AIAA SciTech Forum 7-11 January 2019, San Diego, California AIAA Scitech 2019 Forum

Check for updates

High-Order Hyperbolic Navier-Stokes Reconstructed Discontinuous Galerkin Method

Lingquan Li * Jialin Lou[†] and Hong Luo[‡] North Carolina State University, Raleigh, NC 27695, United States

and

Hiroaki Nishikawa §

National Institute of Aerospace, Hampton, VA 23666, United States

In this paper, we develop a reconstructed discontinuous Galerkin hyperbolic Navier-Stokes method based on the Galerkin formulation with a full basis matrix used as a test function. In the hyperbolic discontinuous Galerkin method for model equations, the formulation with a full basis matrix is known to be robust and leads to accurate explicit time-stepping schemes. Its extensions to a hyperbolic Navier-Stokes system, however, present a challenge in expressing derivatives of the conservative variables in terms of the gradient variables (corresponding to the gradients of the primitive variables) and their derivatives. To overcome the difficulty, we propose defining polynomial approximations in the primitive variables, and performing the weak formulation for the conservative system with pseudo/physical time derivatives expressed in terms of the primitive variables. This approach greatly simplifies the Galerkin formulation for the hyperbolic Navier-Stokes system. The resulting discretization is further improved by the reconstructed discontinuous Galerkin methodology, i.e., higher-order without introducing extra degrees of freedom.

1. Introduction

This paper reports further progress in the development of the hyperbolic reconstructed discontinuous Galerkin (rDG) for the Navier-Stokes (NS) equations. The hyperbolic rDG method is an efficient discretization method that combines the hyperbolic diffusion/viscous formulation and the rDG discretization methods. The hyperbolic diffusion/viscous formulation is a first-order system formulation of diffusion/viscous terms with extra variables, called the gradient variables, added to form a system and with pseudo time terms added to render the system hyperbolic [1, 2]. The rDG method is a general framework for constructing efficient high-order schemes with reconstruction techniques, having the finite-volume (FV) and discontinuous Galerkin (DG) methods as special cases [3, 4]. As we have shown in our previous developments [5, 6, 7, 8, 9], the two approaches can be combined systematically to simplify the discretization of diffusion/viscous terms, improve gradient accuracy, accelerate iterative convergence, and achieve higher-order accuracy with fewer numbers of degrees of freedom than DG methods. These advantages have been demonstrated for diffusion with scalar and tensor diffusion coefficients [5, 6], nonlinear diffusion [7] advection-diffusion equations [8], and the NS equations [9]. In Ref.[9], we developed a new hyperbolic Navier-Stokes (HNS) system, HNS20G, with the gradients of the primitive variables as the gradient variables in order to simplify high-order discretizations. The new system enables straightforward high-order primitive-variable reconstruction with the gradient variables, and led to high-order FV discretizations of up to fourth-order accuracy in the inviscid limit. In this paper, we extend the methodology to the rDG discretization method based on the Galerkin formulation in such a manner as to allow extensions to arbitrary order of accuracy.

The hyperbolic rDG method consists of two steps: hyperbolic DG and rDG. In the hyperbolic DG, high-order derivatives in the polynomials of the primary variables are replaced by the gradient variables and their derivatives, and all cross derivatives are unified (i.e., a single unknown to represent equivalent cross derivatives). This step eliminates all discrete equations for high-order derivatives of the primary polynomials. Moreover, the gradients of the gradient variables can be used to build even higher-order polynomials for the primary variables. This unique polynomial construction couples discrete variables and generates a matrix of basis functions for the solution polynomial representation. Further improvement comes in the next rDG step. The polynomials used in the flux and source evaluations are upgraded by reconstructing higher-order derivatives from the underlying DG polynomials. The resulting hyperbolic rDG schemes are known to outperform conventional DG schemes of the same number of discrete unknowns: higher-order inviscid and gradient accuracy and stiffness due to second-derivative viscous terms completely eliminated

^{*}Ph.D Student, Department of Mechanical and Aerospace Engineering, Student Member AIAA.

[†]Ph.D Student, Department of Mechanical and Aerospace Engineering, Student Member AIAA.

^{*}Professor, Department of Mechanical and Aerospace Engineering, Associate Fellow AIAA.

[§]Associate Research Fellow, National Institute of Aerospace, Associate Fellow AIAA.

[5, 6, 7, 8, 9]. However, since the HNS system has the conservative variables as the primary variables and the gradients of the primitive variables as the gradient variables, the derivative replacement in the hyperbolic DG can be quite complicated, especially for higher-order discretizations [10]. Currently, only P_0 -based hyperbolic rDG schemes have been developed for the HNS system [9], which are nothing but high-order FV schemes.

The hyperbolic DG method is based on a weak formulation. The Galerkin formulation corresponds to choosing the basis matrix as a test function. A simplified approach is possible: choose a simplified version of the basis matrix such as the diagonal part of of basis matrix, which is a Petrov-Galerkin-type formulation. In the previous papers, the Petrov-Galerkin was employed for diffusion in Refs. [5, 6], and later, the Galerkin formulation was found to be more robust for advection-diffusion problems [8]. Further later, it was discovered that the Galerkin formulation enables accurate unsteady advection-diffusion computations by explicit time-stepping schemes [11]. For the HNS system, only the Petrov-Galerkin formulation based on a P_0 solution representation has been considered so far because of the complication in expressing the derivatives of the conservative variables in terms of the derivatives of the primitive variables. This approach generates only FV schemes and cannot realize the full potential of embracing the whole hierarchy of the rDG methods. In this paper, we overcome this limitation by defining polynomial approximations exclusively for the primitive variables, and performing the Galerkin formulation for the conservative HNS system with pseudo/physical time derivatives expressed in terms of the primitive variables. This formulation paves the way for a systematic construction of a wide range of rDG schemes of arbitrary order. A slight complication arises in the solutiondependent mass matrices for the time derivatives. We will discuss how they can be efficiently evaluated to minimize the overhead. Numerical results are presented for verification with manufactured solutions and realistic viscous flow problems to demonstrate improved robustness and high-order accuracy.

The paper is organized as follows. In Section 2, we review the target HNS system. In Section 3, the hyperbolic rDG discretization based on the consistent Galerkin discretization is discussed. In Section 4, time integration schemes used for unsteady simulations is described. In Section 6, numerical results are presented for verification with manufactured solutions, and realistic viscous flow problems. In Section 5, boundary conditions implemented for the presented method are discussed. Finally, the paper concludes with remarks.

2. Hyperbolic Navier-Stokes System: HNS20G

Consider the compressible NS equations:

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}_k}{\partial x_k} = \frac{\partial \mathbf{G}_k}{\partial x_k},\tag{2.1}$$

where k indicates the coordinate direction, $\mathbf{F}_1 = \mathbf{F}_x$, $\mathbf{F}_2 = \mathbf{F}_y$, $\mathbf{F}_3 = \mathbf{F}_z$, $x_1 = x$, $x_2 = y$, and $x_3 = z$,

$$\mathbf{U} = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ \rho w \\ \rho e \end{pmatrix}, \quad \mathbf{F}_{x} = \begin{pmatrix} \rho u \\ \rho u^{2} + p \\ \rho u v \\ \rho u w \\ u(\rho e + p) \end{pmatrix}, \quad \mathbf{F}_{y} = \begin{pmatrix} \rho v \\ \rho u v \\ \rho v v \\ \rho v^{2} + p \\ \rho v w \\ v(\rho e + p) \end{pmatrix}, \quad \mathbf{F}_{z} = \begin{pmatrix} \rho w \\ \rho u w \\ \rho v w \\ \rho w w \\ \rho w^{2} + p \\ w(\rho e + p) \end{pmatrix}, \quad (2.2)$$

$$\mathbf{G}_{x} = \begin{pmatrix} 0 \\ \tau_{xx} \\ \tau_{yx} \\ \tau_{zx} \\ \tau_{zx} \\ u\tau_{xx} + v\tau_{xy} + w\tau_{xz} - q_{x} \end{pmatrix}, \quad \mathbf{G}_{y} = \begin{pmatrix} 0 \\ \tau_{xy} \\ \tau_{yy} \\ \tau_{zy} \\ u\tau_{yx} + v\tau_{yy} + w\tau_{yz} - q_{y} \end{pmatrix}, \quad \mathbf{G}_{z} = \begin{pmatrix} 0 \\ \tau_{xz} \\ \tau_{yz} \\ \tau_{zz} \\ u\tau_{zx} + v\tau_{zy} + w\tau_{zz} - q_{z} \end{pmatrix},$$

$$(2.3)$$

 $\mathbf{v} = (u, v, w)$ is the velocity vector, t is the physical time, ρ is the density, p is the pressure, and e is the specific total energy. The viscous stress tensor τ and the heat flux q are given, under Stokes' hypothesis, by

$$\boldsymbol{\tau} = \begin{pmatrix} \tau_{xx} & \tau_{xy} & \tau_{xz} \\ \tau_{yx} & \tau_{yy} & \tau_{yz} \\ \tau_{zx} & \tau_{zy} & \tau_{zz} \end{pmatrix} = -\frac{2}{3}\mu(\operatorname{div} \mathbf{v})\mathbf{I} + \mu\left(\operatorname{grad} \mathbf{v} + (\operatorname{grad} \mathbf{v})^T\right), \quad \mathbf{q} = \begin{pmatrix} q_x \\ q_y \\ q_z \end{pmatrix} = -\frac{\mu}{Pr(\gamma - 1)}\operatorname{grad} T, \quad (2.4)$$

where I is the identity matrix, T is the temperature, γ is the ratio of specific heats, Pr is the Prandtl number, μ is the viscosity defined by Sutherland's law, and the superscript T denotes the transpose.

In the hyperbolic method, the NS equations are discretized in the form of a first-order hyperbolic system. A hyperbolic NS system was first introduced in Ref.[12], with the viscous stresses and the heat fluxes as the gradient variables. The system involves 14 unknowns, and is called the HNS14 system. Later, a third-order edge-based scheme was developed based on HNS14, but limitations were pointed out, i.e. lower-order accuracy in the velocity gradients and the density gradient [13]. New systems, HNS17 and HNS20, were then introduced in Ref.[14] to overcome these limitations. HNS17 uses the velocity gradients scaled by the viscosity, instead of the viscous stresses, as the gradient variables. However, HNS17 by itself does not guarantee the same order of accuracy in the velocity and the velocity gradients. The problem lies in a weaker coupling in the velocity and its gradients in the discrete level, and the idea of an artificial hyperbolic dissipation was introduced to bring the strong coupling in the discrete equations and achieve the same order of accuracy [14]. Yet there is another drawback in the HNS17 system: it does not lead to accurate density gradients, and thus high-order reconstruction for all the primitive variables cannot be performed. To ensure high-order accuracy on the gradients of all primitive variables, which in turn leads to higher-order inviscid approximations, the HNS20 system was constructed with the density gradient introduced as additional variables in the form of an artificial mass diffusion in the continuity equation [14]. In HNS20, however, the gradients of the primitive variables are obtained not directly but by dividing the gradient variables by the local viscosity, except for the density. This scaling causes complications in getting higher-order derivatives of the primitive variables [10]. For arbitrarily high-order discretizations in the framework of the rDG method, it is desirable to directly carry the gradients of the primitive variables as the gradient variables, instead of those scaled by the viscosity and the heat fluxes. For this purpose, the HNS20G system was introduced in Ref.[9] with the gradients of the primitive variables used as the gradient variables. In this study, we discretize the HNS20G system [9]:

$$\mathbf{P}^{-1}\frac{\partial \mathbf{U}}{\partial \tau} + \frac{\partial \mathbf{F}_k}{\partial x_k} = \mathbf{S}$$
(2.5)

where \mathbf{U} and \mathbf{F}_k are redefined here as

$$\mathbf{U} = [\rho, \rho \mathbf{v}, \rho e, \mathbf{g}_u, \mathbf{g}_v, \mathbf{g}_w, \mathbf{h}, \mathbf{r}]^T,$$
(2.6)

$$\mathbf{F} = \begin{pmatrix} \rho \mathbf{v}^{T} - \nu_{\rho} \mathbf{r}^{T} \\ \rho \mathbf{v} \otimes \mathbf{v} + p \mathbf{I} - \mu_{v} \tilde{\boldsymbol{\tau}} \\ \mathbf{v}^{T} (\rho e + p) - (\mu_{v} \tilde{\boldsymbol{\tau}} \mathbf{v})^{T} - \frac{\mu_{h}}{\gamma(\gamma - 1)} \mathbf{h} \\ -u \mathbf{I} \\ -v \mathbf{I} \\ -w \mathbf{I} \\ -T \mathbf{I} \\ -\rho \mathbf{I} \end{pmatrix}$$
(2.7)

and

$$\widetilde{\boldsymbol{\tau}} = \frac{3}{4} (\mathbf{g} + \mathbf{g}^T) - \frac{1}{2} tr(\mathbf{g}) \mathbf{I}, \quad \mu_v = \frac{4}{3} \mu, \quad \mu_h = \frac{\gamma}{Pr} \mu,$$
(2.8)

$$\mathbf{S} = \begin{bmatrix} 0 & \mathbf{0} & -\mathbf{g}_u & -\mathbf{g}_v & \mathbf{g}_w & -\mathbf{h} & -\mathbf{r} \end{bmatrix}^T$$
(2.9)

$$\mathbf{P} = diag\left(1, 1, 1, 1, 1, \frac{1}{T_v}, \frac{1}{T_h}, \frac{1}{T_h}, \frac{1}{T_h}, \frac{1}{T_r}, \frac{1}{T_r}, \frac{1}{T_r}, \frac{1}{T_r}\right),$$
(2.10)

Note that the HNS20G system is equivalent to the NS equations (2.1) in the pseudo steady state or as soon as the pseudo time term is dropped, and the gradient variables $(\mathbf{g}_u, \mathbf{g}_v, \mathbf{g}_w, \mathbf{h}, \mathbf{r})$ are equivalent to the gradients of the primitive variables:

$$\mathbf{r} = \nabla \rho = (\rho_x, \rho_y, \rho_z), \tag{2.12}$$

$$\mathbf{g} = [\mathbf{g}_u^T, \mathbf{g}_v^T, \mathbf{g}_w^T]^T = [(\nabla u)^T, (\nabla v)^T, (\nabla w)^T]^T = \begin{pmatrix} g_{ux} & g_{uy} & g_{uz} \\ g_{vx} & g_{vy} & g_{vz} \\ g_{wx} & g_{wy} & g_{wz} \end{pmatrix},$$
(2.13)

$$\mathbf{h} = \nabla T = (h_x, h_y, h_z). \tag{2.14}$$

$$T_r = \frac{L_r^2}{\nu_{\rho}}, \quad T_v = \frac{\rho L_r^2}{\mu_v}, \quad T_h = \frac{\rho L_r^2}{\mu_h},$$
 (2.15)

where the length scale L_r is defined as in Refs.[15, 16], and ν_{ρ} is an artificial viscosity associated with the artificial hyperbolic mass diffusion added to the continuity equation [2, 14]. In this paper, we focus on steady problems, and therefore ignore the physical time derivative in the rest of the paper.

3. Hyperbolic Reconstructed Discontinuous Galerkin Discretization

3.1. DG Formulation by Primitive Variables

Consider the weak formulation with a test function Ψ over an element Ω_e with a boundary Γ_e : multiply the HNS20G system by Ψ , and integrate by parts over the element to get

$$\int_{\Omega_e} \Psi \mathbf{P}^{-1} \frac{d\mathbf{U}_h}{d\tau} d\Omega = \int_{\Omega_e} \left[\frac{\partial \Psi}{\partial x_k} \mathbf{F}_k(\mathbf{U}_h) + \Psi \mathbf{S}(\mathbf{U}_h) \right] d\Omega - \int_{\Gamma_e} \Psi \mathbf{F}_k(\mathbf{U}_h) n_k d\Gamma, \quad \forall \Psi \in V_h^p.$$
(3.1)

In the previous work, we define U_h to be a set of cell averages, i.e., P_0 approximation, and separately define higherorder polynomials for the primitive variables

$$\mathbf{W} = [\rho, \mathbf{v}, T, \mathbf{g}_u, \mathbf{g}_v, \mathbf{g}_w, \mathbf{h}, \mathbf{r}]^T,$$
(3.2)

for evaluating the flux and source terms. In this approach, it is not simple to perform the hyperbolic DG construction, where derivatives of the conservative variables need to be expressed in terms of the derivatives of the primitive variables. In this study, we overcome the difficulty by working only with high-order polynomials of the primitive variables, and thus define the numerical approximation \mathbf{W}_h in the broken Sobolev space V_h^n :

$$\mathbf{W}_h \in V_h^p, \quad V_h^p = \{ v_h \in [L_2(\Omega)]^k : v_h|_{\Omega_i} \in [V_p^k] \forall \Omega_i \in \Omega \},$$
(3.3)

where k is the dimension of the unknown vector and V_p is the space of all polynomials of degree $\leq p$. To avoid the conservative variables, we perform the weak formulation in the following form of HNS20G:

$$\mathbf{P}^{-1}\frac{\partial \mathbf{U}}{\partial \mathbf{W}}\frac{\partial \mathbf{W}}{\partial \tau} + \frac{\partial \mathbf{F}_k}{\partial x_k} = \mathbf{S},\tag{3.4}$$

which leads to

$$\int_{\Omega_e} \Psi \mathbf{P}^{-1} \frac{\partial \mathbf{U}}{\partial \mathbf{W}} \frac{d\mathbf{W}_h}{d\tau} d\Omega = \int_{\Omega_e} \left[\frac{\partial \Psi}{\partial x_k} \mathbf{F}_k(\mathbf{W}_h) + \Psi \mathbf{S}(\mathbf{W}_h) \right] d\Omega - \int_{\Gamma_e} \Psi \mathbf{F}_k(\mathbf{W}_h) n_k d\Gamma, \ \forall \Psi \in V_h^p.$$
(3.5)

The solution is expressed as a polynomial of the form:

$$\mathbf{W}_h(x, y, z, \tau, t) = \mathbf{C}(x, y, z) \mathbf{V}(\tau, t),$$
(3.6)

where C is a basis matrix, and V is a vector of unknown polynomial coefficients. In the hyperbolic DG/rDG methods, C is not diagonal, but a square matrix with nonzero off-diagonal elements as we will discuss later. In the Galerkin discretization, the basis matrix C is used as a test function, which results in

$$\mathbf{M}_{p}\frac{d\mathbf{V}}{d\tau} = \mathbf{R}\left(\mathbf{W}_{h}\right),\tag{3.7}$$

where

$$\mathbf{M}_{p} = \int_{\Omega_{e}} \mathbf{C}^{T} \mathbf{P}^{-1} \frac{\partial \mathbf{U}}{\partial \mathbf{W}} \mathbf{C} \, d\Omega, \tag{3.8}$$

and

$$\mathbf{R}(\mathbf{W}_{h}) = \int_{\Omega_{e}} \left[\frac{\partial \mathbf{C}^{T}}{\partial x_{k}} \mathbf{F}_{k}(\mathbf{W}_{h}) + \mathbf{C}^{T} \mathbf{S}(\mathbf{W}_{h}) \right] d\Omega - \int_{\Gamma_{e}} \mathbf{C}^{T} \mathbf{F}_{k}(\mathbf{W}_{h}) n_{k} \, d\Gamma,$$
(3.9)

where the integrals are evaluated by Gaussian quadrature formulas of appropriate orders and the interface flux is evaluated by the sum of HLLC-inviscid and the upwind HNS20G viscous flux as described in Ref.[9]. It is important to note that this formulation is different from that in Ref.[9] in that the volume integral will not vanish here and equations are coupled in the integrands due to the effect of nonzero off-diagonal terms. Although this formulation requires only the primitive variables as desired, the mass matrices now involve the Jacobian matrix $\partial U/\partial W$ and the preconditioning matrix P^{-1} , and thus depend on the solution. First, we discuss the hyperbolic DG and rDG methods.

3.2. Efficient Hyperbolic Discontinuous Galerkin: Hyperbolic DG

The hyperbolic DG method is an efficient construction that reduces the number of discrete unknowns by sharing common unknowns, or equivalently that builds higher-order polynomials for the primitive variables by using the gradient variables and their derivatives to represent the higher-order terms. Suppose we have P_0 approximations for all the primitive variables:

$$\mathbf{V} = \left(\overline{\rho}, \overline{\mathbf{v}}, \overline{T}, \Delta x \overline{\mathbf{g}_x}, \Delta y \overline{\mathbf{g}_y}, \Delta z \overline{\mathbf{g}_z}, \Delta x \overline{h_x}, \Delta y \overline{h_y}, \Delta z \overline{h_z}, \Delta x \overline{r_x}, \Delta y \overline{r_y}, \Delta z \overline{r_z}\right)^T,$$
(3.10)

where the overbar indicates a cell-averaged quantity. Then, we can construct a linear polynomial for the density by using the density gradient variables, $(\overline{r_x}, \overline{r_y}, \overline{r_z})$: :

$$\begin{pmatrix} \rho \\ r_x \\ r_y \\ r_z \end{pmatrix}_h = \begin{pmatrix} B_1 & B_2 & B_3 & B_4 \\ 0 & \frac{1}{\Delta x} & 0 & 0 \\ 0 & 0 & \frac{1}{\Delta y} & 0 \\ 0 & 0 & 0 & \frac{1}{\Delta z} \end{pmatrix} \begin{pmatrix} \overline{\rho} \\ \overline{r_x} \Delta x \\ \overline{r_y} \Delta y \\ \overline{r_z} \Delta z \end{pmatrix},$$
(3.11)

where

$$B_1 = 1, \quad B_2 = \frac{x - x_c}{\Delta x}, \quad B_3 = \frac{y - y_c}{\Delta y}, \quad B_4 = \frac{z - z_c}{\Delta z}$$

Similarly, for the *x*-velocity, we have

$$\begin{pmatrix} u \\ g_{ux} \\ g_{uy} \\ g_{uz} \end{pmatrix}_{h} = \begin{pmatrix} B_{1} & B_{2} & B_{3} & B_{4} \\ 0 & \frac{1}{\Delta x} & 0 & 0 \\ 0 & 0 & \frac{1}{\Delta y} & 0 \\ 0 & 0 & 0 & \frac{1}{\Delta z} \end{pmatrix} \begin{pmatrix} \overline{u} \\ \overline{g_{ux}}\Delta x \\ \overline{g_{uy}}\Delta y \\ \overline{g_{uz}}\Delta z \end{pmatrix},$$
(3.12)

and for the temperature,

$$\begin{pmatrix} T \\ h_x \\ h_y \\ h_z \end{pmatrix}_h = \begin{pmatrix} B_1 & B_2 & B_3 & B_4 \\ 0 & \frac{1}{\Delta x} & 0 & 0 \\ 0 & 0 & \frac{1}{\Delta y} & 0 \\ 0 & 0 & 0 & \frac{1}{\Delta z} \end{pmatrix} \begin{pmatrix} \overline{T} \\ \overline{h_x} \Delta x \\ \overline{h_y} \Delta y \\ \overline{h_z} \Delta z \end{pmatrix},$$
(3.13)

It follows from these examples that the basis matrix C is a block diagonal matrix:

$$\mathbf{C} = \begin{bmatrix} \mathbf{C}_{4 \times 4} & & & \\ & \mathbf{C}_{4 \times 4} & & \\ & & \mathbf{C}_{4 \times 4} & \\ & & & \mathbf{C}_{4 \times 4} \end{bmatrix},$$
(3.14)

where

$$\mathbf{C}_{4\times4} = \begin{pmatrix} B_1 & B_2 & B_3 & B_4 \\ 0 & \frac{1}{\Delta x} & 0 & 0 \\ 0 & 0 & \frac{1}{\Delta y} & 0 \\ 0 & 0 & 0 & \frac{1}{\Delta z} \end{pmatrix},$$
(3.15)

for \mathbf{V} reordered as follows:

$$\mathbf{V} = (\boldsymbol{\rho}_1, \mathbf{u}_1, \mathbf{v}_1, \mathbf{w}_1, \mathbf{T}_1)^T, \qquad (3.16)$$

where

$$\boldsymbol{\rho}_1 = (\overline{\rho}, r_x \Delta x, r_y \Delta y, r_z \Delta z), \quad \mathbf{u}_1 = (\overline{u}, g_{ux} \Delta x, g_{uy} \Delta y, g_{uz} \Delta z), \quad \mathbf{v}_1 = (\overline{v}, g_{vx} \Delta x, g_{vy} \Delta y, g_{vz} \Delta z), \quad (3.17)$$

$$\mathbf{w}_1 = (\overline{w}, g_{wx}\Delta x, g_{wy}\Delta y, g_{wz}\Delta z), \quad \mathbf{T}_1 = (\overline{T}, h_x\Delta x, h_y\Delta y, h_z\Delta z).$$
(3.18)

5 of 16

American Institute of Aeronautics and Astronautics

Note that this basis matrix C is used as a test function in the DG discretization, in contrast to the method in Ref.[9], where the diagonal part is used as a test function and thus it results in FV schemes. Due to the off-diagonal elements, the resulting discretization will not be FV schemes, with coupling among the discrete equations and the non-vanishing volume integrals. In the case of the unsteady equations, it will generate physical time derivative terms in a consistent manner and enables the use of explicit time-stepping schemes for time-accurate computations [11].

The above construction is called HNS(P0P1+P0). It involves 20 discrete unknowns, and achieves second-order accuracy for the inviscid terms, and first-order accuracy for the viscous terms and the gradients. This scheme is comparable to a conventional DG(P₁) scheme applied to the NS equations, which also involves 20 discrete unknowns. However, HNS(P0P1+P0) has one-order lower order of accuracy in viscous dominated regions, e.g., first-order accurate in a boundary-layer, as already pointed out for advection-diffusion problems in Ref.[8]. The efficient hyperbolic DG method can be extended systematically to higher-order, such as HNS(P₀P₂+P₁), HNS(P₀P₃+P₂), and so on. However, the order of approximation to the viscous terms will always be one-order lower than the corresponding conventional DG schemes.

In this work, we do not pursue higher-order hyperbolic DG schemes. Instead, we employ the rDG method to upgrade the overall accuracy without increasing the discrete unknowns any further.

3.3. Efficient Hyperbolic Reconstructed Discontinous Galerkin: Hyperbolic rDG

In the rDG method, higher-order accuracy is achieved by upgrading the order of the polynomial used to evaluate the flux and source terms through higher-order derivative reconstructions. The efficient construction in the previous section is, in fact, a variant of the rDG method, but no reconstruction techniques were needed because the derivatives are directly available from the gradient variables. To further improve accuracy, we now consider reconstructing higherorder derivatives by a reconstruction technique, and construct a higher-order reconstructed polynomial \mathbf{W}_{h}^{R} :

$$\mathbf{W}_{h}^{R}(x, y, z, \tau, t) = \mathbf{C}^{R}(x, y, z)\mathbf{V}^{R}(\tau, t), \qquad (3.19)$$

where \mathbf{C}^{R} is a basis matrix, and \mathbf{V}^{R} is the unknown coefficient vector containing reconstructed derivatives, which is then used to evaluate the residual:

$$\mathbf{R}\left(\mathbf{W}_{h}^{R}\right) = \int_{\Omega_{e}} \left[\frac{\partial \mathbf{C}^{T}}{\partial x_{k}}\mathbf{F}_{k}(\mathbf{W}_{h}^{R}) + \mathbf{C}^{T}\mathbf{S}(\mathbf{W}_{h}^{R})\right] d\Omega - \int_{\Gamma_{e}} \mathbf{C}^{T}\mathbf{F}_{k}(\mathbf{W}_{h}^{R})n_{k} d\Gamma.$$
(3.20)

For example, if the second derivatives ρ_{xx}^c , ρ_{yy}^c , ρ_{zz}^c , ρ_{yz}^c , ρ_{yz}^c are reconstructed from the gradient variables $\overline{r_x}$, $\overline{r_y}$, and $\overline{r_z}$ at the centroid as indicated by the superscript, then we can construct a higher-order polynomial for the density and the density gradient variables as follows:

$$\begin{pmatrix} \rho \\ r_x \\ r_y \\ r_z \end{pmatrix}_h^R = \begin{pmatrix} B_1 & B_2 & B_3 & B_4 & B_5 & B_6 & B_7 & B_8 & B_9 & B_{10} \\ 0 & \frac{1}{\Delta x} & 0 & 0 & \frac{B_2}{\Delta x} & 0 & 0 & \frac{B_3}{\Delta x} & \frac{B_4}{\Delta x} & 0 \\ 0 & 0 & \frac{1}{\Delta y} & 0 & 0 & \frac{B_3}{\Delta y} & 0 & \frac{B_2}{\Delta y} & 0 & \frac{B_4}{\Delta x} \\ 0 & 0 & 0 & \frac{1}{\Delta z} & 0 & 0 & \frac{B_4}{\Delta z} & \frac{B_2}{\Delta z} & 0 & \frac{B_3}{\Delta z} \end{pmatrix} \begin{pmatrix} \overline{\rho} \\ \overline{r_x} \Delta x \\ \overline{r_y} \Delta y \\ \overline{r_z} \Delta z \\ \rho_{xx}^c \Delta x^2 \\ \rho_{xy}^c \Delta x^2 \\ \rho_{xy}^c \Delta x \Delta y \\ \rho_{xz}^c \Delta x \Delta z \\ \rho_{yz}^c \Delta y \Delta z \end{pmatrix}.$$
(3.21)

where

$$B_{5} = \frac{1}{2}B_{2}^{2} - \frac{1}{\Omega_{e}}\int_{\Omega_{e}}\frac{1}{2}B_{2}^{2}d\Omega, \quad B_{6} = \frac{1}{2}B_{3}^{2} - \frac{1}{\Omega_{e}}\int_{\Omega_{e}}\frac{1}{2}B_{3}^{2}d\Omega, \quad B_{7} = \frac{1}{2}B_{4}^{2} - \frac{1}{\Omega_{e}}\int_{\Omega_{e}}\frac{1}{2}B_{4}^{2}d\Omega,$$
$$B_{8} = B_{2}B_{3} - \frac{1}{\Omega_{e}}\int_{\Omega_{e}}B_{2}B_{3}d\Omega, \quad B_{9} = B_{2}B_{4} - \frac{1}{\Omega_{e}}\int_{\Omega_{e}}B_{2}B_{4}d\Omega, \quad B_{10} = B_{3}B_{4} - \frac{1}{\Omega_{e}}\int_{\Omega_{e}}B_{3}B_{4}d\Omega.$$

Scheme	DoF/Cell	Inviscid	Viscous	Gradient
$DG(P_1)$	20	$O(h^2)$	$O(h^2)$	O(h)
$DG(P_2)$	50	$O(h^3)$	$O(h^3)$	$O(h^2)$
$HNS(P_0P_1+P_0)$	20	$O(h^2)$	O(h)	O(h)
$HNS(P_0P_2+P_0P_1(VR))$	20	$O(h^3)$	$O(h^2)$	$O(h^2)$

Table 1. Comparison of the HNS schemes and conventional DG schemes for the 3D NS equations. DoF/Cell is the degrees of freedom within each cell.

The density is now represented by a quadratic polynomial, and the corresponding gradient variables by linear polynomials. Applying a similar reconstruction to other primitive variables, we arrive at the following basis matrix:

$$\mathbf{W}_{h}^{R}(x, y, z) = \begin{bmatrix} \mathbf{C}_{4 \times 10} & & & \\ & \mathbf{C}_{4 \times 10} & & \\ & & \mathbf{C}_{4 \times 10} & \\ & & & \mathbf{C}_{4 \times 10} \end{bmatrix} \mathbf{V}^{R},$$
(3.22)

where

$$\mathbf{C}_{4\times10} = \begin{pmatrix} B_1 & B_2 & B_3 & B_4 & B_5 & B_6 & B_7 & B_8 & B_9 & B_{10} \\ 0 & \frac{1}{\Delta x} & 0 & 0 & \frac{B_2}{\Delta x} & 0 & 0 & \frac{B_3}{\Delta x} & \frac{B_4}{\Delta x} & 0 \\ 0 & 0 & \frac{1}{\Delta y} & 0 & 0 & \frac{B_3}{\Delta y} & 0 & \frac{B_2}{\Delta y} & 0 & \frac{B_4}{\Delta x} \\ 0 & 0 & 0 & \frac{1}{\Delta z} & 0 & 0 & \frac{B_4}{\Delta z} & \frac{B_2}{\Delta z} & 0 & \frac{B_3}{\Delta z} \end{pmatrix},$$
(3.23)

for \mathbf{V}^R defined by

$$\mathbf{V}^{R} = (\boldsymbol{\rho}_{2}, \mathbf{u}_{2}, \mathbf{v}_{2}, \mathbf{w}_{2}, \mathbf{T}_{2})^{T}, \qquad (3.24)$$

where

$$\boldsymbol{\rho}_{2} = (\overline{\rho}, \ r_{x}\Delta x, \ r_{y}\Delta y, \ r_{z}\Delta z, \ \rho_{xx}^{c}\Delta x^{2}, \ \rho_{yy}^{c}\Delta y^{2}, \ \rho_{zz}^{c}\Delta z^{2}, \ \rho_{xy}^{c}\Delta x\Delta y, \ \rho_{xz}^{c}\Delta x\Delta z, \ \rho_{yz}^{c}\Delta y\Delta z),$$
(3.25)

and similarly for other variables.

For the derivative reconstruction, we employ the variational reconstruction for computing the derivatives [17, 18]. The variational reconstruction for computing the gradient of a function is denoted by $P_0P_1(VR)$, and it is applied to the gradient variables here to obtain their gradients that are equivalent to the second derivatives of the density. The resulting scheme is called HNS($P_0P_2+P_0P_1(VR)$), which now outperforms the conventional second-order P_1 DG: it has the same 20 degrees of freedom, and achieves third-order accuracy in the inviscid terms, second-order in the viscous terms, and second-order in the gradients.

Compare it with a conventional $DG(P_1)$ scheme, which requires 20 degrees of freedom for second- and first-order accurate solutions and gradients, respectively; or a conventional $DG(P_2)$, which requires 50 degrees of freedom for third- and second-order accurate solutions and gradients, respectively.

Table 1 shows the comparison between the HNS-rDG schemes and conventional DG schemes applied to the original NS system. Note that the HNS-rDG schemes developed in this paper and those in Ref.[9] are different. The former are based on the Galerkin formulation, and the latter are based on the Petrov-Galerkin formulation with a diagonal part of the basis matrix. $HNS(P_0P_1+P_0)$ is comparable to the conventional $DG(P_1)$ scheme, but has a lower order of accuracy in the viscous terms. This is improved by $HNS(P_0P_2+P_0P_1(VR))$, which increases the order of accuracy by one order for all while keeping the same 20 degrees of freedom: one-order higher accuracy for the inviscid and gradients, and matching the second-order accuracy in the viscous terms with the conventional $DG(P_1)$ scheme.

4. Pseudo Time Integration Scheme (Nonlinear Solver)

To solve the semi-discrete pseudo-time equation 3.7:

$$\mathbf{M}_{p}\frac{d\mathbf{V}}{d\tau} = \widehat{\mathbf{R}}(\mathbf{V}),\tag{4.1}$$

we perform an implicit time discretization as

$$\frac{\mathbf{M}_{p}}{\Delta \tau} \left(\mathbf{V}^{m+1} - \mathbf{V}^{m} \right) = \widehat{\mathbf{R}} \left(\mathbf{V}^{m+1} \right), \tag{4.2}$$

where m denotes the pseudo-time level, $\Delta \tau$ the pseudo-time step, and the residual $\widehat{\mathbf{R}}$ is a nonlinear function of V. Linearize the residual will give

$$\frac{\mathbf{M}_{p}}{\Delta \tau} \left(\mathbf{V}^{m+1} - \mathbf{V}^{m} \right) = \widehat{\mathbf{R}} \left(\mathbf{V}^{m} \right) + \frac{\partial \widehat{\mathbf{R}}}{\partial \mathbf{V}} |^{m} \left(\mathbf{V}^{m+1} - \mathbf{V}^{m} \right).$$
(4.3)

The fully discretized form can be written as

$$\left(\frac{\mathbf{M}_{p}}{\Delta\tau} - \frac{\partial \widehat{\mathbf{R}}}{\partial \mathbf{V}}\right)|^{m} \left(\mathbf{V}^{m+1} - \mathbf{V}^{m}\right) = \widehat{\mathbf{R}} \left(\mathbf{V}^{m}\right).$$
(4.4)

The Jacobian matrix $\frac{\partial \hat{\mathbf{R}}}{\partial \mathbf{V}}$ is estimated using an automatic differentiation toolkit TAPENADE. Note that if $\Delta \tau$ tends to infinity, the scheme reduces to the standard Newton's method with a property of quadratic convergence for solving a system of nonlinear equations. In this work, in order to guarantee the robustness of implicit solver, we begin with a small $\Delta \tau$ with a few time steps then increase it to an infinite number. By using an edge-based data structure, the left-hand-side matrix is stored in upper, lower and diagonal forms, which can be expressed as

$$\mathbf{U} = -\frac{\partial \widehat{\mathbf{R}}(\mathbf{V}_i, \mathbf{V}_j, n_{ij})}{\partial \mathbf{V}_j}, \quad \mathbf{L} = -\frac{\partial \widehat{\mathbf{R}}(\mathbf{V}_i, \mathbf{V}_j, n_{ij})}{\partial \mathbf{V}_i}, \quad \mathbf{D} = \frac{\mathbf{M}}{\Delta t} - \sum_j \frac{\partial \widehat{\mathbf{R}}(\mathbf{V}_i, \mathbf{V}_j, n_{ij})}{\partial \mathbf{V}_i}.$$
(4.5)

where U, L and D represent the upper, lower and diagonal matrix, respectively. Since the reconstruction is not taken into consideration for the Jacobian matrix, this makes the implicit method suitable for parallelization. The linear system is relaxed by the symmetric Gauss-Seidel relaxation schemes.

5. Boundary Conditions

The boundary conditions implemented for real flow problems are presented in this chapter.

$$\Phi_{jb} = \Phi\left(\mathbf{V}_{j}, \mathbf{V}_{b}\right),\tag{5.1}$$

where Φ denotes the numerical flux, j the element adjacent to the boundary, and b denotes the ghost state of the boundary.

5.1. Free stream

$$\mathbf{V}_{b} = \left[\rho_{\infty}, u_{\infty}, v_{\infty}, w_{\infty}, p_{\infty}, \mathbf{g}_{\infty}, \mathbf{h}_{\infty}, \mathbf{r}_{\infty}\right].$$
(5.2)

5.2. Subsonic outflow

$$\mathbf{V}_{b} = \left[\rho_{j}, u_{j}, v_{j}, w_{j}, p_{\infty}, \mathbf{g}_{\infty}, \mathbf{h}_{\infty}, \mathbf{r}_{\infty}\right].$$
(5.3)

5.3. Symmetric boundary

$$\mathbf{V}_{b} = \left[\rho_{j}, \mathbf{v}_{j} - 2(\mathbf{v}_{j} \cdot \mathbf{n}_{jb})\mathbf{n}_{jb}, p_{j}, \mathbf{g}_{j} - 2(\mathbf{g}_{j} \cdot \mathbf{n}_{jb})\mathbf{n}_{jb}, \mathbf{h}_{j} - 2(\mathbf{h}_{j} \cdot \mathbf{n}_{jb})\mathbf{n}_{jb}, \mathbf{r}_{j} - 2(\mathbf{r}_{j} \cdot \mathbf{n}_{jb})\mathbf{n}_{jb}\right].$$
(5.4)

5.4. No-slip viscous adiabatic wall

$$\mathbf{V}_{b} = \left[\rho_{j}, 0, 0, 0, p_{j}, (\mathbf{g}_{j} \cdot \mathbf{n}_{jb})\mathbf{n}_{jb}, \mathbf{h}_{j} - \mathbf{h}_{j} \cdot \mathbf{n}_{jb}, \mathbf{r}\right].$$
(5.5)

6. Results

A manufactured solution for Navier-Stokes equation is first tested here for verification. The laminar flow past a sphere is also presented to verify that feasibility of the new method when it comes to real flow problems with real boundary conditions.

6.1. Accuracy Verification

We verify the HNS-rDG schemes implemented in a 3D code by methods of manufactured solutions for 2D problems on hexahedral and highly distorted prismatic mesh. The second levels of each set of meshes are shown in Figure 1. The following functions are made the exact solutions by introducing source terms into the Navier-Stokes equations:

$$\rho = cr0 + crs \cdot sin(crx \cdot x + cry \cdot y),
u = cu0 + cus \cdot sin(cux \cdot x + cuy \cdot y),
v = cv0 + cvs \cdot sin(cvx \cdot x + cvy \cdot y),
w = 0,
p = cp0 + cps \cdot sin(cpx \cdot x + cpy \cdot y),$$
(6.1)

A constant viscosity $\mu = \rho_{inf} v_{inf} L/Re$ is used here for a verification purpose. The subscript denotes the free stream values. The free stream state is taken as $\rho_{inf} = cr0$ and $v_{inf} = sqrt(cu0^2 + cv0^2)$ here. cr0, crs, crx, cry, cu0, cus, cux, cuy, cv0, cvs, cvx, cvy, cp0, cps, cpx and cpy are all taken as constants to generate a smooth solution over the domain. The Reynolds number is taken as 10 and 10^5 . The artificial mass diffusion coefficient ν_{ρ} is taken as h^4 , where $h = \sqrt{1/nelem}$ for 2D problem and L_r is taken as $1/2\pi$. The order of convergence is shown in Figures 2 and 3, from which we can see that Re = 10, the HNS(P₀P₁+P₀) method gives first-order of convergence for both primary variables and their gradients, and the HNS(P₀P₂+P₀P₀(VR)) gives second-order. However, when it comes to $Re = 10^5$, the order of convergence of the gradients remains the same, one-order higher of convergence can be expected for primitive variables. The result is consistent with Table 1.



b) Highly distorted prismatic mesh

Figure 1. Second level of each set of mesh used for manufactured solution



a) Order of convergence for density

b) Order of convergence for velocity



c) Order of convergence for temperature

Figure 2. Order of convergence for primitive variables



a) Order of convergence for the gradients of density

b) Order of convergence for the gradients of velocity



c) Order of convergence for the gradients of temperature

Figure 3. Order of convergence for gradients

6.2. Laminar Flow Past a Sphere

We verify the HNS-rDG schemes implemented in a 3D code on the flow past a sphere at Reynolds number of 100. This is a benchmark for bluff-body flows and has been investigated numerically and experimentally by other authors. The Mach number is taken as 0.5. The mesh is composed of 200, 416 tetrahedral elements, as is shown in FIgure 4. The residual of the momentum equation is shown in 5, from where we can see that the presented method converges faster than DG(P₁) for the first 250 iteration steps. However, the residual of HNS(P₀P₁+P₀) method stops dropping further after 250 iteration step. The residual of HNS(P₀P₂+P₀P₁(VR)) method, however, keeps dropping due to a more accurate approximation of the right-hand-side resulted from the reconstruction.

Mach contours are compared in Figure 6. Although no big difference can be observed from the Mach contours, we can see the diffence when it comes to the recirculation bubble length, as is shown in Table 2. Our result shows that $HNS(P_0P_2+P_0P_1(VR))$ gives the closest result to the result in [19] with a relative error of 8.26%. The drag coefficient is compared in Table 3. As we can seen from the table, the $HNS(P_0P_2+P_0P_1(VR))$ gives a closest to Ruben et al.'s results of 1.178 [20] with an relative error of 1.93%. The comparison of pressure coefficient is shown in Figure 7. Our results are in good agreement with the result in [21].

When the Reynolds number is increased to Re = 250, a non-axisymmetric flow occurs but the flows remains steady [22]. A comparison of the streamlines is shown in Figure 8, from where a good agreement can be observed.



Figure 4. Mesh used for flow past a sphere



Figure 5. The residual of momentum equation versus iteration for flow past a sphere



a) **DG**(**P**₁)



b) HNS($P_0P_1+P_0$))



c) $HNS(P_0P_2+P_0P_1(VR))$

Figure 6. Mach contours for flow past a sphere

American Institute of Aeronautics and Astronautics

	X_s/D for $Re = 100$
$DG(P_1)$	1.0781
$HNS(P_0P_1+P_0)$	1.0283
$HNS(P_0P_2+P_0P_1(VR))$	0.9678
Gilmanov et al. (2003)	0.894
Ruben et al. (2009)	0.85

	C_d for $Re = 100$
$DG(P_1)$	0.9985
$HNS(P_0P_1+P_0)$	1.1433
$HNS(P_0P_2+P_0P_1(VR))$	1.2007
Gilmanov et al. (2003)	1.153
Ruben et al. (2009)	1.178

Table 3. Comparison of the drag coefficient.



Figure 7. Pressure distribution for flow past a sphere

American Institute of Aeronautics and Astronautics



a) Reference [19]



b) $HNS(P_0P_2+P_0P_1(VR))$

Figure 8. Streamlines for flow past a sphere at Re=250

7. Concluding Remarks

A reconstructed discontinuous Galerkin hyperbolic Navier-Stokes method based on the Galerkin formulation with a full matrix is implemented. A manufactured solution is tested to show that the presented method gives consistent order of convergence to our previous study on model equations. The laminar flow past a sphere is also tested to verify the feasibility of our method when it comes to real flow problems. This work will later be extended to unsteady flows.

Acknowledgments

This work has been funded by the U.S. Army Research Office under the contract/grant number W911NF-16-1-0108 with Dr. Matthew Munson as the program manager.

References

- Nishikawa, H., "A First-Order System Approach for Diffusion Equation. I: Second Order Residual Distribution Schemes," J. Comput. Phys., Vol. 227, 2007, pp. 315–352.
- [2] Nakashima, Y., Watanabe, N., and Nishikawa, H., "Hyperbolic Navier-Stokes Solver for Three-Dimensional Flows," 54th AIAA Aerospace Sciences Meeting, AIAA Paper 2016-1101, San Diego, CA, 2016.
- [3] Luo, H., Luo, L., Nourgaliev, R., Mousseau, V., and Ding, N., "A Reconstructed Discontinuous Galerkin Method for the Compressible Navier-Stokes Equations on Arbitrary Grids," J. Comput. Phys., Vol. 229, No. 19, 2010, pp. 6961 – 6978.

- [4] Liu, X., Xia, Y., Cheng, J., and Luo, H., "Development and assessment of a reconstructed discontinuous Galerkin method for the compressible turbulent flows on hybrid grids," AIAA Paper 2016-1359, 2016.
- [5] Lou, J., Luo, H., and Nishikawa, H., "Discontinuous Galerkin Methods for Hyperbolic Advection-Diffusion Equation on Unstructured Grids," Proc. of The 9th International Conference on Computational Fluid Dynamics, Istanbul, Turkey, 2016.
- [6] Lou, J., Liu, X., Luo, H., and Nishikawa, H., "Reconstructed Discontinuous Galerkin Methods for Hyperbolic Diffusion Equations on Unstructured Grids," 55th AIAA Aerospace Sciences Meeting, AIAA Paper 2017-0310, Grapevine, Texas, 2017.
- [7] Lou, J., Li, L., Luo, H., and Nishikawa, H., "First-Order Hyperbolic System Based Reconstructed Discontinuous Galerkin Methods for Nonlinear Diffusion Equations on Unstructured Grids," 56th AIAA Aerospace Sciences Meeting, AIAA Paper 2018-2094, Kissimmee, Florida, 2018.
- [8] Lou, J., Li, L., Luo, H., and Nishikawa, H., "Reconstructed Discontinuous Galerkin Methods for Linear Advection-Diffusion Equations Based on First-Order Hyperbolic System," J. Comput. Phys., Vol. 369, 2018, pp. 103–124.
- [9] Li, L., Lou, J., Luo, H., and Nishikawa, H., "A New Hyperbolic Navier-Stokes Formulation and Efficient Reconstructed Discontinuous Galerkin Discretizations," 48th AIAA AIAA Fluid Dynamics Conference, AIAA Paper, Atlanta, GA, 2018, to be published as AIAA paper.
- [10] Liu, Y. and Nishikawa, H., "Third-Order Inviscid and Second-Order Hyperbolic Navier-Stokes Solvers for Three-Dimensional Inviscid and Viscous Flows," 46th AIAA Fluid Dynamics Conference, AIAA Paper 2016-3969, Washington, D.C., 2016.
- [11] Lou, J., Li, Lu, H., and Nishikawa, H., "Explicit Hyperbolic Reconstructed Discontinuous Galerkin Methods for Time-Dependent Problems," 48th AIAA AIAA Fluid Dynamics Conference, AIAA Paper, Atlanta, GA, 2018, to be published as AIAA paper.
- [12] Nishikawa, H., "New-Generation Hyperbolic Navier-Stokes Schemes: O(1/h) Speed-Up and Accurate Viscous/Heat Fluxes," Proc. of 20th AIAA Computational Fluid Dynamics Conference, AIAA Paper 2011-3043, Honolulu, Hawaii, 2011.
- [13] Nishikawa, H., "First, Second, and Third Order Finite-Volume Schemes for Navier-Stokes Equations," Proc. of 7th AIAA Theoretical Fluid Mechanics Conference, AIAA Aviation and Aeronautics Forum and Exposition 2014, AIAA Paper 2014-2091, Atlanta, GA, 2014.
- [14] Nishikawa, H., "Alternative Formulations for First-, Second-, and Third-Order Hyperbolic Navier-Stokes Schemes," Proc. of 22nd AIAA Computational Fluid Dynamics Conference, AIAA Paper 2015-2451, Dallas, TX, 2015.
- [15] Liu, Y. and Nishikawa, H., "Third-Order Inviscid and Second-Order Hyperbolic Navier-Stokes Solvers for Three-Dimensional Unsteady Inviscid and Viscous Flows," 55th AIAA Aerospace Sciences Meeting, AIAA Paper 2017-0738, Grapevine, Texas, 2017.
- [16] Nishikawa, H. and Liu, Y., "Hyperbolic Advection-Diffusion Schemes for High-Reynolds-Number Boundary-Layer Problems," J. Comput. Phys., Vol. 352, 2018, pp. 23–51.
- [17] Li, L., Liu, X., Lou, J., Luo, H., Nishikawa, H., and Ren, Y., "A Finite Volume Method Based on Variational Reconstruction for Compressible Flows on Arbitrary Grids," 23rd AIAA Computational Fluid Dynamics Conference, AIAA Paper 2017-3097, Denver, Colorado, 2017.
- [18] Li, L., Liu, X., Lou, J., Luo, H., Nishikawa, H., and Ren, Y., "A Discontinuous Galerkin Method Based on Variational Reconstruction for Compressible Flows on Arbitrary Grids," 56th AIAA Aerospace Sciences Meeting, AIAA Paper 2018-0831, Kissimmee, Florida, 2018.
- [19] Gilmanov, A., Sotiropoulos, F., and Balaras, E., "A general reconstruction algorithm for simulating flows with complex 3D immersed boundaries on Cartesian grids," *Journal of Computational Physics*, Vol. 191, No. 2, 2003, pp. 660–669.
- [20] Campregher, R., Militzer, J., Mansur, S. S., and Silveira Neto, A. d., "Computations of the flow past a still sphere at moderate reynolds numbers using an immersed boundary method," *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, Vol. 31, No. 4, 2009, pp. 344–352.
- [21] Johnson, T. A., "Numerical and Experimental Investigation of Flow past a Sphere up to a Reynolds Number of 300," *Ph. D dissertation, University of Iowa*, 1996.
- [22] Johnson, T. and Patel, V., "Flow past a sphere up to a Reynolds number of 300," *Journal of Fluid Mechanics*, Vol. 378, 1999, pp. 19–70.