AIAA 2015-2451

22nd Computational Fluid Dynamics Conference, 22 - 26 June, Dallas, Texas, 2015

Alternative Formulations for First-, Second-, and Third-Order Hyperbolic Navier-Stokes Schemes

Hiroaki Nishikawa*

National Institute of Aerospace, Hampton, VA 23666, USA

In this paper, we introduce new hyperbolic formulations for the compressible Navier-Stokes equations to enable accurate gradient prediction for density, velocity, pressure, and temperature; construction of a superior hyperbolic finite-volume scheme, which yields one-order-higher accuracy in the high-Reynolds-number limit; and a more economical source-term discretization for third-order accuracy, which does not require computation and storage of second-derivatives. This paper also introduces artificial hyperbolic diffusion and dissipation, which are essential to the new formulations. Numerical results are presented for the first-, second-, and third-order finite-volume schemes applied to the new hyperbolic systems.

I. Introduction

This paper reports further progress in the development of first-, second-, and third-order hyperbolic finite-volume schemes for the Navier-Stokes (NS) equations on unstructured grids. In the previous development [1], we presented first-, second-, and third-order hyperbolic finite-volume schemes for the compressible and incompressible NS equations, demonstrating high-order accurate gradients and accelerated convergence for realistic viscous flow problems on irregular grids. However, it was found for the compressible case that the hyperbolic NS (HNS) schemes had certain limitations. First, although the developed schemes can produce first-, second-, and third-order accurate viscous stresses and heat fluxes on irregular grids, they cannot promise the same order of accuracy for all velocity gradient components. Second, construction of Scheme II [2, 3], which has superior features over the scheme used in the previous study, is not straightforward for the compressible NS equations because the velocity gradients cannot be obtained directly from the viscous stresses and also because accurate density gradients are not available. Scheme II is sought because it has a very attractive feature that it gives one-order-higher accuracy in the advection limit, i.e., first- and second-order schemes yield second- and third-order accuracy in the advection limit, as demonstrated in Ref.[3]. For viscous flow simulations, it would enable construction of a scheme equivalent to a third-order inviscid and second-order viscous scheme by a second-order algorithm. Finally, the source term discretization used in the previous study requires computation and storage of the second derivatives of source terms to achieve third-order accuracy, which we would wish to avoid totally, especially in three dimensions. In this paper, we overcome these limitations by introducing new hyperbolic formulations for the compressible NS equations. We first construct a new hyperbolic system by introducing the velocity gradients, instead of the viscous stresses, as extra variables. The new system requires more variables than the previous system, but it enables a fully hyperbolic formulation by which the computation of the second derivatives can be avoided in the source term discretization. Accuracy of the velocity gradients is, however, not guaranteed by the new system alone, apparently, because of the symmetry of the viscous stress tensor. To resolve the issue, we introduce the idea of *artificial hyperbolic dissipation* as a means to break the symmetry and bring strong coupling among the momentum equations and the equations for the velocity gradients. We then extend the new system further by introducing *artificial hyperbolic diffusion* in the continuity equation in order to obtain accurate density gradients, and thereby enable Scheme II for the compressible Navier-Stokes equations. Numerical results show that the schemes based on the new hyperbolic NS systems successfully overcome the limitations encountered by the previous schemes.

*Associate Research Fellow (hiro@nianet.org), National Institute of Aerospace, 100 Exploration Way, Hampton, VA 23666 USA

II. Compressible Navier-Stokes Equations

Consider the compressible NS equations:

$$\partial_t \rho + \mathrm{div}(\rho \mathbf{v}) = 0, \tag{II.1}$$

$$\partial_t(\rho \mathbf{v}) + \mathrm{div}(\rho \mathbf{v} \otimes \mathbf{v}) = -\mathrm{grad}\, p + \mathrm{div}\, \boldsymbol{\tau}, \tag{II.2}$$

$$\partial_t(\rho E) + \mathrm{div}(\rho \mathbf{v} H) = \mathrm{div}\,(\boldsymbol{\tau} \mathbf{v}) - \mathrm{div}\, \mathbf{q}, \tag{II.3}$$

where $\otimes$ denotes the dyadic product, ρ is the density, $\mathbf{v}$ is the velocity vector, p is the pressure, E is the specific total energy, and $H = E + p/\rho$ is the specific total enthalpy. The viscous stress tensor, $\boldsymbol{\tau}$, and the heat flux, $\mathbf{q}$, are given by

$$\boldsymbol{\tau} = -\frac{2}{3}\mu(\mathrm{div}\, \mathbf{v})\mathbf{I} + \mu \left(\mathrm{grad}\, \mathbf{v} + (\mathrm{grad}\, \mathbf{v})^t\right), \quad \mathbf{q} = -\frac{\mu}{Pr(\gamma - 1)}\mathrm{grad}\, T, \tag{II.4}$$

where $\mathbf{I}$ is the identity matrix, the superscript t denotes the transpose, T is the temperature, γ is the ratio of specific heats, Pr is the Prandtl number, and μ is the viscosity defined by Sutherland's law. Stokes' hypothesis has been assumed. All the quantities are assumed to have been nondimensionalized by their free-stream values except that the velocity and the pressure are scaled by the free-stream speed of sound and the free-stream dynamic pressure, respectively (see Ref.[4]). Thus, the viscosity is given by the following form of Sutherland's law:

$$\mu = \frac{M_\infty}{Re_\infty} \frac{1 + C/\tilde{T}_\infty}{T + C/\tilde{T}_\infty} T^{\frac{3}{2}}, \tag{II.5}$$

where $\tilde{T}_\infty$ is the dimensional free stream temperature, and $C = 110.5$ [K] is the Sutherland constant. The ratio of the free stream Mach number, M_∞, to the free stream Reynolds number, Re_∞, arises from the nondimensionalization. The system is closed by the nondimensionalized equation of state for ideal gases: $\gamma p = \rho T$.

III. A New Hyperbolic Formulation: HNS17

A hyperbolic formulation is constructed by introducing additional variables, which we call the gradient variables since these extra variables are typically associated with the gradients of the main variables. In the previous work [1,5], we constructed a hyperbolic NS system by including the viscous stresses and the heat fluxes as the gradient variables. In this paper, we propose to use the velocity gradients instead of the viscous stresses:

$$\partial_\tau \rho + \mathrm{div}(\rho \mathbf{v}) = 0, \tag{III.1}$$

$$\partial_\tau(\rho \mathbf{v}) + \mathrm{div}(\rho \mathbf{v} \otimes \mathbf{v}) = -\mathrm{grad}\, p + \mathrm{div}\, \boldsymbol{\tau}, \tag{III.2}$$

$$\partial_\tau(\rho E) + \mathrm{div}(\rho \mathbf{v} H) = \mathrm{div}\,(\boldsymbol{\tau} \mathbf{v}) - \mathrm{div}\, \mathbf{q}, \tag{III.3}$$

$$\frac{T_v}{\mu_v}\partial_\tau \mathbf{g} = \mathrm{grad}\, \mathbf{v} - \frac{\mathbf{g}}{\mu_v}, \tag{III.4}$$

$$\frac{T_h}{\mu_h}\partial_\tau \mathbf{q} = -\frac{1}{\gamma(\gamma - 1)}\mathrm{grad}\, T - \frac{\mathbf{q}}{\mu_h}, \tag{III.5}$$

where τ is a pseudo time, μ_v and μ_h are scaled viscosities,

$$\mu_v = \frac{4}{3}\mu, \quad \mu_h = \frac{\gamma \mu}{Pr}, \tag{III.6}$$

and T_v and T_h are relaxation times associated with the velocity gradients and the heat flux, respectively. The system is constructed such that we have in the pseudo steady state or when the pseudo time derivatives are dropped,

$$\mathbf{q} = -\frac{\mu_h}{\gamma(\gamma - 1)}\mathrm{grad}\, T, \quad \mathbf{g} = \mu_v\, \mathrm{grad}\, \mathbf{v}, \tag{III.7}$$

and thus the viscous stress tensor is expressed by

$$\boldsymbol{\tau} = -\frac{1}{2} tr(\mathbf{g})\mathbf{I} + \frac{3}{4}\left(\mathbf{g} + \mathbf{g}^t\right), \tag{III.8}$$

where $tr(\mathbf{g})$ denotes the trace of $\mathbf{g}$. Note that T_v and T_h are free parameters and we define them by

$$T_v = \frac{L^2}{\nu_v}, \quad T_h = \frac{L^2}{\nu_h}, \quad \nu_v = \frac{\mu_v}{\rho}, \quad \nu_h = \frac{\mu_h}{\rho}. \tag{III.9}$$

where L is a length scale defined as in Ref.[1]. In this work, we focus on steady problems, but the formulation is valid for unsteady problems with the physical time derivatives incorporated as source terms. See Refs.[6, 7] for details on the hyperbolic method for unsteady problems. Note that the new hyperbolic NS system consists of the original NS equations and the extra equations for the velocity gradients and the heat fluxes. The total number of equations is 10 in two dimensions and 17 in three dimensions in contrast to 9 in two dimensions and 14 in three dimensions for the system employed in the previous study. To distinguish the new system from the previous one, we call the new system HNS17 and the previous system HNS14. It should be noted that including the velocity gradients as variables does not guarantee by itself the equal of order of accuracy for all variables [1]. However, the equal order of accuracy can be achieved by an improved discretization as we will discuss later in Section VI.F.

IV. An Extended Hyperbolic Formulation: HNS20

HNS17, although useful, is not fully versatile because it does not produce accurate gradients for the density, which are required for construction of Scheme II [2, 3]. A fully versatile hyperbolic system, denoted by HNS20, can be constructed by introducing *artificial hyperbolic diffusion* into HNS17 as follows:

$$\partial_\tau \rho + \mathrm{div}(\rho \mathbf{v}) = \mathrm{div}\, \mathbf{r}, \tag{IV.1}$$

$$\partial_\tau (\rho \mathbf{v}) + \mathrm{div}(\rho \mathbf{v} \otimes \mathbf{v}) = -\mathrm{grad}\, p + \mathrm{div}\, \boldsymbol{\tau}, \tag{IV.2}$$

$$\partial_\tau (\rho E) + \mathrm{div}(\rho \mathbf{v} H) = \mathrm{div}\,(\boldsymbol{\tau} \mathbf{v}) - \mathrm{div}\, \mathbf{q}, \tag{IV.3}$$

$$\frac{T_v}{\mu_v} \partial_\tau \mathbf{g} = \mathrm{grad}\, \mathbf{v} - \frac{\mathbf{g}}{\mu_v}, \tag{IV.4}$$

$$\frac{T_h}{\mu_h} \partial_\tau \mathbf{q} = -\frac{1}{\gamma(\gamma-1)} \mathrm{grad}\, T - \frac{\mathbf{q}}{\mu_h}, \tag{IV.5}$$

$$\frac{T_\rho}{\nu_\rho} \partial_\tau \mathbf{r} = \mathrm{grad}\, \rho - \frac{\mathbf{r}}{\nu_\rho}, \tag{IV.6}$$

where ν_ρ is an artificial hyperbolic-diffusion coefficient, $T_\rho = L^2/\nu_\rho$, and $\mathbf{r}/\nu_\rho$ gives the density gradient in the pseudo steady state or as soon as we set $\partial_\tau \mathbf{r} = 0$. Note that the diffusion term in the continuity equation and Equation (IV.6) form a hyperbolic diffusion system. It is artificial because we set ν_ρ to be extremely small, e.g., $\nu_\rho = 10^{-12}$. As demonstrated in Ref.[3], hyperbolic schemes are capable of producing accurate gradients even with a vanishingly small diffusion coefficient. Therefore, the above system is expected to produce accurate density gradients while not affecting the continuity equation. As a result, this system allows us to obtain accurate gradients for all the primitive variables in the pseudo steady state or when the pseudo time derivatives are dropped:

$$\mathrm{grad}\, \rho = \frac{\mathbf{r}}{\nu_\rho}, \quad \mathrm{grad}\, \mathbf{v} = \frac{\mathbf{g}}{\mu_v}, \quad \mathrm{grad}\, T = -\frac{\gamma(\gamma-1)\mathbf{q}}{\mu_h}, \tag{IV.7}$$

and, therefore, it enables construction of Scheme II.

In two dimensions, we have

$$\mathbf{v} = \begin{bmatrix} u \\ v \end{bmatrix}, \quad \mathbf{g} = \begin{bmatrix} g_{ux} & g_{uy} \\ g_{vx} & g_{vy} \end{bmatrix}, \quad \boldsymbol{\tau} = \begin{bmatrix} \tau_{xx} & \tau_{xy} \\ \tau_{yx} & \tau_{yy} \end{bmatrix}, \quad \mathbf{q} = \begin{bmatrix} q_x \\ q_y \end{bmatrix}, \quad \mathbf{r} = \begin{bmatrix} r_x \\ r_y \end{bmatrix}, \tag{IV.8}$$

and HNS20 is written as

$$\mathbf{P}^{-1}\partial_t\mathbf{U} + \partial_x\mathbf{F} + \partial_y\mathbf{G} = \mathbf{S}, \tag{IV.9}$$

where

$$\mathbf{U}=\begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho E \\ g_{ux} \\ g_{uy} \\ g_{vx} \\ g_{vy} \\ q_x \\ q_y \\ r_x \\ r_y \end{bmatrix}, \mathbf{F}=\begin{bmatrix} \rho u - r_x \\ \rho u^2 + p - \tau_{xx} \\ \rho uv - \tau_{xy} \\ \rho uH - \tau_{xx}u - \tau_{xy}v + q_x \\ -u \\ 0 \\ -v \\ 0 \\ \dfrac{T}{\gamma(\gamma-1)} \\ 0 \\ -\rho \\ 0 \end{bmatrix}, \mathbf{G}=\begin{bmatrix} \rho v - r_y \\ \rho uv - \tau_{xy} \\ \rho v^2 + p - \tau_{yy} \\ \rho vH - \tau_{yx}u - \tau_{yy}v + q_y \\ 0 \\ -u \\ 0 \\ -v \\ 0 \\ \dfrac{T}{\gamma(\gamma-1)} \\ 0 \\ -\rho \end{bmatrix}, \mathbf{S}=\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ -g_{ux}/\mu_v \\ -g_{uy}/\mu_v \\ -g_{vx}/\mu_v \\ -g_{vy}/\mu_v \\ -q_x/\mu_h \\ -q_y/\mu_h \\ -r_x/\nu_\rho \\ -r_y/\nu_\rho \end{bmatrix}, \tag{IV.10}$$

$$\mathbf{P}^{-1} = \mathrm{diag}(1,1,1,1,T_v/\mu_v,T_v/\mu_v,T_v/\mu_v,T_v/\mu_v,T_h/\mu_h,T_h/\mu_h,T_\rho/\nu_\rho,T_\rho/\nu_\rho). \tag{IV.11}$$

The projected flux $\mathbf{F}_n$ in the direction of an arbitrary unit vecor $\mathbf{n} = (n_x, n_y)$, which is relevant to the finite-volume discretization, is written as a sum of the inviscid flux $\mathbf{F}_n^i$, the viscous flux $\mathbf{F}_n^v$, and the artificial hyperbolic diffusion flux $\mathbf{F}_n^a$:

$$\mathbf{F}_n = \mathbf{F}n_x + \mathbf{G}n_y = \mathbf{F}_n^i + \mathbf{F}_n^v + \mathbf{F}_n^a, \tag{IV.12}$$

where

$$\mathbf{F}_n^i = \begin{bmatrix} \rho u_n \\ \rho u_n u + p n_x \\ \rho u_n v + p n_y \\ \rho u_n H \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{F}_n^v = \begin{bmatrix} 0 \\ -\tau_{nx} \\ -\tau_{ny} \\ -\tau_{nv} + q_n \\ -u n_x \\ -u n_y \\ -v n_x \\ -v n_y \\ \dfrac{T}{\gamma(\gamma-1)} n_x \\ \dfrac{T}{\gamma(\gamma-1)} n_y \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{F}_n^a = \begin{bmatrix} -r_n \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -\rho n_x \\ -\rho n_y \end{bmatrix}, \tag{IV.13}$$

$$\tau_{nx} = \tau_{xx} n_x + \tau_{xy} n_y, \quad \tau_{ny} = \tau_{yx} n_x + \tau_{yy} n_y, \quad \tau_{nv} = \tau_{nx} u + \tau_{ny} v, \tag{IV.14}$$

$$u_n = u n_x + v n_y, \quad q_n = q_x n_x + q_y n_y, \quad r_n = r_x n_x + r_y n_y. \tag{IV.15}$$

Observe that the viscous flux $\mathbf{F}_n^v$ is identical to that of HNS17, and thus the only difference between HNS17 and HNS20 is the artificial hyperbolic diffusion flux, $\mathbf{F}_n^a$. For the sake of simplicity, we will treat these fluxes separately when we construct numerical fluxes.

V. Special Hyperbolic Formulation for Source Terms

A special treatment is required for source terms to achieve third-order accuracy. A point source quadrature formula, which is typically employed in second-order schemes, is known to destroy third-order accuracy [8]. Three techniques are available to achieve third-order accuracy: an extended Galerkin formula [9], a general divergence formulation of source terms [8], and a fully hyperbolic formulation of hyperbolic diffusion systems [2, 3]. The best choice for the hyperbolic NS schemes is the fully hyperbolic formulation where the source terms are hyperbolized as well. The main advantage is that computation and storage of the second derivatives of the source terms, which is required by the other two techniques, is not required. For the two-dimensional HNS17/20 system (a slight modification is required in three dimensions, and it will be reported elsewhere), it can be formulated as follows:

$$\mathbf{P}^{-1}\partial_t \mathbf{U} + \partial_x \mathbf{F} + \partial_y \mathbf{G} = \mathbf{0}, \tag{V.1}$$

where $\mathbf{F}$ and $\mathbf{G}$ now include source term fluxes, which we write in the form of the normal flux as

$$\mathbf{F}_n = \mathbf{F}_n^i + \mathbf{F}_n^v + \mathbf{F}_n^a + \mathbf{F}_n^{g_u} + \mathbf{F}_n^{g_v} + \mathbf{F}_n^q + \mathbf{F}_n^r, \tag{V.2}$$

$$\mathbf{F}_n^{g_u} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ -\omega_{g_u}(y-y_j) \\ \omega_{g_u}(x-x_j) \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \mathbf{F}_n^{g_v} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -\omega_{g_v}(y-y_j) \\ \omega_{g_v}(x-x_j) \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \mathbf{F}_n^{q} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -\omega_{q}(y-y_j) \\ \omega_{q}(x-x_j) \\ 0 \\ 0 \end{bmatrix}, \mathbf{F}_n^{r} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -\omega_{r}(y-y_j) \\ \omega_{r}(x-x_j) \end{bmatrix}, \tag{V.3}$$

$$\omega_{g_u} = \left(\frac{g_{uy}}{\mu_v}\right) n_x - \left(\frac{g_{ux}}{\mu_v}\right) n_y, \quad \omega_{g_v} = \left(\frac{g_{vy}}{\mu_v}\right) n_x - \left(\frac{g_{vx}}{\mu_v}\right) n_y, \tag{V.4}$$

$$\omega_q = \left(\frac{q_y}{\mu_h}\right) n_x - \left(\frac{q_x}{\mu_h}\right) n_y, \quad \omega_r = \left(\frac{r_y}{\nu_\rho}\right) n_x - \left(\frac{r_x}{\nu_\rho}\right) n_y, \tag{V.5}$$

where x_j and y_j are the x and y coordinates of the node at which the residual will be defined. Observe that these fluxes do not contain derivatives. This is the reason that second derivatives are not needed, and only first derivatives are required in the finite-volume discretization, meaning a huge saving in the computational work especially in three dimensions. It is straightforward to show (see Ref.[2]) that the above system is equivalent to Equation (IV.9) under the following conditions:

$$\frac{\partial}{\partial x}\left(\frac{g_{uy}}{\mu_v}\right) - \frac{\partial}{\partial y}\left(\frac{g_{ux}}{\mu_v}\right) = 0, \quad \frac{\partial}{\partial x}\left(\frac{g_{vy}}{\mu_v}\right) - \frac{\partial}{\partial y}\left(\frac{g_{vx}}{\mu_v}\right) = 0, \tag{V.6}$$

$$\frac{\partial}{\partial x}\left(\frac{q_y}{\mu_h}\right) - \frac{\partial}{\partial y}\left(\frac{q_x}{\mu_h}\right) = 0, \quad \frac{\partial}{\partial x}\left(\frac{r_y}{\nu_\rho}\right) - \frac{\partial}{\partial y}\left(\frac{r_x}{\nu_\rho}\right) = 0, \tag{V.7}$$

which are satisfied in the pseudo steady state or simply when the pseudo-time derivatives are dropped. One can prove these conditions by ignoring the pseudo-time derivatives in Equations (IV.4), (IV.5), (IV.6), solving for $\mathbf{g}/\mu_v$, $\mathbf{q}/\mu_h$, and $\mathbf{r}/\nu_\rho$, and then substituting them into the above equations. Each flux can be analyzed as a 2×2 system, and can be shown to be hyperbolic with real eigenvalues and linearly independent eigenvectors [2, 3]. We therefore discretize these source terms by upwind fluxes defined independently.

The fully hyperbolic formulation may be constructed for HNS14 as well, but it is very difficult to avoid derivatives in the source fluxes and thus it requires computation and storage of second derivatives in the discretization. This difficulty comes from the fact that the constraint on the viscous stresses is, unlike Equations (V.6) and (V.7), not first order but second order as shown in Ref.[1].

VI. Node-Centered Edge-Based Finite-Volume Discretization

VI.A. Discretization

We discretize the HNS20 system by the node-centered edge-based finite-volume method as described in Refs.[1, 5], and obtain the residual at a node j in the form:

$$\mathbf{Res}_j = -\mathbf{P}_j \left(\sum_{k\in\{k_j\}} \mathbf{\Phi}_{jk} A_{jk} - \mathbf{S}_j V_j \right), \tag{VI.1}$$

where $\mathbf{P}_j$ is the preconditioning matrix evaluated at the node j, V_j is the measure of the dual control volume around the node j, $\{k_j\}$ is a set of edge-connected neighbors of j, $\mathbf{\Phi}_{jk}$ is a numerical flux, and A_{jk} is the magnitude of the directed area vector, i.e., $A_{jk} = |\mathbf{n}_{jk}| = |\mathbf{n}_{jk}^{\ell} + \mathbf{n}_{jk}^{r}|$ (see Figure 1). The numerical flux is evaluated by the left and right states reconstructed with linear LSQ gradients for second-order accuracy as we will discuss later.

As discovered in Ref.[10] for hyperbolic systems, the edge-based finite-volume discretization yields third-order accuracy on simplex grids if the solution and the flux are linearly extrapolated with quadratic LSQ gradients. Although extensions to source terms and the viscous terms require special techniques and additional work, we can avoid all that with the fully hyperbolic formulation (V.1). Because it is a hyperbolic system and has no source terms, we can directly apply the third-order edge-based discretization to obtain the residual in the form:

$$\mathbf{Res}_j = -\mathbf{P}_j \left(\sum_{k\in\{k_j\}} \mathbf{\Phi}'_{jk} A_{jk} \right), \tag{VI.2}$$

where $\mathbf{\Phi}'_{jk}$ includes numerical fluxes for the hyperbolized source terms (V.3). One may notice at this point that the third-order edge-based scheme is highly efficient since the third-order residual can be computed, just like the second-order residual, in a loop over edges with *a single flux evaluation per edge*.

Note that we have dropped all the pseudo-time derivatives at this point. As mentioned repeatedly, the hyperbolic formulation becomes equivalent to the Navier-Stokes equations as soon as the pseudo-time derivatives are dropped. Therefore, the residuals as defined in Equations (VI.1) and (VI.2) represent *consistent* discretizations of the steady Navier-Stokes equations. In effect, we have discretized the Navier-Stokes equations consistently by discretizing the hyperbolic Navier-Stokes system, HNS20.

VI.B. Numerical Flux

The numerical flux is constructed as a sum of an upwind inviscid flux, an upwind viscous flux, and an upwind artificial diffusion flux:

$$\mathbf{\Phi}_{jk} = \mathbf{\Phi}_{jk}^{i} + \mathbf{\Phi}_{jk}^{v} + \mathbf{\Phi}_{jk}^{a}, \tag{VI.3}$$

$$\begin{aligned}\mathbf{\Phi}^i_{jk} &= \frac{1}{2}\left[(\mathbf{F}^i_n)_R + (\mathbf{F}^i_n)_L\right] - \frac{1}{2}\left|\mathbf{A}^i_n\right|\Delta\mathbf{U}, &\text{(VI.4)}\\
\mathbf{\Phi}^v_{jk} &= \frac{1}{2}\left[(\mathbf{F}^v_n)_R + (\mathbf{F}^v_n)_L\right] - \frac{1}{2}\mathbf{P}^{-1}\left|\mathbf{P}\mathbf{A}^v_n\right|\Delta\mathbf{U},, &\text{(VI.5)}\\
\mathbf{\Phi}^a_{jk} &= \frac{1}{2}\left[(\mathbf{F}^a_n)_R + (\mathbf{F}^a_n)_L\right] - \frac{1}{2}\mathbf{P}^{-1}\left|\mathbf{P}\mathbf{A}^a_n\right|\Delta\mathbf{U}, &\text{(VI.6)}\end{aligned}$$

where the subscripts L and R indicate values at the left and right sides of the edge midpoint, $\Delta\mathbf{U} = \mathbf{U}_R - \mathbf{U}_L$, and $\mathbf{A}^i_n$, $\mathbf{A}^v_n$, and $\mathbf{A}^a_n$ are the normal flux Jacobians,

$$\mathbf{A}^i_n = \frac{\partial \mathbf{F}^i_n}{\partial \mathbf{U}}, \quad \mathbf{A}^v_n = \frac{\partial \mathbf{F}^v_n}{\partial \mathbf{U}}, \quad \mathbf{A}^a_n = \frac{\partial \mathbf{F}^a_n}{\partial \mathbf{U}}. \tag{VI.7}$$

Note that the dissipation term has been constructed based on the well-known procedure in the local preconditioning technique [11]. Namely, the dissipation matrix is constructed based on the preconditioned flux-Jacobian, $|\mathbf{P}\mathbf{A}^v_n|$, and then it is multiplied by $\mathbf{P}^{-1}$ to cancel the effect of $\mathbf{P}_j$ by which the residual will be multiplied from the left as in Equations (VI.1) and (VI.2). The interface quantities needed to evaluate the dissipation matrix are computed by the Roe-averages [12] in the inviscid flux, and otherwise by the arithmetic averages. The dissipation term for the viscous flux can be simplified and implemented as a single vector, and similarly for the artificial diffusion flux:

$$|\mathbf{P}\mathbf{A}^v_n|\,\Delta\mathbf{U} = \begin{bmatrix} 0 \\ \rho\{a_{nv}\Delta u_n n_x + a_{mv}(\Delta u - \Delta u_n n_x)\} \\ \rho\{a_{nv}\Delta u_n n_y + a_{mv}(\Delta v - \Delta u_n n_y)\} \\ \rho\{a_{nv}u_n\Delta u_n + a_{mv}(u\Delta u + v\Delta v - u_n\Delta u_n)\} + \dfrac{\rho a_h \Delta T}{\gamma(\gamma-1)} \\ + \dfrac{\tau_{nn}\Delta\tau_{nn}}{\rho a_h(Pr_n+1)} + \dfrac{\tau_{nx}\Delta\tau_{nx} + \tau_{ny}\Delta\tau_{ny} - \tau_{nn}\Delta\tau_{nn}}{\rho a_h(Pr_m+1)} \\ a_{nv}\Delta\tau_{nn}n_x n_x + \dfrac{4}{3}a_{mv}(\Delta\tau_{nx} - \Delta\tau_{nn}n_x)n_x \\ a_{nv}\Delta\tau_{nn}n_x n_y + \dfrac{4}{3}a_{mv}(\Delta\tau_{nx} - \Delta\tau_{nn}n_x)n_y \\ a_{nv}\Delta\tau_{nn}n_y n_x + \dfrac{4}{3}a_{mv}(\Delta\tau_{ny} - \Delta\tau_{nn}n_y)n_x \\ a_{nv}\Delta\tau_{nn}n_y n_y + \dfrac{4}{3}a_{mv}(\Delta\tau_{ny} - \Delta\tau_{nn}n_y)n_y \\ a_h\left(\Delta q_n - \dfrac{\tau_{nn}\Delta u_n}{Pr_n+1} - \dfrac{\tau_{nx}\Delta u + \tau_{ny}\Delta v - \tau_{nn}\Delta u_n}{Pr_m+1}\right)n_x \\ a_h\left(\Delta q_n - \dfrac{\tau_{nn}\Delta u_n}{Pr_n+1} - \dfrac{\tau_{nx}\Delta u + \tau_{ny}\Delta v - \tau_{nn}\Delta u_n}{Pr_m+1}\right)n_y \\ 0 \\ 0 \end{bmatrix}, \quad |\mathbf{P}\mathbf{A}^a_n|\,\Delta\mathbf{U} = \begin{bmatrix} \dfrac{\nu_\rho}{L}\Delta\rho \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \dfrac{\nu_\rho}{L}\Delta r_n n_x \\ \dfrac{\nu_\rho}{L}\Delta r_n n_y \end{bmatrix}, \tag{VI.8}$$

where

$$a_{nv} = \sqrt{\frac{\nu_v}{T_v}}, \quad a_{mv} = \sqrt{\frac{3\nu_v}{4T_v}}, \quad a_h = \sqrt{\frac{\nu_h}{T_h}}, \quad \tau_{nn} = \tau_{nx}n_x + \tau_{ny}n_y, \quad Pr_n = \frac{a_{nv}}{a_h}, \quad Pr_m = \frac{a_{mv}}{a_h}. \tag{VI.9}$$

For the third-order scheme, the numerical flux has additional contributions from the source-term fluxes:

$$\mathbf{\Phi}'_{jk} = \mathbf{\Phi}^i_{jk} + \mathbf{\Phi}^v_{jk} + \mathbf{\Phi}^a_{jk} + \mathbf{\Phi}^{g_u}_{jk} + \mathbf{\Phi}^{g_v}_{jk} + \mathbf{\Phi}^q_{jk} + \mathbf{\Phi}^r_{jk}, \tag{VI.10}$$

where $\mathbf{\Phi}_{jk}^{g_u}$ is an upwind flux defined by

$$\mathbf{\Phi}_{jk}^{g_u} \;=\; \frac{1}{2}\left[(\mathbf{F}_n^{g_u})_R + (\mathbf{F}_n^{g_u})_L\right] - \frac{1}{2}\mathbf{P}^{-1}\left|\mathbf{P}\mathbf{A}_n^{g_u}\right|\Delta\mathbf{U}, \quad \mathbf{A}_n^{g_u} = \frac{\partial \mathbf{F}_n^{g_u}}{\partial \mathbf{U}}, \tag{VI.11}$$

and similarly for $\mathbf{\Phi}_{jk}^{g_v}$, $\mathbf{\Phi}_{jk}^{q}$, and $\mathbf{\Phi}_{jk}^{r}$. See Ref.[3] for implementation details of these fluxes.

VI.C. Kappa Scheme for Reconstruction

For second- and third-order accuracy, the solution needs to be reconstructed at the midpoint of each edge. The reconstruction is performed in the following variables:

$$\mathbf{W} = [\rho, u, v, T, g_{ux}, g_{uy}, g_{vx}, g_{vy}, q_x, q_y, r_x, r_y]^t, \tag{VI.12}$$

to define two states, $\mathbf{W}_L$ and $\mathbf{W}_R$, which are constructed from the two end nodes of the edge, j and k, respectively (see Figure 1). If necessary, the conservative variables, $\mathbf{U}_L$ and $\mathbf{U}_R$, can be computed from $\mathbf{W}_L$ and $\mathbf{W}_R$, respectively. In the previous work, we employed a simple linear extrapolation:

$$\mathbf{W}_L = \mathbf{W}_j + \frac{1}{2}\nabla\mathbf{W}_j \cdot \Delta\mathbf{l}_{jk}, \qquad \mathbf{W}_R = \mathbf{W}_k - \frac{1}{2}\nabla\mathbf{W}_k \cdot \Delta\mathbf{l}_{jk}, \tag{VI.13}$$

where $\Delta\mathbf{l}_{jk} = (x_k - x_j, y_k - y_j)$, $\nabla\mathbf{W}_j$ is a least-squares (LSQ) gradient of $\mathbf{W}$ computed at the node j by linear and quadratic fits for second- and third-order schemes, respectively, and similarly for $\nabla\mathbf{W}_k$. In this work, we employ the kappa scheme [13, 14]:

$$\mathbf{W}_L \;=\; \mathbf{W}_j + \left[\frac{1-\kappa}{2}(\partial_{jk}\mathbf{W}_j - \Delta\mathbf{W}_{jk}) + \frac{1+\kappa}{2}\Delta\mathbf{W}_{jk}\right], \tag{VI.14}$$

$$\mathbf{W}_R \;=\; \mathbf{W}_k - \left[\frac{1-\kappa}{2}(\partial_{jk}\mathbf{W}_k - \Delta\mathbf{W}_{jk}) + \frac{1+\kappa}{2}\Delta\mathbf{W}_{jk}\right], \tag{VI.15}$$

where κ is a parameter, and

$$\partial_{jk}\mathbf{W}_j \,=\, \nabla\mathbf{W}_j \cdot \Delta\mathbf{l}_{jk}, \quad \partial_{jk}\mathbf{W}_k \,=\, \nabla\mathbf{W}_k \cdot \Delta\mathbf{l}_{jk}, \quad \Delta\mathbf{W}_{jk} = \frac{1}{2}\left(\mathbf{W}_k - \mathbf{W}_j\right). \tag{VI.16}$$

While the linear extrapolation (VI.13) is reproduced by $\kappa = 0$, we take $\kappa = 1/2$ in this work, which reconstructs the edge-midpoint value exactly for quadratic solutions on arbitrary grids if the LSQ gradients are computed exactly for quadratic solutions. In other words, the kappa scheme will be equivalent to a quadratic extrapolation with the choice $\kappa = 1/2$ on arbitrary grids. That is precisely the case for the third-order scheme in which the LSQ gradients are computed by a quadratic fit. It is emphasized, however, that the quadratic extrapolation with $\kappa = 1/2$ is employed here not to improve the order of accuracy but just to provide accurate face values for the dissipation terms.

In general, the left and right fluxes are evaluated differently in the second- and third-order schemes. In the second-order scheme, it is common to evaluate the numerical fluxes by the reconstructed solutions. For example, the inviscid flux is evaluated as

$$\mathbf{\Phi}_{jk}^{i} = \frac{1}{2}\left[\mathbf{F}_n^{i}(\mathbf{U}_R) + \mathbf{F}_n^{i}(\mathbf{U}_L)\right] - \frac{1}{2}\left|\mathbf{A}_n^{i}\right|\Delta\mathbf{U}, \tag{VI.17}$$

and similarly for other fluxes. In the third-order scheme, the left and right fluxes need to be linearly extrapolated to the midpoint [10, 15] . For example, the inviscid flux is evaluated as

$$\mathbf{\Phi}_{jk}^{i} = \frac{1}{2}\left[(\mathbf{F}_n^{i})_R + (\mathbf{F}_n^{i})_L\right] - \frac{1}{2}\left|\mathbf{A}_n^{i}\right|\Delta\mathbf{U}, \tag{VI.18}$$

where

$$(\mathbf{F}_n^{i})_L = (\mathbf{F}_n^{i})_j + \frac{1}{2}\left(\frac{\partial \mathbf{F}_n^{i}}{\partial \mathbf{W}}\right)_j \nabla\mathbf{W}_j \cdot \Delta\mathbf{l}_{jk}, \tag{VI.19}$$

$$(\mathbf{F}_n^{i})_R = (\mathbf{F}_n^{i})_k - \frac{1}{2}\left(\frac{\partial \mathbf{F}_n^{i}}{\partial \mathbf{W}}\right)_k \nabla\mathbf{W}_k \cdot \Delta\mathbf{l}_{jk}. \tag{VI.20}$$

and similarly for other fluxes. Note that although it is possible to employ the kappa scheme for the flux extrapolation as well, e.g., again for the inviscid flux as an example,

$$(\mathbf{F}_n^i)_L = (\mathbf{F}_n^i)_j + \left[\frac{1-\kappa}{2}(\partial_{jk}(\mathbf{F}_n^i)_j - \Delta(\mathbf{F}_n^i)_{jk}) + \frac{1+\kappa}{2}\Delta(\mathbf{F}_n^i)_{jk}\right], \tag{VI.21}$$

$$(\mathbf{F}_n^i)_R = (\mathbf{F}_n^i)_k - \left[\frac{1-\kappa}{2}(\partial_{jk}(\mathbf{F}_n^i)_k - \Delta(\mathbf{F}_n^i)_{jk}) + \frac{1+\kappa}{2}\Delta(\mathbf{F}_n^i)_{jk}\right], \tag{VI.22}$$

where $\Delta(\mathbf{F}_n^i)_{jk} = \frac{1}{2}\left((\mathbf{F}_n^i)_k - (\mathbf{F}_n^i)_j\right)$ and

$$\partial_{jk}(\mathbf{F}_n^i)_j = \left(\frac{\partial \mathbf{F}_n^i}{\partial \mathbf{W}}\right)_j \nabla \mathbf{W}_j \cdot \Delta \mathbf{l}_{jk}, \quad \partial_{jk}(\mathbf{F}_n^i)_k = \left(\frac{\partial \mathbf{F}_n^i}{\partial \mathbf{W}}\right)_k \nabla \mathbf{W}_k \cdot \Delta \mathbf{l}_{jk}, \tag{VI.23}$$

we have no choice but $\kappa = 0$ for the flux extrapolation because any other value of κ will destroy third-order accuracy [15]. It is quite interesting that the choice $\kappa = 1/2$ yields a quadratic flux extrapolation, which is exact for quadratic fluxes, but then the residual will not be exact for quadratic fluxes, and thus third-order accuracy will be lost. Degraded accuracy with $\kappa = 1/2$ has been confirmed by numerical results as we will show later. Third-order accuracy of the node-centered edge-based discretization is a very special property that holds on simplex grids and with the linear flux extrapolation ($\kappa = 0$). See Ref.[15] for details.

VI.D. Scheme II

As demonstrated for model equations in the previous work [2, 3], replacing the LSQ gradients by those obtained from the gradient variables results in a superior scheme. The resulting scheme is called Scheme II, and in contrast, the one based on all LSQ gradients is called Scheme I [2,3]. For HNS20, the gradients of the primitive variables (ρ, u, v, T) can be obtained directly from the gradient variables. Therefore, we do not need to perform the LSQ gradient computation for them, and thus we can construct Scheme II by setting

$$\nabla \rho_j = \frac{\mathbf{r}_j}{\nu_\rho}, \quad \nabla \mathbf{v}_j = \frac{\mathbf{g}_j}{(\mu_v)_j}, \quad \nabla T_j = -\gamma(\gamma-1)\frac{\mathbf{q}_j}{(\mu_h)_j}. \tag{VI.24}$$

In this way, the first-order scheme, where all LSQ gradients are turned off and set equal to zero, can have linearly reconstructed solutions for (ρ, u, v, T) at face. These are the variables relevant to the inviscid terms. In the high-Reynolds-number limit, the viscous contributions will be negligibly small but will produce first-order accurate gradients, and as a result the first-order scheme will reduce to a second-order inviscid scheme. Likewise, the second-order scheme will reduce to a third-order inviscid scheme with second-order gradients generated by the second-order hyperbolic viscous scheme, provided the inviscid fluxes are linearly extrapolated as described in the previous section. For this reason, we will apply the flux extrapolation ($\kappa = 0$) to the inviscid flux not only in the third-order scheme but also in the second-order scheme.

VI.E. Boundary Conditions and Flux Quadrature

In this work, we consider weakly imposing boundary conditions, including viscous wall boundaries. The solutions at boundary nodes are, therefore, all determined by solving the residual equations. In the node-centered edge-based finite-volume method, the residual needs to be closed at a boundary node, where a boundary condition is imposed weakly, by integrating the flux along the dual boundary faces. It is well known that second-order accuracy can be lost unless the boundary flux quadrature is designed to make the residual exact for linear fluxes, i.e., fluxes varying linearly in space. Formulas that ensure the exactness for linear fluxes on triangles and tetrahedra have been known for a long time [16], and those for other types of elements, such as hexahedra, prisms, pyramids, etc., and their derivations are available in Appendix B in Ref.[17]. A general formula that guarantees the exactness for linear and quadratic fluxes and thus works for both second- and third-order schemes has recently been derived in Ref.[15], and it is used here for both second- and third-order schemes. The boundary flux quadrature requires the flux computed at each boundary node, and it is evaluated by the numerical flux in Section VI.B with the nodal solution as the left state and a ghost state, where physical conditions are imposed, as the right state.

VI.F. Artificial Hyperbolic Dissipation

Numerical experiments for two-dimensional computations show that the order of accuracy is deteriorated by one order for the velocity gradients, u_y and v_x. It is interesting, however, that the design order of accuracy is achieved in the viscous stress component, $\tau_{xy} = 3(\partial_y u + \partial_x v)/4$. It indicates that simply carrying the velocity gradients as variables does not guarantee their order of accuracy, and that these components are coupled only in the form of τ_{xy} by symmetry. In fact, it is known from the previous work [2] that the order of accuracy of the gradient variables can deteriorate by one order if the system decouples in the discrete level. In the case of the upwind flux, the dissipation term has been identified as an essential term for strong coupling in Ref.[2]. Therefore, in order to generate strong coupling for all components of $\mathbf{g}$, we propose to add the dissipation term of the upwind hyperbolic flux developed for diffusion. Consider the following hyperbolic diffusion system:

$$\begin{aligned} \partial_\tau(\rho\mathbf{v}) &= \operatorname{div}\mathbf{g}, && \text{(VI.25)} \\ \frac{T_v}{\mu_v}\partial_\tau\mathbf{g} &= \operatorname{grad}\mathbf{v} - \frac{\mathbf{g}}{\mu_v}, && \text{(VI.26)} \end{aligned}$$

which we write in two dimensions

$$\mathbf{P}^{-1}\partial_t\mathbf{U} + \partial_x\mathbf{F}^d + \partial_y\mathbf{G}^d = \mathbf{S}^d, \tag{VI.27}$$

where

$$\mathbf{F}^d = \begin{bmatrix} 0 \\ -g_{ux} \\ -g_{vx} \\ 0 \\ -u \\ 0 \\ -v \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{G}^d = \begin{bmatrix} 0 \\ -g_{uy} \\ -g_{vy} \\ 0 \\ 0 \\ -u \\ 0 \\ -v \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{S}^d = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ -g_{ux}/\mu_v \\ -g_{uy}/\mu_v \\ -g_{vx}/\mu_v \\ -g_{vy}/\mu_v \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \tag{VI.28}$$

The upwind flux can be constructed as

$$\mathbf{\Phi}^d_{jk} = \frac{1}{2}\left[(\mathbf{F}^d_n)_R + (\mathbf{F}^d_n)_L\right] - \frac{1}{2}\mathbf{P}^{-1}\left|\mathbf{P}\mathbf{A}^d_n\right|\Delta\mathbf{U}, \quad \mathbf{A}^d_n = \frac{\partial\mathbf{F}^d_n}{\partial\mathbf{U}} = \frac{\partial\left(\mathbf{F}^d n_x + \mathbf{G}^d n_y\right)}{\partial\mathbf{U}}. \tag{VI.29}$$

Note that it is wrong to add this flux to the numerical flux, $\mathbf{\Phi}_{jk}$ or $\mathbf{\Phi}'_{jk}$, which will make the discretization inconsistent. To ensure strong coupling, it suffices to add the dissipation term alone as *artificial hyperbolic dissipation*:

$$\mathbf{\Phi}_{jk} = \mathbf{\Phi}^i_{jk} + \mathbf{\Phi}^v_{jk} + \mathbf{\Phi}^a_{jk} - \frac{1}{2}\mathbf{P}^{-1}\left|\mathbf{P}\mathbf{A}^d_n\right|\Delta\mathbf{U}, \tag{VI.30}$$

and similarly in the numerical flux for the third-order scheme, $\mathbf{\Phi}'_{jk}$. The dissipation term has the order property and thus does not affect the consistency of the scheme. But it does introduce strong coupling among the variables that is critical for the gradient accuracy [2]. Unlike conventional numerical dissipation, the artificial hyperbolic dissipation is here introduced to raise the order of accuracy by one order.

Numerical experiments indicate, as we will show later, that the third-order scheme does not suffer from the accuracy deterioration in the velocity gradients, and thus does not require the artificial hyperbolic dissipation. It implies that the strong coupling necessary for accuracy has already been introduced in some other form, which would be the fully hyperbolic formulation used in the third-order scheme (not used in others). In fact, the upwind fluxes employed to discretize the source fluxes, e.g., Equation (VI.11), have dissipation terms, and these terms can actually ensure strong coupling in the discrete level. Nevertheless, in this work, we keep the artificial hyperbolic dissipation in all schemes. Note also that the fully hyperbolic formulation may be employed in the first- and second-order schemes to achieve the design accuracy in the velocity gradients, instead of the artificial hyperbolic dissipation. However, we do not consider such an option here because the artificial hyperbolic dissipation is simpler to implement.

VII. Jacobin-Free Newton-Krylov Solver

The node-centered edge-based discretization leads to a global system of residual equations:

$$0 = \mathbf{Res}(\mathbf{U}_h), \tag{VII.1}$$

where $\mathbf{U}_h$ denotes the global solution vector for which the system is to be solved. To solve the nonlinear system, we consider a Jacobin-Free Newton-Krylov solver based on the Generalized Conjugate Residual (GCR) method with the implicit defect-correction solver developed in the previous work [1] employed as a variable preconditioner. The preconditioning is performed by a multi-color Gauss-Seidel relaxation to reduce the linear preconditioner-residual with the exact Jacobian of the first-order scheme (Scheme I), by one order of magnitude or until the maximum of 50 is reached. The GCR projection is performed to reduce the target linear residual:

$$\frac{\mathbf{Res}(\mathbf{U}_h^n + \epsilon \Delta \mathbf{U}_h) - \mathbf{Res}(\mathbf{U}_h^n)}{\epsilon} = -\mathbf{Res}(\mathbf{U}_h^n), \tag{VII.2}$$

where ϵ is a small parameter as defined in Ref.[18], by one order of magnitude or to reach the maximum of 10 projections, and then the solution is updated as

$$\mathbf{U}_h^{n+1} = \mathbf{U}_h^n + \Delta \mathbf{U}_h. \tag{VII.3}$$

The parameter setting employed in this work is by no means optimal in any sense. Ideally, the preconditioning should be performed by a multigrid method to reduce the grid dependence on the convergence of a relaxation scheme. It is also possible to employ the Newton-Krylov solver as a smoother in a nonlinear multigrid method.

VIII. Numerical Results

The first-, second-, and third-order hyperbolic NS schemes based on the HNS20 system have been tested for a set of steady laminar cases, and compared with the results obtained by a conventional scheme. The first-, second-, and third-order hyperbolic schemes built upon Scheme II are referred to as HNS20-II(1st), HNS20-II(2nd), HNS20-II(3rd), respectively; those based on Scheme I are referred to as HNS20(1st), HNS20(2nd), HNS20(3rd). The conventional scheme is the same second-order node-centered edge-based finite-volume used for comparison in the previous work [1]. It is based on the Roe inviscid flux and the alpha-damping viscous flux, and referred to as Alpha4/3 [1]. Convergence is taken to be achieved by the residual reduction of six orders of magnitude unless otherwise stated.

VIII.A. Accuracy Verification

Accuracy verification has been performed by the method of manufactured solutions as described in Ref.[1], with the parameters: $M_\infty = 0.3$, $Pr = 3/4$, $\gamma = 1.4$, $Re_\infty = 50$ and 10^8, and $T_\infty = 300$ [K]. The discretization error is measured in L_1 norm over a series of 15 irregular triangular grids with 1024, 2304, 4096, 6400, 9216, 12544, 16384, 20736, 25600, 30976, 36864, 43264, 50176, 57600, and 65536 nodes in a square domain with the Dirichlet condition. For all cases, the discrete problem has been fully solved in the sense that the residuals have been reduced by ten orders of magnitude (or to a machine zero). For the third-order scheme, we employ a two-step implementation of a quadratic LSQ fit as described in Ref.[3].

Results for $Re_\infty = 50$ are shown in Figure 2. First-, second-, and third-order accuracy of the hyperbolic schemes have been verified for all variables. Some strange behavior is seen for the velocity components and the pressure in the case of the first-order scheme; but eventually the errors go down with first-order accuracy. It is observed also that the order of accuracy for the velocity gradients has been improved successfully by the artificial hyperbolic dissipation. To illustrate the improvement in the velocity gradients, results obtained without the artificial hyperbolic dissipation are shown in Figure 3. As clearly shown in Figures 3(l) and 3(m), the order of accuracy in $\partial_y u$ and $\partial_x v$ has been deteriorated by one order, i.e., inconsistent with the first-order scheme and only first-order accuracy with the second-order scheme. However, as confirmed in Figure 3(g), the design order of accuracy is achieved in the viscous stress component, $\tau_{xy} = 3(\partial_y u + \partial_x v)/4$, which is the surprising result mentioned in Ref.[2]. Also demonstrated in Figures 3(l) and 3(m) is that the accuracy deterioration does not occur in the third-order scheme as expected from the strong coupling introduced by the fully hyperbolic formulation. Furthermore, we also show in Figure 3 the results for the third-order scheme with the quadratic flux extrapolation. The kappa scheme has been applied not only to the solution but also to the flux with $\kappa = 1/2$, so that the flux extrapolation becomes a quadratic extrapolation. Observe that third-order accuracy is completely lost in all variables despite the improved accuracy in the numerical fluxes. Not shown, but we have obtained similar results with other nonzero values of κ. These results confirm that the special property of the edge-based scheme on simplex elements, which guarantees third-order accuracy, requires the flux extrapolation not to be quadratic and it must be linear corresponding to $\kappa = 0$ as shown theoretically in Ref.[15]. It should be noted that improved accuracy in the fluxes, e.g., by a high-order flux reconstruction, does not necessarily lead to improved solution accuracy in the case of the node-centered edge-based finite-volume scheme [4]. In this $Re_\infty = 50$ case, although not shown, similar results have been obtained with Scheme I for HNS20 as well as with schemes for HNS17.

Next, we consider a high-Reynolds-number case, $Re_\infty = 10^8$. For high Reynolds numbers, Scheme II is known to achieve second- and third-order accuracy by first- and second-order schemes as demonstrated for advection-diffusion problems in Ref.[3]. The reason for the improved accuracy lies in the fact that the hyperbolic diffusion scheme has the ability to deliver accurate gradients even with a negligibly small diffusion coefficient and these gradients are used in the reconstruction for the advective term. To achieve the same for the Navier-Stokes equations, the gradients need to be computed for all the primitive variables relevant to the inviscid terms; and it is possible with HNS20 (not with HNS17). As can be seen in Figure 4, HNS20-II(1st) and HNS20-II(2nd) yield, indeed, second- and third-order accuracy, respectively for the primitive variables, ρ, u, v, T, and p. Note also that the velocity gradients are successfully computed to the design accuracy even with the negligibly small viscosity of $O(1/Re_\infty)$. Similar results are obtained for other gradients, and therefore not shown. These results indicate that HNS20-II(2nd) can provide third-order accuracy in the region where the viscous terms are negligible, serving as an alternative to a finite-volume scheme of third-order inviscid and second-order viscous schemes. A major advantage of HNS20-II(2nd) is that it is built entirely upon a second-order algorithm; for example, it does not require quadratic LSQ gradients to achieve third-order accuracy for the inviscid terms.

VIII.B. Cylinder

We consider a laminar flow over a cylinder of unit diameter with $M_\infty = 0.2$, $Pr = 3/4$, $\gamma = 1.4$, $Re_\infty = 40/(\text{unit grid length})$, and $T_\infty = 300$ [K]. The outer boundary is located at the distance of 100 times the diameter. Free stream boundary condition is applied at inflow, and the back-pressure condition is applied at outflow. On the cylinder surface, the no-slip and adiabatic conditions are imposed weakly through the ghost state. For this problem, a steady solution exists; a pair of wake vortices are created and they extend in the x direction by the distance approximately 2.2 times the diameter [19]. The grids are triangular and fully unstructured with 3200, 12800, and 51200 nodes. See Figure 5. Again, we employ the two-step implementation of a quadratic LSQ fit [3] for the third-order scheme.

We observe in Figure 6 that the implicit solver converges faster in CPU time for HNS20-II than for the conventional one for fine grids. As pointed out in the previous work, the difference in the slopes, 1.5 for HNS20-II and 2 for the conventional scheme, is more important and meaningful than the actual CPU time, which comes from the fact that the former involves only first-order derivatives while the latter involves second-order derivatives [20]. As the grid gets finer, the hyperbolic solver eventually gets faster than the conventional solver.

Figures 7-11 show pressure contours with streamlines and viscous stress contours. First, the results in the viscous stress contours clearly show the advantage of the hyperbolic schemes in predicting accurate gradients on irregular grids. For the conventional scheme, the viscous stresses are computed by linear LSQ gradients and it leads to noisy contours as can be seen in Figure 7(b). On the other hand, the viscous stress contours predicted by the hyperbolic schemes are much smoother as in Figures 8(b), 9(b), 10(b), and 11(b). Second, observe that the pressure contours are much improved by Scheme II especially for the second-order hyperbolic scheme. Compare Figures 8(a) and 9(a). Observe also that the recirculation region is more accurately predicted by Scheme II than by Scheme I: the wake extends much closer to the correct distance.

VIII.C. Flat Plate

As a preliminary study towards high-Reynolds-number applications, we consider a laminar flow over a flat plate at zero incidence with $M_\infty = 0.15$, $Pr = 3/4$, $\gamma = 1.4$, $Re_\infty = 10^4/(\text{unit grid length})$, and $T_\infty = 300$

[K]. The domain is taken to be a square and the right half of the bottom boundary is taken as a flat plate. Note that the length of the flat plate is 2.0 in the grid, and the Reynolds number based on the flat plate length is, therefore, 2×10^4. The grids used are all triangular grids with 816, 3264, 13289, 53156, and 212624 nodes. These grids are regular in the sense that they have been generated from non-uniform Cartesian grids of 34×24, 68×48, 137×97, 274×194, 548×388 nodes, and therefore all triangles are right triangles. See Figure 12(a) for the coarsest grid, and Figure 12(b) for Mach number contours obtained by the conventional scheme on that grid. For these regular grids, a quadratic LSQ fit is performed on a compact stencil with the edge-connected neighbors except at boundary nodes where at least one additional node must be added in the direction normal to the boundary to enable a quadratic fit. For this problem, we present results for Scheme I only.

Velocity profiles obtained by the conventional and hyperbolic schemes are compared in Figure 13. No significant differences can be seen in the tangential velocity component, but the comparison of the transverse velocity reveals that the hyperbolic schemes, especially the third-order one, produce more accurate profiles than the conventional scheme. Grid convergence of the drag coefficient is shown in Figure 14. The drag coefficient is defined for a finite flat plate of length 2.0, and a theoretical estimate based on the Blasius solution, Equation (7.40) in Ref.[21], is about 0.0096. In computing the drag coefficient, we employed the trapezoidal rule to integrate the viscous stress over the plate for the conventional scheme, and Simpson's rule for the hyperbolic schemes with the edge-midpoint value estimated by the Hermite interpolation [22]. The latter is possible because the gradients of the viscous stresses are available through the LSQ gradients of $\mathbf{g}$ in the hyperbolic schemes. The results demonstrate as shown in Figure 14 that the hyperbolic schemes provide significant improvements over the conventional scheme even on such regular grids. These results are expected because the second- and third-order hyperbolic schemes can produce second- and third-order accurate viscous stresses, respectively, while the conventional scheme relies on the LSQ gradients whose order of accuracy is typically one order lower on irregular stencils such as the ones at the boundary nodes in the grids used here. Finally, as shown in Figure 15 for the last three fine grids, the hyperbolic schemes still exhibit a slower increase in the CPU time with the number of nodes in comparison with the conventional scheme; the second-order hyperbolic scheme actually converged faster than the conventional scheme for the last two fine grids. As is well known from previous studies, the speed-up in CPU time is a result of the reduced number of iterations, which can be observed in Figure 16 that shows the residual history in the finest-grid case. These results imply that the hyperbolic schemes can achieve faster convergence even for high-Reynolds-number turbulent flows on sufficiently fine grids, for example, those used in large-eddy-simulations or direct numerical simulations. Note that the schemes are compared here on the same grid and therefore the hyperbolic schemes have a potential for both accelerating computations and delivering superior accuracy on a given grid. The potential is even greater if the comparison is performed for target accuracy; the hyperbolic schemes will be even more efficient because the accuracy requirement can be met on coarser grids than those required by the conventional scheme.

IX. Concluding Remarks

New hyperbolic formulations have been introduced for the compressible Navier-Stokes equations. One of the new formulations, HNS17, has been constructed with the velocity gradients and the heat fluxes as extra variables. We have shown that the equal order of accuracy for the velocity gradients is not guaranteed by the new system, but can be achieved either by a technique of artificial hyperbolic dissipation or by a special hyperbolic formulation of source terms. The other new formulation, HNS20, has been introduced to enable accurate gradient computations for all the primitive variables, (ρ, u, v, T). HNS20 has been constructed from HNS17 by adding an artificial hyperbolic diffusion system for the density. This formulation enable construction of a superior hyperbolic finite-volume scheme called Scheme II. In particular, Scheme II yields one-order-higher order of accuracy in the high-Reynolds-number limit (e.g., in inviscid regions): first- and second-order hyperbolic Navier-Stokes schemes will reduce to second- and third-order inviscid schemes when the inviscid terms dominate. Hence, the second-order HNS20-II scheme has a potential for serving as a practical alternative to a third-order inviscid and second-order viscous scheme for high-Reynolds-number turbulent flow simulations. For both HNS17 and HNS20, we have introduced a special hyperbolic formulation of source terms in order to achieve third-order accuracy without computation and storage of the second-derivatives of the source terms, which would be required otherwise. These advantages have been demonstrated through numerical experiments, including laminar flows over a cylinder and a flat plate. It has been shown also that the flux extrapolation required in the third-order scheme must be linear corresponding to the kappa scheme with $\kappa = 0$, and any other value, including $\kappa = 1/2$ that gives a quadratic reconstruction, will destroy third-order accuracy.

The new hyperbolic Navier-Stokes system, HNS20, is expected to serve as a main target system for other discretization methods as well. In the residual-distribution method, it will enable construction of a highly-efficient compact second-order hyperbolic scheme [23] for the compressible Navier-Stokes equations. Also, its third-order version [23] can be extended, without introducing additional LSQ gradients, to the compressible Navier-Stokes equations via HNS20. In the active flux method, it has been shown in Ref.[24] that a uniformly accurate third-order advection-diffusion scheme can be constructed by incorporating the advective term into a hyperbolic diffusion system as a source term. The strategy is made applicable to the compressible Navier-Stokes equations by HNS20 because all derivatives in the inviscid terms can be expressed in terms of the gradient variables.

Future work includes construction of second- and third-order unsteady hyperbolic Navier-Stokes schemes. For unsteady schemes, the physical time derivatives need to be treated as source terms. Hence, for third-order accuracy, it must be discretized either by an extended Galerkin formula [9] or a general divergence formulation of source terms [8]. In either case, the second-derivatives of the variables $(\rho, \rho\mathbf{v}, \rho E)$ will be required, which should be at least first-order accurate. A special feature of the HNS20 schemes is that these second-derivatives are already available from the LSQ gradients of the gradient variables, and therefore computation and storage of the second-derivatives are not required. This implies a huge computational saving since it would otherwise require computation of 30 second derivatives (6 for each) in three dimensions. A further study is necessary to determine the optimal value of the artificial hyperbolic diffusion coefficient, ν_ρ, which has to be defined such that it has no impact on the continuity equation. Although the value of 10^{-12} has been shown to be very effective for the problems considered in this paper, a more sophisticated definition may be necessary for other problems. Another area for further investigation is the weak boundary condition. Although numerical results presented in this work do not show any significant disadvantage of the weak condition, a comparative study between strong and weak boundary conditions would be desirable to determine the relative merits and also to verify the order of accuracy at boundary nodes.

Acknowledgments

This work has been funded by the U.S. Army Research Office under the contract/grant number W911NF-12-1-0154. Support by Software CRADLE is greatly acknowledged.

References

[1] 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.

[2] Nishikawa, H., "First-, Second-, and Third-Order Finite-Volume Schemes for Diffusion," *J. Comput. Phys.*, Vol. 256, 2014, pp. 791–805.

[3] Nishikawa, H., "First, Second, and Third Order Finite-Volume Schemes for Advection-Diffusion," *J. Comput. Phys.*, Vol. 273, 2014, pp. 287–309.

[4] Masatsuka, K., "I do like CFD, VOL.1, Second Edition," `http://www.cfdbooks.com`, 2013.

[5] 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.

[6] Mazaheri, A. and Nishikawa, H., "First-Order Hyperbolic System Method for Time-Dependent Advection-Diffusion Problems," *NASA-TM–2014–218175*, March 2014.

[7] Mazaheri, A. and Nishikawa, H., "Very efficient high-order hyperbolic schemes for time-dependent advection-diffusion problems: Third-, fourth-, and sixth-order," *Computers and Fluids*, Vol. 102, 2014, pp. 131–147.

[8] Nishikawa, H., "Divergence Formulation of Source Term," *J. Comput. Phys.*, Vol. 231, 2012, pp. 6393–6400.

[9] Pincock, B. and Katz, A., "High-Order Flux Correction for Viscous Flows on Arbitrary Unstructured Grids," *J. Sci.Comput.*, Vol. 61, 2014, pp. 454–476.

[10] Katz, A. and Sankaran, V., "Mesh quality effects on the accuracy of CFD solutions on unstructured meshes," *J. Comput. Phys.*, Vol. 230, 2011, pp. 7670–7686.

[11] van Leer, B., Lee, W.-T., and Roe, P. L., "Characteristic Time-Stepping or Local Preconditioning of the Euler Equations," *Proc. of 10th AIAA Computational Fluid Dynamics Conference*, AIAA Paper 91-1552, Hawaii, 1991.

[12] Roe, P. L., "Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes," *J. Comput. Phys.*, Vol. 43, 1981, pp. 357–372.

[13] van Leer, B., "Upwind-Difference Methods for Aerodynamic Problems Governed by the Euler Equations," *Fluid Mechanics, Lectures in Applied Mathematics*, Vol. 22, 1985, pp. 327–335.

[14] Burg, C. O. E., "Higher Order Variable Extrapolation For Unstructured Finite Volume RANS Flow Solvers," AIAA Paper 2005-4999, 2005.

[15] Nishikawa, H., "Accuracy-preserving boundary flux quadrature for finite-volume discretization on unstructured grids," *J. Comput. Phys.*, Vol. 281, 2015, pp. 518–555.

[16]Luo, H., Baum, J. D., and Löhner, R., "An Improved Finite Volume Scheme for Compressible Flows on Unstructured Grids," *Proc. of 33rd Aerospace Sciences Meeting and Exhibit*, AIAA Paper 95-0348, 1995.

[17]Nishikawa, H., "Beyond Interface Gradient: A General Principle for Constructing Diffusion Schemes," *Proc. of 40th AIAA Fluid Dynamics Conference and Exhibit*, AIAA Paper 2010-5093, Chicago, 2010.

[18]Nishikawa, H., Diskin, B., Thomas, J. L., and Hammond, D. H., "Recent Advances in Agglomerated Multigrid," *51st AIAA Aerospace Sciences Meeting*, AIAA Paper 2013-863, Texas, 2013.

[19]Coutanceau, M. and Bouard, R., "Experimental determination of the main features of the viscous flow in the wake of a circular cylinder in uniform translation. Part 1. Steady flow," *J. Fluid Mech.*, Vol. 79, 1977, pp. 231–256.

[20]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.

[21]Dyke, V., *Perturbation Methods in Fluid Mechanics*, The Parabolic Press, 1975.

[22]Nishikawa, H., "Higher-Order Discretization of Diffusion Terms in Residual-Distribution Methods," *VKI Lecture Series 2006-01: CFD - High Order Discretization Methods*, edited by H. Deconinck and M. Ricchiuto, von Karman Institute for Fluid Dynamics, Belgium, 2006.

[23]Mazaheri, A. and Nishikawa, H., "High-Order Hyperbolic Residual-Distribution Schemes on Arbitrary Triangular Grids," *Proc. of 22nd AIAA Computational Fluid Dynamics Conference*, AIAA Paper 2015-2445, Dallas, TX, 2015.

[24]Nishikawa, H., "Active Flux for Advection Diffusion," *Proc. of 22nd AIAA Computational Fluid Dynamics Conference*, AIAA Paper 2015-2450, Dallas, TX, 2015.

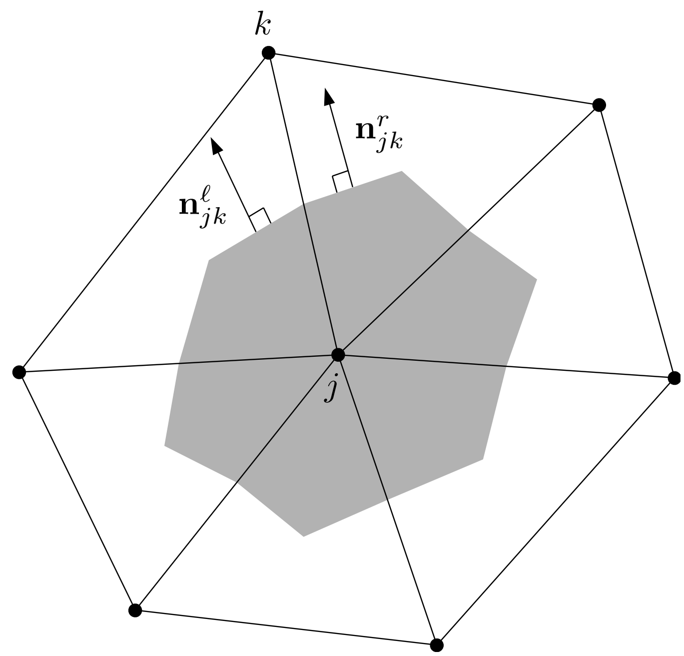
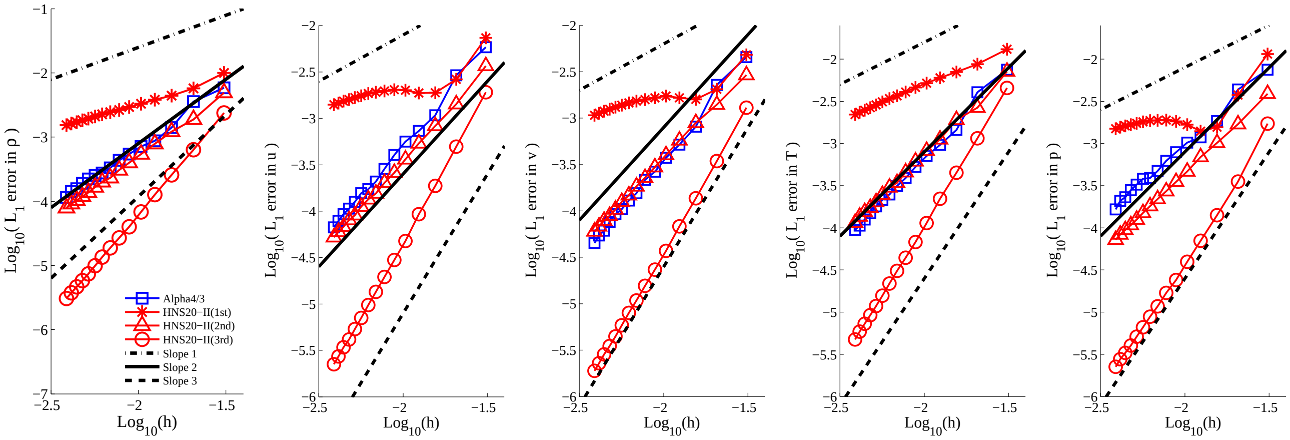
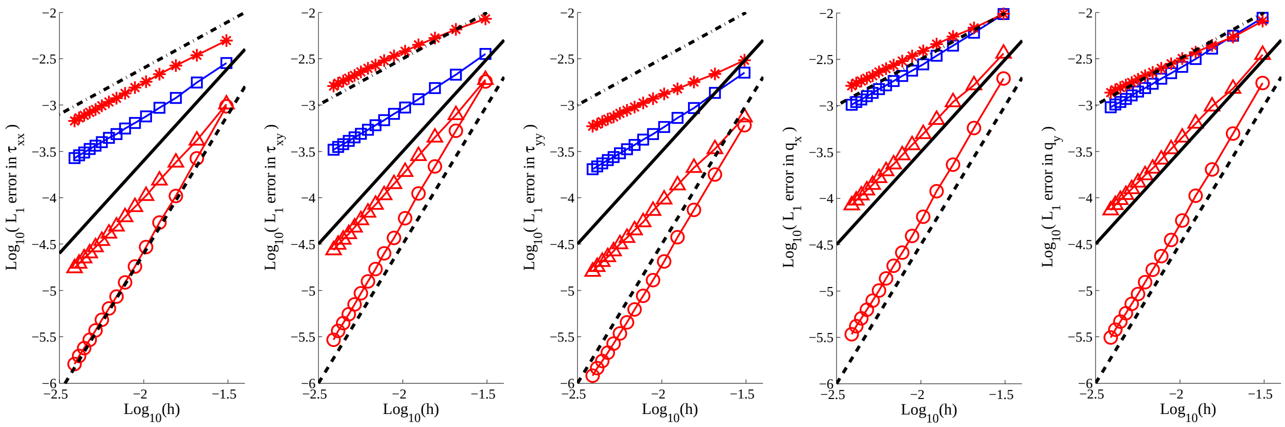


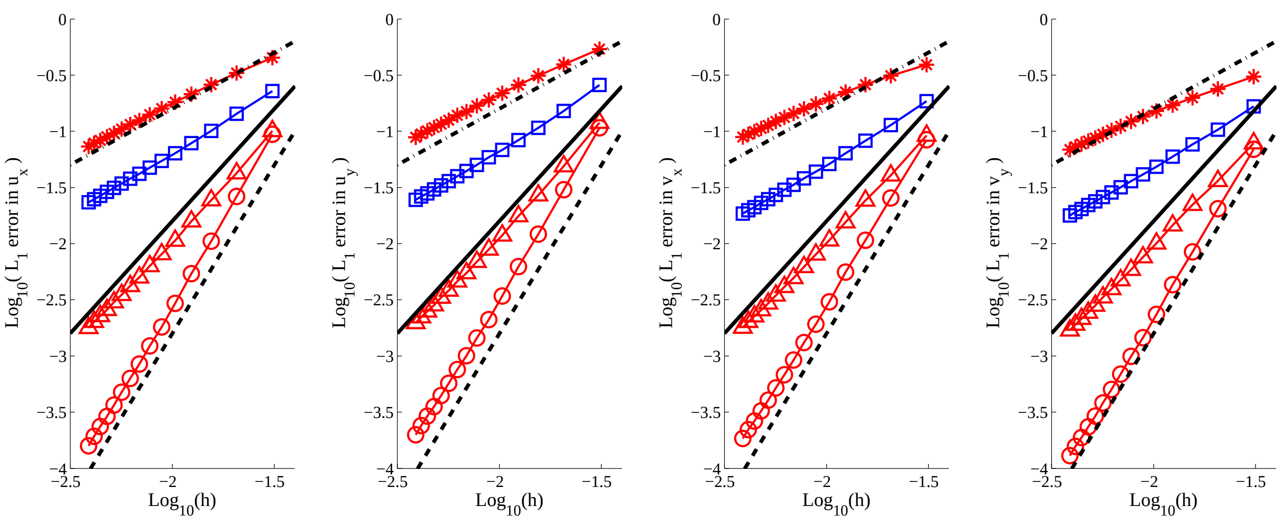
Figure 1: Dual control volume for the node-centered finite-volume method with scaled outward normals associated with an edge, $\{j, k\}$.



(a) Density, ρ. (b) Velocity, u. (c) Velocity, v. (d) Temperature, T. (e) Pressure, p.

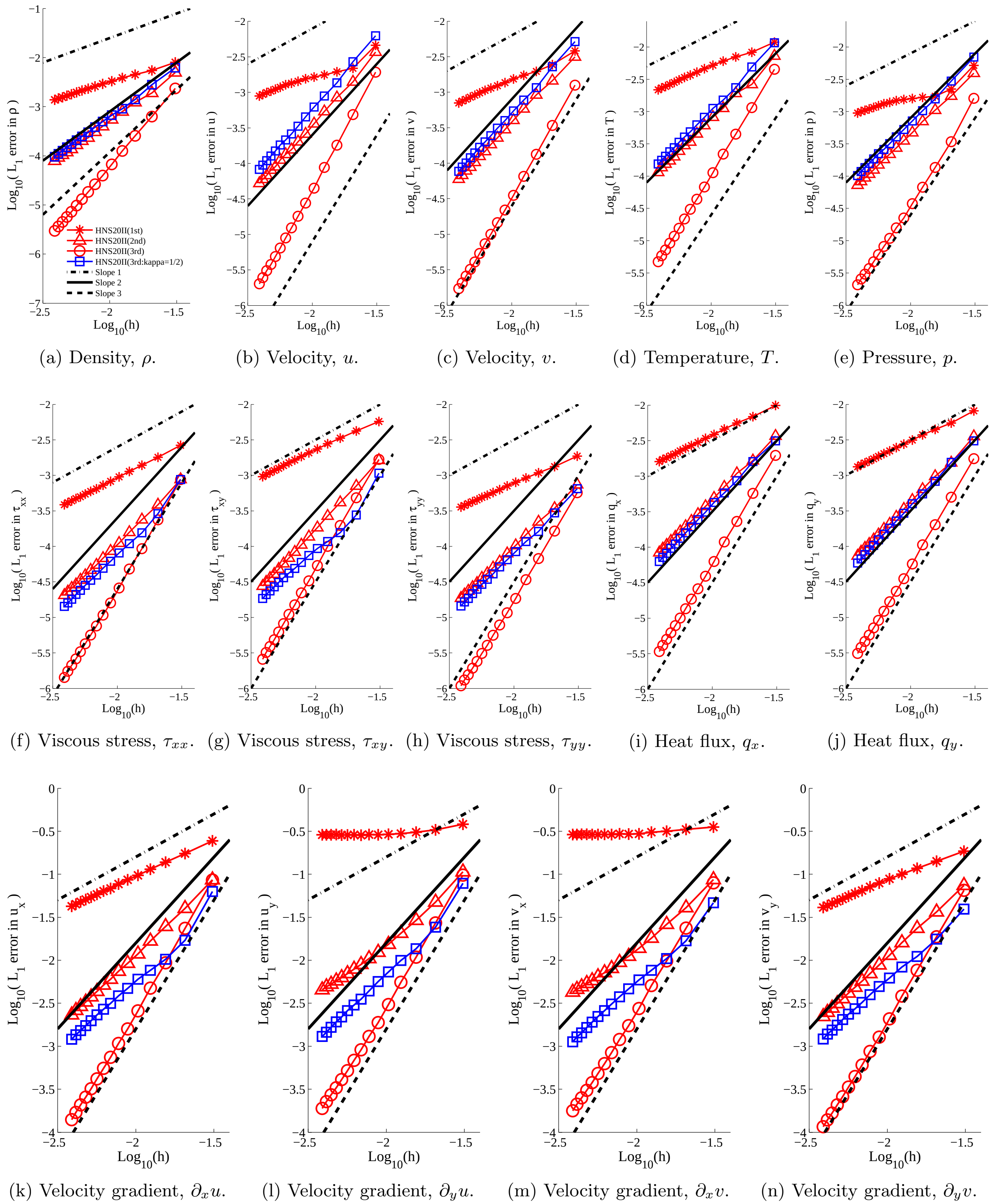


(f) Viscous stress, τ_{xx}. (g) Viscous stress, τ_{xy}. (h) Viscous stress, τ_{yy}. (i) Heat flux, q_x. (j) Heat flux, q_y.



(k) Velocity gradient, $\partial_x u$. (l) Velocity gradient, $\partial_y u$. (m) Velocity gradient, $\partial_x v$. (n) Velocity gradient, $\partial_y v$.

Figure 2: Error convergence results for $Re_\infty = 50$. Squares: conventional second-order scheme; stars: HNS20-II(1st); triangles: HNS20-II(2nd); circles: HNS20-II(3rd).

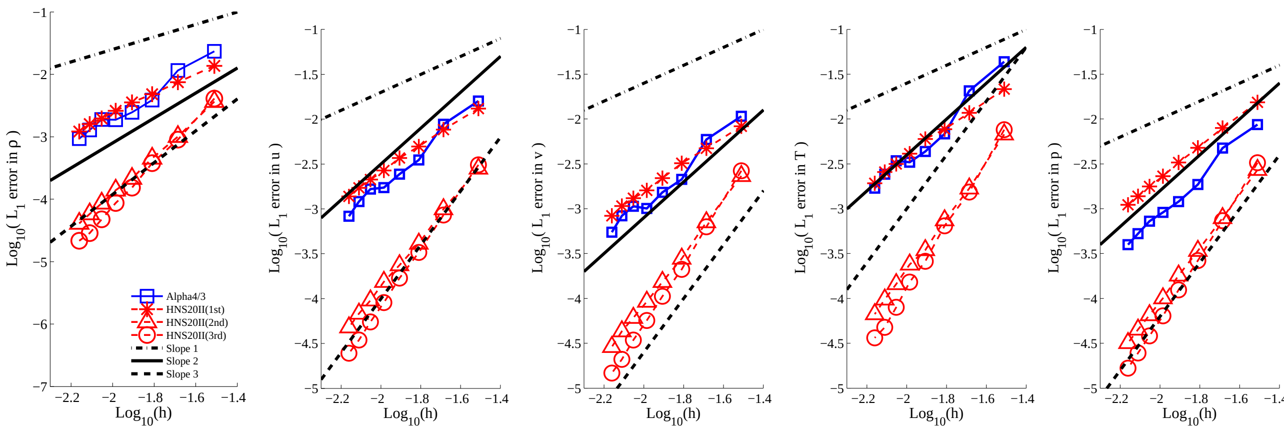


(a) Density, ρ. (b) Velocity, u. (c) Velocity, v. (d) Temperature, T. (e) Pressure, p.

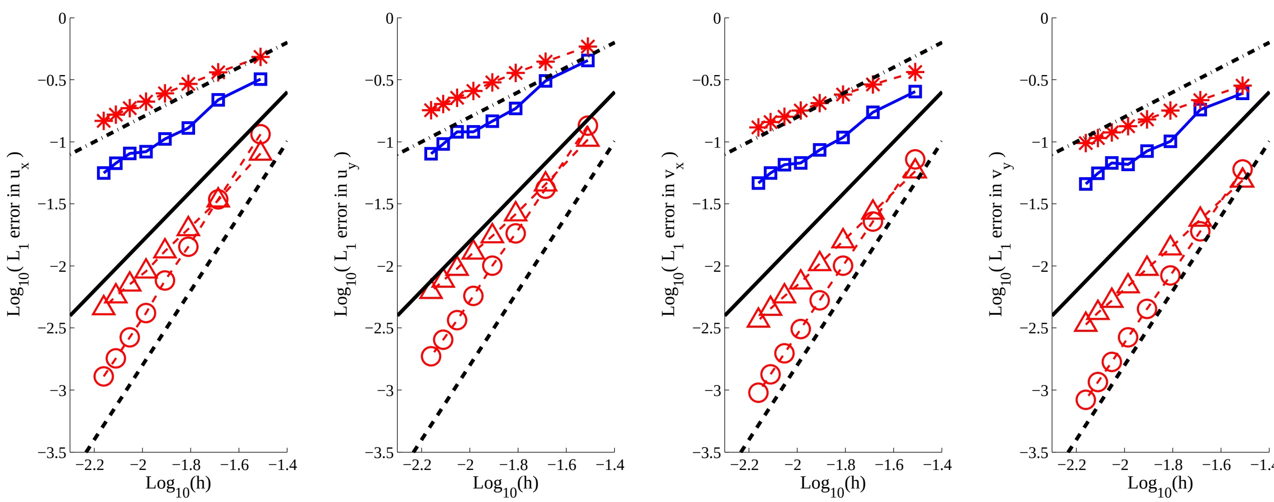
(f) Viscous stress, τ_{xx}. (g) Viscous stress, τ_{xy}. (h) Viscous stress, τ_{yy}. (i) Heat flux, q_x. (j) Heat flux, q_y.

(k) Velocity gradient, $\partial_x u$. (l) Velocity gradient, $\partial_y u$. (m) Velocity gradient, $\partial_x v$. (n) Velocity gradient, $\partial_y v$.

Figure 3: Additional error convergence results for $Re_\infty = 50$. Squares: HNS20-II(3rd) with the quadratic flux extrapolation given by the kappa scheme at $\kappa = 1/2$, showing that third-order accuracy has been completely lost. The rest of the results is for HNS20-II without the artificial hyperbolic dissipation. Stars: HNS20-II(1st); triangles: HNS20-II(2nd); circles: HNS20-II(3rd), showing that the design accuracy is lost for $\partial_y u$ and $\partial_x v$, but not for $\tau_{xy} = 3(\partial_y u + \partial_x v)/4$, without the artificial hyperbolic dissipation in the case of the first- and second-order schemes.



(a) Density, ρ. (b) Velocity, u. (c) Velocity, v. (d) Temperature, T. (e) Pressure, p.



(f) Velocity gradient, $\partial_x u$. (g) Velocity gradient, $\partial_y u$. (h) Velocity gradient, $\partial_x v$. (i) Velocity gradient, $\partial_y v$.

Figure 4: Error convergence results for $Re_\infty = 10^8$. Squares: conventional second-order scheme; stars: HNS20-II(1st); triangles: HNS20-II(2nd); circles: HNS20-II(3rd).

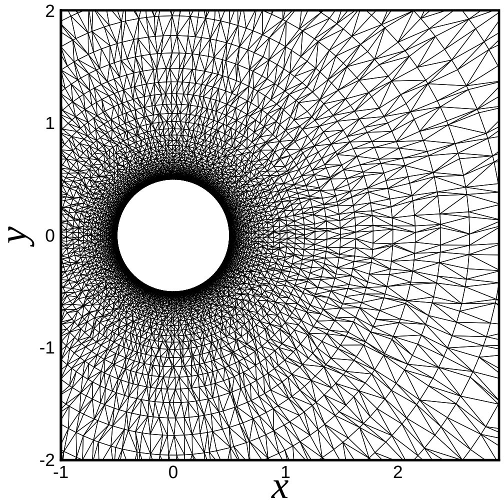


Figure 5: The coarsest grid for the cylinder case. The outer boundary is located at a distance 100 times the diameter.

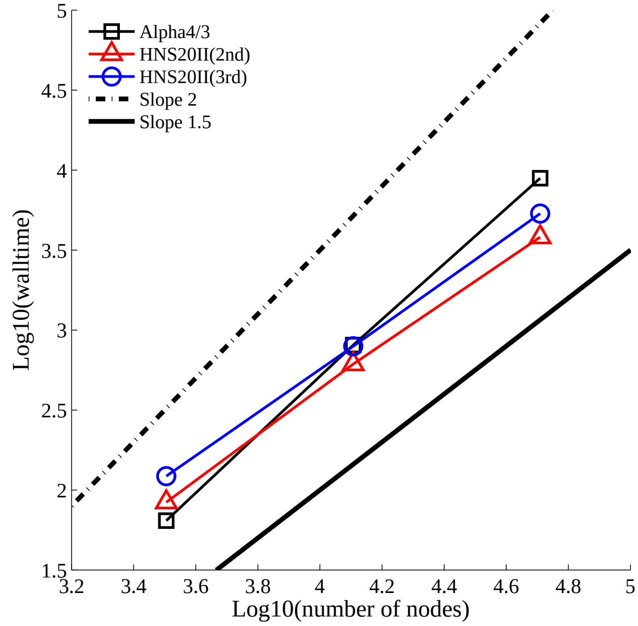
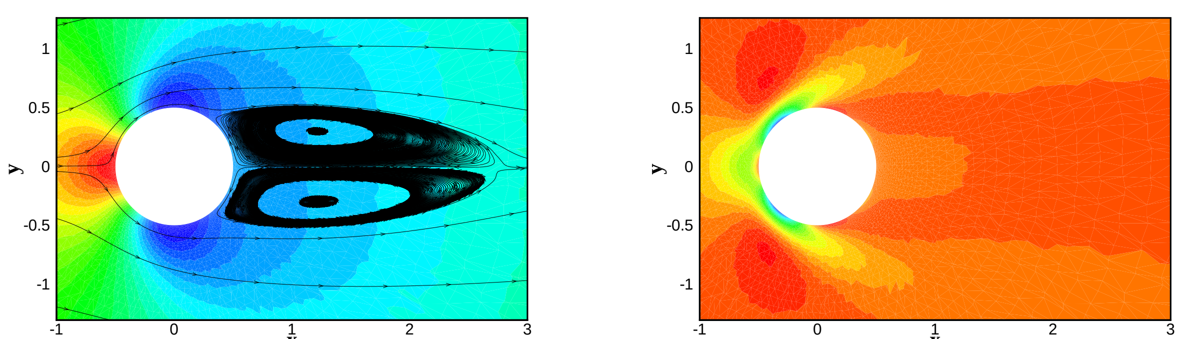


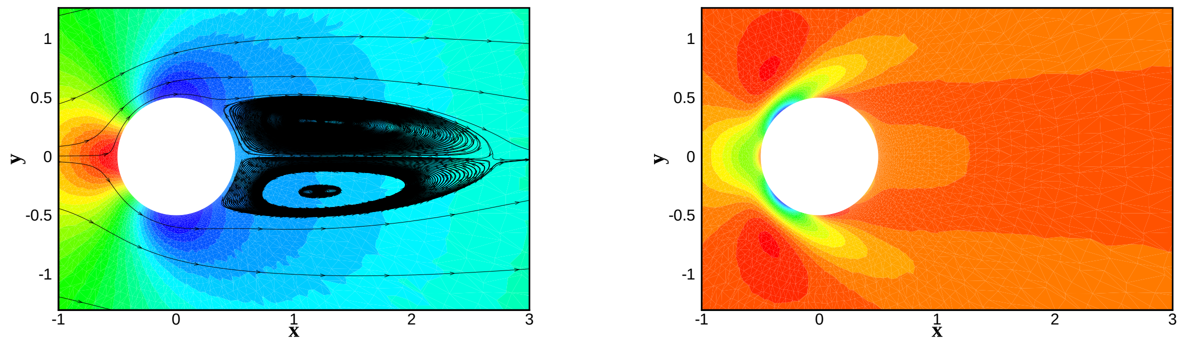
Figure 6: CPU time versus number of nodes.



(a) Pressure contours and streamlines.

(b) Viscous stress τ_{xx} contours.

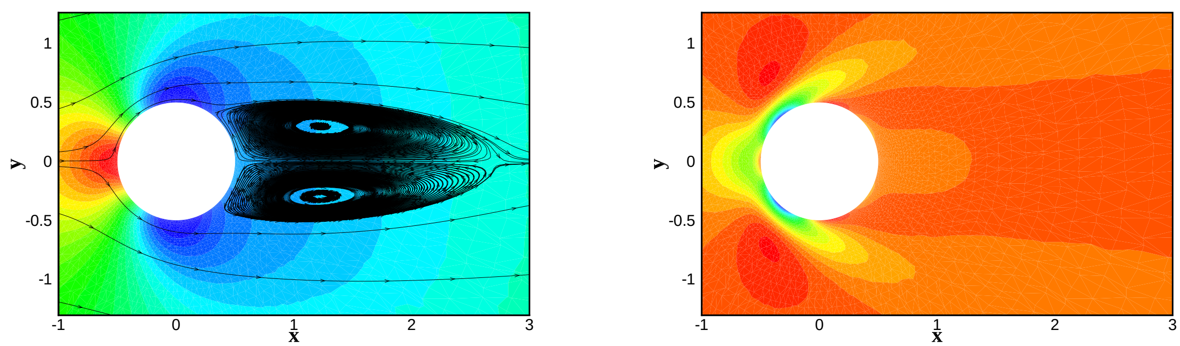
Figure 7: Results for Alpha4/3 (conventional scheme) on the grid with 12800 nodes.



(a) Pressure contours and streamlines.

(b) Viscous stress τ_{xx} contours.

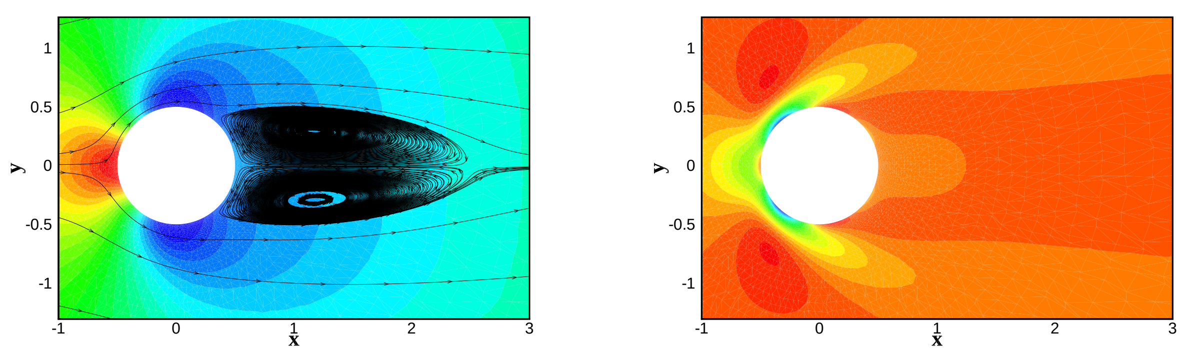
Figure 8: Results for HNS20(2nd) on the grid with 12800 nodes.



(a) Pressure contours and streamlines.

(b) Viscous stress τ_{xx} contours.

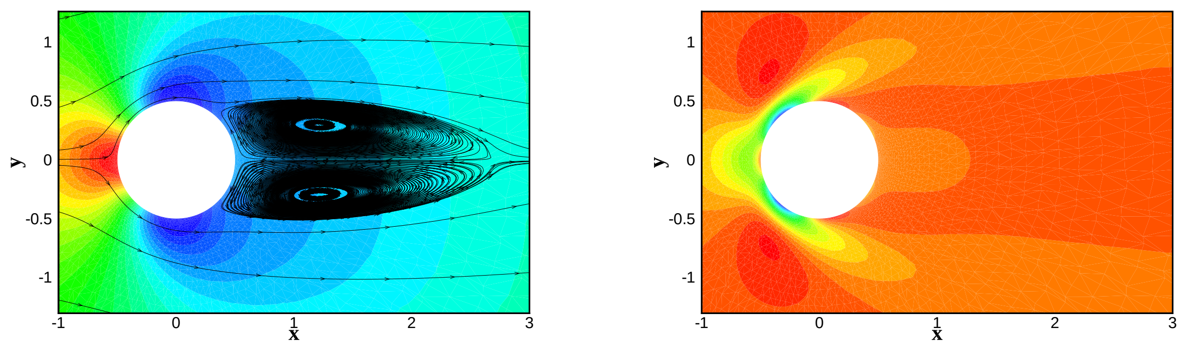
Figure 9: Results for HNS20-II(2nd) on the grid with 12800 nodes.



(a) Pressure contours and streamlines.

(b) Viscous stress τ_{xx} contours.

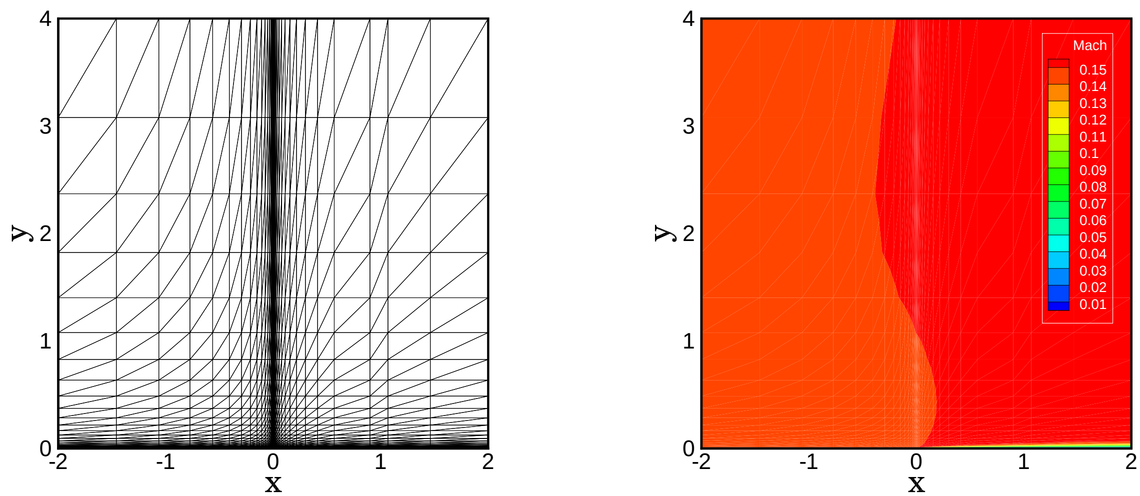
Figure 10: Results for HNS20(3rd) on the grid with 12800 nodes.



(a) Pressure contours and streamlines.

(b) Viscous stress τ_{xx} contours.

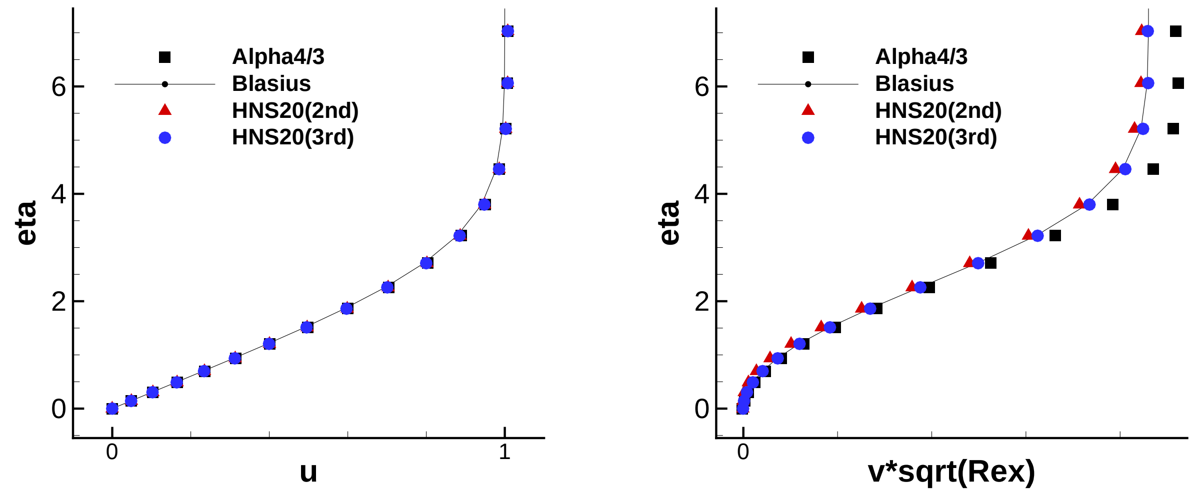
Figure 11: Results for HNS20-II(3rd) on the grid with 12800 nodes.



(a) The coarsest 34×24 grid with 816 nodes.

(b) Mach number contours.

Figure 12: Grid and solution for the flat-plate problem. A flat plate has a length 2.0, and it is located at the bottom of the domain starting at $x = 0$ and ending at $x = 2.0$. All grids have been generated from non-uniform Cartesian grids by inserting diagonals, and all triangles are right triangles. A vertical grid line has been introduced precisely at $x = 0.9$ to sample the solution for comparison.



(a) Tangential component of the velocity. (b) Transverse component of the velocity.

Figure 13: Velocity profiles on flat plate at $x = 0.9$ obtained on the 68×48 grid for $M_\infty = 0.15$ and $Re_\infty = 10^4$. The Blasius solution has been computed at the actual grid points, and the data are connected by straight lines. The vertical axis is taken as the boundary-layer coordinate, $\eta = y\sqrt{Re_x}/x$, where Re_x is the Reynolds number based on the distance along the plate from the leading edge, which has been used also to scale the transverse velocity.

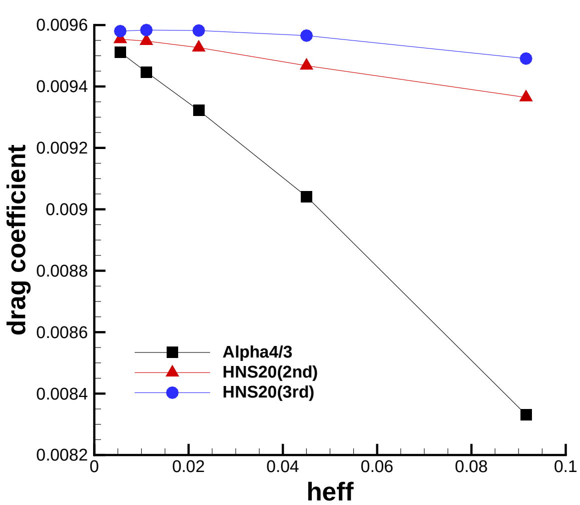


Figure 14: Grid convergence of the drag coefficient.

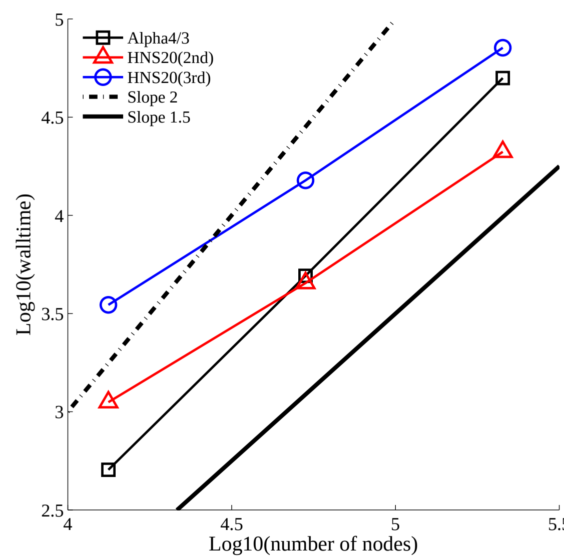
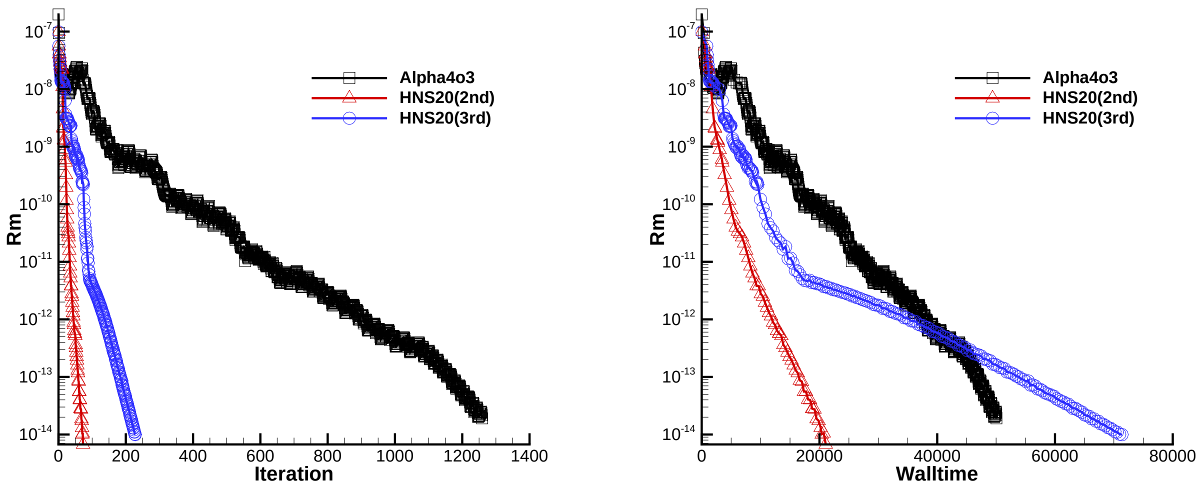


Figure 15: CPU time versus number of nodes.



(a) Residual norm versus iteration.

(b) Residual norm versus CPU time.

Figure 16: Residual convergence for the finest grid. Rm denotes the maximum L_1 residual norm.