Downloaded by Hiroaki Nishikawa on July 4, 2018 | http://arc.aiaa.org | DOI: 10.2514/6.2018-4160



A New Formulation of Hyperbolic Navier-Stokes Solver based on Finite Volume Method on Arbitrary Grids

Lingquan Li^{*}, Jialin Lou[†], Hong Luo[‡] North Carolina State University, Raleigh, NC, 27695, USA

Hiroaki Nishikawa[§]

National Institute of Aerospace, Hampton, VA, 23666, USA

In this paper, we propose a new hyperbolic Navier-Stokes formulation and develop efficient highorder reconstructed discontinuous Galerkin schemes for solving the compressible Navier-Stokes equations. The new formulation uses the gradients of the primitive variables as additional variables to form a hyperbolic viscous system. This formulation enables simple and economical solution reconstruction, and paves the way to the development of a family of truely efficient high-order hyperbolic Navier-Stokes schemes: more efficient than discontinuous Galerkin schemes for comparable orders of accuracy despite the increase in the number of variables/equations due to the hyperbolic formulation at the partial-differential-equation level. With 20 degrees of freedom per cell, the method achieves, on irregular grids, third-order error convergence in the solution and gradients, and fourthorder in the inviscid terms. The eigen-structure of the hyperbolic viscous system is analyzed and a suitable dissipation matrix is derived. A class of reconstructed discontinuous Galerkin discretizations are constructed for the hyperbolic Navier-Stokes sytem, and solved by an implicit solver based on the first-order backward Euler pseudo-time integration scheme with exact or approximate Jacobians. The methods are implemented in a three-dimensional unstructured code, verified by the method of manufactured solutions on hexahedral, regular, irregular and highly distorted prismatic grids, and tested for a Couette flow problem.

I. Introduction

Today, Computational Fluid Dynamics (CFD) is applied to a variety of applications in science and engineering. Second-order algorithms dominate practical three-dimensional CFD codes, but there is a growing need for higherorder accuracy especially in high-fidelity time-dependent turbulent-flow simulations over complex geometries. Beside many challenges, e.g., high-order grid generation and adaptation, high-performance computing, robust and efficient solver developments, etc., there exits a classic challenge about how to deal with the viscous terms in Navier-Stokes (NS) equations. Although high-order methods have been well developed for hyperbolic systems such as the inviscid terms, they have been less studied for the viscous terms. In particular, the widely-used and studied finite-volume (FV) and Discontinuous Garlerkin (DG) methods become less certain and advantageous for elliptic or parabolic problems, requiring extra efforts needed to deal with gradients in the viscous fluxes. Commonly used high-order viscous methods are the BR2 [1] and DDG [2] methods. A reconstructed discontinuous Galerkin (rDG) method using an inter-cell reconstruction is also developed [3]. All of these methods, however, are substantially more demanding in computational efforts than the classical continuous finite element methods that are more suited for solving elliptic problems. In this work, we explore an alternative to these high-order viscous discretization methods: the hyperbolic Navier-Stokes (HNS) method, in combination with the rDG method to drive the cost of high-order schemes down to the level at which applications to practical problems would be within reach with currently available resources. Ultimately, the developed method is expected to provide an efficient and accurate high-order unstructured-grid solver for scalable high-order CFD computations on fully adaptive grids with ever-increasing parallelism.

The HNS method is a first-order hyperbolic system reformulation approach, which is originally developed for diffusion equations [4] and the advection-diffusion equations [5]. In the HNS method, diffusion/viscous terms are reformulated as a hyperbolic system, and discretized by methods for hyperbolic systems, e.g., upwind methods. Thus, methods developed for hyperbolic systems, including high-order methods, are directly applicable to diffusion and

^{*}Graduate Research Assistant, Department of Mechanical and Aerospace Engineering, AIAA Student Member. 11132@ncsu.edu

[†]Graduate Research Assistant, Department of Mechanical and Aerospace Engineering, AIAA Student Member.

[‡]Professor, Department of Mechanical and Aerospace Engineering, Associate Fellow AIAA.

[§]Associate Research Fellow, Associate Fellow AIAA.

viscous terms. Moreover, the method has been demonstrated to bring various other advantages: high-order and noisefree gradients on severely distorted grids, upgraded accuracy in the advection/inviscid schemes, and convergence acceleration by the elimination of numerical stiffness due to second-order derivatives. These advantages have been demonstrated for various applications, including the two-dimensional compressible/incompressible NS equations [6, 7, 8], three-dimensional compressible NS equations [9, 10, 11, 12, 13, 14] with a proper handling high-Reynolds-number boundary-layer flows [11, 15], unsteady problems [16, 17], magnetohydrodynamic models [18, 19], third-order partialdifferential equations [20] and diffusion with tensor coefficients [21] and with discontinuous coefficients [22]. In terms of the discretization method, the hyperbolic method has been applied to the residual-distribution methods [4, 5], the finite-volume methods [6, 7, 8, 9, 10, 11, 12, 23, 24, 25], the high-order active-flux method [16], high-order compact methods [26], and high-order rDG methods [21, 27, 28, 29, 30]. This paper is a sequel to the high-order hyperbolic rDG method development, and presents the extension to the compressible NS equations.

For diffusion equations, a hyperbolic system is formed typically with the solution gradients introduced as additional variables, which are called the gradient variables. In three dimensions, it requires three additional gradient variables per solution variable, meaning three additional equations are added to each equation in a target system. This increase in variables and equations presents a challenge in applying the hyperbolic method to DG methods. For example, if the three-dimensional compressible NS system is formulated as a hyperbolic system with 20 equations instead of 5 [12], even a second-order DG scheme method would result in 80 discrete equations. To minimize the number of discrete equations, we have explored an efficient reconstruction of recycling the gradient variables to form high-order polynomials of the solution variables for model equations [21, 27, 28, 29, 30]. The recycling strategy dramatically reduces the number of discrete equations, and leads to efficient hyperbolic DG schemes. For the NS equations, it can bring the number of discrete equations down to 20 from 80 without losing second-order accuracy and even with an upgrade of the inviscid accuracy to third-order. For example, the HNS formulation in Ref.[12] involves the density gradient as gradient variables. If DG(P1) method is used, the density gradient (ρ_x , ρ_y , ρ_z), and their gradients, i.e., second-order derivatives ($\rho_{xx}, \rho_{xy}, \rho_{xz}, \rho_{yx}, \rho_{yy}, \rho_{yz}, \rho_{zx}, \rho_{zy}, \rho_{zz}$) would be introduced as discrete unknowns. Then, these quantities can be used to build a quadratic polynomial for the density, and thus only the cell-average needs to be stored for the density. This particular construction is denoted by DG(P0P2)+DG(P1), and leads to a third-order inviscid and second-order viscous scheme. However, this efficient construction is not straightforward for other conservative or primitive variables because other gradient variables are the velocity gradients scaled by the viscosity and the heat fluxes. The derivatives of the velocity and temperature, which are needed to build the polynomials for the velocity and temperature, can be quite complicated if obtained from the gradient variables as they will involve derivatives of the viscosity. To make the recycling trivial for all variables, we here propose a new HNS formulation, HNS20G, with the gradients of the primitive variables as additional unknowns. A similar hyperbolic formulation has been employed for nonlinear or variable-coefficient diffusion problems [22, 27, 31]; this paper presents, for the first time, its extension to the compressible NS equations.

To further reduce the cost of the hyperbolic DG schemes, we focus on the rDG method, which has been widely used for compressible inviscid and viscous flow problems, as well as for turbulent problems [32, 33, 34, 35, 36, 37, 38, 39, 40]. The rDG method is a general framework that includes the FV and DG methods as special cases: define an underlying high-order polynomial of on each element, *reconstruct a higher-order polynomial*, and solve a Riemann problem that arises from the discontinuous representation of solution on each element interface. An rDG scheme with a polynomial of degree n and a reconstructed polynomial of degree n + 1 is denoted by rDG(PnPn+1). As discussed in the previous papers [21, 27, 28, 29, 30], high-order schemes that combine the hyperbolic approach and the rDG method require less number of discrete unknowns and achieve higher-order accuracy than conventional DG schemes. The reconstruction step is a key to accuracy and stability of the rDG schemes, and a recently developed high-order reconstruction technique called the variational reconstruction (VR) provides excellent performance of the rDG schemes as demonstrated for compressible inviscid flow [41, 42]. This VR reconstruction method is used in the present paper for computing high-order derivatives of the gradient variables in the new HNS system; derivatives of the primitive variables are provided by the gradient variables and their derivatives that are already available. As a result, the rDG schemes will have the same number of degrees of freedom as a FV scheme applied to the HNS formulation or conventional P_1 DG schemes for the NS equations, i.e. 20, and this is true for arbitrarily high-order accuracy.

In this paper, we present a new HNS formulation and demonstrate that it enables an efficient construction of highorder rDG schemes for solving the compressible NS equations, with 20 degrees of freedom for arbitrarily high-order accuracy. The paper focuses on schemes up to fourth-order accuracy in the inviscid terms, which eliminate both dispersion and dissipation errors from a second-order scheme and are expected to provide significant improvements for high-fidelity turbulent-flow simulations. An implicit solver is used to solve the residual equations, based on an approximate or exact Jacobian with a pseudo time term, which is equivalent to the first-order backward Euler pseudotime integration scheme. The pseudo time step is taken to be infinity in many cases, and thus it can be considered simply as an implicit nonlinear solver. The HNS rDG solver is developed for the 3D NS equations and implemented in a 3D code, but here we focus on 2D problems, and perform accuracy verification studies and solve a simple viscousflow problem to demonstrate that the HNS rDG schemes provide accurate solutions for a realistic viscous boundary conditions. The remainder of this paper is organized as follows. Section 2 describes the target NS equations, Section 3 presents the new HNS formulation and the construction of dissipation matrices are given, Section 4 discusses the discretization methods, Section 5 describes the implicit solver, Section 6 presents numerical results, and Section 7 closes the paper with remarks.

II. Compressible Navier-Stokes Equations

We consider the compressible NS equations:

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

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

$$\begin{pmatrix} \rho e \end{pmatrix} \begin{pmatrix} u(\rho e + p) \end{pmatrix} \begin{pmatrix} v(\rho e + p) \end{pmatrix} \begin{pmatrix} w(\rho e + p) \end{pmatrix}$$

$$\mathbf{G}_{x} = \begin{pmatrix} 0 \\ \tau_{xx} \\ \tau_{yx} \\ \tau_{zx} \\ u\tau_{xx} + v\tau_{xy} + w\tau_{xz} - q_{x} \end{pmatrix}, \quad \mathbf{G}_{y} = \begin{pmatrix} 0 \\ \tau_{xy} \\ \tau_{zy} \\ u\tau_{yx} + v\tau_{yy} + w\tau_{yz} - q_{y} \end{pmatrix}, \quad (2.3)$$

$$\mathbf{G}_{z} = \begin{pmatrix} 0 \\ \tau_{xz} \\ \tau_{yz} \\ \tau_{zz} \\ u\tau_{zx} + v\tau_{zy} + w\tau_{zz} - q_{z} \end{pmatrix}, \quad (2.4)$$

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

$$\boldsymbol{\tau} = \begin{pmatrix} \tau_{xx} & \tau_{xy} & \tau_{xz} \\ \tau_{yx} & \tau_{yy} & \tau_{yz} \\ \tau_{zx} & \tau_{zy} & \tau_{zz} \end{pmatrix} = -\frac{2}{3}\mu(\operatorname{div} \mathbf{v})\mathbf{I} + \mu\left(\operatorname{grad} \mathbf{v} + (\operatorname{grad} \mathbf{v})^T\right), \quad \mathbf{q} = \begin{pmatrix} q_x \\ q_y \\ q_z \end{pmatrix} = -\frac{\mu}{Pr(\gamma - 1)}\operatorname{grad} T, \quad (2.5)$$

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, which is taken as a constant in this paper, and the superscript T denotes the transpose.

In this paper, we focus on the spatial discretization, and therefore consider only the spatial part in the rest of the paper:

$$\frac{\partial \mathbf{F}_k}{\partial x_k} = \frac{\partial \mathbf{G}_k}{\partial x_k}.$$
(2.6)

For unsteady problems, the physical time derivative can be discretized in time and then treated as a source term as demonstrated in Ref.[11]. For many practical problems, implicit-time stepping schemes would be required, and then the steady solver developed in this paper will serve as a nonlinear solver for a system of unsteady residual equations that needs to be solved at every time step. Unsteady extensions will be discussed in a subsequent paper.

III. New Hyperbolic Navier-Stokes System: HNS20G

The HNS formulation used in Ref.[12] uses the velocity gradients scaled by the viscosity, the heat fluxes, and the density gradients as additional variables, thus resulting in 20 equations and thus termed HNS20. As we will illustrate later, the scaled velocity gradients and the heat fluxes introduce complications in the efficient construction of hyperbolic rDG schemes. To simply the construction, we propose to form a hyperbolic viscous system with the gradients of the primitive variables as additional variables.

3.1. HNS20G

The new HNS formulation is given in the form of a preconditioned conservative system:

$$\mathbf{P}^{-1}\frac{\partial \mathbf{U}}{\partial \tau} + \frac{\partial \mathbf{F}_k}{\partial x_k} = \mathbf{S}$$
(3.1)

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

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

$$\mathbf{F} = \begin{pmatrix} \rho \mathbf{v}^{T} - \nu_{\rho} \mathbf{r}^{T} \\ \rho \mathbf{v} \otimes \mathbf{v} + p \mathbf{I} - \mu_{v} \tilde{\boldsymbol{\tau}} \\ \mathbf{v}^{T} (\rho e + p) - (\mu_{v} \tilde{\boldsymbol{\tau}} \mathbf{v})^{T} - \frac{\mu_{h}}{\gamma(\gamma - 1)} \mathbf{h} \\ -u \mathbf{I} \\ -v \mathbf{I} \\ -w \mathbf{I} \\ -T \mathbf{I} \\ -\rho \mathbf{I} \end{pmatrix}$$
(3.3)

$$\widetilde{\boldsymbol{\tau}} = \frac{3}{4} (\mathbf{g} + \mathbf{g}^T) - \frac{1}{2} tr(\mathbf{g}) \mathbf{I}, \quad \mu_v = \frac{4}{3} \mu, \quad \mu_h = \frac{\gamma}{Pr} \mu, \quad (3.4)$$

$$\mathbf{S} = \begin{bmatrix} 0 & \mathbf{0} & -\mathbf{g}_u & -\mathbf{g}_v & \mathbf{g}_w & -\mathbf{h} & -\mathbf{r} \end{bmatrix}^T$$
(3.5)

$$\mathbf{P} = diag\left(1, 1, 1, 1, 1, \frac{1}{T_v}, \frac{1}{T_h}, \frac{1}{T_h}, \frac{1}{T_h}, \frac{1}{T_r}, \frac{1}{T_r}, \frac{1}{T_r}, \frac{1}{T_r}\right).$$
(3.6)

Note that the system is equivalent to the target steady Navier-Stokes equations (2.6) in the pseudo steady state or as soon as the pseudo time term is dropped with an infinite pseud-otime step, and the additional variables are equivalent to the gradients of the primitive variables (thus called the gradient variables):

$$\mathbf{r} = \nabla \rho = (\rho_x, \rho_y, \rho_z), \tag{3.7}$$

$$\mathbf{g} = [\mathbf{g}_u^T, \mathbf{g}_v^T, \mathbf{g}_w^T]^T = [(\nabla u)^T, (\nabla v)^T, (\nabla w)^T]^T = \begin{pmatrix} g_{ux} & g_{uy} & g_{uz} \\ g_{vx} & g_{vy} & g_{vz} \\ g_{wx} & g_{wy} & g_{wz} \end{pmatrix},$$
(3.8)

$$\mathbf{h} = \nabla T = (h_x, h_y, h_z). \tag{3.9}$$

To emphasize that this system uses the gradients of the primitive variables as additional variables, we term the new system HNS20G. The preconditioned conservative system simplifies the construction of a conservative scheme and a numerical flux [8], and the viscous system is now hyperbolic in the pseudo time as we will discuss below. The relaxation times T_r , T_v , and T_h are defined as

$$T_r = \frac{L_r^2}{\nu_{\rho}}, \quad T_v = \frac{\rho L_r^2}{\mu_v}, \quad T_h = \frac{\rho L_r^2}{\mu_h},$$
 (3.10)

where the length scale L_r is defined as in Refs.[11, 15], and ν_{ρ} is an artificial viscosity associated with the artificial hyperbolic mass diffusion added to the continuity equation [7, 12].

The flux projected along an arbitrary direction $\mathbf{n} = (n_x, n_y, n_z)^T$ is split into the inviscid part, viscous part, and artificial hyperbolic mass diffusion part:

$$\mathbf{F}_n = \mathbf{F}_x n_x + \mathbf{F}_y n_y + \mathbf{F}_z n_z = \mathbf{F}_n^i + \mathbf{F}_n^v + \mathbf{F}_n^a,$$
(3.11)

where

$$\mathbf{F}_{n}^{i} = \begin{pmatrix} \rho u_{n} \\ \rho u_{n} \mathbf{v}^{T} + \rho \mathbf{n} \\ u_{n} (\rho e + p) \\ \mathbf{0} \end{pmatrix}, \quad \mathbf{F}_{n}^{v} = \begin{pmatrix} 0 \\ -\mu_{v} \tilde{\boldsymbol{\tau}}_{n} \\ -\mu_{v} \tilde{\boldsymbol{\tau}}_{nv} - \frac{\mu_{h}}{\gamma(\gamma - 1)} h_{n} \\ -\nu \mathbf{n} \\ -v \mathbf{n} \\ -w \mathbf{n} \\ -T \mathbf{n} \\ \mathbf{0} \\ \mathbf{0} \end{pmatrix}, \quad \mathbf{F}_{n}^{a} = \begin{pmatrix} -\nu_{\rho} \mathbf{r}_{n} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ -\rho \mathbf{n} \end{pmatrix}, \quad (3.12)$$

where

$$u_n = \mathbf{v} \cdot \mathbf{n} = un_x + vn_y + wn_z, \quad h_n = \mathbf{h} \cdot \mathbf{n} = h_x n_x + h_y n_y + h_z n_z, \tag{3.13}$$

$$\widetilde{\boldsymbol{\tau}}_{n} = \begin{bmatrix} \widetilde{\tau}_{nx} \\ \widetilde{\tau}_{ny} \\ \widetilde{\tau}_{nz} \end{bmatrix} = \boldsymbol{\tau} \mathbf{n} = \begin{bmatrix} \widetilde{\tau}_{xx} n_{x} + \widetilde{\tau}_{xy} n_{y} + \widetilde{\tau}_{xz} n_{z} \\ \widetilde{\tau}_{yx} n_{x} + \widetilde{\tau}_{yy} n_{y} + \widetilde{\tau}_{yz} n_{z} \\ \widetilde{\tau}_{zx} n_{x} + \widetilde{\tau}_{zy} n_{y} + \widetilde{\tau}_{zz} n_{z} \end{bmatrix}, \quad \widetilde{\tau}_{nv} = \widetilde{\boldsymbol{\tau}}_{n} \cdot \mathbf{v} = \widetilde{\tau}_{nx} u + \widetilde{\tau}_{ny} v + \widetilde{\tau}_{nz} w. \quad (3.14)$$

The inviscid part is hyperbolic, so it can be estimated by using classic Riemann solvers. The HLLC flux [43] is used in the present work. In the HNS2G system, the viscous part is also hyperbolic as shown in the next section.

3.2. Dissipation Matrix for Viscous Flux

An important difference between HNS20 and HNS20G is that the viscosity does not appear in the flux for HNS20 but does appear in the case of HNS20G. Therefore, the flux Jacobian, which is required to construct a dissipation term, involves derivatives of the viscosity. This is the complication we get in return for a simplified high-order scheme construction as discussed later. However, we can avoid the complication by considering the flux Jacobian with a frozen viscosity [27]:

$$\mathbf{A}_{n}^{v} = \mathbf{P} \frac{\partial \mathbf{F}_{n}^{v}}{\partial \mathbf{U}} = \begin{pmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} & \mathbf{A}_{13} & \mathbf{A}_{14} & \mathbf{A}_{15} & \mathbf{0} \\ \mathbf{A}_{21} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{pmatrix},$$
(3.15)

where

5 of 28

where the viscosity is frozen in computing the derivative, $\frac{\partial \mathbf{F}_n^v}{\partial \mathbf{U}}$. The eigenvalues of the matrix can be obtained as

$$\lambda_1 = -a_{nv}, \quad \lambda_2 = a_{nv}, \quad \lambda_3 = -a_{mv}, \quad \lambda_4 = a_{mv}, \tag{3.22}$$

$$\lambda_5 = -a_{lv}, \quad \lambda_6 = a_{lv}, \quad \lambda_7 = -a_h, \quad \lambda_8 = a_h, \quad \lambda_{9,\dots,20} = 0,$$
(3.23)

where

$$a_{nv} = \sqrt{\frac{\nu_v}{T_v}}, \quad a_{mv} = a_{lv} = \sqrt{\frac{3\nu_v}{4T_v}}, \quad a_h = \sqrt{\frac{\nu_h}{T_h}}.$$
 (3.24)

corresponding to the normal viscous, shear viscous and heating waves, respectively, with a set of linearly independent left and right eigenvectors, l and r. Therefore, the system is hyperbolic. Then, the upwind flux can be constructed

Downloaded by Hiroaki Nishikawa on July 4, 2018 | http://arc.aiaa.org | DOI: 10.2514/6.2018-4160

$$\begin{aligned} |\mathbf{A}_{n}^{v}|\Delta\mathbf{U} &= \sum_{k=1}^{8} |\lambda_{k}| (\mathbf{I}_{k}^{T}\Delta\mathbf{U})\mathbf{r}_{k} \\ & 0 \\ \rho[(a_{nv} - a_{mv})\Delta u_{n}\mathbf{n} + a_{mv}\Delta\mathbf{v}] \\ \rho(a_{nv} - a_{mv})u_{n}\Delta u_{n} + \rho a_{mv}(\mathbf{v}\cdot\Delta\mathbf{v}) + \frac{\rho a_{h}\Delta T}{\gamma(\gamma-1)} + \frac{\mu_{v}^{2}}{\rho a_{h}} \left(\frac{\tilde{\tau}_{nn}\Delta\tilde{\tau}_{nn}}{Pr_{n}+1} + \frac{\tilde{\tau}_{n}\cdot\Delta\tilde{\tau}_{n}-\tilde{\tau}_{nn}\Delta\tilde{\tau}_{nn}}{Pr_{m}+1}\right) \\ & \left[\left(a_{nv} - \frac{4}{3}a_{mv}\right)\Delta\tilde{\tau}_{nn}n_{x} + \frac{4}{3}a_{mv}\Delta\tilde{\tau}_{nx}\right]\mathbf{n} \\ & \left[\left(a_{nv} - \frac{4}{3}a_{mv}\right)\Delta\tilde{\tau}_{nn}n_{y} + \frac{4}{3}a_{mv}\Delta\tilde{\tau}_{nz}\right]\mathbf{n} \\ & \left[\left(a_{nv} - \frac{4}{3}a_{mv}\right)\Delta\tilde{\tau}_{nn}n_{z} + \frac{4}{3}a_{mv}\Delta\tilde{\tau}_{nz}\right]\mathbf{n} \\ & a_{h}\left[\Delta h_{n} + \frac{\gamma(\gamma-1)\mu_{v}}{\mu_{h}}\left(\frac{\tilde{\tau}_{nn}\Delta u_{n}}{Pr_{n}+1} + \frac{\tilde{\tau}_{n}\cdot\Delta\mathbf{v}-\tilde{\tau}_{nn}\Delta u_{n}}{Pr_{m}+1}\right)\right]\mathbf{n} \\ & \mathbf{0} \end{aligned}$$

where

$$\Delta \mathbf{U} = \mathbf{U}_R - \mathbf{U}_L, \quad Pr_n = \frac{a_{nv}}{a_h}, \quad Pr_m = \frac{a_{mv}}{a_h}.$$
(3.26)

This dissipation term is similar to the one for HNS20 [12], but has different scalings for the viscous stresses and heat fluxes. Finally, this dissipation term will be multiplied by \mathbf{P}^{-1} from the left to cancel the effect of the preconditioning [8].

To ensure strong coupling among the system in the discrete level, an artificial hyperbolic dissipation is added to the upwind scheme for the viscous flux [7]. Consider the following hyperbolic diffusion system:

$$\mathbf{P}^{-1}\frac{\partial \mathbf{U}}{\partial \tau} + \frac{\partial \mathbf{F}_x^d}{\partial x} + \frac{\partial \mathbf{F}_y^d}{\partial y} + \frac{\partial \mathbf{F}_z^d}{\partial z} = \mathbf{S}^d, \tag{3.27}$$

The flux projected along an arbitrary direction n is given by

$$\mathbf{F}_{n}^{d} = \begin{pmatrix} 0 \\ -\mu_{v}g_{un} \\ -\mu_{v}g_{vn} \\ -\mu_{v}g_{wn} \\ 0 \\ -u\mathbf{n} \\ -v\mathbf{n} \\ -w\mathbf{n} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \qquad (3.29)$$

where

Downloaded by Hiroaki Nishikawa on July 4, 2018 | http://arc.aiaa.org | DOI: 10.2514/6.2018-4160

$$\begin{cases} g_{un} = g_{ux}n_x + g_{uy}n_y + g_{uz}n_z, \\ g_{vn} = g_{vx}n_x + g_{vy}n_y + g_{vz}n_z, \\ g_{wn} = g_{wx}n_x + g_{wy}n_y + g_{wz}n_z. \end{cases}$$
(3.30)

The flux Jacobian with a frozen viscosity is given by

(3.33)

The

Therefore, the artificial dissipation term is given by

$$|\mathbf{A}_{n}^{av}|\Delta \mathbf{U} = a_{nv} \begin{pmatrix} 0 \\ -u\Delta\rho + \Delta(\rho u) \\ -v\Delta\rho + \Delta(\rho v) \\ -w\Delta\rho + \Delta(\rho w) \\ 0 \\ \Delta g_{un} \mathbf{n} \\ \Delta g_{vn} \mathbf{n} \\ \Delta g_{wn} \mathbf{n} \\ \Delta g_{wn} \mathbf{n} \\ 0 \\ 0 \end{pmatrix} = a_{nv} \begin{pmatrix} 0 \\ \rho\Delta u \\ \rho\Delta v \\ \rho\Delta w \\ 0 \\ \Delta g_{un} \mathbf{n} \\ \Delta g_{un} \mathbf{n} \\ \Delta g_{wn} \mathbf{n} \\ \Delta g_{wn} \mathbf{n} \\ 0 \\ 0 \end{pmatrix}.$$
(3.34)

Finally, the upwind viscous flux across an interface is given by

$$\mathbf{F}_{n}^{v}(\mathbf{U}_{L},\mathbf{U}^{R}) = \frac{1}{2} \left(\mathbf{F}_{n}^{v}(\mathbf{U}_{L}) + \mathbf{F}_{n}^{v}(\mathbf{U}_{R}) \right) - \frac{1}{2} \mathbf{P}^{-1} \left(|\mathbf{A}_{n}^{v}| + |\mathbf{A}_{n}^{av}| \right) \left(\mathbf{U}_{R} - \mathbf{U}_{L} \right).$$
(3.35)

3.3. Dissipation Matrix for Artificial Mass Diffusion Flux

The remaining component is the artificial mass diffusion term, which is shared by HNS20 and HNS20G and required to obtain high-order accurate density gradients [7, 12]. The corresponding flux is given by

and the flux Jacobian

$$\mathbf{A}_{n}^{a} = \mathbf{P} \frac{\partial \mathbf{F}_{n}^{a}}{\partial \mathbf{U}} = \begin{pmatrix} 0 & \mathbf{0} & -\nu_{\rho} n_{x} & -\nu_{\rho} n_{y} & -\nu_{\rho} n_{z} \\ \mathbf{0} & \mathbf{O} & \mathbf{0} & \mathbf{0} \\ -\frac{n_{x}}{T_{\rho}} & \mathbf{0} & 0 & 0 & 0 \\ -\frac{n_{y}}{T_{\rho}} & \mathbf{0} & 0 & 0 & 0 \\ -\frac{n_{z}}{T_{\rho}} & \mathbf{0} & 0 & 0 & 0 \end{pmatrix}.$$
(3.37)

The dissipation term is given, as described in Ref.[7, 12], by

$$|\mathbf{A}_{n}^{a}|\Delta \mathbf{U} = a_{\rho} \begin{pmatrix} \Delta \rho \\ \mathbf{0} \\ \Delta r_{n} \mathbf{n} \end{pmatrix}, \qquad (3.38)$$

where

$$a_{\rho} = \sqrt{\frac{\nu_{\rho}}{T_{\rho}}}.$$
(3.39)

Therefore, the artificial mass diffusion flux is given by

$$\mathbf{F}_{n}^{a}(\mathbf{U}_{L},\mathbf{U}_{R}) = \frac{1}{2} \left(\mathbf{F}_{n}^{a}(\mathbf{U}_{L}) + \mathbf{F}_{n}^{a}(\mathbf{U}_{R}) \right) - \frac{1}{2} \mathbf{P}^{-1} |\mathbf{A}_{n}^{a}| (\mathbf{U}_{R} - \mathbf{U}_{L}).$$
(3.40)

IV. Discretization

In this section, we present efficient rDG schemes for the HNS20G system. First, we discuss the underlying DG discretization, which can be very expensive for HNS20G. Then, we illustrate the recycling procedure that dramatically reduces the number of discrete unknwons/equations by taking advantage of the gradient variables. Finally, we introduce the rDG method to bring the discretization discretization further down, and develop very efficient HNS rDG schemes.

4.1. Discontinous Galerkin

We begin by formulating the discontinuous Galerkin discretization 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,

$$\mathbf{P}^{-1}\frac{d}{d\tau}\int_{\Omega_e} \boldsymbol{\Psi} \mathbf{U}_h d\Omega = \int_{\Omega_e} \left[\frac{\partial \boldsymbol{\Psi}}{\partial x_k} \mathbf{F}_k(\mathbf{U}_h) + \boldsymbol{\Psi} \mathbf{S}(\mathbf{U}_h)\right] d\Omega - \int_{\Gamma_e} \boldsymbol{\Psi} \mathbf{F}_k(\mathbf{U}_h) n_k d\Gamma, \quad \forall \boldsymbol{\Psi} \in V_h^p.$$
(4.1)

where **P** can be assumed to be constant over the element without affecting accuracy and thus has been taken out of the integral, and U_h is the solution we seek for in the broken Soblev space V_h^n :

$$\mathbf{U}_h \in V_h^p, \quad V_h^p = \{ v_h \in [L_2(\Omega)]^k : v_h|_{\Omega_i} \in [V_p^k] \forall \Omega_i \in \Omega \},$$
(4.2)

where k is the dimension of the unknown vector and V_p is the space of all polynomials of degree $\leq p$. The solution can be expressed as a polynomial of the form:

$$\mathbf{U}_{h}(x, y, z, \tau) = \mathbf{C}(x, y, z)\mathbf{V}(\tau), \tag{4.3}$$

where C is a basis matrix, and V is a vector of unknown polynomial coefficients. For example, in the P_0 DG method, we have

$$\mathbf{U}_{h} = \left(\rho, \rho \mathbf{v}, \rho e, \mathbf{g}_{x}, \mathbf{g}_{y}, \mathbf{g}_{z}, h_{x}, h_{y}, h_{z}, r_{x}, r_{y}, r_{z}\right)^{T},$$

$$(4.4)$$

$$\mathbf{V} = \left(\overline{\rho}, \overline{\rho \mathbf{v}}, \overline{\rho e}, \Delta x \overline{\mathbf{g}_x}, \Delta y \overline{\mathbf{g}_y}, \Delta z \overline{\mathbf{g}_z}, \Delta x \overline{h_x}, \Delta y \overline{h_y}, \Delta z \overline{h_z}, \Delta x \overline{r_x}, \Delta y \overline{r_y}, \Delta z \overline{r_z}\right)^T,$$
(4.5)

$$\mathbf{C} = \begin{bmatrix} 1 & & & \\ & \mathbf{I} & & & \\ & & 1 & & \\ & & \mathbf{C}_{sub} & \\ & & & \mathbf{C}_{sub} \end{bmatrix}, \quad \mathbf{C}_{sub} = \begin{bmatrix} \frac{1}{\Delta x} & & \\ & \frac{1}{\Delta y} & \\ & & \frac{1}{\Delta z} \end{bmatrix}, \quad (4.6)$$

where $\mathbf{g}_x = (g_{ux}, g_{vx}, g_{wx}), \mathbf{g}_y = (g_{uy}, g_{vy}, g_{wy}), \mathbf{g}_z = (g_{uz}, g_{vz}, g_{wz}),$ the overbar indicates the cell-averaged value, $\Delta x = 0.5(x_{max} - x_{min}), \Delta x = 0.5(y_{max} - y_{min})$ and $\Delta z = 0.5(z_{max} - z_{min}). x_{max}, x_{min}, y_{max}, y_{min}, z_{max}$ and z_{min} are the maximum and minimum coordinates in the element Ω_e in x-, y- and z- directions. Note that the variables $(\mathbf{g}_u, \mathbf{g}_v, \mathbf{g}_w)$ have been rearranged in the vector \mathbf{U}_h for convenience; compare Equation (4.4) and Equation (3.2).

Choose the basis matrix for the test function:

$$\Psi = \mathbf{C}^T, \tag{4.7}$$

then,

$$\mathbf{P}^{-1}\frac{d}{d\tau}\int_{\Omega_e}\mathbf{C}^T\mathbf{C}\mathbf{V}d\Omega = \int_{\Omega_e}\left[\frac{\partial\mathbf{C}^T}{\partial x_k}\mathbf{F}_k(\mathbf{U}_h) + \mathbf{C}^T\mathbf{S}(\mathbf{U}_h)\right]d\Omega - \int_{\Gamma_e}\mathbf{C}^T\mathbf{F}_k(\mathbf{U}_h)n_kd\Gamma.$$
(4.8)

which can be written into the semi-discrete form:

$$\mathbf{P}^{-1}\mathbf{M}\frac{d\mathbf{V}}{d\tau} = \mathbf{R}(\mathbf{U}_h),\tag{4.9}$$

where

$$\mathbf{P^{-1}M} = \mathbf{P^{-1}} \int_{\Omega_e} \mathbf{C}^T \mathbf{C} d\Omega, \quad \mathbf{R}(\mathbf{U}_h) = \int_{\Omega_e} \left[\frac{\partial \mathbf{C}^T}{\partial x_k} \mathbf{F}_k(\mathbf{U}_h) + \mathbf{C}^T \mathbf{S}(\mathbf{U}_h) \right] d\Omega - \int_{\Gamma_e} \mathbf{C}^T \mathbf{F}_k(\mathbf{U}_h) n_k d\Gamma \quad (4.10)$$

The interface flux is evaluated at each quadrature point for a given pair of states, U_L and U_R , computed by the solution polynomials from two adjacent elements, by the HLLC flux [43] for the inviscid part and the sum of the upwind viscous flux (3.35) and the artificial mass diffusion flux (3.40) for the viscous part. This discretization results in a large number of degrees of freedom (or equivalently discrete equations) per cell, with derivatives added to the vector **V**: 80 for P_1 , 200 for P_2 , etc., for the HNS20G system. As shown in the previous papers [21, 27, 28, 29, 30], it is possible to dramatically reduce the number of degrees of freedom by taking advantage of the gradient variables and a reconstruction technique as described in the next sections.

4.2. Efficient Hyperbolic Discontinuous Galerkin: Hyperbolic DG

The first step to reduce the required degrees of freedom is to use or recycle the gradient variables to represent high-order derivatives of the primal variables. This is a simple procedure for model equations as demonstrated in Ref.[21, 27, 28, 29, 30]. However, it is somewhat complicated for the HNS system because the primal variables are the conservative variables (ρ , ρu , ρv , ρw , ρe), and the gradient variables correspond to the gradients of the primitive

variables (ρ, u, v, w, T) . Thus, the derivatives of the conservative variables cannot be directly expressed by these gradient variables; it can be expressed indirectly but the mass matrix will then depend on the solution. Such a construction may be explored in future, but in this paper we consider a simplified procedure, where we keep the P_0 conservative variables in U_h , construct high-order polynomials in the primitive variables and use them to evaluate the flux and source term: $\mathbf{R}(\mathbf{W}_h) = \int_{\Omega_e} \left[\frac{\partial \mathbf{C}^T}{\partial x_k} \mathbf{F}_k(\mathbf{W}_h) + \mathbf{C}^T \mathbf{S}(\mathbf{W}_h) \right] d\Omega - \int_{\Gamma_e} \mathbf{C}^T \mathbf{F}_k(\mathbf{W}_h) n_k d\Gamma,$ where \mathbf{W}_h is a set of higher-order polynomials of the primitive variables:

$$\mathbf{W}_h(x, y, z, \tau) = \mathbf{C}_w(x, y, z) \mathbf{V}_w(\tau), \tag{4.12}$$

(4.11)

with $\mathbf{W} = (\rho, \mathbf{r}, u, \mathbf{g}_u, v, \mathbf{g}_v, w, \mathbf{g}_v, T, \mathbf{h})$, and \mathbf{C}_w is a basis matrix for the primitive variables. In this way, higherorder polynomials can be easily constructed for the primal variables (ρ, u, v, w, T) by directly using the gradient variables $(\mathbf{r}, \mathbf{g}_u, \mathbf{g}_v, \mathbf{g}_w, \mathbf{h})$. For example, for the density, we can construct the following:

$$\begin{pmatrix} \rho \\ r_x \\ r_y \\ r_z \end{pmatrix}_h = \begin{pmatrix} B_1 & B_2 & B_3 & B_4 \\ 0 & \frac{1}{\Delta x} & 0 & 0 \\ 0 & 0 & \frac{1}{\Delta y} & 0 \\ 0 & 0 & 0 & \frac{1}{\Delta z} \end{pmatrix} \begin{pmatrix} \overline{\rho} \\ \overline{r_x} \Delta x \\ \overline{r_y} \Delta y \\ \overline{r_z} \Delta z \end{pmatrix},$$
(4.13)

where

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

The solution has just been upgraded to linear for the density without introducing additional unknowns. Similarly, for the x-velocity, we have

$$\begin{pmatrix} u \\ g_{ux} \\ g_{uy} \\ g_{uz} \\ g_{uz} \\ g_{uz} \\ \end{pmatrix}_{h} = \begin{pmatrix} B_{1} & B_{2} & B_{3} & B_{4} \\ 0 & \frac{1}{\Delta x} & 0 & 0 \\ 0 & 0 & \frac{1}{\Delta y} & 0 \\ 0 & 0 & 0 & \frac{1}{\Delta z} \\ \end{pmatrix} \begin{pmatrix} \overline{\rho u} / \overline{\rho} \\ \overline{g_{ux}} \Delta x \\ \overline{g_{uy}} \Delta y \\ \overline{g_{uz}} \Delta z \\ \end{pmatrix},$$
(4.14)

and for the temperature,

$$\begin{pmatrix} T \\ h_x \\ h_y \\ h_z \end{pmatrix}_h = \begin{pmatrix} B_1 & B_2 & B_3 & B_4 \\ 0 & \frac{1}{\Delta x} & 0 & 0 \\ 0 & 0 & \frac{1}{\Delta y} & 0 \\ 0 & 0 & 0 & \frac{1}{\Delta z} \end{pmatrix} \begin{pmatrix} \frac{(\gamma - 1)\overline{\rho}}{\gamma} \left[\overline{\rho e} - \frac{1}{2\overline{\rho}} (\overline{\rho u}^2 + \overline{\rho v}^2 + \overline{\rho w}^2) \right] \\ \overline{h_x} \Delta x \\ \overline{h_y} \Delta y \\ \overline{h_z} \Delta z \end{pmatrix},$$
(4.15)

Note that the cell-averaged values of the primitive variables are algebraically computed from the conservative variables that we keep in U_h . It follows from these examples that the basis matrix C_w is a block diagonal matrix:

$$\mathbf{C}_{w} = \begin{bmatrix} \mathbf{C}_{4 \times 4} & & & \\ & \mathbf{C}_{4 \times 4} & & \\ & & \mathbf{C}_{4 \times 4} & \\ & & & \mathbf{C}_{4 \times 4} \end{bmatrix},$$
(4.16)

where

$$\mathbf{C}_{4\times4} = \begin{pmatrix} B_1 & B_2 & B_3 & B_4 \\ 0 & \frac{1}{\Delta x} & 0 & 0 \\ 0 & 0 & \frac{1}{\Delta y} & 0 \\ 0 & 0 & 0 & \frac{1}{\Delta z} \end{pmatrix},$$
(4.17)

for \mathbf{V}_w defined by

$$\mathbf{V}_w = (\boldsymbol{\rho}_1, \mathbf{u}_1, \mathbf{v}_1, \mathbf{w}_1, \mathbf{T}_1)^T, \qquad (4.18)$$

$$\boldsymbol{\rho}_1 = (\overline{\rho}, r_x \Delta x, r_y \Delta y, r_z \Delta z), \quad \mathbf{u}_1 = (\overline{u}, g_{ux} \Delta x, g_{uy} \Delta y, g_{uz} \Delta z), \quad \mathbf{v}_1 = (\overline{v}, g_{vx} \Delta x, g_{vy} \Delta y, g_{vz} \Delta z), \quad (4.19)$$

$$\mathbf{w}_1 = (\overline{w}, g_{wx} \Delta x, g_{wy} \Delta y, g_{wz} \Delta z), \quad \mathbf{T}_1 = (\overline{T}, h_x \Delta x, h_y \Delta y, h_z \Delta z), \tag{4.20}$$

with

$$\overline{u} = \overline{\rho u}/\overline{\rho}, \quad \overline{v} = \overline{\rho v}/\overline{\rho}, \quad \overline{w} = \overline{\rho w}/\overline{\rho}, \quad \overline{T} = \frac{(\gamma - 1)\overline{\rho}}{\gamma} \left[\overline{\rho e} - \frac{1}{2\overline{\rho}} (\overline{\rho u}^2 + \overline{\rho v}^2 + \overline{\rho w}^2) \right]. \tag{4.21}$$

This construction is called HNS(P0P1+P0). Note that HNS(P0P1+P0) has 20 unknowns and achieve secondorder accuracy in the inviscid terms, which is comparable to the P_1 DG scheme applied to the original Navier-Stokes equations. In effect, the above efficient construction brings the number of unknowns from 80 to 20 for the P_1 level of accuracy. However, as pointed out in Ref.[28], it is first-order accurate in the viscous limit. Further improvement comes from the rDG method.

4.3. Efficient Hyperbolic Reconstructed Discontinuous 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 in the residual 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 through the gradient variables. To further improve accuracy, we now consider reconstructing higher-order derivatives by a reconstruction technique, and construct higher-order reconstructed polynomials \mathbf{W}_{h}^{R} :

$$\mathbf{W}_{h}^{R}(x, y, z, \tau) = \mathbf{C}_{w}^{R}(x, y, z) \mathbf{V}_{w}^{R}(\tau), \qquad (4.22)$$

where \mathbf{C}_w^R is a basis matrix, and \mathbf{V}_w^R the unknown coefficient vector containing now reconstructed derivatives. For example, if the second derivatives ρ_{xx}^c , ρ_{yy}^c , ρ_{zz}^c , ρ_{xy}^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:

$$\begin{pmatrix} \rho \\ r_x \\ r_y \\ r_z \end{pmatrix}_h^R = \begin{pmatrix} B_1 & B_2 & B_3 & B_4 & B_5 & B_6 & B_7 & B_8 & B_9 & B_{10} \\ 0 & \frac{1}{\Delta x} & 0 & 0 & \frac{B_2}{\Delta x} & 0 & 0 & \frac{B_3}{\Delta x} & \frac{B_4}{\Delta x} & 0 \\ 0 & 0 & \frac{1}{\Delta y} & 0 & 0 & \frac{B_3}{\Delta y} & 0 & \frac{B_2}{\Delta y} & 0 & \frac{B_4}{\Delta x} \\ 0 & 0 & 0 & \frac{1}{\Delta z} & 0 & 0 & \frac{B_4}{\Delta z} & \frac{B_2}{\Delta z} & 0 & \frac{B_3}{\Delta z} \end{pmatrix} \begin{pmatrix} \overline{\rho} \\ \overline{r_x} \Delta x \\ \overline{r_y} \Delta y \\ \overline{r_z} \Delta z \\ \rho_{xx}^c \Delta x^2 \\ \rho_{xy}^c \Delta x^2 \\ \rho_{xy}^c \Delta x \Delta y \\ \rho_{xz}^c \Delta x \Delta z \\ \rho_{yz}^c \Delta y \Delta z \end{pmatrix}.$$
(4.23)

where

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

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:

$$\mathbf{W}_{h}^{R}(x,y,z) = \begin{bmatrix} \mathbf{C}_{4\times10} & & & \\ & \mathbf{C}_{4\times10} & & \\ & & \mathbf{C}_{4\times10} & \\ & & & \mathbf{C}_{4\times10} \end{bmatrix} \mathbf{V}^{R},$$
(4.24)

$$\mathbf{C}_{4\times10} = \begin{pmatrix} B_1 & B_2 & B_3 & B_4 & B_5 & B_6 & B_7 & B_8 & B_9 & B_{10} \\ 0 & \frac{1}{\Delta x} & 0 & 0 & \frac{B_2}{\Delta x} & 0 & 0 & \frac{B_3}{\Delta x} & \frac{B_4}{\Delta x} & 0 \\ 0 & 0 & \frac{1}{\Delta y} & 0 & 0 & \frac{B_3}{\Delta y} & 0 & \frac{B_2}{\Delta y} & 0 & \frac{B_4}{\Delta x} \\ 0 & 0 & 0 & \frac{1}{\Delta z} & 0 & 0 & \frac{B_4}{\Delta z} & \frac{B_2}{\Delta z} & 0 & \frac{B_3}{\Delta z} \end{pmatrix},$$
(4.25)

for \mathbf{V}_w^R defined by

$$\mathbf{V}_{w}^{R} = \left(\boldsymbol{\rho}_{2}, \mathbf{u}_{2}, \mathbf{v}_{2}, \mathbf{w}_{2}, \mathbf{T}_{2}\right)^{T}, \qquad (4.26)$$

where

$$\boldsymbol{\rho}_{2} = (\overline{\rho}, \ r_{x}\Delta x, \ r_{y}\Delta y, \ r_{z}\Delta z, \ \rho_{xx}^{c}\Delta x^{2}, \ \rho_{yy}^{c}\Delta y^{2}, \ \rho_{zz}^{c}\Delta z^{2}, \ \rho_{xy}^{c}\Delta x\Delta y, \ \rho_{xz}^{c}\Delta x\Delta z, \ \rho_{yz}^{c}\Delta y\Delta z),$$
(4.27)

and similarly for other variables. It is emphasized that the reconstructed polynomials are used only in the flux and source terms, not in the weak formulation:

$$\mathbf{P}^{-1}\mathbf{M}\frac{d\mathbf{V}}{d\tau} = \mathbf{R}(\mathbf{W}_h^R),\tag{4.28}$$

and thus the mass matrix M is still based on the underlying polynomial basis matrix C defined for the conservative variables. For the derivative reconstruction, we employ the variational reconstruction for computing the derivatives [41, 42]. The variational reconstruction for computing the gradient of a function is denoted by P0P1(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(P0P2+P0P1(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.

An even more efficient scheme is HNS(P0P3+P0P2(VR)) with

Scheme	DoF/Cell	Inviscid	Viscous	Gradient
DG(P1)	20	$O(h^2)$	$O(h^2)$	O(h)
DG(P2)	50	$O(h^3)$	$O(h^3)$	$O(h^2)$
HNS(P0P1+P0)	20	$O(h^2)$	O(h)	O(h)
HNS(P0P2+P0P1(VR))	20	$O(h^3)$	$O(h^2)$	$O(h^2)$
HNS(P0P3+P0P2(VR))	20	$O(h^4)$	$O(h^3)$	$O(h^3)$

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.

$$B_{11} = \frac{1}{6}B_2^3 - \frac{1}{\Omega_e}\int_{\Omega_e}\frac{1}{6}B_2^3d\Omega, \quad B_{12} = \frac{1}{6}B_3^3 - \frac{1}{\Omega_e}\int_{\Omega_e}\frac{1}{6}B_3^3d\Omega, \quad B_{13} = \frac{1}{6}B_4^3 - \frac{1}{\Omega_e}\int_{\Omega_e}\frac{1}{6}B_4^3d\Omega,$$

$$B_{14} = \frac{1}{2}B_2^2B_3 - \frac{1}{\Omega_e}\int_{\Omega_e}\frac{1}{2}B_2^2B_3d\Omega, \quad B_{15} = \frac{1}{2}B_2^2B_4 - \frac{1}{\Omega_e}\int_{\Omega_e}\frac{1}{2}B_2^2B_4d\Omega, \quad B_{16} = \frac{1}{2}B_3^2B_2 - \frac{1}{\Omega_e}\int_{\Omega_e}\frac{1}{2}B_3^2B_2d\Omega,$$

$$B_{17} = B_2B_3B_4 - \frac{1}{\Omega_e}\int_{\Omega_e}B_2B_3B_4d\Omega,$$

$$B_{18} = \frac{1}{2}B_4^2B_2 - \frac{1}{\Omega_e}\int_{\Omega_e}\frac{1}{2}B_4^2B_2d\Omega, \quad B_{19} = \frac{1}{2}B_3^2B_4 - \frac{1}{\Omega_e}\int_{\Omega_e}\frac{1}{2}B_3^2B_4d\Omega, \quad B_{20} = \frac{1}{2}B_4^2B_3 - \frac{1}{\Omega_e}\int_{\Omega_e}\frac{1}{2}B_4^2B_3d\Omega,$$

$$(4.30)$$

and the superscript c indicates that the basis function is evaluated by the centroid coordinates. In this case, we reconstruct not only the gradient of the gradient variables, but also their second derivatives, which correspond to third-order derivatives of the density, so that a cubic polynomial can be reconstructed for the density. Here, a higher-order version of the variational reconstruction, P0P2(VR), is used to compute the gradients and second-derivatives of the gradient variables. The resulting scheme achieves, with the same 20 degrees of freedom, fourth-order accuracy in the inviscid terms, third-order accuracy in the viscous terms and the solution gradients. Compare it with a conventional P_1 DG scheme, which requires 20 degrees of freedom for second- and first-order accurate solutions and gradients, respectively; or a conventional P_2 DG, 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. HNS(P0P1+P0) is comparable to the conventional P_1 DG scheme, but has a lower order of accuracy in the viscous terms. This is improved by HNS(P0P2+P0P1(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 P_1 DG scheme. A further improvement is achieved by HNS(P0P3+P0P2(VR)): it gives fourth-order accuracy in the inviscid terms, and third-order accuracy in the viscous terms and gradients with the same 20 degrees of freedom, which is more efficient than the conventional P_2 DG scheme.

Thee HNS-rDG schemes may be considered as high-order FV schemes developed in the rDG framework. It can be systematically extended to higher-order in more than one way. For example, HNS(P0P4+P2P3(VR)) may be developed by a linear (instead of quadratic) VR reconstruction method, or HNS(P0P4+P0P3(VR) by a cubic VR reconstruction method. In the former, the basis matrix used in the weak formulation may no longer be diagonal and the scheme will be a mixed FV-DG scheme. Furthermore, if the primitive variables are used as a set of working variables in the weak formulation, HNS-rDG schemes will be again mixed-FV-DG schemes. Such variants should be explored in future, and compared with the schemes developed in this paper.

V. Implicit Solver

We discretize the pseudo time term by the first-order backward Euler scheme:

$$\mathbf{P}^{-1}\mathbf{M}\frac{\mathbf{V}-\mathbf{V}^m}{\Delta\tau} = \mathbf{R}(\widehat{\mathbf{W}}_h^R),\tag{5.1}$$

g | DOI: 10.2514/6.2018-4160 Downl

Table 2 shows the number of iterations versus 1/h for Re = 10 on hexahedral mesh. The number of iterations is taken at five, six, seven, orders of magnitude residual reductions for HNS(P0P1+P0), HNS(P0P2+P0P1(VR)), and HNS(P0P3+P0P2(VR)), respectively. From the table, we can see that the iteration increases linearly with 1/h for all the three schemes, demonstrating the convergence acceleration feature of the hyperbolic method; it increases quadratically for conventional viscous schemes in such a low-Reynolds-number problem.

6.2. Couette flow

Couette flow is a laminar viscous flow between two parallel plates. The bottom plate with a fixed temperature (T_0) is stationary, while the top plate with a fixed temperature (T_1) is moving at a constant speed of U in the positive x-

where $\Delta \tau$ is a pseudo time step, and m is the previous pseudo time level, and solve the resulting global system of residual equations for V by the following implicit method:

$$\mathbf{V}^{k+1} = \mathbf{V}^k + \Delta \mathbf{V},\tag{5.2}$$

where k is the iteration counter, the correction ΔV is given as a solution to the linearized system:

$$\left(\frac{\mathbf{P}^{-1}\mathbf{M}}{\Delta\tau} - \frac{\partial \overline{\mathbf{R}}}{\partial \mathbf{V}}\right) \Delta \mathbf{V} = \mathbf{R}(\widehat{\mathbf{W}}_{h}^{R}), \tag{5.3}$$

and $\overline{\mathbf{R}}$ is the residual without the reconstruction:

$$\overline{\mathbf{R}} = \mathbf{R}(\mathbf{W}_h). \tag{5.4}$$

The Jacobians for the inviscid HLLC and viscous upwind fluxes are computed exactly. However, the residual Jacobian ignores the VR reconstructed part, and thus approximate for rDG schemes; the solver reduces to Newton's method only when the pseudo time step is taken as infinity and no reconstruction is performed. The linear system is relaxed by the symmetric Gauss-Seidel relaxation scheme. In this study, only one symmetric Gauss-Seidel relaxation is performed per iteration.

VI. Numerical Results

6.1. 2D Manufactured solution

We verify the HNS rDG schemes implemented in a 3D code by methods of manufactured solutions for 2D problems on hexahedral, regular, irregular 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:

$$\begin{cases}
\rho = cr0 + crs \cdot sin(crx \cdot x + cry \cdot y), \\
u = cu0 + cus \cdot sin(cux \cdot x + cuy \cdot y), \\
v = cv0 + cvs \cdot sin(cvx \cdot x + cvy \cdot y), \\
w = 0, \\
p = cp0 + cps \cdot sin(cpx \cdot x + cpy \cdot y),
\end{cases}$$
(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 log of residual and L2 error norms are plotted in Figure 2 and Figure 3. From these two figures, we can see that although the HNS(P0P1+P0) method will come to a steady solution when all the residuals drop to at least four orders of magnitude, HNS(P0P3+P0P2(VR)) will come to a steady state after all the residuals drop at least to seven orders of magnitude. From the comparison, we can see that the solution becomes steady after all the residuals drop to the order of magnitude of the log of error. The mesh convergence is shown in Figure 4 and Figure 5. For viscous limit when
$$Re = 10$$
, all the primitive variables and auxiliary variables converge to first, second and third order of accuracy for HNS(P0P1+P0), HNS(P0P2+P0P1(VR)) and HNS(P0P3+P0P2(VR)), respectively. However, when it comes to inviscid limit for $Re = 10^5$, the gradient variables still converge to first, second and third, but an over convergence occurs for primitive variables as they converge to second-, third- and forth-order of accuracy for these three methods, respectively

1/h	P0P1+P0	P0P2+P0P1(VR)	P0P3+P0P2(VR)
8	21	45	135
16	40	79	250
32	97	184	380
64	200	400	785

Table 2. Iteration numbers versus 1/h, Re=10.

direction. The distance between two plates is H. Under some simplifications with the costant μ and low speed to ensure the nearly incompressible effects, the analytical solution is given [44] as

$$u = \frac{U}{H}z, \quad w = 0, \tag{6.2}$$

$$T = T_0 + \frac{z}{H} + \frac{z}{H}(T_1 - T_0) + \frac{Pr * U^2}{2 * Cp} \frac{z}{H}(1 - \frac{z}{H}),$$
(6.3)

$$p = constant, \quad \rho = \frac{p}{RT}$$
 (6.4)

where Pr denotes the Prandtl number, R is the gas constant and Cp is the specific heat capacity for constant pressure. In our numerical experiment, H = 2, $T_0 = 0.8$ and $T_1 = 0.85$. The Mach number for the moving wall is taken as 0.1. Reynold number is taken as 100 with a constant viscosity $\mu = 0.001$. The computational domain is a rectangle $(0 \le x \le 2H, 0 \le z \le H)$. For this test case, take $\nu_r = 10^{-15}$. Weak Dirichlet boundary conditions are given for all the boudary faces. Hexahedral, prismatic and hybrid mesh are used here with 5 successively refined grids. The second level meshes of each type are shown in Figure 6. The mesh convergence is shown in Figure 7. Errors in the temperature and its gradient in z direction converge at first-order for HNS(P0P1+P0) method. However, a super convergence occurs when it comes to HNS(P0P2+P0P1(VR)) and HNS(P0P3+P0P2(VR)) methods, giving third- and forth- order of accuracy for temperature and its gradient in z direction.

VII. Conclusions

A new formulation of hyperbolic Navier-Stokes method, HNS20G, suitable to the reconstructed discontinuous Galerkin framework has been developed. In this formulation, the hyperbolic viscous system is established by introducing the gradients of density, velocity and temperature as the additional variables. The high order derivatives of the gradient variables are obtained by performing a variational reconstruction to avoid the redundant variables. Three discretization schemes, HNS(P0P1+P0), HNS(P0P2+P0P1(VR)) and HNS(P0P3+P0P2(VR)), have been developed, and the residual equations are solved by an implicit solver. The methods have been verified by the method of manufactured solutions, and the designed orders of accuracy have bene confirmed for all the primitive variables and gradient variables, including one-order higher-order of accuracy in the inviscid limit. In future work, other high-order variants should be explored, the methods will be applied to more realistic three-dimensional flow test cases on arbitrary grids, and extended to unsteady problems.

Acknowledgements

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



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





Figure 2. Residual and L2 error versus iteration manufactured solution, Re = 10.





Figure 3. Residual and L2 error versus iteration for manufactured solution, $Re = 10^5$.







Figure 5. Mesh convergence for manufactured solution, $Re = 10^5$.



Figure 6. Second level hexahedral, prismatic and hybrid mesh for Couette flow



Figure 7. Mesh convergence for Couette flow, Re = 100.

References

- Bassi, F. and Rebay, S., "GMRES discontinuous Galerkin solution of the compressible Navier-Stokes equations," *Discontin*uous Galerkin Methods: Theory, Computations and Applications, edited by B. C. G. Karniadakis and C. W. Shu, Vol. 11, Springer, 2000, pp. 197–208.
- [2] Guo Liu, J. and Wang Shu, C., "A High-Order Discontinuous Galerkin Method for 2D Incompressible Flows," J. Comput. Phys., Vol. 160, 2000, pp. 577–596.
- [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] 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.
- [5] Nishikawa, H., "A First-Order System Approach for Diffusion Equation. II: Unification of Advection and Diffusion," J. Comput. Phys., Vol. 229, 2010, pp. 3989–4016.
- [6] 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.
- [7] 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.
- [8] 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.
- [9] 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.
- [10] Liu, Y. and Nishikawa, H., "Third-Order Edge-Based Hyperbolic Navier-Stokes Scheme for Three-Dimensional Viscous Flows," 23rd AIAA Computational Fluid Dynamics Conference, AIAA Paper 2017-3443, Denver, Colorado, 2017.
- [11] 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.
- [12] 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.
- [13] Nagao, T., Hashimoto, A., and Sato, T., "Study on Time Evolution Method for Hyperbolic Navier-Stokes System," 56th AIAA Aerospace Sciences Meeting, AIAA Paper 2018-0370, Kissimmee, Florida, 2018.
- [14] Wu, Z. and Wang, D., "The Development of a Hyperbolic RANS System for Analysing Turbomachinery Flow Field," Proc. of Shanghai 2017 Global Power and Propulsion Forum, GPPS-17-0198, Shanghai, China, 2017.
- [15] 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.
- [16] Nishikawa, H. and Roe, P. L., "Third-Order Active-Flux Scheme for Advection Diffusion: Hyperbolic Diffusion, Boundary Condition, and Newton Solver," *Computers and Fluids*, Vol. 125, 2016, pp. 71–81.
- [17] Nishikawa, H. and Liu, Y., "Hyperbolic Navier-Stokes Method for High-Reynolds-Number Boundary-Layer Flows," 55th AIAA Aerospace Sciences Meeting, AIAA Paper 2017-0081, Grapevine, Texas, 2017.
- [18] Baty, H. and Nishikawa, H., "Hyperbolic Method for Magnetic Reconnection Process in Steady State Magnetohydrodynamics," *Mon. Not. R. Astron. Soc.*, Vol. 459, 2016, pp. 624–637.
- [19] Kawashima, R., Komurasaki, K., and Schonherr, T., "A Flux-Splitting Method for Hyperbolic-Equation System of Magnetized Electron Fluids in Quasi-Neutral Plasmas," J. Comput. Phys., Vol. 310, 2016, pp. 202–212.
- [20] Mazaheri, A., Ricchiuto, M., and Nishikawa, H., "A First-Order Hyperbolic System Approach for Dispersion," J. Comput. Phys., Vol. 321, September 2016, pp. 593–605.
- [21] 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.
- [22] Nishikawa, H., "On Hyperbolic Merthod for Diffusion with Discontinous Coefficients," J. Comput. Phys., Vol. 367, 2018, pp. 102–108.
- [23] Nishikawa, H., "First, Second, and Third Order Finite-Volume Schemes for Advection-Diffusion," J. Comput. Phys., Vol. 273, 2014, pp. 287–309.
- [24] Nishikawa, H., "First-, Second-, and Third-Order Finite-Volume Schemes for Diffusion," J. Comput. Phys., Vol. 256, 2014, pp. 791–805.

- [25] Lee, E., Ahn, H. T., and Luo, H., "Cell-Centered High-order Hyperbolic Finite Volume Method for Diffusion Equation on Unstructured Grids," J. Comput. Phys., Vol. 355, 2018, pp. 464–491.
- [26] Chamarthi, A. S., Wang, Z., and Kawashima, R., "Weighted Nonlinear Schemes for Magnetized Electron Fluid in Quasineutral plasma," 56th AIAA Aerospace Sciences Meeting, AIAA Paper 2018-2194, Kissimmee, Florida, 2018.
- [27] Lou, J., Li, Lu, 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.
- [28] Lou, J., Li, L., Liu, X., Luo, H., and Nishikawa, H., "Reconstructed Discontinuous Galerkin Methods Based on First-Order Hyperbolic System for Advection-Diffusion Equations," 23rd AIAA Computational Fluid Dynamics Conference, AIAA Paper 2017-3445, Denver, Colorado, 2017.
- [29] 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.
- [30] 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.
- [31] Nishikawa, H. and Nakashima, Y., "Dimensional Scaling and Numerical Similarity in Hyperbolic Method for Diffusion," J. Comput. Phys., Vol. 355, 2018, pp. 121–143.
- [32] 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.
- [33] Liu, X., Xuan, L., Luo, H., and Xia, Y., "A reconstructed discontinuous Galerkin method for the compressible Navier-Stokes equations on three-dimensional hybrid gridss," *Comput. Fluids*, Vol. 152, 2017, pp. 217–230.
- [34] Liu, X., Xia, Y., and Hong Luo, L. X., "A comparative study of Rosenbrock-type and implicit Runge-Kutta time integration for discontinuous Galerkin method for unsteady 3D compressible Navier-Stokes equations," *Commun. Compt. Phys.*, Vol. 20, No. 4, 2016, pp. 1016–1044.
- [35] Xia, Y., Liu, X., Luo, H., and Nourgaliev, R., "A third-order implicit discontinuous Galerkin method based on a Hermite WENO reconstruction for time-accurate solution of the compressible NavierStokes equations," *Int. J. Numer. Meth. Fluids*, Vol. 79, No. 8, 2015, pp. 416–435.
- [36] Cheng, J., Liu, T., and Luo, H., "A hybrid reconstructed discontinuous Galerkin method for compressible flows on arbitrary grids," *Comput. Fluids*, Vol. 139, 2016, pp. 68–79.
- [37] Xia, Y., Lou, J., Hong, L., Edwards, J., and Mueller, F., "OpenACC acceleration of an unstructured CFD solver based on a reconstructed discontinuous Galerkin method for compressible flows," *Int. J. Numer. Meth. Fluids*, Vol. 78, No. 3, 2015, pp. 123–139.
- [38] Xia, Y., Luo, H., Frisbey, M., and Nourgaliev, R., "A Set of Parallel, Implicit Methods for a Reconstructed Discontinuous Galerkin Method for Compressible Flows on 3D Hybrid Grid," *Comput. Fluids*, Vol. 98, 2014, pp. 134–151.
- [39] Wang, C., Cheng, J., Berndt, M., N.Carlson, N., and HongLuo, "Application of Nonlinear Krylov Acceleration to a Reconstructed Discontinuous Galerkin Method for Compressible Flows," Vol. 163, 2018, pp. 32–49.
- [40] Liu, X., Xia, Y., and Luo, H., "A Reconstructed Discontinuous Galerkin Method for Compressible Turbulent Flows on 3D curved Grids," *Comput. Fluids*, Vol. 160, 2018, pp. 26–41.
- [41] 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.
- [42] 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.
- [43] Toro, E. F., Spruce, M., and Speares, W., "Restoration of the Contact Surface in the HLL-Riemann Solver," *Shock Waves*, Volume 39, Issue 10, 1994, pp. 25–34.
- [44] R. Kannan and Z.J. Wang, "The direct discontinuous Galerkin (DDG) viscous flux scheme for the high order spectral volume method," *Computers & Fluids*, Vol. 39, 2010, pp. 2007–2021.