

Finite-Volume Diffusion Schemes for Svard's Eulerian Governing Equations

Emmett Padway*

NASA Langley Research Center, Hampton, VA 23681, USA

This paper contains discussion on the implementation of new diffusion schemes for Svard's Eulerian flow (EF) governing equations. An alpha-damping type scheme and a new hyperbolic Eulerian flow (HEF) discretization is employed, paralleling previous work on hyperbolic Navier-Stokes (HNS). This work shows these equations have simplified hyperbolic discretizations due to the simplicity of the new stress tensor and compares these schemes to established Navier-Stokes (NS) discretizations to verify the predictive utilities of such diffusion schemes.

1. Introduction

The Navier-Stokes(-Fourier) equations have been widely used and treated as the governing equations for viscous fluid flow problems for centuries and are used widely as the backbone of Computational Fluid Dynamics (CFD) software and development. While these equations are an excellent model for the real physics and show good behaviors as approximations for engineering applications, they suffer by lacking certain mathematical properties, and it appears that due to the lack of diffusive flux in the continuity equation, they lack guarantees of well-posedness, existence, uniqueness, and continuity of data.¹

Efforts to address this began with the work of Brenner who did investigations into demonstrating incompatibility between the velocity terms that appear in the continuity and momentum equations.^{2,3} While his formulation of the new governing equations was later rebutted,⁴ it showed the importance of adding diffusive flux to the continuity equations and led to weak well-posedness and the development of a convergent scheme. Subsequently, Svard⁵ developed a model for viscous fluid flow in an effort to address the same issues Brenner identified. Svard's governing equations include a mass diffusion term (allowing for well-posedness) and greatly simplify the stress tensor that appears in the momentum equations. The addition of the mass diffusion is of particular interest to hyperbolic discretizations of the Navier-Stokes equations. These have gained prominence in recent years as Nishikawa developed them first for diffusion equations before extending them to Navier-Stokes⁶⁻¹² (NS) and implementing them in the massively parallel, unstructured-grid computational-fluid-dynamics code from NASA Langley¹³ for unsteady three-dimensional and two-dimensional spacetime discretizations.¹⁴ The Hyperbolic Navier-Stokes (HNS) discretization requires the addition of an artificial mass diffusion term that is defined to be arbitrarily small to not impact the final flow field but large enough to induce coupling thereby allowing the density gradients to achieve design order accuracy. These artificial terms are no longer necessary in the Hyperbolic Eulerian Fluid (HEF) discretization, a new discretization introduced in this work. Furthermore, the simplified stress tensor allows for a simpler viscous diffusion matrix and the selection of a single length scale as opposed to the three length scales required for the HNS20G discretizations.¹² In this work the implementation of the Eulerian fluid governing equations in the spacetime discretization of FUN3D is shown and validated.

1.1. Navier-Stokes Governing Equations

Our target equations are the unsteady compressible NS equations:

$$\begin{aligned}\partial_t \rho + \nabla_{\mathbf{x}} \cdot (\rho \mathbf{v}) &= 0, \\ \partial_t (\rho \mathbf{v}) + \nabla_{\mathbf{x}} \cdot (\rho \mathbf{v} \otimes \mathbf{v}) &= -\nabla_{\mathbf{x}} p + \nabla_{\mathbf{x}} \cdot \boldsymbol{\tau}, \\ \partial_t (E) + \nabla_{\mathbf{x}} \cdot (E \mathbf{v} + p \mathbf{v}) &= \nabla_{\mathbf{x}} \cdot (\boldsymbol{\tau} \mathbf{v}) - \nabla_{\mathbf{x}} \cdot \mathbf{q},\end{aligned}\tag{1.1}$$

*Research Aerospace Engineer, Aeroelasticity Branch, emmett.m.padway@nasa.gov, Senior Member AIAA.

where t is the physical time, $\otimes$ denotes the dyadic product, ρ is the density, $\mathbf{v}$ is the velocity vector, p is the pressure, E is the total energy, $\nabla_{\mathbf{x}}$ is the spatial derivative and $\nabla_{\mathbf{x}} \cdot$ is the spatial divergence. The viscous stress tensor τ , under Stokes' hypothesis is

$$\tau = -\frac{2}{3}\mu(\nabla_{\mathbf{x}} \cdot \mathbf{v})\mathbf{I} + \mu(\nabla_{\mathbf{x}}\mathbf{v} + (\nabla_{\mathbf{x}}\mathbf{v})^T), \quad (1.2)$$

where $\mathbf{I}$ is the identity matrix, μ is the viscosity defined by Sutherland's law, and the superscript T denotes the transpose. All the quantities are assumed to have been nondimensionalized by their freestream values except that the velocity and the pressure are scaled by the freestream speed of sound and the freestream dynamic pressure, respectively.¹⁵ Thus, the viscosity is given by the following form of Sutherland's law:

$$\mu = \frac{M_{\infty}}{Re_{\infty}} \frac{1 + C/\tilde{T}_{\infty}}{T + C/\tilde{T}_{\infty}} T^{\frac{3}{2}}, \quad (1.3)$$

where T is the temperature, $\tilde{T}_{\infty}$ is the dimensional freestream temperature, and $C = 110.5$ [K] is the Sutherland constant. The ratio of the freestream Mach number, M_{∞} , to the freestream Reynolds number per grid unit, Re_{∞} , arises from the non-dimensionalization. The heat flux ($\mathbf{q}$) is similarly changed by the nondimensionalization, and is defined as:

$$\mathbf{q} = -\frac{\mu}{Pr(\gamma - 1)} \nabla_{\mathbf{x}} T, \quad (1.4)$$

where γ is the ratio of specific heats, Pr is the Prandtl number, and the system is closed by the nondimensionalized equation of state for ideal gases:

$$\gamma p = \rho T. \quad (1.5)$$

An earlier paper on spacetime discretizations of Navier-Stokes equations¹⁶ used the alpha-damping discretization to compute the viscous fluxes:

$$\Phi_{jk}^{vis} = \mathbf{f}^{vis}(\mathbf{w}_L, \mathbf{w}_R, \nabla \mathbf{w}), \quad \mathbf{f}^{vis} = [\mathbf{f}_x^{vis}, \mathbf{f}_y^{vis}] \cdot \hat{\mathbf{n}}_{xy} = \begin{bmatrix} 0 \\ -\tau_n \\ -\tau_n \bar{\mathbf{v}} + q_n \end{bmatrix}, \quad (1.6)$$

where the heat flux and viscous stresses are given in 2D at the face by

$$q_n = -\frac{\bar{\mu}}{Pr(\gamma - 1)} \nabla_{\mathbf{x}} T \cdot \hat{\mathbf{n}}_{xy} = -\frac{\gamma \bar{\mu}}{\bar{\rho} Pr(\gamma - 1)} \left[\nabla_{\alpha} \rho - \frac{\bar{p}}{\bar{\rho}} \nabla_{\alpha} p \right] \cdot \hat{\mathbf{n}}_{xy}, \quad (1.7)$$

$$\tau_n = -\bar{\mu} \left[\frac{2}{3} tr(\nabla_{\alpha} \mathbf{v}) \mathbf{I} + \nabla_{\alpha} \mathbf{v} + (\nabla_{\alpha} \mathbf{v})^t \right] \cdot \hat{\mathbf{n}}_{xy}. \quad (1.8)$$

The gradients (∇_{α}) are computed by the alpha-damping scheme,^{17,18}

$$\nabla_{\alpha} \mathbf{w} = \frac{1}{2} \left[\nabla_{\mathbf{x}} \mathbf{w}_j^{LSQ} + \nabla_{\mathbf{x}} \mathbf{w}_k^{LSQ} \right] + \frac{\alpha}{\beta} (\mathbf{w}_R - \mathbf{w}_L) \hat{\mathbf{n}}_{xy}, \quad \beta = |\mathbf{e}_{xy} \cdot \hat{\mathbf{n}}_{xy}|, \quad \mathbf{e}_{xy} = (x_k - x_j, y_k - y_j), \quad (1.9)$$

where α is a damping coefficient and it is set $\alpha = 4/3$.¹⁷⁻¹⁹

1.1.1. Hyperbolic Navier-Stokes System: HNS20G

Previous work derived hyperbolic formulations for the viscous terms: HNS14,⁷ HNS17,⁸ HNS20,⁸ and HNS20G.¹² Here, HNS20G is employed, which was first introduced to simplify the development of high-order reconstructed discontinuous Galerkin methods.¹² In HNS20G, the Navier-Stokes system is extended by adding extra variables $\mathbf{r}$, $\mathbf{g}$, $\mathbf{h}$ corresponding to the gradients of the primitive variables, which are called

the gradient variables:

$$\begin{aligned}
\partial_\tau \rho + \partial_t \rho + \nabla_{\mathbf{x}} \cdot (\rho \mathbf{v}) &= \nabla_{\mathbf{x}} \cdot (\nu_r \mathbf{r}), \\
\partial_\tau (\rho \mathbf{v}) + \partial_t (\rho \mathbf{v}) + \text{div}(\rho \mathbf{v} \otimes \mathbf{v}) &= -\nabla_{\mathbf{x}} p + \nabla_{\mathbf{x}} \cdot (\mu_v \tilde{\tau}), \\
\partial_\tau (\rho E) + \partial_t (\rho E) + \nabla_{\mathbf{x}} \cdot (\rho \mathbf{v} H) &= \nabla_{\mathbf{x}} \cdot (\mu_v \tilde{\tau} \mathbf{v}) + \nabla_{\mathbf{x}} \cdot \left(\frac{\mu_h}{\gamma(\gamma-1)} \mathbf{h} \right), \\
T_v \partial_\tau \mathbf{g} &= \nabla_{\mathbf{x}} \mathbf{v} - \mathbf{g}, \\
T_h \partial_\tau \mathbf{h} &= \nabla_{\mathbf{x}} T - \mathbf{h}, \\
T_r \partial_\tau \mathbf{r} &= \nabla_{\mathbf{x}} \rho - \mathbf{r}.
\end{aligned} \tag{1.10}$$

Note that τ is a pseudotime variable, and $\tilde{\tau} = -\frac{1}{2} \text{tr}(\mathbf{g}) \mathbf{I} + \frac{3}{4} (\mathbf{g} + \mathbf{g}^t)$ is the stress tensor computed using the HNS variables without the viscosity coefficient,

$$T_r = \frac{L_r^2}{\nu_r}, \quad T_v = \frac{L_v^2}{\nu_v}, \quad T_h = \frac{L_h^2}{\nu_h}, \quad \nu_v = \frac{4\mu}{3\rho}, \quad \nu_h = \frac{\gamma\mu}{\rho Pr}, \quad \nu_r = \min(10^{-12}, V_{min}^{4/3}) M_\infty, \tag{1.11}$$

where V_{min} is the minimum volume element in the mesh and the length scale may be taken as $L_r = L_v = L_h = 1/(2\pi)$, or as in other works it is modified with Reynolds number:^{11,12}

$$\begin{aligned}
L_r &= L_d = \frac{1}{2\pi}, \\
L_v &= \frac{L_d}{\sqrt{Re_{L_d}^{v\infty}}}, \quad Re_{L_d}^{v\infty} = \frac{U_\infty L_d}{\nu_{v\infty}}, \\
L_h &= \frac{L_d}{\sqrt{Re_{L_d}^{h\infty}}}, \quad Re_{L_d}^{h\infty} = \frac{U_\infty L_d}{\nu_{h\infty}},
\end{aligned} \tag{1.12}$$

where U_∞ is the freestream flow speed, and the subscript ∞ refers to the freestream state. The latter is more suitable for high Reynolds number flows. Note that L_r is different from that in previous works.^{11,12} As shown earlier,^{9,20} it has been observed that the order of accuracy of the density gradient is reduced by one order if L_r is too small, and therefore, it is sometimes necessary to keep $L_r = L_d = \frac{1}{2\pi}$. This discretization was implemented in the spacetime discretization in FUN3D and provides a good comparison for the HEF system developed in this work as they use the same principles. Note that the HNS20G system is equivalent to the original Navier-Stokes system in a pseudosteady state or when the pseudotime derivatives are dropped. Then, the gradient variables $\mathbf{h}$ and $\mathbf{g}$ correspond to the temperature and velocity gradients:

$$\begin{aligned}
\mathbf{h} &= \nabla_{\mathbf{x}} T, \\
\mathbf{g} &= \nabla_{\mathbf{x}} \mathbf{v}.
\end{aligned} \tag{1.13}$$

The viscous part of HNS20G is hyperbolic in pseudotime and therefore, can be discretized by an upwind method similar to the inviscid part. Once the viscous part is discretized, the pseudotime terms are dropped to arrive at a consistent discretization of the original Navier-Stokes system in a first-order form.

In two dimensions, the HNS20G system is written as

$$\mathbf{P}^{-1} \frac{\partial \mathbf{u}}{\partial \tau} + \mathbf{M} \frac{\partial \mathbf{u}}{\partial t} + \frac{\partial \mathbf{f}_x}{\partial x} + \frac{\partial \mathbf{f}_y}{\partial y} = \mathbf{s}, \tag{1.14}$$

where

$$\mathbf{P}^{-1} = \text{diag}(1, 1, 1, 1, T_r, T_r, T_v, T_v, T_v, T_v, T_h, T_h), \tag{1.15}$$

$$\mathbf{M} = \text{diag}(1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0), \tag{1.16}$$

$$\mathbf{u} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho E \\ g_{ux} \\ g_{uy} \\ g_{vx} \\ g_{vy} \\ h_x \\ h_y \\ r_x \\ r_y \end{bmatrix}, \quad \mathbf{f}_x = \mathbf{f}_x^{inv} + \mathbf{f}_x^{vis} = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uH \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ -\mu_v \tilde{\tau}_{xx} \\ -\mu_v \tilde{\tau}_{xy} \\ -\mu_v \tilde{\tau}_{xx}u - \mu_v \tilde{\tau}_{xy}v + \frac{\mu_h}{\gamma(\gamma-1)}h_x \\ -u \\ 0 \\ -v \\ 0 \\ -T \\ 0 \\ -\rho \\ 0 \end{bmatrix}, \quad (1.17)$$

$$\mathbf{f}_y = \mathbf{f}_y^{inv} + \mathbf{f}_y^{vis} = \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho vH \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ -\mu_v \tilde{\tau}_{yx} \\ -\mu_v \tilde{\tau}_{yy} \\ -\mu_v \tilde{\tau}_{yx}u - \mu_v \tilde{\tau}_{yy}v + \frac{\mu_h}{\gamma(\gamma-1)}h_y \\ 0 \\ -u \\ 0 \\ -v \\ 0 \\ -T \\ 0 \\ -\rho \end{bmatrix}, \quad \mathbf{s} = \begin{bmatrix} s_1(x, y, t) \\ s_2(x, y, t) \\ s_3(x, y, t) \\ s_4(x, y, t) \\ -g_{ux} \\ -g_{uy} \\ -g_{vx} \\ -g_{vy} \\ -h_x \\ -h_y \\ -r_x \\ -r_y \end{bmatrix}, \quad (1.18)$$

where s_1, s_2, s_3, s_4 are forcing terms, typically set to 0 except in the case of verification via the method of

manufactured solutions (MMS). The dissipation vector of an upwind flux for the viscous part is given by:

$$\left| \mathbf{P} \frac{\partial \mathbf{f}_n^{vis}}{\partial \mathbf{u}} \right| \Delta \mathbf{u} = \begin{pmatrix} 0 \\ \rho[(a_{nv} - a_{mv})\Delta u_n \mathbf{n} + a_{mv}\Delta \mathbf{v}] \\ \rho \mathbf{v} \cdot \{(a_{nv} - a_{mv})\Delta u_n \mathbf{n} + a_{mv}\Delta \mathbf{v}\} + \frac{\rho a_h \Delta T}{\gamma(\gamma-1)} + \frac{\mu_v^2}{\rho a_h} (\mathcal{L}_\tau \cdot \Delta \tilde{\boldsymbol{\tau}}_n) \\ \left[\left(a_{nv} - \frac{4}{3} a_{mv} \right) \Delta \tilde{\tau}_{nn} n_x + \frac{4}{3} a_{mv} \Delta \tilde{\tau}_{nx} \right] \mathbf{n} \\ \left[\left(a_{nv} - \frac{4}{3} a_{mv} \right) \Delta \tilde{\tau}_{nn} n_y + \frac{4}{3} a_{mv} \Delta \tilde{\tau}_{ny} \right] \mathbf{n} \\ a_h [\Delta h_n + (\mathcal{L}_\tau \cdot \Delta \mathbf{v})] \mathbf{n} \\ 0 \end{pmatrix}, \quad (1.19)$$

where $\mathbf{f}_n^{vis} = (\mathbf{f}_x^{vis}, \mathbf{f}_y^{vis}) \cdot \mathbf{n}$, Δ denotes a jump across two states, e.g., $\Delta \mathbf{v} = \mathbf{v}_R - \mathbf{v}_L$, $\mathbf{v} = (u, v)^t$, $\mathbf{n} = (n_x, n_y)^t$ is a unit vector, and

$$u_n = \mathbf{v} \cdot \mathbf{n}, \quad h_n = \mathbf{h} \cdot \mathbf{n}, \quad \tilde{\boldsymbol{\tau}}_n = \tilde{\boldsymbol{\tau}} \mathbf{n}, \quad \mathcal{L}_\tau = \frac{\tilde{\tau}_{nn} \mathbf{n}}{Pr_n + 1} + \frac{\tilde{\boldsymbol{\tau}}_n - \tilde{\tau}_{nn} \mathbf{n}}{Pr_m + 1}, \quad (1.20)$$

$$\tilde{\tau}_{nm} = \tilde{\tau}_{nx} m_x + \tilde{\tau}_{ny} m_y, \quad \tilde{\tau}_{nn} = \tilde{\tau}_{nx} n_x + \tilde{\tau}_{ny} n_y, \quad (1.21)$$

$$\tilde{\tau}_{nx} = \tilde{\tau}_{xx} n_x + \tilde{\tau}_{xy} n_y, \quad \tilde{\tau}_{ny} = \tilde{\tau}_{xy} n_x + \tilde{\tau}_{yy} n_y, \quad (1.22)$$

$$Pr_n = \frac{a_{nv}}{a_h}, \quad Pr_m = \frac{a_{mv}}{a_h}, \quad Pr_l = \frac{a_{lv}}{a_h} = \frac{a_{mv}}{a_h} = Pr_m, \quad (1.23)$$

$$a_{nv} = \sqrt{\frac{\nu_v}{T_v}}, \quad a_{mv} = \sqrt{\frac{3\nu_v}{4T_v}}, \quad a_h = \sqrt{\frac{\nu_h}{T_h}}. \quad (1.24)$$

All quantities in the dissipation matrix $\left| \mathbf{P} \frac{\partial \mathbf{f}_n^{vis}}{\partial \mathbf{u}} \right|$ are evaluated with arithmetic averages of the two states, e.g., $\mathbf{v} = (\mathbf{v}_L + \mathbf{v}_R)/2$.

1.2. Eulerian Viscous Fluids Governing Equations

Consider the Eulerian fluid model using Svard's derivation,¹

$$\begin{aligned} \frac{\partial \rho}{\partial t} + \nabla_{\mathbf{x}} \cdot (\rho \mathbf{v}) &= \nabla_{\mathbf{x}} \cdot (\nu \nabla_{\mathbf{x}} \rho), \\ \frac{\partial \rho \mathbf{v}}{\partial t} + \nabla_{\mathbf{x}} \cdot (\rho \mathbf{v} \otimes \mathbf{v}) + \nabla_{\mathbf{x}} p &= \nabla_{\mathbf{x}} \cdot (\nu \nabla_{\mathbf{x}} \rho \mathbf{v}), \\ \frac{\partial E}{\partial t} + \nabla_{\mathbf{x}} \cdot (E \mathbf{v} + p \mathbf{v}) &= \nabla_{\mathbf{x}} \cdot (\nu \nabla_{\mathbf{x}} E), \\ p &= \rho R T, \\ \nu &= \alpha \frac{\mu}{\rho} + \beta(\rho, T). \end{aligned} \quad (1.25)$$

Note that these differ from the Navier-Stokes equations due to the presence of a mass diffusion term in the continuity equation and the changes to the diffusion terms in the momentum and energy equations. Rather than the more complex Navier-Stokes equations' stress tensor, the Eulerian fluid model defines the

diffusive terms to be momentum and energy diffusion for the momentum and energy equations, respectively. To maintain simplicity and correspondence to the NS equations these viscosity coefficients are taken as $(\alpha, \beta) = (1, 0)$. It is possible to decompose this discretized governing equation into inviscid and viscous fluxes, and the viscous fluxes on edge jk with normal n_{jk} and magnitude A_{jk} become in 2D:

$$F_v = \begin{bmatrix} \nu \nabla_{\mathbf{x}} \rho \\ \nu \nabla_{\mathbf{x}} \rho \mathbf{v} \\ \nu \nabla_{\mathbf{x}} E \end{bmatrix} \cdot \vec{n}_{\mathbf{x}} = \begin{bmatrix} \nu(\rho_x n_x + \rho_y n_y) \\ \nu((\rho u)_x n_x + (\rho u)_y n_y) \\ \nu((\rho v)_x n_x + (\rho v)_y n_y) \\ \nu(E_x n_x + E_y n_y) \end{bmatrix} A_{jk} \quad (1.26)$$

and in such a case, it is possible to use the alpha-damping scheme. By using equation 1.9, one can compute the gradients of density, momentum, and energy at the interface and evaluate the Eulerian fluid stress tensor.

1.2.1. Hyperbolic Eulerian Flow System: HEF20

Alternatively, one can follow the HNS20G system to derive a hyperbolic system (HEF20) with some key differences. First, since there is mass diffusion in the underlying governing equations, there is no need to add an artificial mass diffusion to the Eulerian fluid flow model. Second, it is useful to define our auxiliary variables as the gradients of density, velocity, and energy rather than temperature for ease of implementation and of derivation of the viscous flux Jacobian; this does mean that temperature gradient calculations will be more cumbersome when required. Third, since there is a more simplified stress tensor, it is shown that a simpler dissipation matrix for the diffusion system is obtained, with only one viscous wave speed that will allow for only one relaxation time as opposed to the three (T_v, T_h, T_r) in HNS20G. Finally, since the stress tensor is simplified, there is no need for the additional viscous dissipation that HNS20G adds to maintain design order accuracy in the velocity gradients (as without this dissipation in HNS20G the stress terms will be the quantities that maintain design order accuracy). The derivation begins from the conservation equation:

$$\frac{\partial}{\partial t} \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ E \\ T_r \rho_x \\ T_r \rho_y \\ T_v u_x \\ T_v u_y \\ T_v v_x \\ T_v v_y \\ T_h E_x \\ T_h E_y \end{bmatrix} + \nabla_{\mathbf{x}} \cdot \begin{bmatrix} \rho \mathbf{v} \\ \rho u \mathbf{v} + p \\ \rho v \mathbf{v} + p \\ (E + p) \mathbf{v} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \nabla_{\mathbf{x}} \cdot \begin{bmatrix} \nu \nabla_{\mathbf{x}} \rho \\ \nu(u \nabla_{\mathbf{x}} \rho + \rho \nabla_{\mathbf{x}} u) \\ \nu(v \nabla_{\mathbf{x}} \rho + \rho \nabla_{\mathbf{x}} v) \\ \nu \nabla_{\mathbf{x}} E \\ \rho \\ \rho \\ u \\ u \\ v \\ v \\ E \\ E \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ r_x \\ r_y \\ u_x \\ u_y \\ v_x \\ v_y \\ E_x \\ E_y \end{bmatrix}, \quad (1.27)$$

where later it will be shown that the different values of T_r , T_v , and T_h all take the same value as mentioned above. Where the unified flux at a face is given by:

$$F \cdot n = \Phi_{jk} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ E \\ T_r \rho_x \\ T_r \rho_y \\ T_v u_x \\ T_v u_y \\ T_v v_x \\ T_v v_y \\ T_h E_x \\ T_h E_y \end{bmatrix} n_t + \begin{bmatrix} \rho(un_x + vn_y) - \nu(r_x n_x + r_y n_y) \\ \rho u(un_x + vn_y) + pn_x - \nu[\rho(u_x n_x + u_y n_y) + (r_x n_x + r_y n_y)u] \\ \rho v(un_x + vn_y) + pn_y - \nu[\rho(v_x n_x + v_y n_y) + (r_x n_x + r_y n_y)v] \\ (E + p)(un_x + vn_y) - \nu(E_x n_x + E_y n_y) \\ -\rho n_x \\ -\rho n_y \\ -un_x \\ -un_y \\ -vn_x \\ -vn_y \\ -En_x \\ -En_y \end{bmatrix}. \quad (1.28)$$

For simplicity it is helpful to examine only the spatial fluxes and define the following variables $un = un_x + vn_y$, $u_x n = u_x n_x + u_y n_y$, $v_x n = v_x n_x + v_y n_y$, $w_x n = w_x n_x + w_y n_y$, $r_x n = r_x n_x + r_y n_y$, $E_x n = E_x n_x + E_y n_y$ and then rewrite the flux as below.

$$F_{\mathbf{x}} \cdot n_{\mathbf{x}} = \begin{bmatrix} \rho(un) - \nu(r_x n) \\ \rho u(un) + pn_x - \nu[\rho(u_x n) + (r_x n)u] \\ \rho v(un) + pn_y - \nu[\rho(v_x n) + (r_x n)v] \\ (E + p)(un) - \nu(E_x n) \\ -\rho n_x \\ -\rho n_y \\ -un_x \\ -un_y \\ -vn_x \\ -vn_y \\ -En_x \\ -En_y \end{bmatrix} \quad (1.29)$$

Paralleling the work done for the spacetime discretization of HNS20G, it is advisable to split the flux:

$$\Phi_{jk} = \Phi_{jk}^{inv} |(\hat{n}_x, \hat{n}_y)| + \Phi_{jk}^{vis} |(\hat{n}_x, \hat{n}_y)| + \Phi_{jk}^{time} |\hat{n}_t|, \quad (1.30)$$

where $\hat{\mathbf{n}}_{jk} = \mathbf{n}_{jk}/|\mathbf{n}_{jk}| = (\hat{n}_x, \hat{n}_y, \hat{n}_t)$, Φ_{jk}^{inv} , and Φ_{jk}^{vis} are the spatial inviscid and viscous fluxes respectively and Φ_{jk}^{time} is the temporal flux, and then define an upwind viscous flux using a dissipation matrix defined from the viscous system alone. The viscous numerical flux Φ_{jk}^{vis} is defined as:

$$\Phi_{jk}^{vis} = \frac{1}{2} (F_{jn}^{vis} + F_{kn}^{vis}) - \frac{1}{2} \mathbf{P}^{-1} \left| \mathbf{P} \frac{\partial \mathbf{f}_{\mathbf{n}}}{\partial \mathbf{u}} \right| \Delta \mathbf{u}. \quad (1.31)$$

The A matrix $\left(\left|P\frac{\partial f_n}{\partial \mathbf{u}}\right|\right)$ for the viscous system is computed by first linearizing the viscous flux with an assumption of constant viscosity for simplicity and this returns:

$$A = - \begin{bmatrix} 0 & 0 & 0 & 0 & \nu n_x & \nu n_y & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{u}{\rho}\nu(r_x n) & \frac{\nu}{\rho}(r_x n) & 0 & 0 & \nu u n_x & \nu u n_y & \nu \rho n_x & \nu \rho n_y & 0 & 0 & \nu n_x & \nu n_y \\ \frac{v}{\rho}\nu(r_x n) & 0 & \frac{\nu}{\rho}(r_x n) & 0 & \nu v n_x & \nu v n_y & 0 & 0 & \nu \rho n_x & \nu \rho n_y & \nu n_x & \nu n_y \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \nu n_x & \nu n_y \\ n_x & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ n_y & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -\frac{u}{\rho}n_x & \frac{n_x}{\rho} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -\frac{u}{\rho}n_y & \frac{n_y}{\rho} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -\frac{v}{\rho}n_x & 0 & \frac{n_x}{\rho} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -\frac{v}{\rho}n_y & 0 & \frac{n_y}{\rho} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & n_x & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & n_y & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \quad (1.32)$$

The final step is to diagonalize this matrix, and then use the eigenvalue and eigenvector decomposition to obtain the dissipation matrix. Finally, using the local preconditioning approach, as in previous HNS formulations, returns:

$$|PA| = \begin{bmatrix} a\nu & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & a\nu & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & a\nu & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & a\nu & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{\nu n_x^2}{a\nu} & \frac{\nu n_x n_y}{a\nu} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{\nu n_x n_y}{a\nu} & \frac{\nu n_y^2}{a\nu} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{\nu n_x^2}{a\nu} & \frac{\nu n_x n_y}{a\nu} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{\nu n_x n_y}{a\nu} & \frac{\nu n_y^2}{a\nu} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{\nu n_x^2}{a\nu} & \frac{\nu n_x n_y}{a\nu} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{\nu n_x n_y}{a\nu} & \frac{\nu n_y^2}{a\nu} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{\nu n_x^2}{a\nu} & \frac{\nu n_x n_y}{a\nu} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{\nu n_x n_y}{a\nu} & \frac{\nu n_y^2}{a\nu} \end{bmatrix}, \quad (1.33)$$

where $a\nu$ is the viscous wave speed defined as $a\nu = \sqrt{\nu/T_h}$, where $T_h = \frac{L_h^2}{\nu}$ and L_h is defined as $L_h = \frac{L_d}{\sqrt{Re_{L_d}}}$, with $Re_{L_d} = \frac{U_\infty L_d}{\nu}$.

2. Nonlinear Solver

The nonlinear solver here is the the node-centered finite-volume solver in FUN3D²¹ modified to work for two-dimensional spacetime solvers for laminar flows (four unknowns per node or 12 unknowns for the hyperbolic discretizations) as opposed to at least five unknowns used for three-dimensional compressible flows with turbulence modeling. This solver uses an implicit defect-correction method²¹ defined by

$$\mathbf{U} = \mathbf{U} + \omega \Delta \mathbf{U}, \quad (2.1)$$

where ω is a scalar underrelaxation parameter defined by a realizability check, $\mathbf{U}$ is a global vector of the discrete unknowns (conservative variables stored at the vertices of the mesh), and $\Delta \mathbf{U}$, computed by relaxing the following system of linear equations.

$$[\mathbf{P}] \Delta \mathbf{U} = -\mathbf{R}(\mathbf{U}), \quad (2.2)$$

where $\mathbf{R}(\mathbf{U})$ is a global vector of the residuals defined at each node. Here, $[\mathbf{P}]$ is the exact derivative of the residual operator with zero least-squares (LSQ) gradients (denoted by $\tilde{\mathbf{R}}$) augmented with a diagonal term

to improve the diagonal dominance of the Jacobian matrix.

$$[P] = \left[\frac{\partial \tilde{\mathbf{R}}}{\partial \mathbf{U}} \right] + \mathbf{D}, \quad (2.3)$$

and $\mathbf{D}$ is the block diagonal matrix with each 4x4 block defined by the diagonal matrix with the ratio of the dual volume to the local pseudo time step: e.g., at a node j ,

$$\mathbf{D}_j = \mathbf{I}_{4 \times 4} \frac{V_j}{CFL \Delta \tau_j}, \quad (2.4)$$

where $\mathbf{I}_{4 \times 4}$ is the identity matrix, CFL is the Courant-Friedrichs-Levy number²² used to maintain stability, and V_j is the dual volume at node j . The local pseudo-time-step $\Delta \tau_j$ is defined by

$$\Delta \tau_j = \frac{V_j}{\sum_{k \in \{k_j\}} \left[|\mathbf{v} \cdot \hat{\mathbf{n}}_{\mathbf{xy}}| + \sqrt{\frac{\gamma p}{\rho}} + 1 \right] A_{jk}}, \quad (2.5)$$

where the quantities at the face with area A_{jk} are computed by the arithmetic average of the values stored at nodes j and k . The linear system is relaxed by a point-implicit multi-color Gauss-Seidel relaxation scheme with a specified number of relaxations.

As in earlier work,^{14, 16} this spacetime solver is implemented as a new governing equation option in the FUN3D code.²³

3. Verification

3.1. Exponential MMS

We verify the implementation of the residual through the method of manufactured solutions. The manufactured solution, used here for the unknowns at each node, is constructed by the following function:

$$a = a_0 + a_{scale} \exp(a_x x + a_y y + a_t t), \quad (3.1)$$

where a_0, a_{scale}, a_x, a_y , and a_t are user defined values that vary for each unknown - note that in our implementation z is the temporal coordinate. See Table 1 for the values used in this study. This set of functions is made to be an exact solution to the Eulerian flow equations, using the source term in Equation 3.2 that is computed numerically at each node.

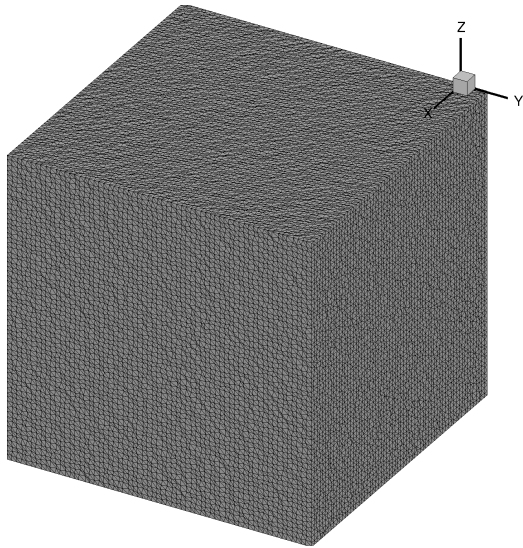
$$\mathbf{s}(x, y, z) = \frac{\partial \mathbf{f}_x(\mathbf{w}_{exact})}{\partial x} + \frac{\partial \mathbf{f}_y(\mathbf{w}_{exact})}{\partial y} + \frac{\partial \mathbf{u}(\mathbf{w}_{exact})}{\partial z} \quad (3.2)$$

The order of the discretization error is measured by $DE = ||u_{exact} - u||_1$, using a sequence of hierarchically refined regular tetrahedral grids (in which the nodes are then perturbed to obtain irregularity) in a unit cubic domain with $n \times n \times n$ nodes, where $n = 16, 32, 64, 128$, and 256 ; an example of one of these grids is shown in Figure 1 along with the solution and gradient variable fields for comparison, which show the expected improved behavior for the hyperbolic discretization. This case is in a flow with conditions $Re_\infty = 3900$ and dimensional freestream temperature $\tilde{T}_\infty = 460^\circ R$. Figure 2 shows the discretization error results that demonstrate that both the alpha-damping and hyperbolic discretizations of the Eulerian fluid model are 2nd order accurate as measured by the discretization error of the primitive variables. This is not sufficient to verify the accuracy of the HEF discretization as it should maintain 2nd order accuracy in the gradient variables as well, as this is shown in Figure 3

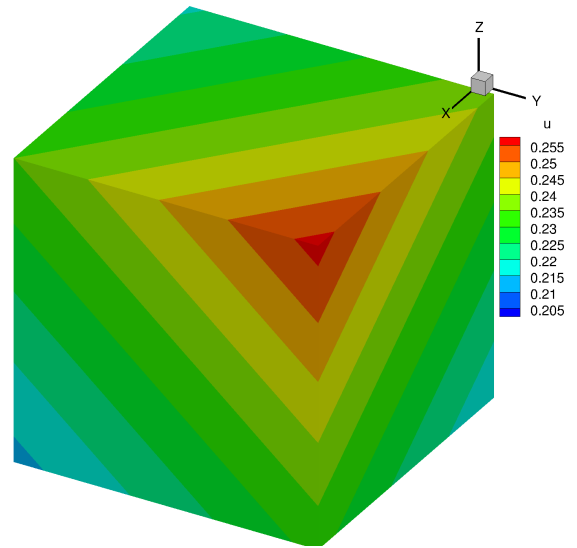
3.2. Boundary Layer MMS

Next, it is helpful to consider a more realistic case with the following boundary-layer-type functions:

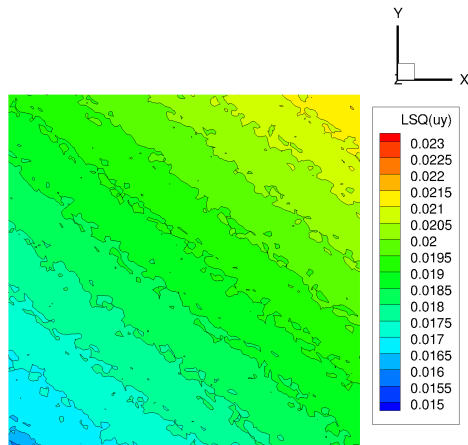
$$\mathbf{w}_{exact} = \begin{bmatrix} 1.0 \\ 1.0 \\ 0.1 \\ 1.0 \end{bmatrix} - \begin{bmatrix} 0.1 \\ 1.0 \\ 0.1 \\ 0.1 \end{bmatrix} \exp(-\eta), \quad \eta = y \sqrt{\frac{R(z)}{x - 0.2}}, \quad (3.3)$$



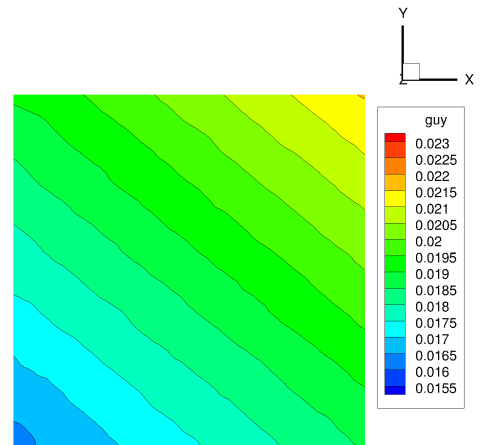
(a) Perturbed tetrahedral mesh.



(b) X-velocity solution from HEF discretization.



(c) Least-squares y-gradient of x-velocity.



(d) Hyperbolic Eulerian flow y-gradient variables of x-velocity.

Figure 1. Mesh, solution and gradient values on 64^3 mesh.

Table 1. Parameters used to define the exact solution.

variable	a_0	a_{scale}	a_x	a_y	a_t
ρ	.5	1.0	0.525	0.550	0.575
u	.1	0.1	0.125	0.150	0.175
v	.2	0.2	0.225	0.250	0.275
p	.4	0.714	0.425	0.450	0.475

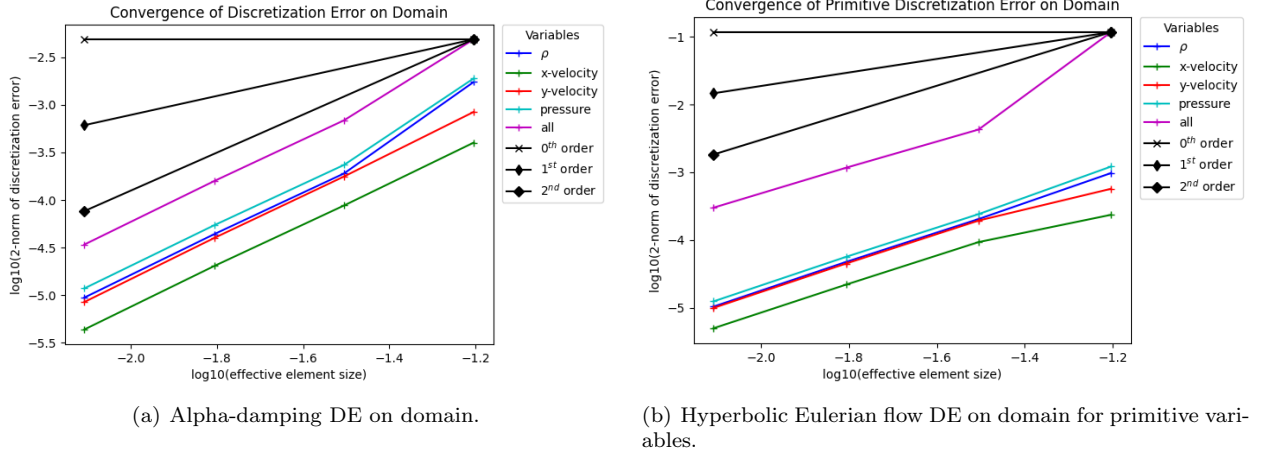


Figure 2. Primitive variable DE convergence for alpha-damping and hyperbolic discretizations.

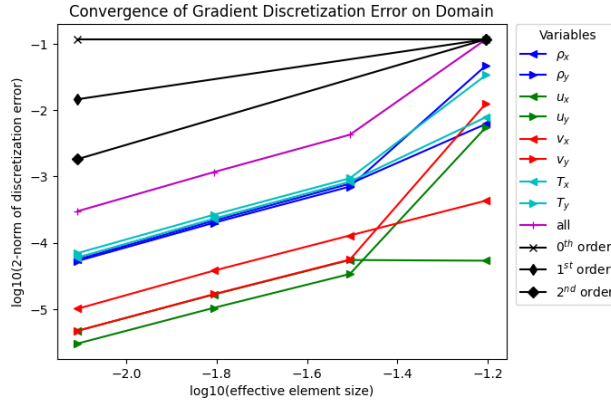


Figure 3. Hyperbolic Eulerian flow DE on domain for gradient variables.

where $R(z)$ is a time-dependent parameter – once again z is the temporal coordinate – defined by

$$R(z) = 10^5 [1 + 0.75 \sin(4\pi z)] . \quad (3.4)$$

The Eulerian fluid governing equations are solved on an adapted mesh created by using the *refine* package developed at NASA Langley.^{24,25} To minimize interpolation error, this mesh was generated by performing ten cycles of solving the alpha-damping discretization of the Eulerian fluid governing equations and adapting the minimized interpolation error. Figure 4 shows the final adapted grid with 69,076 nodes and 378,209 tetrahedra, generated in the earlier work.¹⁶ Figures 5 and 6 show a comparison in the XY and XZ planes, respectively, between the least-squares and HEF provided gradients, and these show as before that the HEF provided gradients are much smoother than the least squares computed ones. This is an adapted mesh, which is highly irregular – having accurate gradients on these meshes is not a trivial benefit – and there is an

interest in comparing the y-gradients of the x-velocity as those are the gradients that would drive the skin friction calculation on this sort of problem.

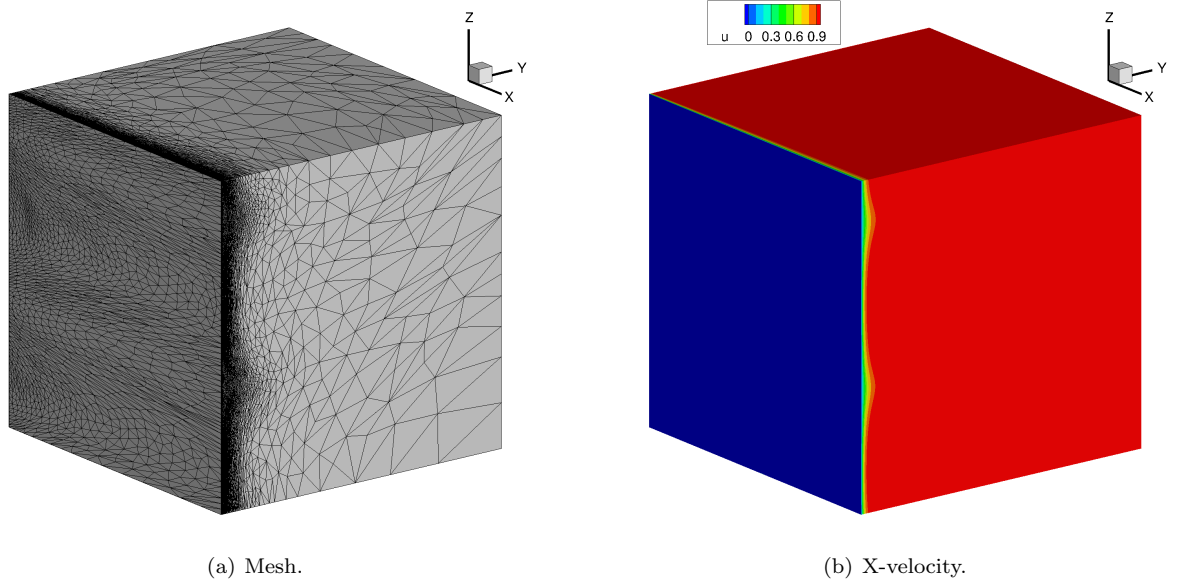


Figure 4. Mesh and solution on adapted mesh.

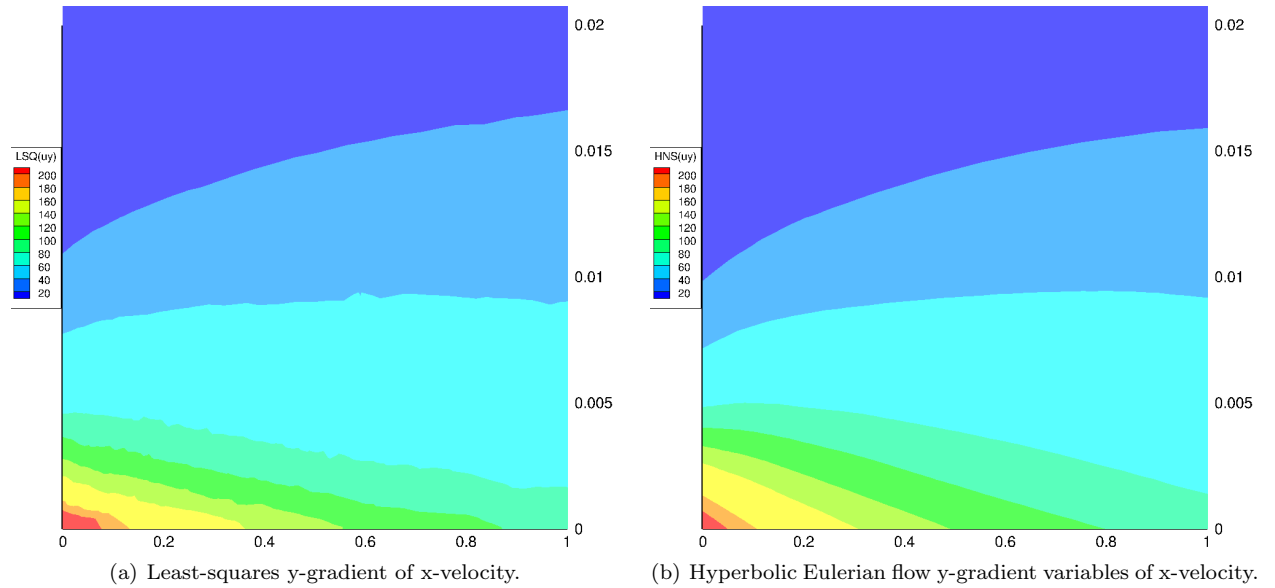
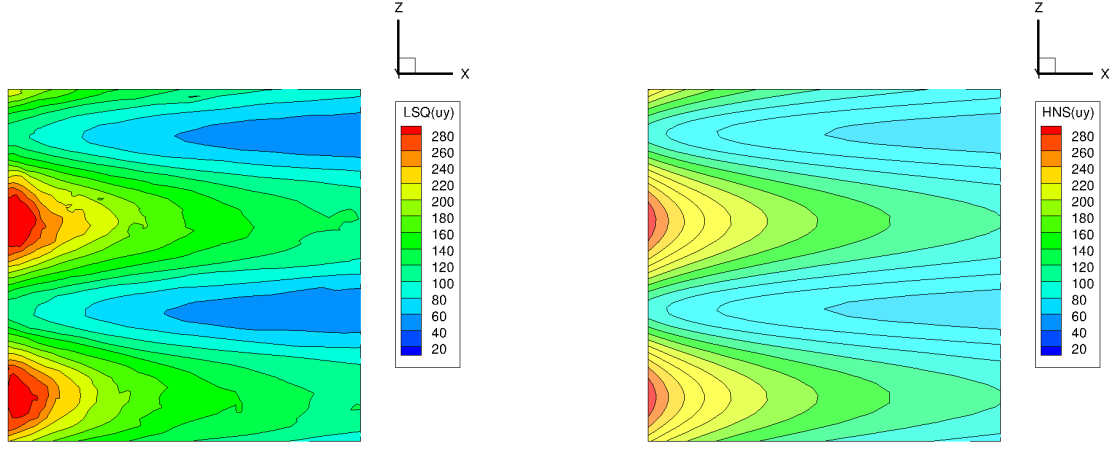


Figure 5. XY plane at $z = 0.619$.

The contour planes demonstrate better solution accuracy, but a natural follow-up asks for a comparison in time to solution. Figure 7 shows that there is a pronounced speedup in time to solution, this is driven by a steep decrease in the number of the nonlinear iterations required to solve the system. This is similar to observations in earlier work,¹⁴ which also showed a noticeable improvement in time to solution. One intricacy to note is that in the EF formulation, every equation now has an $O(h^2)$ conditioning term that can retard the solution process, as opposed to the NS formulation, which lacks mass diffusion in continuity so has better conditioning in that equation; this can explain the greater increase in performance when comparing HEF to EF and HNS to NS (which showed approximately a factor of 2 speedup rather than the factor of approximately 5 shown here).



(a) Least-squares y-gradient of x-velocity. (b) Hyperbolic Eulerian flow y-gradient variables of x-velocity.

Figure 6. XZ plane at $y = 0.5$.

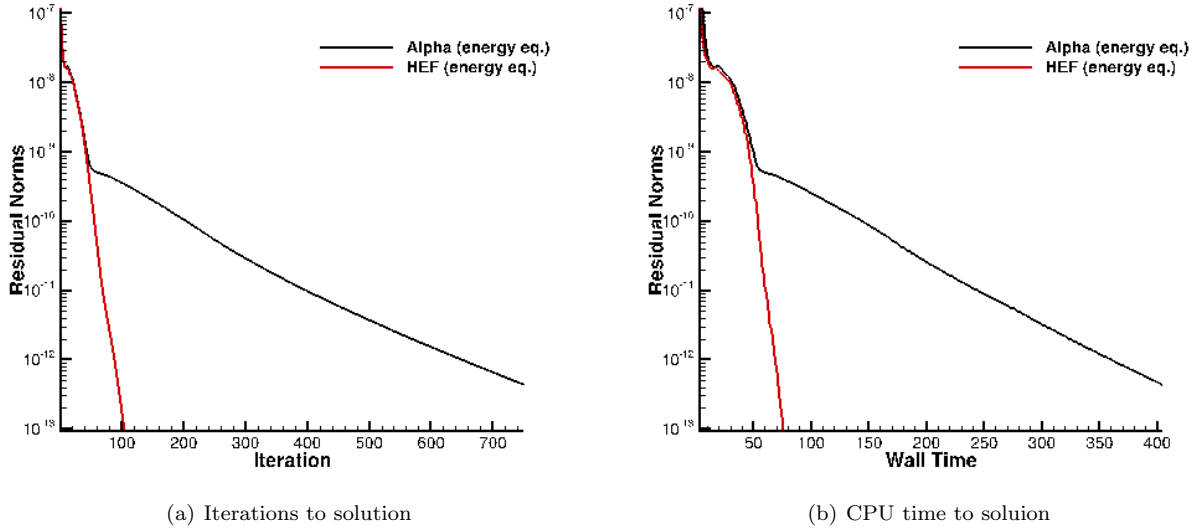


Figure 7. Comparison of CPU time and iterations to convergence for boundary layer MMS.

3.3. Subsonic Cylinder in Cross Flow

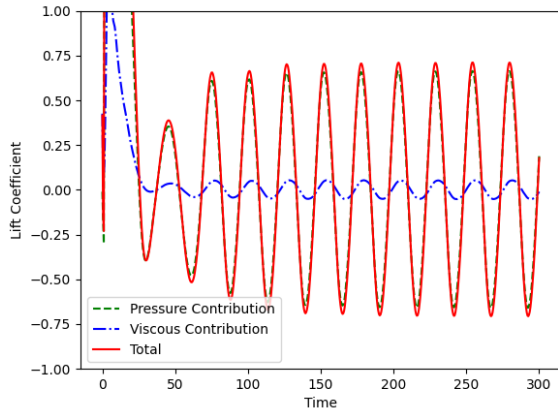
The final case in this paper shows the results of a more realistic application of the spacetime viscous solver. Next, we consider a viscous flow over a circular cylinder of unit diameter at the freestream Mach number $M_\infty = 0.2$ and the freestream Reynolds number per diameter $Re_\infty = 200$. The circular cylinder is centered at the origin, and the outer boundary is a square whose side is 100 units. This two-dimensional domain is then extended from $z = 0$ to $z = 300$ units. The grid with 15,720,940 nodes and 91,499,780 tetrahedra was generated after performing 13 grid adaptation cycles using numerical solutions produced by the alpha-damping solver.

Table 2 shows that the Strouhal number (St), and engineering coefficients ($c_{L_{RMS}}$ and $c_{D_{avg}}$) correspond between the Navier-Stokes (the second and third rows) and the Eulerian flow (the final two rows) models to a high degree of accuracy, which is a promising and necessary result in that they validate both the

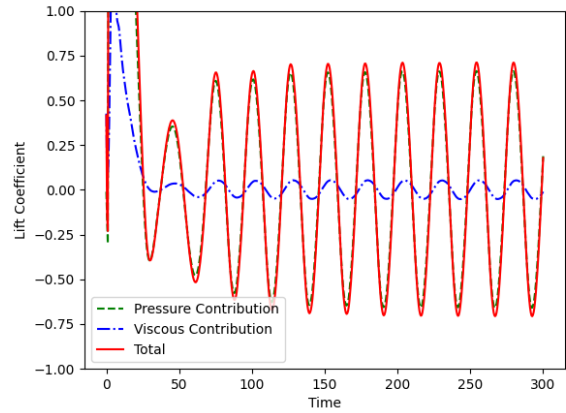
implementation and the governing equations. Figure 8 shows that the change from the alpha-damping to the hyperbolic viscous discretization has almost no effect on the shedding behavior of the cylinder. While Figure 9 compares the gradient in the normal direction of the x-direction velocity on the surface of the cylinder and shows the greatly improved quality of the gradients obtained by the HNS discretization - they are smoother and less dissipative than those obtained by least-squares. As shown for the boundary layer MMS problem in Figure 7, Figure 10 shows a marked decrease in time and iterations to solution. The number of iterations is decreased by a factor of 15 with the HNS solution actually reaching the target residual reduction, while the alpha-damping solution process has reached the maximum iteration count without reaching the desired residual tolerance. This corresponds to a decrease in runtime by approximately a factor of 3. This shows that the expected benefits in solver performance are maintained even on semi-realistic geometries.

Table 2. Comparison of engineering quantities to reference values.

	St	$c_{L_{rms}}$	$c_{D_{avg}}$
reference ²⁶	0.1957	0.4244	1.3365
NS values	0.1961	0.4941	1.3376
EF values	0.1964	0.4991	1.3699
HEF values	0.1969	0.4986	1.3697



(a) C_L evolution for alpha-damping discretization of the Eulerian fluid equations.



(b) C_L evolution for hyperbolic discretization of the Eulerian fluid equations.

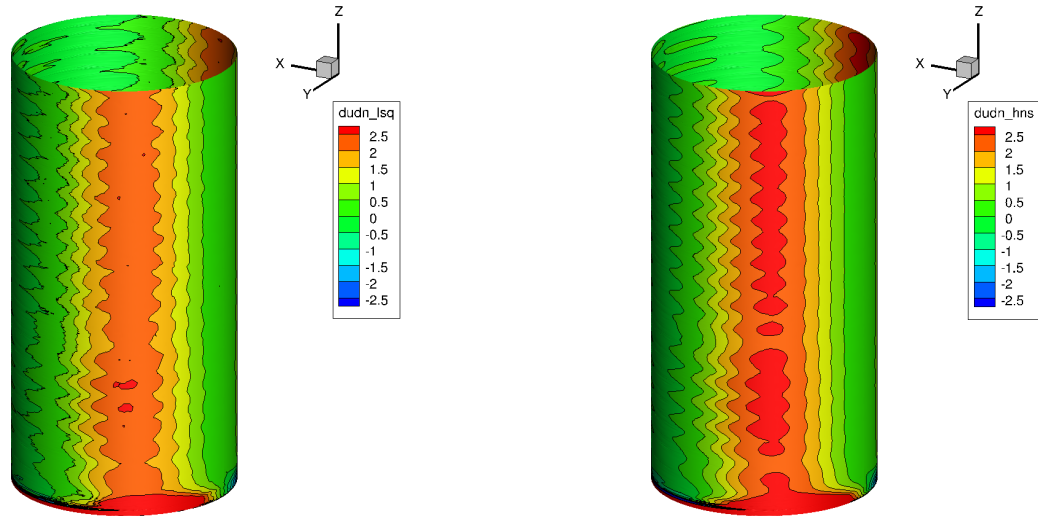
Figure 8. c_L as a function of z (through time).

4. Conclusions

This work demonstrated the development, possibility, and promise of Eulerian fluid and Hyperbolic Eulerian fluid formulations for finite-volume schemes. We demonstrated very close agreement between the results of the Navier-Stokes and the Eulerian fluid governing equations with the benefit being a simpler viscous discretization. Furthermore, the hyperbolic discretization shows two main benefits over the typical diffusion equations with a pronounced decrease in time and iterations to solution and the improved quality of the gradients on skewed meshes.

5. Acknowledgements

I would like to thank Kyle Anderson for his discussions and encouragement to pursue this area of interest as well as directing me to Svard's initial paper, Hiro Nishikawa for his assistance with deriving the HEF dissipation matrix, the NASA Langley K-Cluster for the compute time, and U.S. Army Research Office under the contract/grant number W911NF-19-1-0429 for funding the HNS portion of this work that was completed and published at the National Institute of Aerospace prior to the author joining NASA Langley.



(a) Least-squares gradients of x-velocity in normal direction. (b) Hyperbolic gradients of x-velocity in normal direction.

Figure 9. Comparison of gradient computation methods.

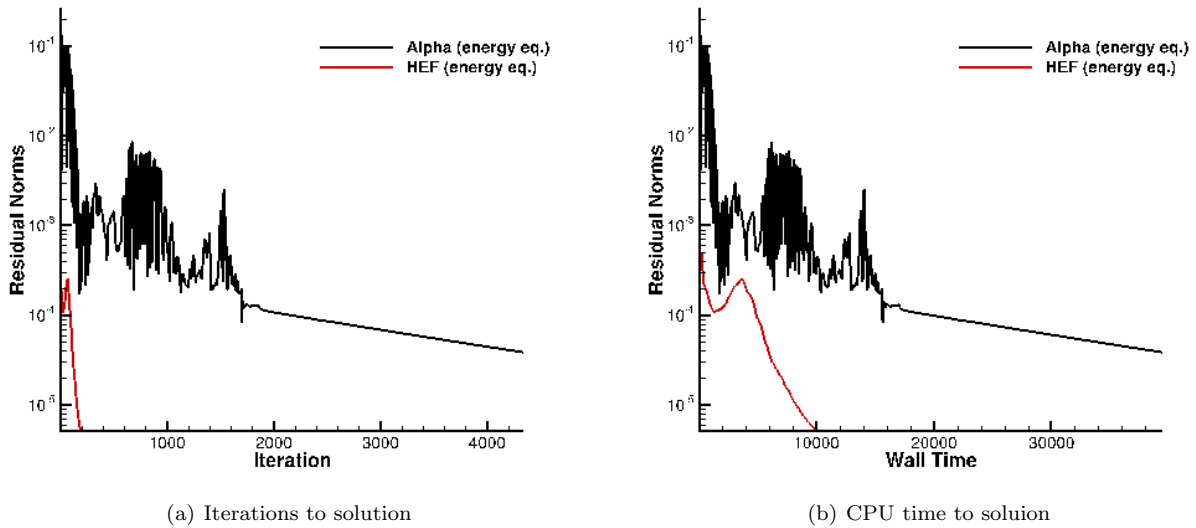


Figure 10. Comparison of CPU time and iterations to convergence for subsonic cylinder in crossflow.

References

- ¹Svärd, M., "A new Eulerian model for viscous and heat conducting compressible flows," *Physica A: Statistical Mechanics and its Applications*, Vol. 506, 2018, pp. 350–375.
- ²Brenner, H., "Kinematics of Volume Transport," *Physica A: Statistical Mechanics and its Applications*, Vol. 349, No. 1, 2005, pp. 11–59.
- ³Brenner, H., "Navier–Stokes Revisited," *Physica A: Statistical Mechanics and its Applications*, Vol. 349, No. 1, 2005, pp. 60–132.
- ⁴Bardow, A. and Christian Öttinger, H., "Consequences of the Brenner Modification to the Navier–Stokes Equations for Dynamic Light Scattering," *Physica A: Statistical Mechanics and its Applications*, Vol. 373, No. C, 2007, pp. 88–96.
- ⁵Svärd, M., "A New Eulerian Model for Viscous and Heat Conducting Compressible Flow," 2018.
- ⁶Nishikawa, 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.
- ⁷Nishikawa, 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.

⁸Nishikawa, 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.

⁹Liu, 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.

¹⁰Liu, 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.

¹¹Nishikawa, H. and Liu, Y., “Hyperbolic Advection-Diffusion Schemes for High-Reynolds-Number Boundary-Layer Problems,” *Journal of Computational Physics*, Vol. 352, 2018, pp. 23–51.

¹²Li, L., Lou, J., Nishikawa, H., and Luo, H., “Reconstructed Discontinuous Galerkin Methods for Compressible Flows based on a New Hyperbolic Navier-Stokes System,” *Journal of Computational Physics*, Vol. 427, 2021, pp. 110058.

¹³“FUN3D online manual,” https://fun3d.larc.nasa.gov/papers/FUN3D_INTG_Manual-14.0.pdf.

¹⁴Padway, E. and Nishikawa, H., *An Adaptive Space-Time Hyperbolic Navier-Stokes Solver for Two-Dimensional Unsteady Viscous Flows*.

¹⁵Masatsuka, K., “I do like CFD, VOL.1, Second Edition,” <http://www.cfdbooks.com>, 2013.

¹⁶Padway, E. and Nishikawa, H., *An Adaptive Space-Time Edge-Based Solver for Two-Dimensional Unsteady Viscous Flows*.

¹⁷Nishikawa, 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.

¹⁸Nishikawa, H., “Two Ways to Extend Diffusion Schemes to Navier-Stokes Schemes: Gradient Formula or Upwinding,” *20th AIAA Computational Fluid Dynamics Conference*, AIAA Paper 2011-3044, Honolulu, Hawaii, 2011.

¹⁹Jalali, A., Sharbatdar, M., and Ollivier-Gooch, C., “Accuracy Analysis of Unstructured Finite Volume Discretization Schemes for Diffusive Fluxes,” Vol. 101, September 2014, pp. 220–232.

²⁰Nishikawa, H., “First-, Second-, and Third-Order Finite-Volume Schemes for Diffusion,” *Journal of Computational Physics*, Vol. 256, 2014, pp. 791–805.

²¹Anderson, W. K. and Bonhaus, D. L., “An Implicit Upwind Algorithm for Computing Turbulent Flows on Unstructured Grids,” *Computers & Fluids*, Vol. 23, No. 1, 1994, pp. 1–21.

²²Courant, R., Friedrichs, K., and Lewy, H., “Über die partiellen Differenzengleichungen der mathematischen Physik,” *Mathematische annalen*, Vol. 100, No. 1, 1928, pp. 32–74.

²³Anderson, W. K., Biedron, R. T., Carlson, J.-R., Delaga, J. M., Jr., C. T. D., Gnoffo, P. A., Hammond, D. P., Jacobson, K. E., Jones, W. T., Kleb, B., Lee-Rausch, E. M., Nastac, G. C., Nielsen, E. J., Padway, E., Park, M. A., Rumsey, C. L., Thomas, J. L., Thompson, K. B., Walden, A. C., Wang, L., Wood, S. L., and Wood, W. A., “FUN3D Manual: 14.0.2,” Tech. Rep. NASA/TM-20230004211, NASA Langley Research Center, Hampton, Virginia, November 2023.

²⁴“Refine Repository,” <https://github.com/nasa/refine>, Accessed: 2020-06-08.

²⁵Park, M. A., *Anisotropic Output-Based Adaptation with Tetrahedral Cut Cells for Compressible Flows*, Ph.D. thesis, Massachusetts Institute of Technology, Sept. 2008.

²⁶Rajani, B., Kandasamy, A., and Majumdar, S., “Numerical simulation of laminar flow past a circular cylinder,” *Applied Mathematical Modelling*, Vol. 33, No. 3, 2009, pp. 1228–1247.