions of the boundary control points, $\psi \left( \| x - x_{b_j} \| \right)$ is the $j$th radius basis function for the displacement field $s$, and $a_j$ is the corresponding unknown coefficient in that basis. With the given boundary displacements, $s(x_{b_i}), i = 1, 2, \ldots, n_b$, the unknown coefficients $a_j$ can be determined by solving the following system of $n_b$ equations:

$$s(x_{b_i}) = \sum_{j=1}^{n_b} a_j \psi \left( \| x_{b_i} - x_{b_j} \| \right).$$  

(23)

Once the coefficients $a_j$ are determined, the interior displacement of any point in the domain can be calculated by Equation (22). Concerning further details of the RBF interpolation, we refer to the work of Reference [31].

4. Time Stepping Schemes

By assembling all the elemental contributions together, the semi-discrete form of Equation (18) governing the evolution in time of the discrete solution can be written as

$$M \frac{\partial \hat{q}}{\partial t} = R,$$  

(24)

where $M$ is the block diagonal matrix in which each block corresponds to a specific element, $\hat{q}$ is the vector of all the solution coefficients to be solved, and $R$ is the vector of the residual. Both explicit and implicit time stepping schemes are used in the present paper for discretizing the semi-discrete Equation (24) in time. Specifically, an implicit LU-SGS scheme will be mainly addressed after having a brief description of the explicit Runge–Kutta scheme.

4.1. Explicit Runge–Kutta Scheme

The explicit three-stage third-order TVD Runge–Kutta scheme [5] is used to advance the solution of Equation (24) from time $t$ to time $t + \Delta t$, which reads:

$$\hat{q}^{(1)} = \hat{q}^n + \Delta t M^{-1} R(\hat{q}^n),$$

$$\hat{q}^{(2)} = \frac{3}{4} \hat{q}^n + \frac{1}{4} \left[ \frac{\hat{q}^{(1)} + \Delta t M^{-1} (\hat{q}^n) R(\hat{q}^{(1)})}{\Delta t M^{-1} R(\hat{q}^n)} \right],$$

$$\hat{q}^{n+1} = \frac{1}{3} \hat{q}^n + \frac{2}{3} \left[ \frac{\hat{q}^{(2)} + \Delta t M^{-1} (\hat{q}^n) R(\hat{q}^{(2)})}{\Delta t M^{-1} R(\hat{q}^n)} \right].$$  

(25)
In addition to the preconditioning mentioned, the local time stepping is also used to improve the convergence speed to steady state solutions. In specific, the local time step $\Delta t_i$ is computed for each element $\Omega_i$ by

$$\Delta t_i = \frac{CFL}{2p+1} \cdot \frac{|\Omega_i|}{\int_{\partial \Omega_i} \max(|u'| + c'), \max(|u'| + c')^-} d\sigma,$$

where $p$ is the approximate order as mentioned, and CFL is the Courant number. In general, CFL could be 1 for 2D and less than 1 for 3D applications of the DG method.

### 4.2. Implicit LU-SGS Scheme

The implicit backward Euler time discretization of Equation (24) can be written as

$$\left( \frac{M}{\Delta t} - \frac{\partial R^n}{\partial \tilde{q}} \right) \Delta \tilde{q}^n = R^n,$$  

(27)

where $\Delta \tilde{q}^n = \tilde{q}^{n+1} - \tilde{q}^n$, in which the superscripts $n$ and $n+1$ denote the current and the next time steps, respectively. $\frac{\partial R^n}{\partial \tilde{q}}$ is the Jacobian matrix of the DG spatial discretization, which is computed analytically (except for the dissipative part of the numerical flux using divided differencing [34]) in this paper.

Recall that $M = \text{diag}(M_1, M_2, \ldots, M_N_c)$, and let $C_i$ be a set of the indexes of the neighboring connected elements of $\Omega_i$, define $A = \frac{M}{\Delta t} - \frac{\partial R^n}{\partial \tilde{q}}$ as the global system matrix, and the Equation (27) can then be written as

$$A \Delta \tilde{q}^n = R^n,$$

(28)

in which

$$A = [A_{ij}] \text{ with } A_{ij} = \begin{cases} \frac{M_i}{\Delta t} - \sum_{k \in C_i} \frac{\partial R^n_i}{\partial \tilde{q}_j} & i = j \\ -\frac{\partial R^n_j}{\partial \tilde{q}_j} & i \neq j \end{cases}.$$

The resultant system of Equation (28) encapsulates the implicit iteration schemes, and it can be solved iteratively to converge to a steady state. The traditional LU-SGS scheme consists of a forward and a backward sweeping through all the computational elements in a sequential order [16], which can be written as

Forward sweep : $\Delta \tilde{q}^n_i = A_i^{-1} \left[ R^n_i - \sum_{j \in C_i} A_{ij} \Delta \tilde{q}^n_j \right]$, $i = 1, 2, \ldots, N_c - 1, N_c$.  

(29)

Backward sweep : $\Delta \tilde{q}^n_i = \Delta \tilde{q}^n_i - A_i^{-1} \sum_{j \in C_i} A_{ij} \Delta \tilde{q}^n_j$, $i = N_c, N_c - 1, \ldots, 2, 1$.  

(30)

### 5. Results

In this section, the performance of the present DG method is investigated by simulating a set of typical two- and three-dimensional flow problems. The test cases include flows over a symmetric NACA0012 airfoil, an asymmetric RAE2822 airfoil, a hemispherical headform, as well as more practical flows over the ONERA M6 wing. All the simulations are carried out with the present implicit LU-SGS algorithm in view of having implicit accelerations. The explicit scheme is used only for the cases with comparisons between implicit and explicit schemes.

#### 5.1. Low-Speed Flows over a NACA0012 Airfoil

In order to verify whether our preconditioned high-order DG method can be used for nearly incompressible flow simulations, flows over a symmetric NACA0012 airfoil with zero angle of attack ($\alpha = 0^\circ$) at four selected Mach numbers, 0.1, 0.01, 0.001, and 0.0001, are first simulated. Without loss of generality, the order of DG method is fixed to be fourth
In order to verify whether our preconditioned high-order DG method can be used, the simulations, and a fully unstructured mesh having only 3527 triangle elements (see Figure 4) is used in view of adopting a fourth-order accurate DG method.

![Computational mesh around NACA0012 airfoil.](image)

**Figure 4.** Computational mesh around NACA0012 airfoil.

The behaviors of low speed in convergence (stiffness problem [18,19]) at low Mach numbers can be learned from the convergence histories of three representative Mach numbers, as shown in Figure 5. It seems that the high-order DG method itself (without preconditioning) has the ability to cope with the flows at relatively low Mach numbers (see $Ma = 0.1$ in Figure 5a), which is different from low order methods [35]. However, difficulties in convergence appear as the Mach number further reduces. Such a situation is greatly ameliorated, as illustrated in Figure 5b,c, through the help of preconditioning.

![The convergence histories of flows over NACA0012 airfoil.](image)

**Figure 5.** The convergence histories of flows over NACA0012 airfoil. (a) $Ma = 0.1$; (b) $Ma = 0.01$; and (c) $Ma = 0.001$.

The effects of very low speed on the accuracy of the solutions can be learned from the corresponding contours of the Mach number. The typical contours of $Ma = 0.0001$ are presented in Figure 6. It can be noted that the developed code, without preconditioning, fails to obtain the correct physical solution (reaching unphysical solution [18], on the left-hand side) in view of the chaotic, asymmetric, and poor smoothness of the contours, while the code with preconditioning can capture smooth contours of the Mach number (see the right-hand side of Figure 6).
5.2. Flows over an Asymmetric RAE2822 Airfoil

The capabilities of the developed code equipped with shock capturing, described in Section 2.3, are further verified to deal with both transonic compressible and nearly incompressible flows over an asymmetric RAE2822 airfoil. The computational domain is discretized by a fully unstructured mesh having 4386 triangle elements, as shown in Figure 7. A set of simulations with different flow conditions was carried out with different accurate orders. Here, numerical results of third-order ($p = 2$) accurate simulations are presented in Figure 8 for the nearly incompressible flow ($Ma = 0.01$ and $\alpha = 1.89^\circ$), and in Figure 9 for the transonic shocked flow ($Ma = 0.729$ and $\alpha = 2.31^\circ$). It can be noted from Figure 8 that the obtained surface pressure coefficients (see Figure 8a) are in good agreement with other researchers’ results [35,36] and the Mach contours are distributed smoothly (see Figure 8b). Additionally included in Figure 9a are the experimental data [37] and other researchers’ results [38]. Quite a good agreement can also be noted in view of the strength or location of the captured shock. Frankly, such shocked solution is based on the strength and the affected region of the artificial viscosity. As mentioned in the References [8,28], the region functioned with artificial viscosity is as narrow as possible for the given mesh. The results can be further improved when using local refinement of the mesh near the shock, but this is not considered here.

Figure 6. Contours of the Mach number for flows over NACA0012 airfoil. (a) Without preconditioning; (b) with preconditioning.

Figure 7. Computational mesh around RAE2822 airfoil.

(a) (b)
Figure 8. Computational results for the nearly incompressible flow around a RAE2822 airfoil. (a) Surface pressure coefficients (Cao [35], Puoti [36]); (b) Contours of Mach number.

Figure 9. Computational results for the transonic flow around a RAE2822 airfoil. (a) Surface pressure coefficients (Experiment [37], Yasue [38]); (b) Contours of Mach number.

It should be mentioned that based on the preconditioned Euler equation, the shock capturing is implemented and only functions when shock appears, and hence the performance of the developed code in convergence is still kept for the nearly incompressible flow. Actually, both explicit and implicit time marching schemes described in Section 4 are used for the code developed. The implicit speed up of convergence histories is remarkable for the above nearly incompressible flow, as illustrated in Figure 10, in view of being consuming in both iteration and time.
5.3. Flows over a Hemispherical Headform

It is pointed out that the above relatively accurate results of the flows over the airfoils are obtained with the help of the curved boundary treatment presented in Section 3. In order to have a view of the effect of curved boundary treatment on the numerical solution, a typical flow over a hemispherical headform [35–39] at the low Mach number 0.01 with zero angle of attack is simulated. Following the computational domain defined in the Reference [39], a relatively coarse mesh, having 27,976 tetrahedral elements (see Figure 11), is used in the present third-order \( p = 2 \) accurate simulation. As shown in Figure 12a, a meaningful high-order accurate solution can be obtained with the present curved boundary treatment, while it seems that the DG method fails to obtain the correct solution without any curved boundary treatment, judging by the poor smoothness of the Mach contours evidenced in Figure 12b. Such a phenomenon can also be observed in the distribution of corresponding surface pressure coefficients against the ratio of distance along the boundary of the head diameter, as shown in Figure 13. Additionally in Figure 13 are the experimental data [40] and other researchers’ results [35,39], which are included in order to have a view of agreement. In particular, the deviation between the numerical results with or without curved boundary treatment can be noted, particularly on the surface with relatively coarse meshes, which indicates that an appropriate curved boundary treatment is most helpful for seeking high-order accurate numerical solutions.

Figure 10. Comparisons of convergence histories. (a) Vs iteration; (b) Vs time.

Figure 11. Computational mesh of the hemispherical headform.
5.3. Flows over a Hemispherical Headform

Along the boundary of the head diameter, as shown in Figure 13. Additionally in Figure 13 are the experimental data [40] and other researchers’ results [35,39], which are included in order to have a view of agreement. In particular, the deviation between the numerical results with or without curved boundary treatment can be noted, particularly on the surface with relatively coarse meshes, which indicates that an appropriate curved boundary treatment is most helpful for seeking high-order accurate numerical solutions. Nearly incompressible flow at Mach number $Ma = 0.01$ and typical transonic shocked flow at $Ma = 0.84$ with the same angle of attack ($\alpha = 3.06^\circ$) are selected and tested to show the abilities of coping with both nearly incompressible and compressible flows for the methods developed. A relatively coarse mesh, having only 62,897 tetrahedron elements, is generated and used in the present third-order ($p=2$) accurate flow simulations mentioned. The corresponding meshes on the surface of the wing and in the symmetric plane are presented in Figure 14.

5.4. Flows over the ONERA M6 Wing

Finally, flows over an aerodynamic ONERA M6 wing [41–43] are considered to validate the developed DG code in relatively more practical simulations. Nearly incompressible flow at Mach number $Ma = 0.01$ and typical transonic shocked flow at $Ma = 0.84$ with the same angle of attack ($\alpha = 3.06^\circ$) are selected and tested to show the abilities of coping with both nearly incompressible and compressible flows for the methods developed. A relatively coarse mesh, having only 62,897 tetrahedron elements, is generated and used in the present third-order ($p=2$) accurate flow simulations mentioned. The corresponding meshes on the surface of the wing and in the symmetric plane are presented in Figure 14.
5.4. Flows over the ONERA M6 Wing

Finally, flows over an aerodynamic ONERA M6 wing [41–43] are considered to validate the developed DG code in relatively more practical simulations. Nearly incompressible flow at Mach number $M_a = 0.01$ and typical transonic shocked flow at $M_a = 0.84$ with the same angle of attack ($\alpha = 3.06^\circ$) are selected and tested to show the abilities of coping with both nearly incompressible and compressible flows for the methods developed. A relatively coarse mesh, having only 62,897 tetrahedron elements, is generated and used in the present third-order ($p = 2$) accurate flow simulations mentioned.

The corresponding meshes on the surface of the wing and in the symmetric plane are presented in Figure 14.

In Figure 15, the obtained contours of the pressure coefficient on the surface of the wing are first presented for the mentioned transonic case, together with the isolines of the pressure coefficient in the symmetric plane, in order to have a global view of captured flow fields. The corresponding distributions of the surface pressure coefficient at six typical sections along the wingspan are also presented in Figure 16. It can be observed that the obtained results at the sections are all in a reasonable agreement with the numerical results [41] and experimental data [42] in view of the strength or location of the captured shock.

![Figure 15. Contours of pressure coefficient ($Ma = 0.84$, $\alpha = 3.06^\circ$).](image)
Figure 15. Contours of pressure coefficient ($Ma = 0.84, \alpha = 3.06^\circ$). (a) (b) (c) (d) (e) (f)

Figure 16. Distributions of surface pressure coefficients at six wingspan sections (Mani [41], Experiment [42]). (a) $\eta = 0.20$; (b) $\eta = 0.44$; (c) $\eta = 0.65$; (d) $\eta = 0.80$; (e) $\eta = 0.90$; and (f) $\eta = 0.95$.

Following the Reference [43], the obtained distributions of the surface pressure coefficient at the specific wingspan section $\eta = 0.80$ are then presented (see Figure 17) for the case of the nearly incompressible flow mentioned. It can be noted that the results captured are in good agreement with other numerical results [43]. The corresponding contours of the surface pressure coefficient of the wing, together with the field isolines of the pressure coefficient in the symmetric plane, are also presented in Figure 18 to have a view of flow fields. As shown in Figure 19, a significant improvement is again achieved in the convergence rate when preconditioning is adopted. The successful flow simulations of
the above tested Mach numbers indicate that the DG methods presented function for both nearly incompressible and transonic shocked flows.

![Graph](image1.png)

**Figure 17.** Distributions of surface pressure coefficients (Turkel [43]).

![Surface Pressure Coefficients](image2.png)

**Figure 18.** Contours of pressure coefficient ($Ma = 0.01, \alpha = 3.06^\circ$).

![Convergence Graph](image3.png)

**Figure 19.** Convergence histories.
6. Conclusions Remarks

A high-order DG method for solving the preconditioned Euler equations with explicit or implicit time marching scheme is presented. A detailed description is given of a practical implementation of a precondition matrix of the type of Weiss and Smith, as well as of the DG spatial discretization scheme employed, with particular emphasis on the artificial viscosity-based shock capturing techniques and the proposed treatment of curved boundaries through adopting a NURBS surface equipped with RBF interpolation in order to propagate the boundary displacement to the interior of the mesh. The method is verified by considering a series of two- and three-dimensional test cases, including flows around a NACA0012 airfoil, a RAE2822 airfoil, a hemispherical headform, and an aerodynamic M6 wing. Numerical tests show that the present DG method functions for both transonic and nearly incompressible flow simulations. Numerical results also show that when a freestream Mach number reduces to an incompressible limit, the preconditioning adopted becomes necessary to accelerate convergence toward a steady state in the process of obtaining relatively accurate solutions. The deviation between the numerical results with or without curved boundary treatment confirms that an appropriate curved boundary treatment is helpful for seeking high-order accurate numerical solutions under the situation of relatively coarse meshes, particularly on the surface and with high curvature. The present DG method is implemented with unstructured meshes, and hence has the ability to cope with practical flows over more complex geometries. However, the computational cost grows rapidly with an increasing order of approximation. Thus, extension for further speedup, such as parallelization, can be considered. In addition, the governing equations can be easily updated with physical viscous terms to treat more practical viscous flows, which will be addressed in future research.

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

Funding: This research was funded by the National Natural Science Foundation of China (grant number 11972189, 12102185 and 12102188), the China Postdoctoral Science Foundation (grant number 2021M701693) and the Natural Science Foundation of Jiangsu Province (grant number BK20190391). The APC was funded by BK20190391.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Birken, P.; Gassner, G.; Haas, M.; Munz, C.D. Preconditioning for modal discontinuous Galerkin methods for unsteady 3D Navier–Stokes equations. J. Comput. Phys. 2013, 240, 20–35. [CrossRef]
2. Franciolini, M.; Fidkowski, K.J.; Crivellini, A. Efficient discontinuous Galerkin implementations and preconditioners for implicit unsteady compressible flow simulations. Comput. Fluids 2020, 203, 104542. [CrossRef]
3. Pazner, W.; Persson, P. Stage-parallel fully implicit Runge–Kutta solvers for discontinuous Galerkin fluid simulations. J. Comput. Phys. 2017, 335, 700–717. [CrossRef]
4. Nicolas, C.; Roland, B.; Eric, S. Performance of DG methods based on different variables for low Mach number flows. Commun. Nonlinear Sci. Numer. Simul. 2020, 85, 105580. [CrossRef]
5. Cockburn, B.; Shu, C.W. The Runge–Kutta Discontinuous Galerkin Method for Conservation Laws V. J. Comput. Phys. 1997, 141, 199–224. [CrossRef]
6. Dumbser, M.; KaSer, M. Arbitrary high order non-oscillatory finite volume schemes on unstructured meshes for linear hyperbolic systems. J. Comput. Phys. 2007, 221, 693–723. [CrossRef]
7. 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]
8. Persson, P.O.; Peraire, J. Sub-Cell Shock Capturing for Discontinuous Galerkin Methods. In Proceedings of the 44th AIAA Aerospace Sciences Meeting and Exhibit, Reno, NV, USA, 9–12 January 2006.

9. Siebenborn, M.; Schulz, V.; Schmidt, S. A curved-element unstructured discontinuous Galerkin method on GPUs for the Euler equations. Comput. Vis. Sci. 2013, 15, 61–73. [CrossRef]

10. Bassi, F.; Bartolo, C.D.; Hartmann, R.; Nick, A. A discontinuous Galerkin method for inviscid low Mach number flows. J. Comput. Phys. 2009, 228, 3996–4011. [CrossRef]

11. Asue, K. Implicit Discontinuous Galerkin Method for RANS Simulation Utilizing Pointwise Relaxation Algorithm. Comput. Mech. 2010, 7, 510–533. [CrossRef]

12. Crivellini, A.; Bassi, F. An implicit matrix-free Discontinuous Galerkin solver for viscous and turbulent aerodynamic simulations. Comput. Fluids 2011, 50, 81–93. [CrossRef]

13. Fernández, P.; Nguyen, N.C.; Peraire, J. The hybridized Discontinuous Galerkin method for Implicit Large-Eddy Simulation of transitional turbulent flows. J. Comput. Phys. 2017, 336, 308–329. [CrossRef]

14. Frangioli, M.; Botti, L.; Colombo, A.; Crivellini, A. p-Multigrid matrix-free discontinuous Galerkin solution strategies on the under-resolved simulation of incompressible turbulent flows. Comput. Fluids 2020, 206, 104558. [CrossRef]

15. Ali, A.; Syed, K.S.; Hassan, A.; Ahmad, I.; Ismail, M.A. On parallel performance of an implicit discontinuous Galerkin compressible flow solver based on different linear solvers. In Proceedings of the IEEE 14th International Multitopic Conference, Karachi, Pakistan, 22–24 December 2011.

16. Yoon, S.; Jameson, A. Lower-upper Symmetric-Gauss-Seidel method for the Euler and Navier-Stokes equations. AIAA J. 1988, 26, 1025–1026. [CrossRef]

17. Zeifang, J.; Kaiser, K.; Beck, A.; Schütz, J.; Munz, C.D. Efficient high-order discontinuous Galerkin computations on low Mach number flows. Commun. Appl. Math. Comput. Sci. 2018, 13, 243–270. [CrossRef]

18. Giro, A.; Bartolo, C.D.; Hartmann, R.; Bassi, F. Discontinuous Galerkin solution of preconditioned Euler equations for very low Mach number flows. Int. J. Numer. Methods Fluids 2010, 63, 449–467. [CrossRef]

19. Giro, A.; Renda, S.; Bartolo, C.D.; Hartmann, R.; Bassi, F. A high-order accurate discontinuous Galerkin finite element method for laminar Mach number flows. Int. J. Numer. Methods Fluids 2013, 72, 43–68. [CrossRef]

20. Landmann, B. A Parallel Discontinuous Galerkin Code for the Navier-Stokes and Reynolds-Averaged Navier-Stokes Equations. Ph.D. Thesis, University of Stuttgart, Stuttgart, Germany, 2008.

21. Hauke, G.; Hughes, T. A comparative study of different sets of variables for solving compressible and incompressible flows. Comput. Methods Appl. Mech. Eng. 1998, 153, 1–44. [CrossRef]

22. Venkateswaran, S.; Weiss, J.M.; Merkle, C.L.; Choi, Y.H. Propulsion-related flow fields using the preconditioned Navier-Stokes equations. In Proceedings of the 28th Joint Propulsion Conference and Exhibit, Nashville, TN, USA, 6–8 July 1992.

23. Weiss, J.M.; Smith, W.A. Preconditioning applied to variable and constant density time-accurate flows on unstructured meshes. In Proceedings of the 25th AIAA Fluid Dynamics Conference, Colorado Springs, CO, USA, 20–23 June 1994.

24. Roe, P.L. Approximate Riemann solvers, parameter vectors, and difference schemes. J. Comput. Phys. 1981, 43, 357–372. [CrossRef]

25. Bassi, F.; Rebay, S. High-order accurate discontinuous finite element solution of the 2D Euler equations. J. Comput. Phys. 1997, 138, 251–283. [CrossRef]

26. Bui, T.T. A parallel, finite-volume algorithm for large-eddy simulation of turbulent flows. Comput. Fluids 2000, 29, 877–915. [CrossRef]

27. Bassi, F.; Crivellini, A.; Rebay, S.; Savini, M. Discontinuous Galerkin solution of the Reynolds-averaged Navier-Stokes and k–ω turbulence model equations. Comput. Fluids 2005, 34, 507–540. [CrossRef]

28. Klockner, A.; Warburton, T.; Hesthaven, J.S. Viscous Shock Capturing in a Time-Explicit Discontinuous Galerkin Method. Math. Model. Nat. Phenom. 2011, 6, 57–83. [CrossRef]

29. Turkel, E.; Fiterman, A.; Vanleer, B. Preconditioning and the limit to the incompressible flow equations. In Computing the Future: Frontiers of Computational Fluid Dynamics 1994; Caughey, D.A., Hafez, M.M., Eds.; Wiley: New York, NY, USA, 1994; pp. 215–234.

30. Hong, L.; Baum, J.D.; Hner, R.L. A discontinuous Galerkin method based on a Taylor basis for the compressible flows on arbitrary grids. J. Comput. Phys. 2008, 227, 8875–8893.

31. Kashi, A.; Hong, L. Curved mesh generation using radial basis functions. In Proceedings of the 46th AIAA Fluid Dynamics Conference, Washington, DC, USA, 13–17 June 2016.

32. Toulorge, T.; Guezaine, C.; Remacle, J.-F.; Lambrecht, J. Robust untangling of curvilinear meshes. J. Comput. Phys. 2013, 254, 8–26. [CrossRef]

33. Xie, Z.Q.; Sevilla, R.; Hassan, O.; Morgan, K. The generation of arbitrary order curved meshes for 3D finite element analysis. Comput. Mech. 2013, 51, 361–374. [CrossRef]

34. Bramkamp, F.D.; Bucker, H.M.; Rasch, A. Using exact Jacobians in an implicit Newton–Krylov method. Comput. Fluids 2006, 35, 1063–1073. [CrossRef]

35. Cao, C.; Chen, H.; Zhang, J. Preconditioned Gridless Methods for Solving Three Dimensional Euler Equations at Low Mach Numbers. Int. J. Modeling Simul. Sci. Comput. 2020, 11, 2050055. [CrossRef]

36. Puoti, V. Preconditioning method for low-speed flows. AIAA J. 2003, 41, 817–830. [CrossRef]
37. Cook, P.H.; Mcdonald, M.A.; Firmin, M.C.P. Airfoil RAE2822—Pressure Distributions, and Boundary Layer and Wake Measurements; AGARD Advisory Report No. 138; Advisory Group for Aerospace Research and Development North Atlantic Treaty Organization: Neuilly sur Seine, France, 1979.

38. Yasue, K.; Ohnishi, N.; Sawada, K. A Pointwise Relaxation Computation of Viscous Compressible Flowfield Using Discontinuous Galerkin Method. In Proceedings of the 36th AIAA Fluid Dynamics Conference and Exhibit, San Francisco, CA, USA, 5–8 June 2006.

39. Hejranfar, K.; Kamali-Moghadam, R. Preconditioned characteristic boundary conditions for solution of the preconditioned Euler equations at low Mach number flows. J. Comput. Phys. 2012, 231, 4384–4402. [CrossRef]

40. Rouse, H.; McNown, J.S. Cavitation and Pressure Distribution, Head Forms at Zero Angle of Yaw, Studies Engineering; State University of Iowa: Iowa City, IA, USA, 1948; Volume 32.

41. Mani, M.; Ladd, J.; Cain, A.; Bush, R. An assessment of one- and two-equation turbulence models for internal and external flows. In Proceedings of the 28th Fluid Dynamics Conference, Snowmass Village, CO, USA, 29 June–2 July 1997.

42. Schmitt, V.; Charpin, F. Pressure Distributions on the ONERA M6 Wing at Transonic Mach Numbers; AGARD Advisory Report No. 138; Advisory Group for Aerospace Research and Development North Atlantic Treaty Organization: Neuilly sur Seine, France, 1979.

43. Turkel, E. Preconditioning methods for low-speed flows. In Proceedings of the 14th AIAA Applied Aerodynamics Conference, New Orleans, LA, USA, 18–20 June 1996.